User's Manual

32

RX660 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX600 Series

Renesas Electronics
www.renesas.com

Rev.1.00 Mar 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 RX660 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	RX660 Group Datasheet	R01DS0393EJ
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX660 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

R01UH0937EJ0100
Rev.1.00
Mar 18, 2022

120-MHz 32-bit RX MCU, on-chip FPU, 709 CoreMark, Supportive of 5V power supply, up to 1-MB flash memory, up to 128-KB SRAM, 32-KB data flash memory, various communications interfaces, including CAN FD, 12-bit A/D converter, 12-bit D/A converter, Analog comparator, RTC, Remote control signal receiver

Features

■ 32-bit RXv3 CPU core

- Maximum operating frequency: 120 MHz
  Capable of 709 CoreMark in operation at 120 MHz
- 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 5.5-V supply
- Deep software standby mode with the RTC continuing to run
- Four low-power modes

■ On-chip code flash memory

- Supports versions with up to 1 Mbyte of ROM
- No waiting for access in 120-MHz operation
- User code is programmable by on-board or off-board programming.
- Programming/erasing as background operations (BGOs)

■ On-chip data flash memory

- 32 Kbytes, reprogrammable up to 100,000 times
- Programming/erasing as background operations (BGOs)

■ On-chip SRAM

- 128 Kbytes of SRAM (no wait states)

■ External address space

- Buses for full-speed data transfer (maximum operating frequency of 40 MHz)
- Four CS areas
- 8- or 16-bit bus space is selectable per area

■ Data transfer

- DMACAa: 8 channels
- DTCb: 1 channel

■ ELC

- Module operation can be initiated by event signals without using interrupts
- Linked operation between modules is possible when the CPU is in sleep mode

■ Reset and supply management

- Power-on reset (POR)
- Low voltage detection (LVD)

■ Clock functions

- The main clock oscillator is connectable to an 8- to 24-MHz external crystal resonator and usable as the PLL reference clock.
- 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

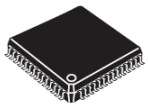
■ Independent watchdog timer

- 120-kHz IWDT-dedicated on-chip oscillator clock operation

■ Useful functions for IEC60730 compliance

- Oscillation-stoppage detection, functions for self-diagnosis and detection of disconnection for the A/D converter, clock frequency accuracy measurement circuit, independent watchdog timer, RAM test-assisting function by DOC, and CRCA, etc.
- Register write protection function that protects important registers against overwriting

■ Remote control signal receiver



PLQP0144KA-B 20 × 20 mm, 0.5 mm pitch
PLQP0100KB-B 14 × 14 mm, 0.5 mm pitch
PLQP0080KB-B 12 × 12 mm, 0.5 mm pitch
PLQP0064KB-C 10 × 10 mm, 0.5 mm pitch
PLQP0048KB-B 7 × 7 mm, 0.5 mm pitch

■ Various communications interfaces

- CAN FD: Compliant with ISO 11898-1:2015, standard frame and extended frame (1 channel)
- SCIk, SCIm, 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)
- The I^2C bus interfaces (RIICa) for transfer at up to 400 kbps (fast mode), capable of SMBus operation (2 channels)
- RSPId (1 channel) for transfer at up to 30 Mbps

■ Up to 19 extended-function timers

- 16-bit MTU3a
- 8-bit TMRb (4 channels), 16-bit CMT (4 channels), 32-bit CMTW (2 channels)

■ 12-bit A/D converter

- Single 12-bit unit (24 channels)
- Self diagnosis, detection of analog input disconnection

■ Analog Comparator (CMPC): 4 channels

■ 12-bit D/A converter (R12DAb): 2 channels

- Usable as a reference voltage for the analog comparator

■ Temperature sensor for measuring temperature within the chip

■ Up to 134 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 (1/8)

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
	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/512 Kbytes • 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.
	Data flash memory	• Capacity: 32 Kbytes • Programming/erasing: 100,000 times
	Unique ID	• 12-byte unique ID for the device
	RAM	• Capacity: 128 Kbytes • 120 MHz, no-wait access
Operating modes		• Operating modes by the mode-setting pins at the time of release from the reset state Single-chip mode Boot mode (for the SCI interface) Boot mode (for the FINE interface) User boot mode • Selection of operating mode by register setting Single-chip mode User boot mode On-chip ROM disabled extended mode On-chip ROM enabled extended mode • Endian selectable

Table 1.1 Outline of Specifications (2/8)

Classification	Module/Function	Description
Clock	Clock generation circuit	• Main clock oscillator, sub-clock oscillator, low-speed/high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • The peripheral module clocks can be set to frequencies above that of the system clock. • Main-clock oscillation stoppage detection • Separate frequency-division and multiplication settings for the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, 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 The following peripheral modules run in synchronization with PCLKA, which runs at up to 120 MHz: MTU, RSPI, SCIs (SCI10 and SCI11), RSCI, and the ECC function control registers in the CAN FD module. Other peripheral modules run in synchronization with PCLKB: Up to 60 MHz ADCLK in the S12AD 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 40 MHz • Multiplication is possible with using the high-speed on-chip oscillator (HOCO) as a reference clock of the PLL circuit
Reset		Nine types of reset • RES# pin reset: Generated when the RES# pin is driven low. • Power-on reset: Generated when the RES# pin is driven high and VCC = AVCC0 rises. • Voltage-monitoring 0 reset: Generated when VCC = AVCC0 falls. • Voltage-monitoring 1 reset: Generated when VCC = AVCC0 falls. • Voltage-monitoring 2 reset: Generated when VCC = AVCC0 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 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 pins and generates an internal reset or internal interrupt in response to the voltage reaching a threshold. • 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 two different levels • Voltage detection circuits 1 and 2 Voltage detection level: Selectable from five different levels Digital filtering (1/2, 1/4, 1/8, and 1/16 LOCO frequency) Capable of generating an internal reset • Two types of timing are selectable for release from reset An internal interrupt can be requested. • Detection of voltage rising above and falling below thresholds is selectable. • Maskable or non-maskable interrupt is selectable Voltage detection monitoring Event linking
Low power consumption	Low power consumption function	• Module stop function • Four low power consumption modes Sleep mode, all-module clock stop mode, software standby mode, and deep software standby mode
Interrupt	Interrupt controller (ICUF)	• Peripheral function interrupts: 256 sources • External interrupts: 16 (pins IRQ0 to IRQ15) • Software interrupts: 2 sources • Non-maskable interrupts: 7 sources • Sixteen levels specifiable for the order of priority • Method of interrupt source selection: The interrupt vectors consist of 256 vectors (128 sources are fixed. The remaining 133 vectors are selected from among the other 128 sources.)

Table 1.1 Outline of Specifications (3/8)

Classification	Module/Function	Description
External bus extension		• The external address space can be divided into four areas (CS0 to CS3), each with independent control of access settings. Capacity of each area: 2 Mbytes (CS0 to CS3) A chip-select signal (CS0# to CS3#) 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). • Bus format: Separate bus, multiplex bus • Wait control • Write buffer facility
DMA	DMA controller (DMACAa)	• 8 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Software trigger, external interrupts, 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
I/O ports	Programmable I/O ports	• I/O ports for the 144-pin LFQFP with no JTAG interface and no sub-clock oscillator I/O pins: 133 Input pin: 1 Pull-up resistors: 133 Open-drain outputs: 133 5-V tolerance: 4 • I/O ports for the 144-pin LFQFP with a sub-clock oscillator, but no JTAG interface I/O pins: 131 Input pin: 1 Pull-up resistors: 131 Open-drain outputs: 131 5-V tolerance: 4 • I/O ports for the 144-pin LFQFP with a JTAG interface, but no sub-clock oscillator I/O pins: 132 Input pin: 1 Pull-up resistors: 132 Open-drain outputs: 132 5-V tolerance: 4 • I/O ports for the 144-pin LFQFP with a JTAG interface and a sub-clock oscillator I/O pins: 130 Input pin: 1 Pull-up resistors: 130 Open-drain outputs: 130 5-V tolerance: 4 • I/O ports for the 100-pin LFQFP with no JTAG interface and no sub-clock oscillator I/O pins: 91 Input pin: 1 Pull-up resistors: 91 Open-drain outputs: 91 5-V tolerance: 4 • I/O ports for the 100-pin LFQFP with a sub-clock oscillator, but no JTAG interface I/O pins: 89 Input pin: 1 Pull-up resistors: 89 Open-drain outputs: 89 5-V tolerance: 4 • I/O ports for the 100-pin LFQFP with a JTAG interface, but no sub-clock oscillator I/O pins: 90 Input pin: 1 Pull-up resistors: 90 Open-drain outputs: 90 5-V tolerance: 4

Table 1.1 Outline of Specifications (4/8)

Classification	Module/Function	Description
I/O ports	Programmable I/O ports	• I/O ports for the 100-pin LFQFP with a JTAG interface and a sub-clock oscillator I/O pins: 88 Input pin: 1 Pull-up resistors: 88 Open-drain outputs: 88 5-V tolerance: 4 • I/O ports for the 80-pin LFQFP with no sub-clock oscillator I/O pins: 71 Input pin: 1 Pull-up resistors: 71 Open-drain outputs: 71 5-V tolerance: 4 • I/O ports for the 80-pin LFQFP with a sub-clock oscillator I/O pins: 69 Input pin: 1 Pull-up resistors: 69 Open-drain outputs: 69 5-V tolerance: 4 • I/O ports for the 64-pin LFQFP with no sub-clock oscillator I/O pins: 55 Input pin: 1 Pull-up resistors: 55 Open-drain outputs: 55 5-V tolerance: 2 • I/O ports for the 64-pin LFQFP with a sub-clock oscillator I/O pins: 53 Input pin: 1 Pull-up resistors: 53 Open-drain outputs: 53 5-V tolerance: 2 • I/O ports for the 48-pin LFQFP I/O pins: 39 Input pin: 1 Pull-up resistors: 39 Open-drain outputs: 39 5-V tolerance: 2
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. • 83 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	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
	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.

Table 1.1 Outline of Specifications (5/8)

Classification	Module/Function	Description
Timers	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 • 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
	Realtime clock (RTCC)*1	• Clock sources: 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 • Time capture function (up to 3 pins) • Event linking by the ELC

Table 1.1 Outline of Specifications (6/8)

Classification	Module/Function	Description
Communication function	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
	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 • 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. • RSCI has some home bus system (HBS) functionality. • Data match detection • Adjustment of the timing of sampling of the RXD signals
	I^2C bus interface (RIICa)	• 2 channels Communication formats I^2C bus format/SMBus format Supports the multi-master • Event linking by the ELC
	CAN FD module (CANFD)*2	• 1 channel • Compliance with the ISO11898-1:2015 specification (standard frame and extended frame)

Table 1.1 Outline of Specifications (7/8)

Classification	Module/Function	Description
Communication function	Serial peripheral interface (RSPId)	• 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) 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
	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.
12-bit A/D converter (S12ADH)		• 12 bits (24 channels × 1 unit) • 12-bit resolution • Minimum conversion time 0.9 µs per channel (when ADCLK operates at 60 MHz) • Operating mode Scan mode (single scan mode, continuous scan mode, or 3 group scan mode) Group A priority control (only for 3 group scan mode) • Sampling variable Sampling time can be set up for each channel. • Conversion function in order of arbitrarily selected channels (Serial conversion of the same channel cannot be allowed) • Double trigger mode (A/D conversion data duplicated) • Three ways to start A/D conversion Software trigger, synchronous trigger (MTU, TMR, ELC), external trigger • Prioritization in group scanning can be controlled among group A, B, and C. • 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 • Detection of analog input disconnection • Event linking by the ELC
12-bit D/A converter (R12DAb)		• 2 channels • 12-bit resolution • Output voltage: 0 V to AVCC0 • Capable of providing as a reference voltage for comparator • Event linking by the ELC
Comparator C (CMPC)		• 4 channels • Function to compare the reference voltage and the analog input voltage • Digital filtering
Temperature sensor		• 1 channel • Relative precision: ± 1.0°C • The voltage of the temperature is converted into a digital value by the 12-bit A/D converter.
Arithmetic unit for trigonometric functions (TFU)		• Sine, cosine, arctangent, $\sqrt{x^2 + y^2}$ Simultaneous calculation of sine and cosine Simultaneous calculation of arctangent and $\sqrt{x^2 + y^2}$

Table 1.1 Outline of Specifications (8/8)

Classification	Module/Function	Description
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.
Operating frequency		Up to 120 MHz
Power supply voltage		VCC = 2.7 to 5.5V AVCC0 = 3.0 to 5.5V (VCC ≤ AVCC0)
Operating temperature		D-version: –40 to +85°C G-version: –40 to +105°C
Package		144-pin LFQFP (PLQP0144KA-B) 100-pin LFQFP (PLQP0100KB-B) 80-pin LFQFP (PLQP0080KB-B) 64-pin LFQFP (PLQP0064KB-C) 48-pin LFQFP (PLQP0048KB-B)
Debugging interface		JTAG*3 and FINE interfaces

Note 1. When the realtime clock is not used, initialize the registers in the realtime clock according to description in section 27.6.7, Initialization Procedure When the Realtime Clock is Not to be Used. The realtime clock cannot be used in products with no sub-clock oscillator. At this time, disable the realtime clock according to description in section 27.6.7, Initialization Procedure When the Realtime Clock is Not to be Used.

Note 2. The product part number differs according to whether or not the CANFD actually supports the CAN FD protocol.

Note 3. The product part number differs according to whether or not the MCU includes a JTAG interface.

Table 1.2 Comparison of Functions for Different Packages

Functions		Products	RX660				
		Package	144-pin LFQFP	100-pin LFQFP	80-pin LFQFP	64-pin LFQFP	48-pin LFQFP
Code Flash Memory Capacity			1 Mbyte/512 Kbytes				
Data Flash Memory Capacity			32 Kbytes				
RAM			128 Kbytes				
External bus	External bus width		16 bits		Not available		
	Address Space		2 Mbytes × 4 areas		Not available		
DMA	DMA controller		Available				
	Data transfer controller		Available				
Oscillator	Main clock oscillator (MOSC)		Available				
	Sub-clock oscillator (SOSC)		Available/Not available				Not available
Timers	Multi-function timer pulse unit 3		Ch. 0 to 8		Ch. 0 to 7		Ch. 0 to 5, and 7
	Port output enable 3		Available				
	8-bit timers		Ch. 0 to 3				Ch. 0 to 2
	Compare match timer		Ch. 0 to 3				
	Compare match timer W		Ch. 0 and 1				
	Realtime clock		Available/Not available*1				Not available*1
	Watchdog timer		Available				
	Independent watchdog timer		Available				
Communication function	Serial communications interfaces (SCIk)		Ch. 0 to 9	Ch. 0 to 6, 8, and 9	Ch. 0, 1, 3 to 6, 8, and 9	Ch. 1, 3 to 6, 8, and 9	Ch. 1, 3 to 6, and 8
	Serial communications interfaces (SCIm)		Ch. 10 and 11				Ch. 10
	Serial communications interfaces (SCIh)		Ch. 12				
	Serial communications interfaces (RSCI)		Ch. 10 and 11				Ch. 10
	I2C bus interfaces (RIIC)		Ch. 0 to 2			Ch. 2	
	Serial peripheral interface (RSPI)		1 channel				
	CAN FD module (CANFD)		1 channel				
	Remote control signal receiver (REMC)		Ch. 0				
Analog	12-bit A/D converter		24 channels		17 channels	14 channels	10 channels
	Comparator C		4 channels				
	12-bit D/A converter*2	Number of channels	2 channels				
		Number of output pins	2 pins	2 pins/1 pin*3	2 pins	1 pin	Not available
	Temperature sensor		Available				
CRC calculator (CRCA)			Available				
Data operation circuit (DOCA)			Available				
Clock frequency accuracy measurement circuit (CAC)			Available				
Event link controller (ELC)			Available				
Off-board programming			Available		Not available		
Debugging interfaces	JTAG interface		Available/Not available		Not available		
	FINE interface		Available				

Note 1. The realtime clock cannot be used in products with no sub-clock oscillator. Disable the realtime clock according to description in section 27.6.7, Initialization Procedure When the Realtime Clock is Not to be Used.
Note 2. The 2-channel analog output of the D/A converters can be used as the input of the comparators in all packages.
Note 3. Products with a JTAG interface that are also in the 100-pin LFQFP have a single D/A converter channel.

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))	JTAG	Sub-clock oscillator	CANFD	Operating temperature (°C)
RX660 (D-version)	R5F56609ADFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56609BDFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56609CDFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56609DDFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56609EDFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Not available	Available*1	–40 to +85
	R5F56609FDFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Not available	Available	–40 to +85
	R5F56609GDFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Available	Available*1	–40 to +85
	R5F56609HDFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Available	Available	–40 to +85
	R5F56609ADFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56609BDFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56609CDFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56609DDFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56609EDFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Not available	Available*1	–40 to +85
	R5F56609FDFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Not available	Available	–40 to +85
	R5F56609GDFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Available	Available*1	–40 to +85
	R5F56609HDFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Available	Available	–40 to +85
	R5F56609ADFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56609BDFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56609CDFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56609DDFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56609ADFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56609BDFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56609CDFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56609DDFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56609ADFL	PLQP0048KB-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56609BDFL	PLQP0048KB-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56604ADFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56604BDFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56604CDFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56604DDFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56604EDFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Not available	Available*1	–40 to +85
	R5F56604FDFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Not available	Available	–40 to +85
	R5F56604GDFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Available	Available*1	–40 to +85
	R5F56604HDFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Available	Available	–40 to +85
	R5F56604ADFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56604BDFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56604CDFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56604DDFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Available	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))	JTAG	Sub-clock oscillator	CANFD	Operating temperature (°C)
RX660 (D-version)	R5F56604EDFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Not available	Available*1	–40 to +85
	R5F56604FDFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Not available	Available	–40 to +85
	R5F56604GDFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Available	Available*1	–40 to +85
	R5F56604HDFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Available	Available	–40 to +85
	R5F56604ADFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56604BDFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56604CDFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56604DDFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56604ADFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56604BDFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Not available	Available	–40 to +85
	R5F56604CDFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +85
	R5F56604DDFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Available	Available	–40 to +85
	R5F56604ADFL	PLQP0048KB-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +85
	R5F56604BDFL	PLQP0048KB-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +85
RX660 (G-version)	R5F56609AGFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56609BGFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56609CGFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56609DGFB	PLQP0144KA-B	1 M	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56609EGFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Not available	Available*1	–40 to +105
	R5F56609FGFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Not available	Available	–40 to +105
	R5F56609GGFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Available	Available*1	–40 to +105
	R5F56609HGFB	PLQP0144KA-B	1 M	128 K	32 K	Available	Available	Available	–40 to +105
	R5F56609AGFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56609BGFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56609CGFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56609DGFP	PLQP0100KB-B	1 M	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56609EGFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Not available	Available*1	–40 to +105
	R5F56609FGFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Not available	Available	–40 to +105
	R5F56609GGFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Available	Available*1	–40 to +105
	R5F56609HGFP	PLQP0100KB-B	1 M	128 K	32 K	Available	Available	Available	–40 to +105
	R5F56609AGFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56609BGFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56609CGFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56609DGFN	PLQP0080KB-B	1 M	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56609AGFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56609BGFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56609CGFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56609DGFM	PLQP0064KB-C	1 M	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56609AGFL	PLQP0048KB-B	1 M	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56609BGFL	PLQP0048KB-B	1 M	128 K	32 K	Not available	Not available	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))	JTAG	Sub-clock oscillator	CANFD	Operating temperature (°C)
RX660 (G-version)	R5F56604AGFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56604BGFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56604CGFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56604DGFB	PLQP0144KA-B	512 K	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56604EGFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Not available	Available*1	–40 to +105
	R5F56604FGFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Not available	Available	–40 to +105
	R5F56604GGFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Available	Available*1	–40 to +105
	R5F56604HGFB	PLQP0144KA-B	512 K	128 K	32 K	Available	Available	Available	–40 to +105
	R5F56604AGFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56604BGFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56604CGFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56604DGFP	PLQP0100KB-B	512 K	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56604EGFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Not available	Available*1	–40 to +105
	R5F56604FGFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Not available	Available	–40 to +105
	R5F56604GGFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Available	Available*1	–40 to +105
	R5F56604HGFP	PLQP0100KB-B	512 K	128 K	32 K	Available	Available	Available	–40 to +105
	R5F56604AGFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56604BGFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56604CGFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56604DGFN	PLQP0080KB-B	512 K	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56604AGFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56604BGFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Not available	Available	–40 to +105
	R5F56604CGFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Available	Available*1	–40 to +105
	R5F56604DGFM	PLQP0064KB-C	512 K	128 K	32 K	Not available	Available	Available	–40 to +105
	R5F56604AGFL	PLQP0048KB-B	512 K	128 K	32 K	Not available	Not available	Available*1	–40 to +105
	R5F56604BGFL	PLQP0048KB-B	512 K	128 K	32 K	Not available	Not available	Available	–40 to +105

Note 1. Products with this part number support only CAN 2.0 protocol.

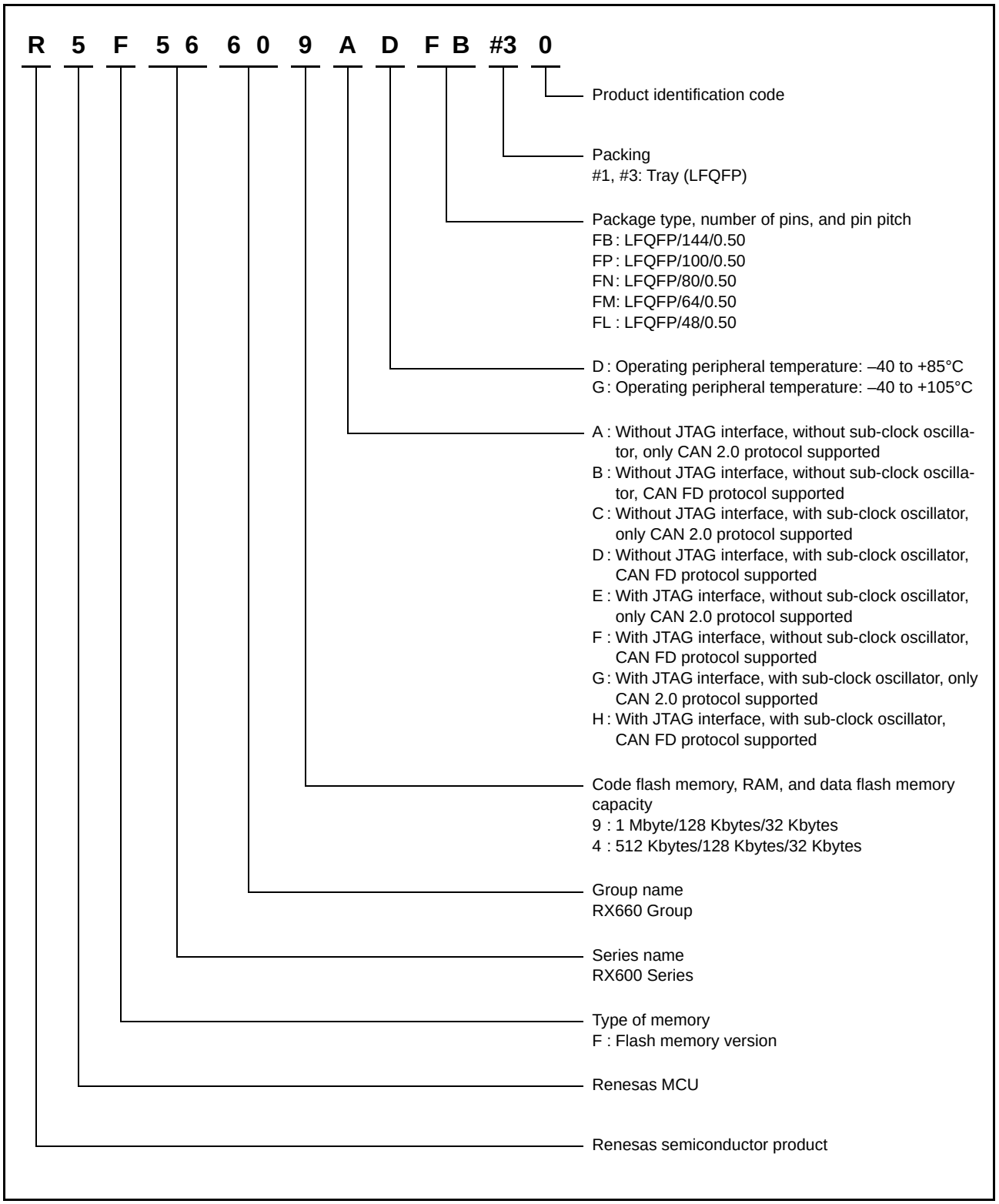
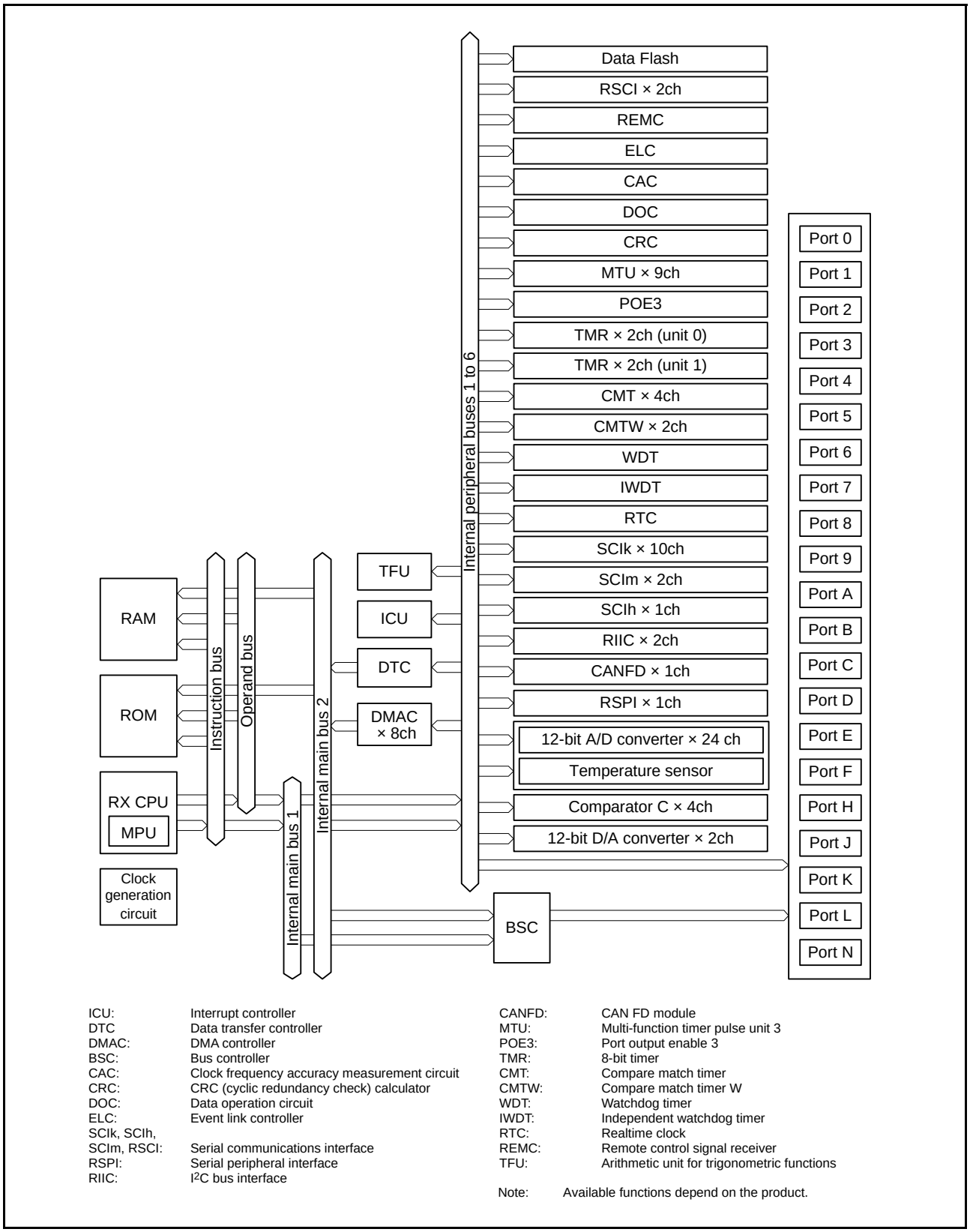


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ICU:	Interrupt controller
DTC	Data transfer controller
DMAC:	DMA controller
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:	Serial peripheral interface
RIIC:	I²C bus interface
CANFD:	CAN FD module
MTU:	Multi-function timer pulse unit 3
POE3:	Port output enable 3
TMR:	8-bit timer
CMT:	Compare match timer
CMTW:	Compare match timer W
WDT:	Watchdog timer
IWDT:	Independent watchdog timer
RTC:	Realtime clock
REMC:	Remote control signal receiver
TFU:	Arithmetic unit for trigonometric functions

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 (1/5)

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 0.47-µF smoothing capacitor used to stabilize the internal power supply. The capacitor should be placed close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
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.
	XCOUT	Output	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCIN	Input	
Clock frequency accuracy measurement	CACREF	Input	Reference clock input pin for the clock frequency accuracy measurement circuit
Operating mode control	MD	Input	Pin for setting the operating mode. For details on how to use this pin, see section 3.1, Operating Mode Types and Selection.
	UB	Input	User boot mode enable pin
System control	RES#	Input	Reset signal input pin. This LSI enters the reset state when this signal goes low.
	EMLE	Input	On-chip emulator enable pin while the JTAG pins are in use If the on-chip emulator is to be used, drive this pin to the high level. If the on-chip emulator is not to be used, drive this pin to the low level.
On-chip emulator	FINED	I/O	Fine interface pin
	TRST#	Input	On-chip emulator 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.
Address bus	A0 to A20	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

Table 1.4 Pin Functions (2/5)

Classifications	Pin Name	I/O	Description
Bus control	RD#	Output	Strobe signal which indicates that reading from the external bus interface space is in progress
	WR#	Output	Strobe signal which indicates that writing to the external bus interface space is in progress, in 1-write strobe mode
	WR0#, 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 CS3#	Output	Select signals for CS areas
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
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

Table 1.4 Pin Functions (3/5)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIk)	• Simple I2C mode		
	SSCL0 to SSCL9	I/O	Input/output pins for the I2C clock
	SSDA0 to SSDA9	I/O	Input/output pins for the I2C 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
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 I2C mode		
	SSCL12	I/O	Input/output pin for the I2C clock
	SSDA12	I/O	Input/output pin for the I2C 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 I2C mode		
	SSCL10, SSCL11	I/O	Input/output pins for the I2C clock
	SSDA10, SSDA11	I/O	Input/output pins for the I2C 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 (4/5)

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, SCL2	I/O	Input/output pins for clocks. Bus can be directly driven by the N-channel open drain
	SDA0, SDA2,	I/O	Input/output pins for data. Bus can be directly driven by the N-channel open drain
CAN FD module	CRX0	Input	Input pins
	CTX0	Output	Output pins
Serial peripheral interface	RSPCKA	I/O	Input/output pin for the RSPI clock
	MOSIA	I/O	Input/output pin for transmitting data from the RSPI master
	MISOA	I/O	Input/output pin for transmitting data from the RSPI slave
	SSLA0	I/O	Input/output pin to select the slave for the RSPI
	SSLA1 to SSLA3	Output	Output pins to select the slave for the RSPI
12-bit A/D converter	AN000 to AN023	Input	Input pins for the analog signals to be processed by the A/D converter
	ADST0	Output	Output pin for A/D conversion status.
	ADTRG0#	Input	Input pin for the external trigger signals that start the A/D conversion
12-bit D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter
Comparator C	COMP0 to COMP3	Output	Comparator detection result output pins
	CVREFC0 to CVREFC3	Input	Analog reference voltage supply pins for comparator C
	CMPC00, CMPC10, CMPC20, CMPC30	Input	Analog input pins for CMPCn0 (n = 0 to 3)
Realtime clock	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins
Remote control signal receiver (REMC)	PMC0	Input	Input pin for external pulse signal

Table 1.4 Pin Functions (5/5)

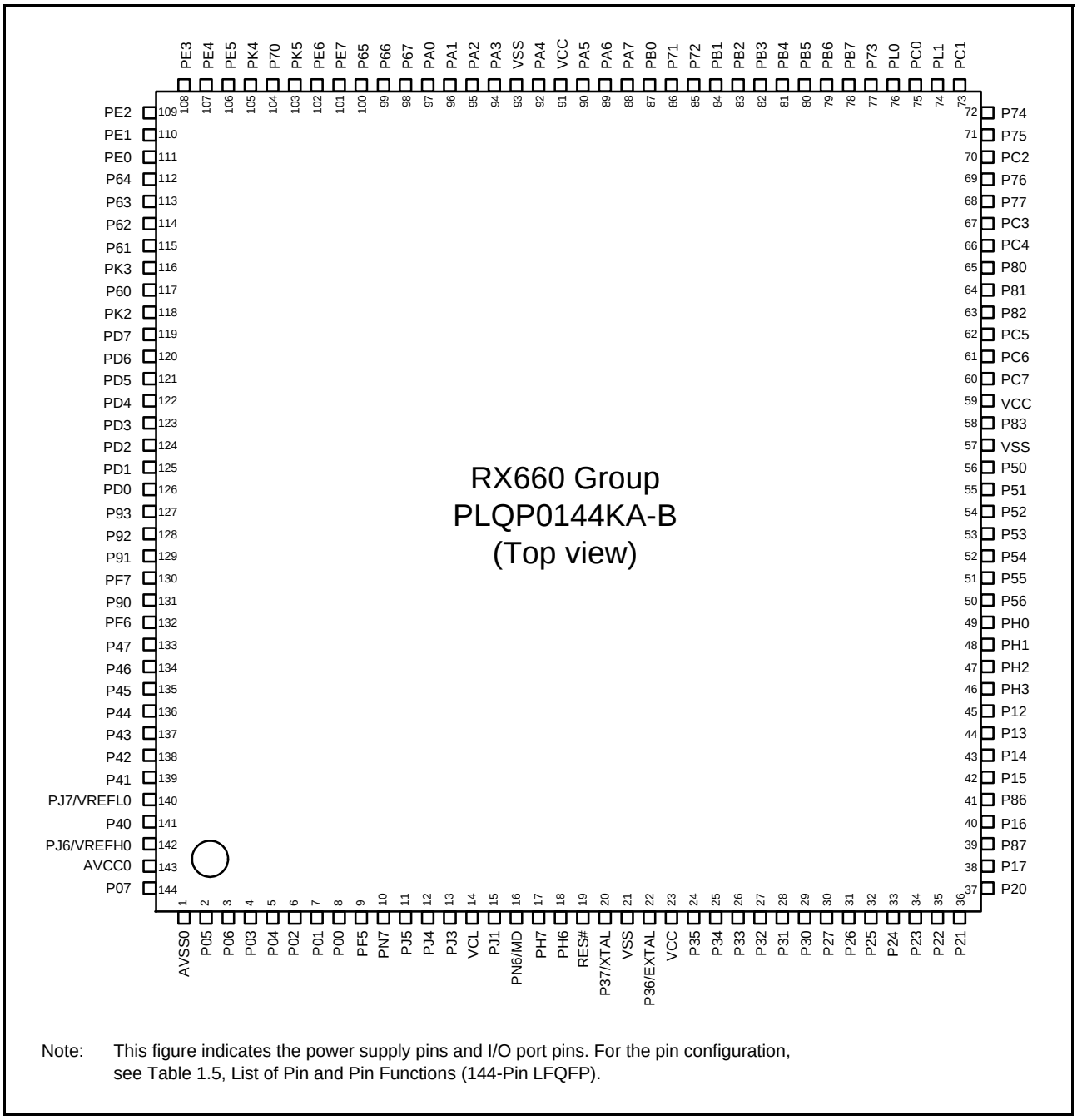
Classifications	Pin Name	I/O	Description
Analog power supply	AVCC0*1	Input	Analog voltage supply pin for the 12-bit A/D converter, 12-bit D/A converter, comparator C, and temperature sensor. Connect the pin to AVSS0 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	AVSS0*1	Input	Analog ground pin for the 12-bit A/D converter, 12-bit D/A converter, comparator C, and temperature sensor. 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. Connect the pin to VREFL0 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin. If the 12-bit A/D converter is not to be used, set this pin to its general-purpose function.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter. Connect the pin to VREFH0 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin. If the 12-bit A/D converter is not to be used, set this pin to its general-purpose function.
I/O ports	P00 to P07	I/O	8-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, PF6, PF7	I/O	3-bit input/output pins
	PH0 to PH3, PH6, PH7	I/O	6-bit input/output pins
	PJ1, PJ3 to PJ7	I/O	6-bit input/output pins
	PK2 to PK5	I/O	4-bit input/output pins
	PL0, PL1	I/O	2-bit input/output pins
	PN6, PN7	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.
- When the pin functions have “-DS” appended to their names, they can also be used as triggers for release from deep software standby.

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

1.5 Pin Assignments

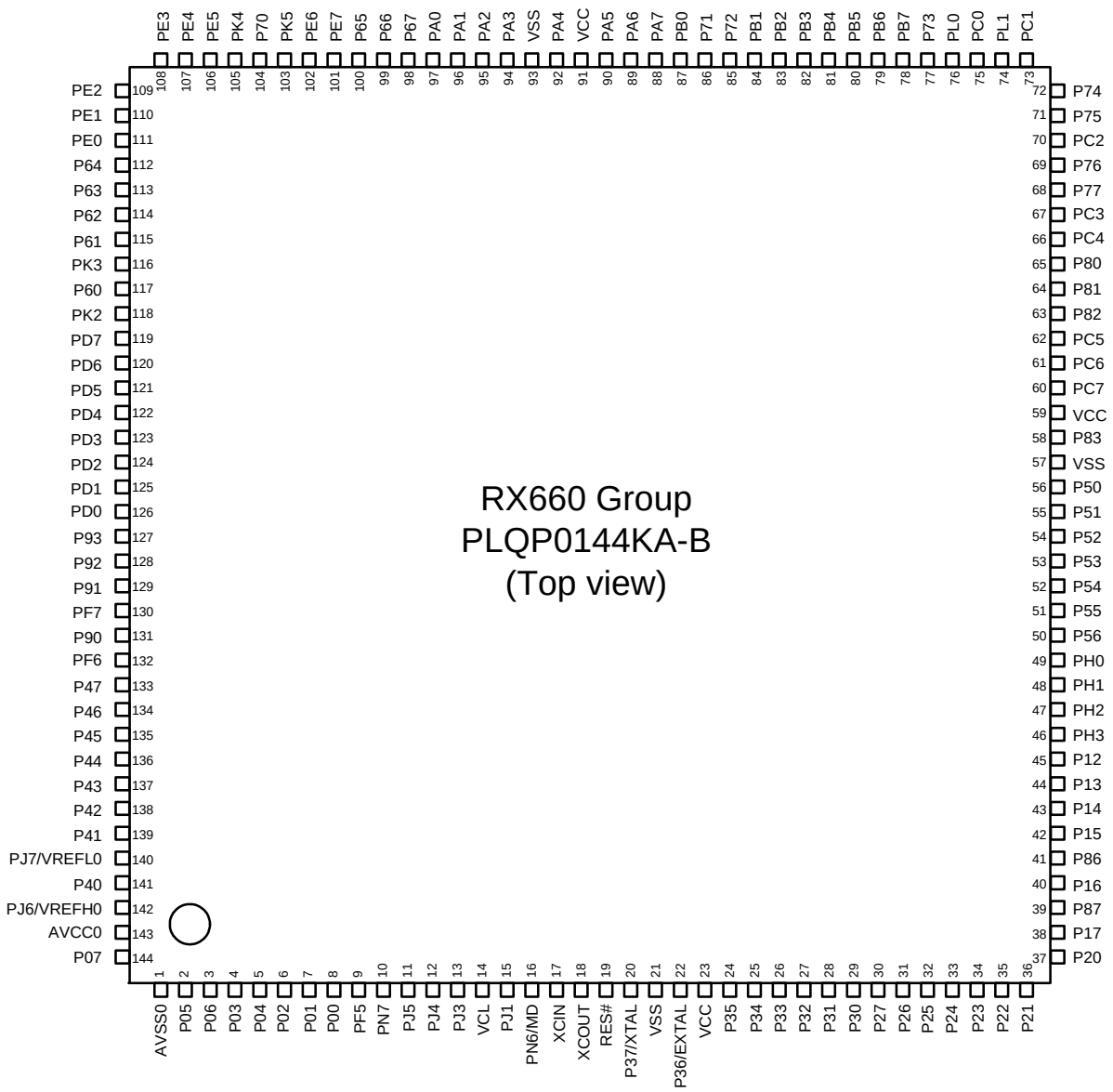
1.5.1 144-Pin LFQFP (without JTAG Interface, without Sub-clock Oscillator)



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 (144-Pin LFQFP).

Figure 1.3 Pin Assignment (144-Pin LFQFP (without JTAG Interface, without Sub-clock Oscillator))

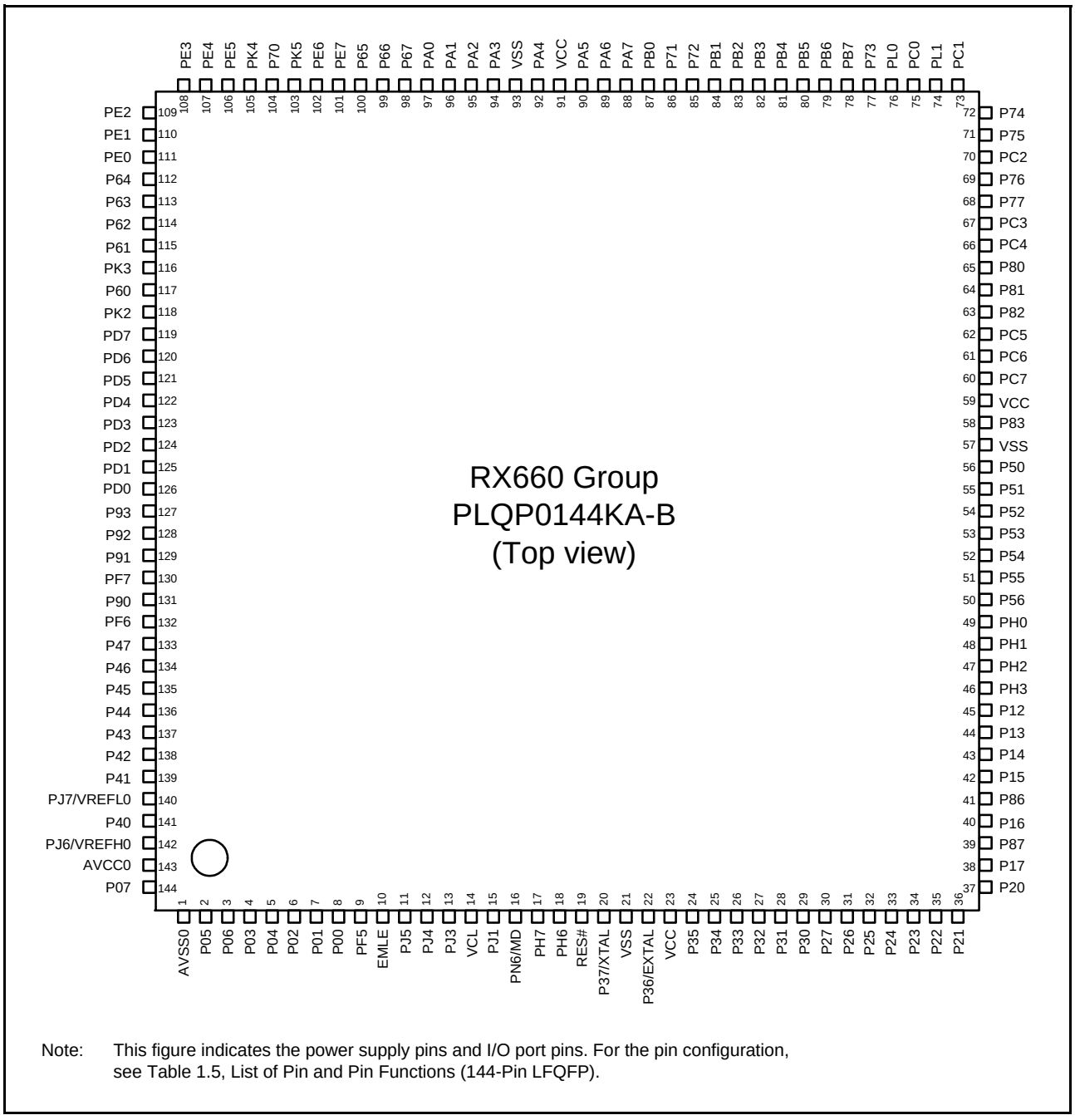
1.5.2 144-Pin LFQFP (without JTAG Interface, with Sub-clock Oscillator)



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 (144-Pin LFQFP).

Figure 1.4 Pin Assignment (144-Pin LFQFP (without JTAG Interface, with Sub-clock Oscillator))

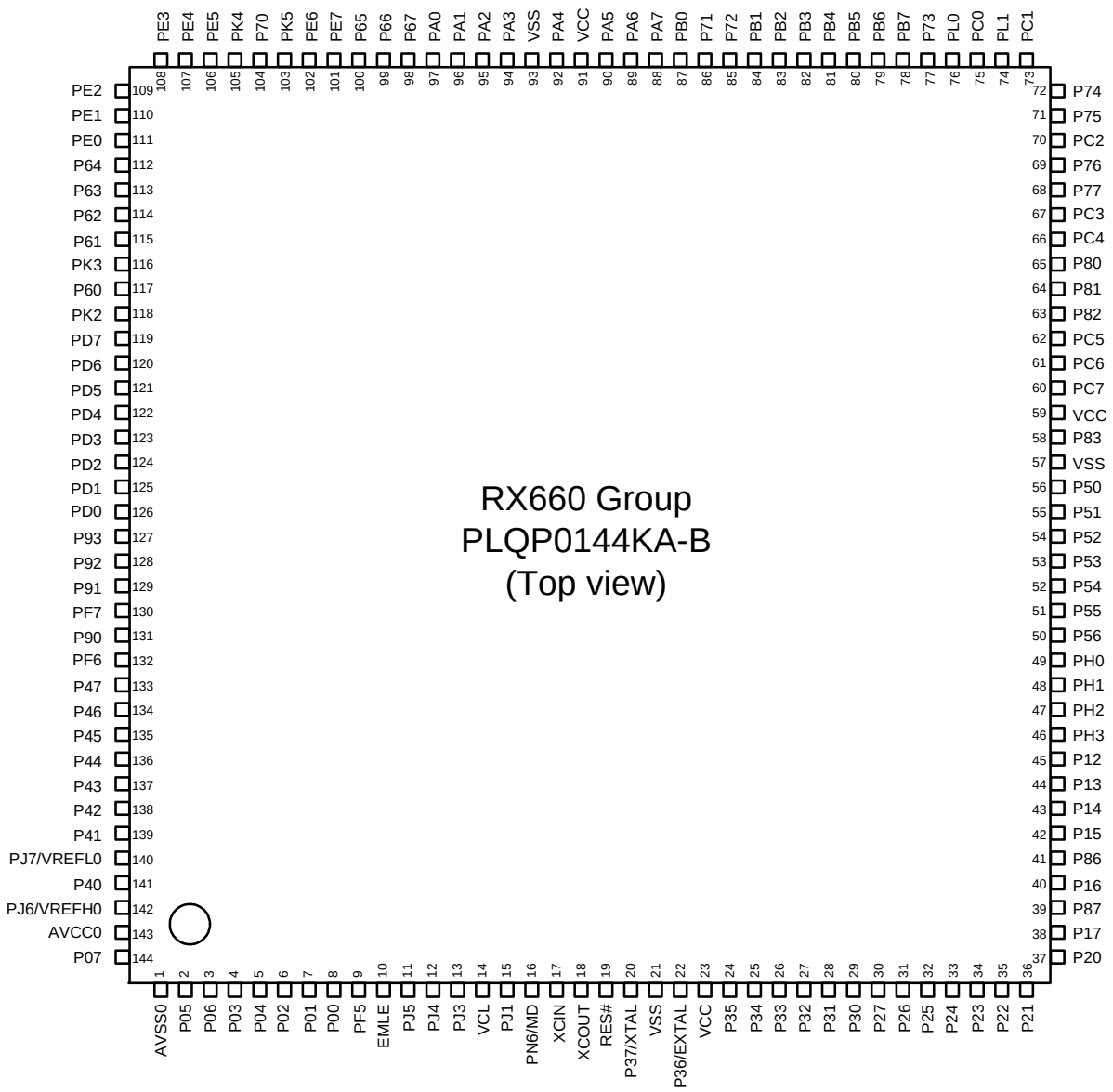
1.5.3 144-Pin LFQFP (with JTAG Interface, without Sub-clock Oscillator)



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 (144-Pin LFQFP).

Figure 1.5 Pin Assignment (144-Pin LFQFP (with JTAG Interface, without Sub-clock Oscillator))

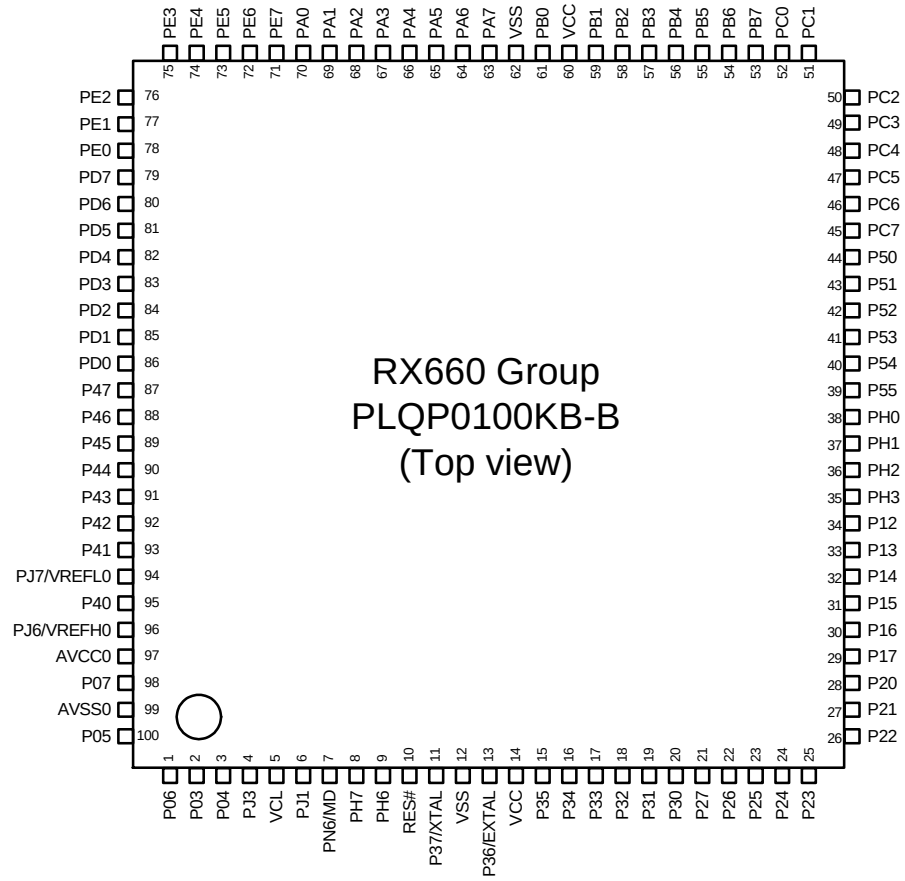
1.5.4 144-Pin LFQFP (with JTAG Interface, with Sub-clock Oscillator)



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 (144-Pin LFQFP).

Figure 1.6 Pin Assignment (144-Pin LFQFP (with JTAG Interface, with Sub-clock Oscillator))

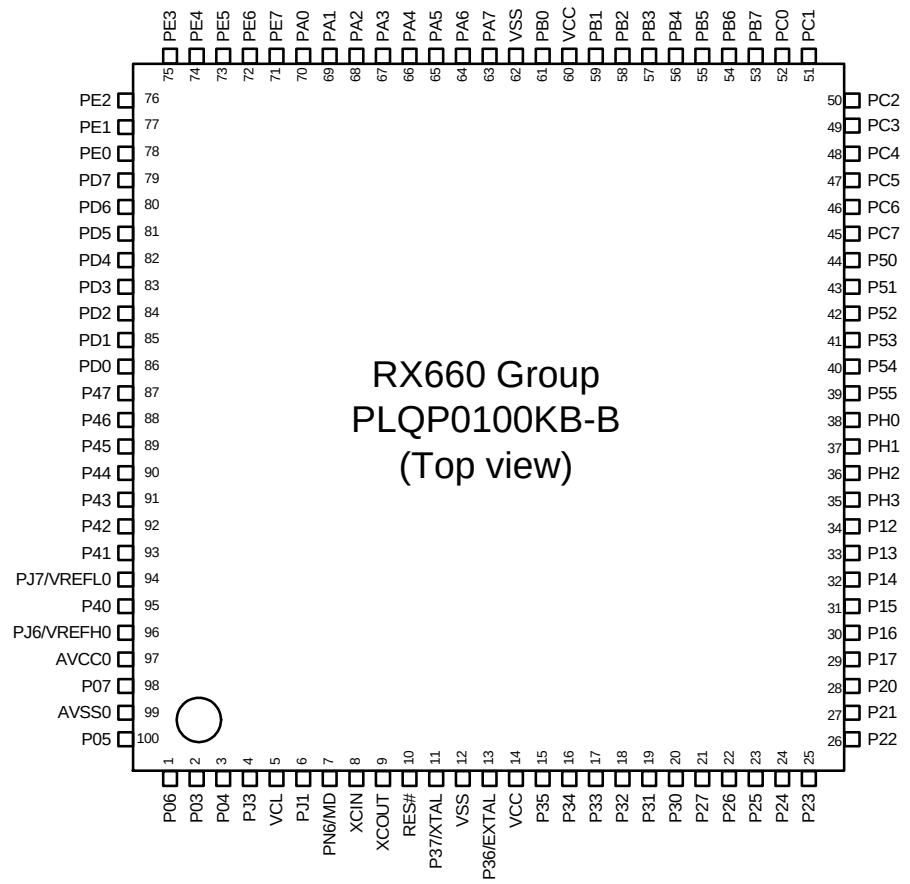
1.5.5 100-Pin LFQFP (without JTAG Interface, without Sub-clock Oscillator)



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 (100-Pin LFQFP).

Figure 1.7 Pin Assignment (100-Pin LFQFP (without JTAG Interface, without Sub-clock Oscillator))

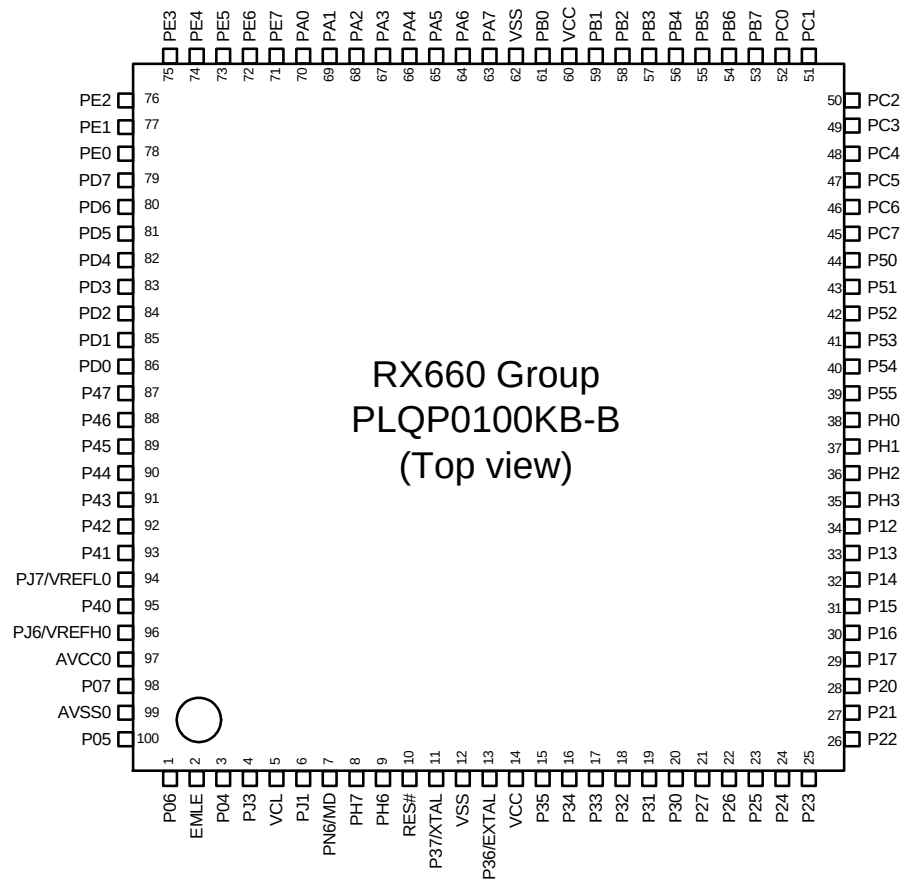
1.5.6 100-Pin LFQFP (without JTAG Interface, with Sub-clock Oscillator)



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 (100-Pin LFQFP).

Figure 1.8 Pin Assignment (100-Pin LFQFP (without JTAG Interface, with Sub-clock Oscillator))

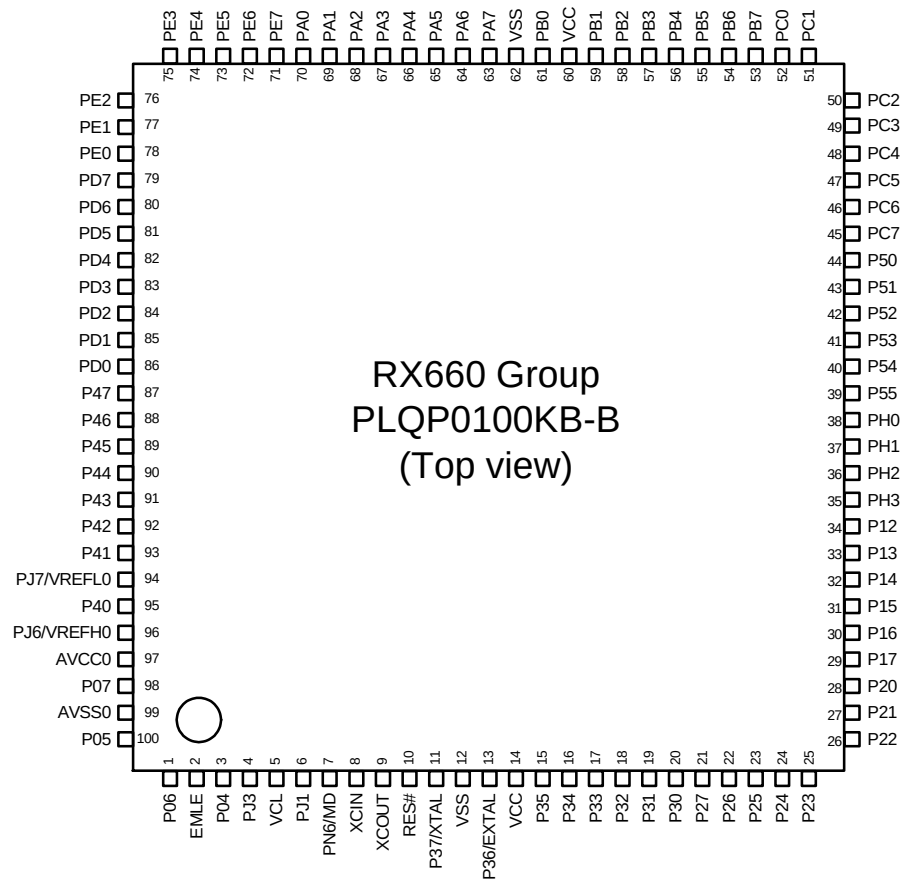
1.5.7 100-Pin LFQFP (with JTAG Interface, without Sub-clock Oscillator)



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 (100-Pin LFQFP).

Figure 1.9 Pin Assignment (100-Pin LFQFP (with JTAG Interface, without Sub-clock Oscillator))

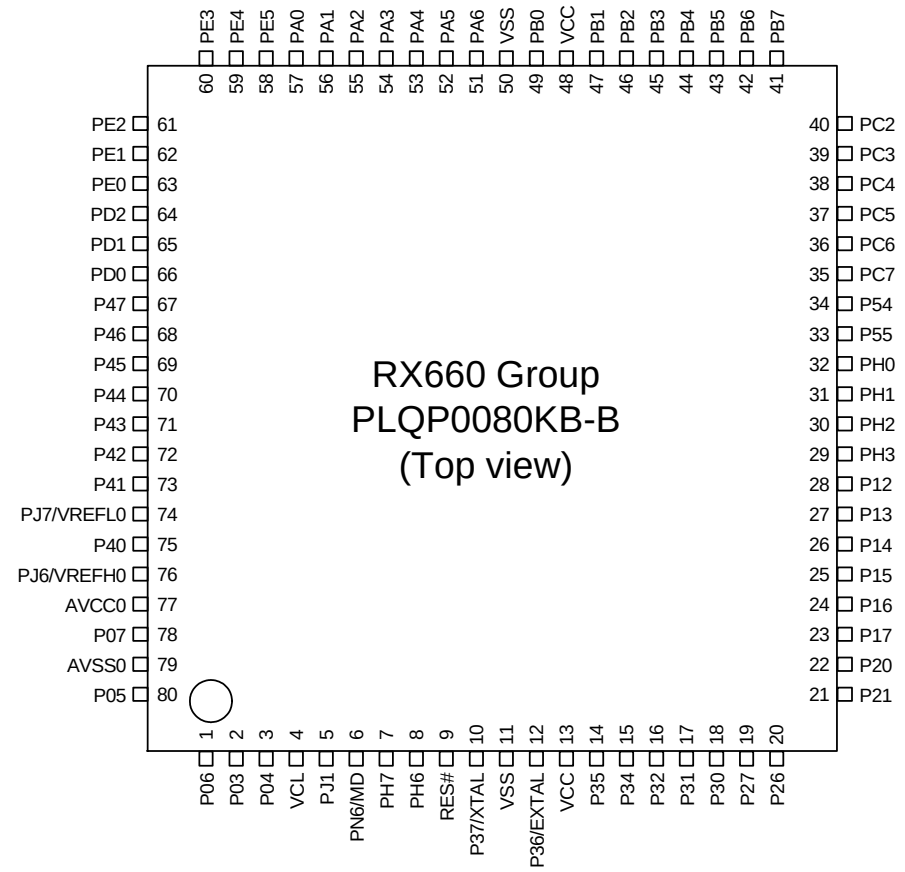
1.5.8 100-Pin LFQFP (with JTAG Interface, with Sub-clock Oscillator)



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 (100-Pin LFQFP).

Figure 1.10 Pin Assignment (100-Pin LFQFP (with JTAG Interface, with Sub-clock Oscillator))

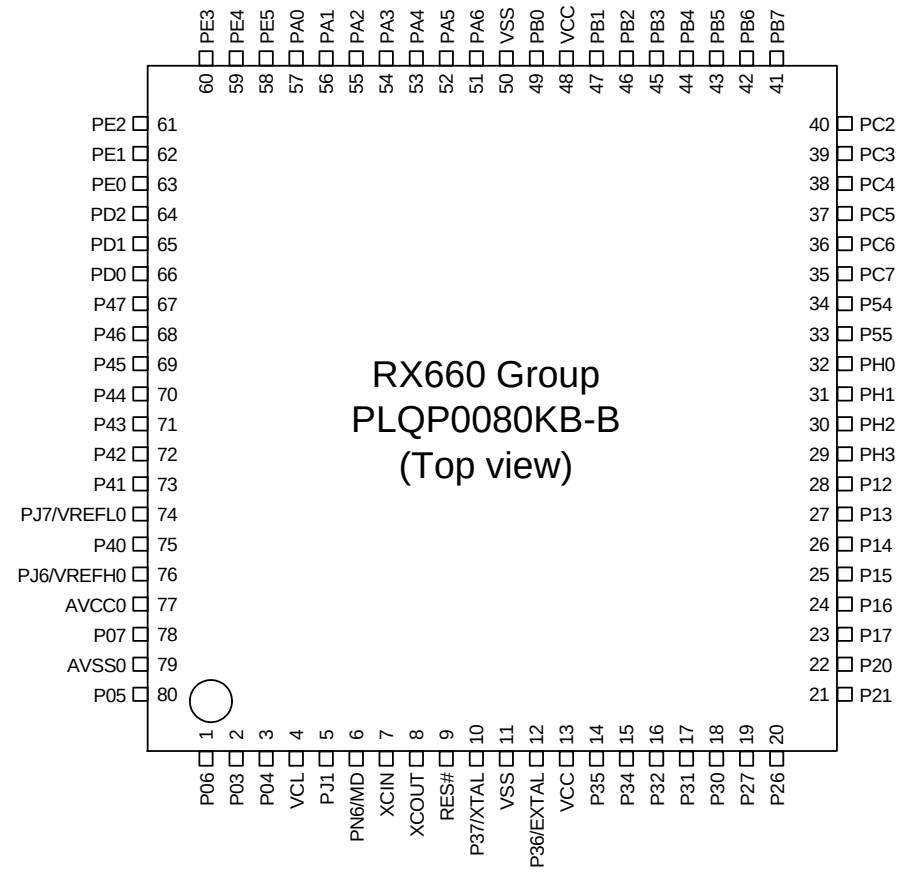
1.5.9 80-Pin LFQFP (without Sub-clock Oscillator)



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 (80-Pin LFQFP).

Figure 1.11 Pin Assignment (80-Pin LFQFP (without Sub-clock Oscillator))

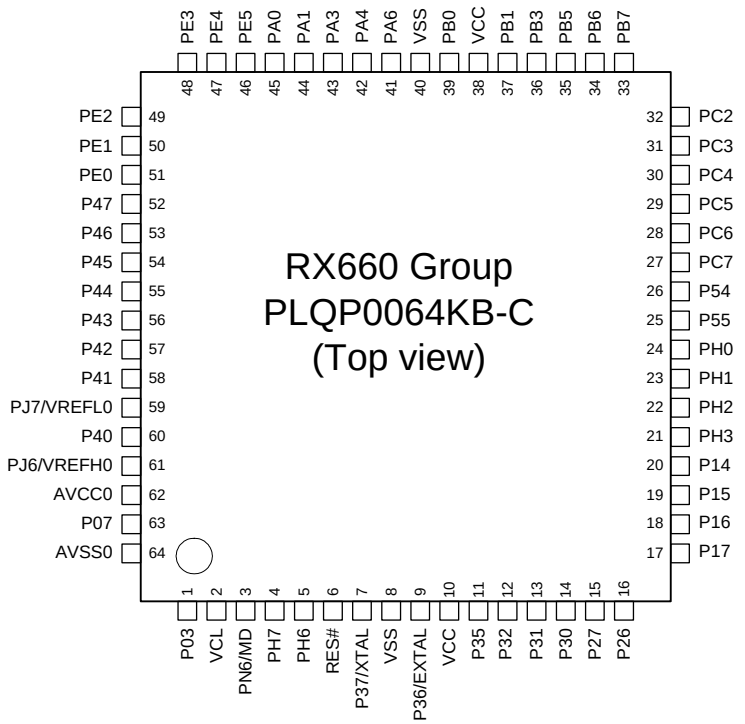
1.5.10 80-Pin LFQFP (with Sub-clock Oscillator)



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 (80-Pin LFQFP).

Figure 1.12 Pin Assignment (80-Pin LFQFP (with Sub-clock Oscillator))

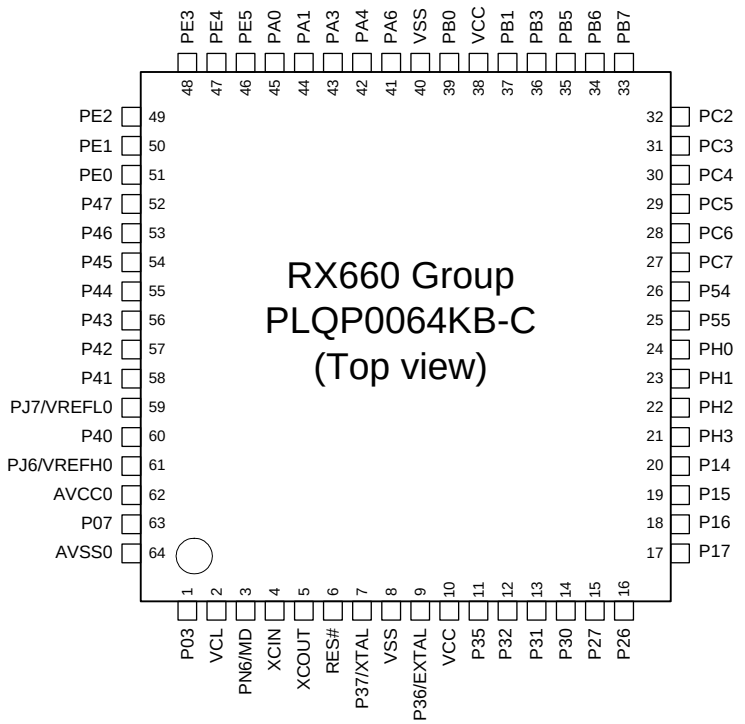
1.5.11 64-Pin LFQFP (without Sub-clock Oscillator)



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 (64-Pin LFQFP).

Figure 1.13 Pin Assignment (64-Pin LFQFP (without Sub-clock Oscillator))

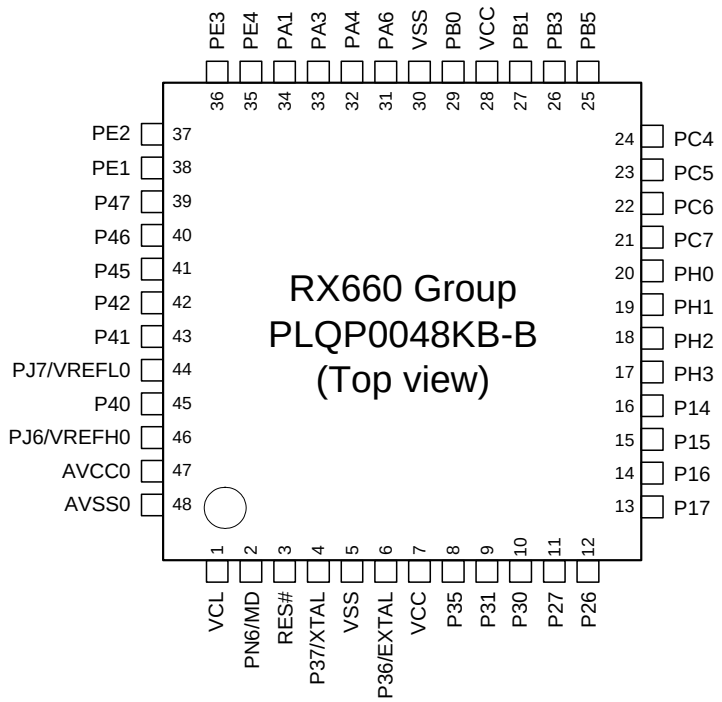
1.5.12 64-Pin LFQFP (with Sub-clock Oscillator)



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 (64-Pin LFQFP).

Figure 1.14 Pin Assignment (64-Pin LFQFP (with Sub-clock Oscillator))

1.5.13 48-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 (48-Pin LFQFP).

Figure 1.15 Pin Assignment (48-Pin LFQFP)

1.6 List of Pin and Pin Functions

1.6.1 144-Pin LFQFP

Table 1.5 List of Pin and Pin Functions (144-Pin LFQFP) (1/6)

Pin No.	Power Supply Clock System Control	I/O Port	Bus	Timer	Communication	Interrupt	A/D, D/A, CMPC
144-Pin LFQFP				(MTU, TMR, RTC, POE, CAC, CMTW)	(SCI, RSCI, RSPI, RIIC, CANFD, REMC)	(IRQ, NMI)	
1	AVSS0						
2		P05				IRQ13	DA1
3		P06					
4		P03				IRQ11	DA0
5		P04					
6		P02		TMCI1	SCK6	IRQ10	
7		P01		TMCI0	RXD6/SMISO6/ SSCL6	IRQ9	
8		P00		TMRI0	TXD6/SMOSI6/ SSDA6	IRQ8	
9		PF5				IRQ4	
10	EMLE*1	PN7*2					
11		PJ5		POE8#	CTS2#/RTS2#/SS2#	IRQ13	
12		PJ4					
13		PJ3		MTIOC3C	CTS6#/RTS6#/SS6#/ CTS0#/RTS0#/SS0#	IRQ11	
14	VCL						
15		PJ1		MTIOC3A			
16	MD/FINED	PN6					
17	XCIN*3	PH7*4					
18	XCOUT*3	PH6*4					
19	RES#						
20	XTAL	P37				IRQ4	
21	VSS						
22	EXTAL	P36				IRQ5	
23	VCC						
24		P35				NMI	
25	TRST#*1	P34		MTIOC0A/TMCI3/ POE10#	SCK6/SCK0	IRQ4	
26		P33		MTIOC0D/TMRI3/ POE4#/POE11#	RXD6/SMISO6/ SSCL6/RXD0/ SMISO0/SSCL0/ CRX0-A	IRQ3-DS	
27		P32		MTIOC0C/TMO3/ RTCIC2*5/RTCOUT*5/ POE0#/POE10#	TXD6/SMOSI6/ SSDA6/TXD0/ SMOSI0/SSDA0/ CTX0-A	IRQ2-DS	
28	TMS*1	P31		MTIOC4D/TMCI2/ RTCIC1*5	CTS1#/RTS1#/SS1#	IRQ1-DS	
29	TDI*1	P30		MTIOC4B/TMRI3/ RTCIC0*5/POE8#	RXD1/SMISO1/ SSCL1	IRQ0-DS	COMP3
30	TCK*1	P27	CS3#	MTIOC2B/TMCI3	SCK1	IRQ7	CVREFC3

Table 1.5 List of Pin and Pin Functions (144-Pin LFQFP) (2/6)

Pin No. 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
31	TDO*1	P26	CS2#	MTIOC2A/TMO1	TXD1/SMOSI1/ SSDA1/CTS3#/ RTS3#/SS3#	IRQ6	CMPC30
32		P25	CS1#	MTIOC4C/MTCLKB	RXD3/SMISO3/ SSCL3	IRQ5	ADTRG0#
33		P24	CS0#	MTIOC4A/MTCLKA/ TMRI1	SCK3	IRQ12	
34		P23		MTIOC3D/MTCLKD	TXD3/SMOSI3/ SSDA3/CTS0#/ RTS0#/SS0#	IRQ3	
35		P22		MTIOC3B/MTCLKC/ TMO0	SCK0	IRQ15	
36		P21		MTIOC1B/TMCI0/ MTIOC4A	RXD0/SMISO0/ SSCL0	IRQ9	
37		P20		MTIOC1A/TMRI0	TXD0/SMOSI0/ SSDA0	IRQ8	
38		P17		MTIOC3A/MTIOC3B/ TMO1/POE8#/ MTIOC4B	SCK1/TXD3/SMOSI3/ SSDA3/MISOA-C/ SDA2	IRQ7	COMP2
39		P87		MTIOC4C	SMOSI10/SSDA10/ TXD10/TXD010-B/ SMOSI010-B/ SSDA010-B	IRQ15	
40		P16		MTIOC3C/MTIOC3D/ TMO2/RTCOUT*5	TXD1/SMOSI1/ SSDA1/RXD3/ SMISO3/SSCL3/ MOSIA-C/SCL2	IRQ6	ADTRG0#
41		P86		MTIOC4D	SMISO10/SSCL10/ RXD10/RXD010-B/ SMISO010-B/ SSCL010-B	IRQ14	
42		P15		MTIOC0B/MTCLKB/ TMCI2	RXD1/SMISO1/ SSCL1/SCK3/ CRX0-C	IRQ5	CMPC20
43		P14		MTIOC3A/MTCLKA/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0-C	IRQ4	CVREFC2
44		P13		MTIOC0B/TMO3	TXD2/SMOSI2/ SSDA2/SDA0	IRQ3	
45		P12		MTIC5U/TMCI1	RXD2/SMISO2/ SSCL2/SCL0	IRQ2	
46		PH3		MTIOC4D/TMCI0			
47		PH2		MTIOC4C/TMRI0/ TOC1		IRQ1	
48		PH1		MTIOC3D/TMO0/TIC1		IRQ0	ADST0
49		PH0		MTIOC3B/CACREF			ADTRG0#
50		P56		MTIOC3C	SCK7	IRQ6	
51	TRDATA3*1	P55	D0[A0/D0]/ WAIT#	MTIOC4D/MTIOC4A/ TMO3	TXD7/SMOSI7/ SSDA7/CRX0-D	IRQ10	
52	TRDATA2*1	P54	ALE/ D1[A1/D1]	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX0-D	IRQ4	
53		P53	BCLK		PMC0	IRQ3	
54		P52	RD#		RXD2/SMISO2/ SSCL2	IRQ2	

Table 1.5 List of Pin and Pin Functions (144-Pin LFQFP) (3/6)

Pin No. 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
55		P51	WR1#/BC1#/ WAIT#		SCK2/PMC0	IRQ1	
56		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2	IRQ0	
57	VSS						
58	TRCLK*1	P83		MTIOC4C	SCK10/SS10#/ CTS10#/SCK010-B/ CTS010#-A/SS010#-A	IRQ3	
59	VCC						
60	UB	PC7	CS0#	MTIOC3A/MTCLKB/ TMO2/CACREF/TOC0	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ TXD010-C/ SMOSI010-C/ SSDA010-C/MISOA-A	IRQ14	
61		PC6	D2[A2/D2]/ CS1#	MTIOC3C/MTCLKA/ TMCI2/TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ RXD010-C/ SMISO010-C/ SSCL010-C/MOSIA-A	IRQ13	
62		PC5	D3[A3/D3]/ CS2#/WAIT#	MTIOC3B/MTCLKD/ TMRI2/MTIOC0C	SCK8/SCK10/ SCK010-C/ RSPCKA-A/PMC0	IRQ5	
63	TRSYNC*1	P82		MTIOC4A	SMOSI10/SSDA10/ TXD10/TXD010-A/ SMOSI010-A/ SSDA010-A	IRQ2	
64	TRDATA1*1	P81		MTIOC3D	SMISO10/SSCL10/ RXD10/RXD010-A/ SMISO010-A/ SSCL010-A	IRQ9	
65	TRDATA0*1	P80		MTIOC3B	SCK10/RTS10#/ SCK010-A/ RTS010#-A/DE010-A	IRQ8	
66		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TMCI1/POE0#/ MTIOC0A	SCK5/CTS8#/RTS8#/ SS8#/SS10#/CTS10#/ RTS10#/CTS010#-B/ RTS010#-B/ SS010#-B/DE010-B/ SSLA0-A/PMC0	IRQ12	
67		PC3	A19	MTIOC4D	TXD5/SMOSI5/ SSDA5/PMC0	IRQ11	
68	TRDATA7*1	P77			SMOSI11/SSDA11/ TXD11/TXD011-A/ SMOSI011-A/ SSDA011-A	IRQ7	
69	TRDATA6*1	P76			SMISO11/SSCL11/ RXD11/RXD011-A/ SMISO011-A/ SSCL011-A	IRQ14	
70		PC2	A18	MTIOC4B	RXD5/SMISO5/ SSCL5/TXDB011-A/ SSLA3-A	IRQ10	
71	TRSYNC1*1	P75			SCK11/RTS11#/ SCK011-A/ RTS011#-A/DE011-A	IRQ13	

Table 1.5 List of Pin and Pin Functions (144-Pin LFQFP) (4/6)

Pin No. 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
72	TRDATA5*1	P74	A20		SS11#/CTS11#/ CTS011#-A/SS011#-A	IRQ12	
73		PC1	A17	MTIOC3A	SCK5/TXD011-C/ SMOSI011-C/ SSDA011-C/ TXDA011-C/SSLA2-A	IRQ12	
74		PL1					
75		PC0	A16	MTIOC3C	CTS5#/RTS5#/SS5#/ RXD011-C/ SMISO011-C/ SSCL011-C/SSLA1-A	IRQ14	
76		PL0					
77	TRDATA4*1	P73	CS3#			IRQ8	
78		PB7	A15	MTIOC3B	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ TXD011-B/ SMOSI011-B/ SSDA011-B	IRQ15	
79		PB6	A14	MTIOC3D	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ RXD011-B/ SMISO011-B/ SSCL011-B	IRQ6	
80		PB5	A13	MTIOC2A/MTIOC1B/ TMRI1/POE4#/TOC2	SCK9/SCK11/ SCK011-B	IRQ13	
81		PB4	A12		CTS9#/RTS9#/SS9#/ SS11#/CTS11#/ RTS11#/CTS011#-B/ RTS011#-B/ SS011#-B/DE011-B	IRQ4	
82		PB3	A11	MTIOC0A/MTIOC4A/ TMO0/POE11#/TIC2	SCK4/SCK6/PMC0	IRQ3	
83		PB2	A10		CTS4#/RTS4#/SS4#/ CTS6#/RTS6#/SS6#	IRQ2	
84		PB1	A9	MTIOC0C/MTIOC4C/ TMCI0	TXD4/SMOSI4/ SSDA4/TXD6/ SMOSI6/SSDA6	IRQ4-DS	COMP1
85		P72	A19/CS2#			IRQ10	
86		P71	A18/CS1#			IRQ1	
87		PB0	A8	MTIC5W/MTIOC3D	RXD4/SMISO4/ SSCL4/RXD6/ SMISO6/SSCL6/ RSPCKA-C	IRQ12	
88		PA7	A7		MISOA-B	IRQ7	
89		PA6	A6	MTIC5V/MTCLKB/ TMCI3/POE10#/ MTIOC3D/MTIOC6B	CTS5#/RTS5#/SS5#/ CTS12#/RTS12#/ SS12#/MOSIA-B	IRQ14	
90		PA5	A5	MTIOC6B	RSPCKA-B	IRQ5	
91	VCC						

Table 1.5 List of Pin and Pin Functions (144-Pin LFQFP) (5/6)

Pin No. 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
92		PA4	A4	MTIC5U/MTCLKA/ TMRI0/MTIOC4C/ MTIOC7C	TXD5/SMOSI5/ SSDA5/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12/ SSLA0-B	IRQ5-DS	CVREFC1/ ADST0
93	VSS						
94		PA3	A3	MTIOC0D/MTCLKD/ MTIC5V/MTIOC4D	RXD5/SMISO5/ SSCL5	IRQ6-DS	CMPC10
95		PA2	A2	MTIOC7A	RXD5/SMISO5/ SSCL5/RXD12/ SMISO12/SSCL12/ RXDX12/SSLA3-B	IRQ10	
96		PA1	A1	MTIOC0B/MTCLKC/ MTIOC7B/MTIOC3B	SCK5/SCK12/ SSLA2-B	IRQ11	ADTRG0#
97		PA0	BC0#/A0	MTIOC4A/CACREF/ MTIOC6D	SSLA1-B	IRQ0	
98		P67		MTIOC7C		IRQ15	
99		P66		MTIOC7D		IRQ14	
100		P65				IRQ13	
101		PE7	D15[A15/D15]/ D7[A7/D7]	MTIOC6A/TOC1		IRQ7	AN015
102		PE6	D14[A14/D14]/ D6[A6/D6]	MTIOC6C/TIC1	CTS4#/RTS4#/SS4#	IRQ6	AN014
103		PK5			TXD4/SMOSI4/ SSDA4		
104		P70			SCK4	IRQ0	
105		PK4			RXD4/SMISO4/ SSCL4		
106		PE5	D13[A13/D13]/ D5[A5/D5]	MTIOC4C/MTIOC2B		IRQ5	AN013/ COMP0
107		PE4	D12[A12/D12]/ D4[A4/D4]	MTIOC4D/MTIOC1A/ MTIOC4A/MTIOC7D		IRQ12	AN012
108		PE3	D11[A11/D11]/ D3[A3/D3]	MTIOC4B/POE8#/ MTIOC1B/TOC3	CTS12#/RTS12#/ SS12#	IRQ11	AN011
109		PE2	D10[A10/D10]/ D2[A2/D2]	MTIOC4A/MTIOC7A/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12	IRQ7-DS	AN010/ CVREFC0
110		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	IRQ9	AN009/ CMPC00
111		PE0	D8[A8/D8]/ D0[A0/D0]	MTIOC3D	SCK12	IRQ8	AN008
112		P64	D3[A3/D3]			IRQ4	
113		P63	D2[A2/D2]/ CS3#			IRQ3	
114		P62	D1[A1/D1]/ CS2#			IRQ2	
115		P61	D0[A0/D0]/ CS1#		CTS9#/RTS9#/SS9#	IRQ1	
116		PK3			RXD9/SMISO9/ SSCL9		
117		P60	CS0#		SCK9	IRQ0	
118		PK2			TXD9/SMOSI9/ SSDA9		

Table 1.5 List of Pin and Pin Functions (144-Pin LFQFP) (6/6)

Pin No. 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
119	TRDATA3*1	PD7	D7[A7/D7]	MTIC5U/POE0#		IRQ7	AN023
120	TRDATA2*1	PD6	D6[A6/D6]	MTIC5V/POE4#/ MTIOC8A		IRQ6	AN022
121	TRCLK*1	PD5	D5[A5/D5]	MTIC5W/POE10#/ MTIOC8C		IRQ5	AN021
122	TRSYNC*1	PD4	D4[A4/D4]	POE11#/MTIOC8B		IRQ4	AN020
123	TRDATA1*1	PD3	D3[A3/D3]	POE8#/MTIOC8D/ TOC2		IRQ3	AN019
124	TRDATA0*1	PD2	D2[A2/D2]	MTIOC4D/TIC2	CRX0-B	IRQ2	AN018
125	TRDATA7*1	PD1	D1[A1/D1]	MTIOC4B/POE0#	CTX0-B	IRQ1	AN017
126	TRDATA6*1	PD0	D0[A0/D0]	POE4#		IRQ0	AN016
127	TRSYNC1*1	P93	A19	POE0#	CTS7#/RTS7#/SS7#	IRQ11	
128	TRDATA5*1	P92	A18	POE4#	RXD7/SMISO7/ SSCL7	IRQ10	
129	TRDATA4*1	P91	A17		SCK7	IRQ9	
130		PF7					
131		P90	A16		TXD7/SMOSI7/ SSDA7	IRQ0	
132		PF6					
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	PJ7					
141		P40				IRQ8-DS	AN000
142	VREFH0	PJ6					
143	AVCC0						
144		P07				IRQ15	ADTRG0#

Note 1. This pin function is not provided for products with no JTAG interface.
Note 2. This pin function is not provided for products with a JTAG interface.
Note 3. This pin function is not provided for products with no sub-clock oscillator.
Note 4. This pin function is not provided for products with a sub-clock oscillator.
Note 5. This pin function is not available in products with no sub-clock oscillator.

1.6.2 100-Pin LFQFP

Table 1.6 List of Pin and Pin Functions (100-Pin LFQFP) (1/5)

Pin No. 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
1		P06					
2	EMLE*1	P03*2				IRQ11*2	DA0*2
3		P04					
4		PJ3		MTIOC3C	CTS6#/RTS6#/SS6#/ CTS0#/RTS0#/SS0#	IRQ11	
5	VCL						
6		PJ1		MTIOC3A			
7	MD/FINED	PN6					
8	XCIN*3	PH7*4					
9	XCOUT*3	PH6*4					
10	RES#						
11	XTAL	P37				IRQ4	
12	VSS						
13	EXTAL	P36				IRQ5	
14	VCC						
15		P35				NMI	
16	TRST#*1	P34		MTIOC0A/TMCI3/ POE10#	SCK6/SCK0	IRQ4	
17		P33		MTIOC0D/TMRI3/ POE4#/POE11#	RXD6/SMISO6/ SSCL6/RXD0/ SMISO0/SSCL0/ CRX0-A	IRQ3-DS	
18		P32		MTIOC0C/TMO3/ RTCIC2*5/RTCOUT*5/ POE0#/POE10#	TXD6/SMOSI6/ SSDA6/TXD0/ SMOSI0/SSDA0/ CTX0-A	IRQ2-DS	
19	TMS*1	P31		MTIOC4D/TMCI2/ RTCIC1*5	CTS1#/RTS1#/SS1#	IRQ1-DS	
20	TDI*1	P30		MTIOC4B/TMRI3/ RTCIC0*5/POE8#	RXD1/SMISO1/ SSCL1	IRQ0-DS	COMP3
21	TCK*1	P27	CS3#	MTIOC2B/TMCI3	SCK1	IRQ7	CVREFC3
22	TDO*1	P26	CS2#	MTIOC2A/TMO1	TXD1/SMOSI1/ SSDA1/CTS3#/ RTS3#/SS3#	IRQ6	CMPC30
23		P25	CS1#	MTIOC4C/MTCLKB	RXD3/SMISO3/ SSCL3	IRQ5	ADTRG0#
24		P24	CS0#	MTIOC4A/MTCLKA/ TMRI1	SCK3	IRQ12	
25		P23		MTIOC3D/MTCLKD	TXD3/SMOSI3/ SSDA3/CTS0#/ RTS0#/SS0#	IRQ3	
26		P22		MTIOC3B/MTCLKC/ TMO0	SCK0	IRQ15	
27		P21		MTIOC1B/TMCI0/ MTIOC4A	RXD0/SMISO0/ SSCL0	IRQ9	
28		P20		MTIOC1A/TMRI0	TXD0/SMOSI0/ SSDA0	IRQ8	

Table 1.6 List of Pin and Pin Functions (100-Pin LFQFP) (2/5)

Pin No. 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
29		P17		MTIOC3A/MTIOC3B/ TMO1/POE8#/ MTIOC4B	SCK1/TXD3/SMOSI3/ SSDA3/MISOA-C/ SDA2	IRQ7	COMP2
30		P16		MTIOC3C/MTIOC3D/ TMO2/RTCOUT*5	TXD1/SMOSI1/ SSDA1/RXD3/ SMISO3/SSCL3/ MOSIA-C/SCL2	IRQ6	ADTRG0#
31		P15		MTIOC0B/MTCLKB/ TMCI2	RXD1/SMISO1/ SSCL1/SCK3/ CRX0-C	IRQ5	CMPC20
32		P14		MTIOC3A/MTCLKA/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0-C	IRQ4	CVREFC2
33		P13		MTIOC0B/TMO3	TXD2/SMOSI2/ SSDA2/SDA0	IRQ3	
34		P12		MTIC5U/TMCI1	RXD2/SMISO2/ SSCL2/SCL0	IRQ2	
35		PH3		MTIOC4D/TMCI0			
36		PH2		MTIOC4C/TMRI0/ TOC1		IRQ1	
37		PH1		MTIOC3D/TMO0/TIC1		IRQ0	ADST0
38		PH0		MTIOC3B/CACREF			ADTRG0#
39		P55	D0[A0/D0]/ WAIT#	MTIOC4D/MTIOC4A/ TMO3	CRX0-D	IRQ10	
40		P54	ALE/ D1[A1/D1]	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX0-D	IRQ4	
41		P53	BCLK		PMC0	IRQ3	
42		P52	RD#		RXD2/SMISO2/ SSCL2	IRQ2	
43		P51	WR1#/BC1#/ WAIT#		SCK2/PMC0	IRQ1	
44		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2	IRQ0	
45	UB	PC7	CS0#	MTIOC3A/MTCLKB/ TMO2/CACREF/TOC0	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ TXD010-C/ SMOSI010-C/ SSDA010-C/MISOA-A	IRQ14	
46		PC6	D2[A2/D2]/ CS1#	MTIOC3C/MTCLKA/ TMCI2/TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ RXD010-C/ SMISO010-C/ SSCL010-C/MOSIA-A	IRQ13	
47		PC5	D3[A3/D3]/ CS2#/WAIT#	MTIOC3B/MTCLKD/ TMRI2/MTIOC0C	SCK8/SCK10/ SCK010-C/ RSPCKA-A/PMC0	IRQ5	
48		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TMCI1/POE0#/ MTIOC0A	SCK5/CTS8#/RTS8#/ SS8#/SS10#/CTS10#/ RTS10#/CTS010#-B/ RTS010#-B/ SS010#-B/DE010-B/ SSLA0-A/PMC0	IRQ12	
49		PC3	A19	MTIOC4D	TXD5/SMOSI5/ SSDA5/PMC0	IRQ11	

Table 1.6 List of Pin and Pin Functions (100-Pin LFQFP) (3/5)

Pin No. 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
50		PC2	A18	MTIOC4B	RXD5/SMISO5/ SSCL5/TXDB011-A/ SSLA3-A	IRQ10	
51		PC1	A17	MTIOC3A	SCK5/TXD011-C/ SMOSI011-C/ SSDA011-C/ TXDA011-C/SSLA2-A	IRQ12	
52		PC0	A16	MTIOC3C	CTS5#/RTS5#/SS5#/ RXD011-C/ SMISO011-C/ SSCL011-C/SSLA1-A	IRQ14	
53		PB7	A15	MTIOC3B	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ TXD011-B/ SMOSI011-B/ SSDA011-B	IRQ15	
54		PB6	A14	MTIOC3D	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ RXD011-B/ SMISO011-B/ SSCL011-B	IRQ6	
55		PB5	A13	MTIOC2A/MTIOC1B/ TMRI1/POE4#/TOC2	SCK9/SCK11/ SCK011-B	IRQ13	
56		PB4	A12		CTS9#/RTS9#/SS9#/ SS11#/CTS11#/ RTS11#/CTS011#-B/ RTS011#-B/ SS011#-B/DE011-B	IRQ4	
57		PB3	A11	MTIOC0A/MTIOC4A/ TMO0/POE11#/TIC2	SCK4/SCK6/PMC0	IRQ3	
58		PB2	A10		CTS4#/RTS4#/SS4#/ CTS6#/RTS6#/SS6#	IRQ2	
59		PB1	A9	MTIOC0C/MTIOC4C/ TMCI0	TXD4/SMOSI4/ SSDA4/TXD6/ SMOSI6/SSDA6	IRQ4-DS	COMP1
60	VCC						
61		PB0	A8	MTIC5W/MTIOC3D	RXD4/SMISO4/ SSCL4/RXD6/ SMISO6/SSCL6/ RSPCKA-C	IRQ12	
62	VSS						
63		PA7	A7		MISOA-B	IRQ7	
64		PA6	A6	MTIC5V/MTCLKB/ TMCI3/POE10#/ MTIOC3D/MTIOC6B	CTS5#/RTS5#/SS5#/ CTS12#/RTS12#/ SS12#/MOSIA-B	IRQ14	
65		PA5	A5	MTIOC6B	RSPCKA-B	IRQ5	
66		PA4	A4	MTIC5U/MTCLKA/ TMRI0/MTIOC4C/ MTIOC7C	TXD5/SMOSI5/ SSDA5/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12/ SSLA0-B	IRQ5-DS	CVREFC1/ ADST0
67		PA3	A3	MTIOC0D/MTCLKD/ MTIC5V/MTIOC4D	RXD5/SMISO5/ SSCL5	IRQ6-DS	CMPC10

Table 1.6 List of Pin and Pin Functions (100-Pin LFQFP) (4/5)

Pin No. 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
68		PA2	A2	MTIOC7A	RXD5/SMISO5/ SSCL5/RXD12/ SMISO12/SSCL12/ RXDX12/SSLA3-B	IRQ10	
69		PA1	A1	MTIOC0B/MTCLKC/ MTIOC7B/MTIOC3B	SCK5/SCK12/ SSLA2-B	IRQ11	ADTRG0#
70		PA0	BC0#/A0	MTIOC4A/CACREF/ MTIOC6D	SSLA1-B	IRQ0	
71		PE7	D15[A15/D15]/ D7[A7/D7]	MTIOC6A/TOC1		IRQ7	AN015
72		PE6	D14[A14/D14]/ D6[A6/D6]	MTIOC6C/TIC1	CTS4#/RTS4#/SS4#	IRQ6	AN014
73		PE5	D13[A13/D13]/ D5[A5/D5]	MTIOC4C/MTIOC2B		IRQ5	AN013/ COMP0
74		PE4	D12[A12/D12]/ D4[A4/D4]	MTIOC4D/MTIOC1A/ MTIOC4A/MTIOC7D		IRQ12	AN012
75		PE3	D11[A11/D11]/ D3[A3/D3]	MTIOC4B/POE8#/ MTIOC1B/TOC3	CTS12#/RTS12#/ SS12#	IRQ11	AN011
76		PE2	D10[A10/D10]/ D2[A2/D2]	MTIOC4A/MTIOC7A/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12	IRQ7-DS	AN010/ CVREFC0
77		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	IRQ9	AN009/ CMPC00
78		PE0	D8[A8/D8]/ D0[A0/D0]	MTIOC3D	SCK12	IRQ8	AN008
79		PD7	D7[A7/D7]	MTIC5U/POE0#		IRQ7	AN023
80		PD6	D6[A6/D6]	MTIC5V/POE4#/ MTIOC8A		IRQ6	AN022
81		PD5	D5[A5/D5]	MTIC5W/POE10#/ MTIOC8C		IRQ5	AN021
82		PD4	D4[A4/D4]	POE11#/MTIOC8B		IRQ4	AN020
83		PD3	D3[A3/D3]	POE8#/MTIOC8D/ TOC2		IRQ3	AN019
84		PD2	D2[A2/D2]	MTIOC4D/TIC2	CRX0-B	IRQ2	AN018
85		PD1	D1[A1/D1]	MTIOC4B/POE0#	CTX0-B	IRQ1	AN017
86		PD0	D0[A0/D0]	POE4#		IRQ0	AN016
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	PJ7					
95		P40				IRQ8-DS	AN000
96	VREFH0	PJ6					
97	AVCC0						
98		P07				IRQ15	ADTRG0#
99	AVSS0						

Table 1.6 List of Pin and Pin Functions (100-Pin LFQFP) (5/5)

Pin No.	Power Supply Clock System Control	I/O Port	Bus	Timer	Communication	Interrupt	
100-Pin LFQFP				(MTU, TMR, RTC, POE, CAC, CMTW)	(SCI, RSCI, RSPI, RIIC, CANFD, REMC)	(IRQ, NMI)	A/D, D/A, CMPC
100		P05				IRQ13	DA1

Note 1. This pin function is not provided for products with no JTAG interface.
Note 2. This pin function is not provided for products with a JTAG interface.
Note 3. This pin function is not provided for products with no sub-clock oscillator.
Note 4. This pin function is not provided for products with a sub-clock oscillator.
Note 5. This pin function is not available in products with no sub-clock oscillator.

1.6.3 80-Pin LFQFP

Table 1.7 List of Pin and Pin Functions (80-Pin LFQFP) (1/4)

Pin No. 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
1		P06				
2		P03			IRQ11	DA0
3		P04				
4	VCL					
5		PJ1	MTIOC3A			
6	MD/FINED	PN6				
7	XCIN*1	PH7*2				
8	XCOUT*1	PH6*2				
9	RES#					
10	XTAL	P37			IRQ4	
11	VSS					
12	EXTAL	P36			IRQ5	
13	VCC					
14		P35			NMI	
15		P34	MTIOC0A/TMCI3/ POE10#	SCK6/SCK0	IRQ4	
16		P32	MTIOC0C/TMO3/ RTCIC2*3/RTCOUT*3/ POE0#/POE10#	TXD6/SMOSI6/ SSDA6/TXD0/ SMOSI0/SSDA0/ CTX0-A	IRQ2-DS	
17		P31	MTIOC4D/TMCI2/ RTCIC1*3	CTS1#/RTS1#/SS1#	IRQ1-DS	
18		P30	MTIOC4B/TMRI3/ RTCIC0*3/POE8#	RXD1/SMISO1/ SSCL1	IRQ0-DS	COMP3
19		P27	MTIOC2B/TMCI3	SCK1	IRQ7	CVREFC3
20		P26	MTIOC2A/TMO1	TXD1/SMOSI1/ SSDA1/CTS3#/ RTS3#/SS3#	IRQ6	CMPC30
21		P21	MTIOC1B/TMCI0/ MTIOC4A	RXD0/SMISO0/ SSCL0	IRQ9	
22		P20	MTIOC1A/TMRI0	TXD0/SMOSI0/ SSDA0	IRQ8	
23		P17	MTIOC3A/MTIOC3B/ TMO1/POE8#/ MTIOC4B	SCK1/TXD3/SMOSI3/ SSDA3/MISOA-C/ SDA2	IRQ7	COMP2
24		P16	MTIOC3C/MTIOC3D/ TMO2/RTCOUT*3	TXD1/SMOSI1/ SSDA1/RXD3/ SMISO3/SSCL3/ MOSIA-C/SCL2	IRQ6	ADTRG0#
25		P15	MTIOC0B/MTCLKB/ TMCI2	RXD1/SMISO1/ SSCL1/SCK3/ CRX0-C	IRQ5	CMPC20
26		P14	MTIOC3A/MTCLKA/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0-C	IRQ4	CVREFC2
27		P13	MTIOC0B/TMO3	SDA0	IRQ3	
28		P12	MTIC5U/TMCI1	SCL0	IRQ2	
29		PH3	MTIOC4D/TMCI0			

Table 1.7 List of Pin and Pin Functions (80-Pin LFQFP) (2/4)

Pin No. 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
30		PH2	MTIOC4C/TMRI0/ TOC1		IRQ1	
31		PH1	MTIOC3D/TMO0/TIC1		IRQ0	ADST0
32		PH0	MTIOC3B/CACREF			ADTRG0#
33		P55	MTIOC4D/MTIOC4A/ TMO3	CRX0-D	IRQ10	
34		P54	MTIOC4B/TMCI1	CTX0-D	IRQ4	
35	UB	PC7	MTIOC3A/MTCLKB/ TMO2/CACREF/TOC0	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ TXD010-C/ SMOSI010-C/ SSDA010-C/MISOA-A	IRQ14	
36		PC6	MTIOC3C/MTCLKA/ TMCI2/TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ RXD010-C/ SMISO010-C/ SSCL010-C/MOSIA-A	IRQ13	
37		PC5	MTIOC3B/MTCLKD/ TMRI2/MTIOC0C	SCK8/SCK10/ SCK010-C/ RSPCKA-A/PMC0	IRQ5	
38		PC4	MTIOC3D/MTCLKC/ TMCI1/POE0#/ MTIOC0A	SCK5/CTS8#/RTS8#/ SS8#/SS10#/CTS10#/ RTS10#/CTS010#-B/ RTS010#-B/ SS010#-B/DE010-B/ SSLA0-A/PMC0	IRQ12	
39		PC3	MTIOC4D	TXD5/SMOSI5/ SSDA5/PMC0	IRQ11	
40		PC2	MTIOC4B	RXD5/SMISO5/ SSCL5/TXDB011-A/ SSLA3-A	IRQ10	
41		PB7	MTIOC3B	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ TXD011-B/ SMOSI011-B/ SSDA011-B	IRQ15	
42		PB6	MTIOC3D	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ RXD011-B/ SMISO011-B/ SSCL011-B	IRQ6	
43		PB5	MTIOC2A/MTIOC1B/ TMRI1/POE4#/TOC2	SCK9/SCK11/ SCK011-B	IRQ13	
44		PB4		CTS9#/RTS9#/SS9#/ SS11#/CTS11#/ RTS11#/CTS011#-B/ RTS011#-B/ SS011#-B/DE011-B	IRQ4	
45		PB3	MTIOC0A/MTIOC4A/ TMO0/POE11#/TIC2	SCK4/SCK6/PMC0	IRQ3	
46		PB2		CTS4#/RTS4#/SS4#/ CTS6#/RTS6#/SS6#	IRQ2	

Table 1.7 List of Pin and Pin Functions (80-Pin LFQFP) (3/4)

Pin No. 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
47		PB1	MTIOC0C/MTIOC4C/ TMCI0	TXD4/SMOSI4/ SSDA4/TXD6/ SMOSI6/SSDA6	IRQ4-DS	COMP1
48	VCC					
49		PB0	MTIC5W/MTIOC3D	RXD4/SMISO4/ SSCL4/RXD6/ SMISO6/SSCL6/ RSPCKA-C	IRQ12	
50	VSS					
51		PA6	MTIC5V/MTCLKB/ TMCI3/POE10#/ MTIOC3D/MTIOC6B	CTS5#/RTS5#/SS5#/ CTS12#/RTS12#/ SS12#/MOSIA-B	IRQ14	
52		PA5	MTIOC6B	RSPCKA-B	IRQ5	
53		PA4	MTIC5U/MTCLKA/ TMRI0/MTIOC4C/ MTIOC7C	TXD5/SMOSI5/ SSDA5/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12/ SSLA0-B	IRQ5-DS	CVREFC1/ ADST0
54		PA3	MTIOC0D/MTCLKD/ MTIC5V/MTIOC4D	RXD5/SMISO5/ SSCL5	IRQ6-DS	CMPC10
55		PA2	MTIOC7A	RXD5/SMISO5/ SSCL5/RXD12/ SMISO12/SSCL12/ RXDX12/SSLA3-B	IRQ10	
56		PA1	MTIOC0B/MTCLKC/ MTIOC7B/MTIOC3B	SCK5/SCK12/ SSLA2-B	IRQ11	ADTRG0#
57		PA0	MTIOC4A/CACREF/ MTIOC6D	SSLA1-B	IRQ0	
58		PE5	MTIOC4C/MTIOC2B		IRQ5	AN013/COMP0
59		PE4	MTIOC4D/MTIOC1A/ MTIOC4A/MTIOC7D		IRQ12	AN012
60		PE3	MTIOC4B/POE8#/ MTIOC1B/TOC3	CTS12#/RTS12#/ SS12#	IRQ11	AN011
61		PE2	MTIOC4A/MTIOC7A/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12	IRQ7-DS	AN010/ CVREFC0
62		PE1	MTIOC4C/MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	IRQ9	AN009/ CMPC00
63		PE0	MTIOC3D	SCK12	IRQ8	AN008
64		PD2	MTIOC4D/TIC2	CRX0-B	IRQ2	AN018
65		PD1	MTIOC4B/POE0#	CTX0-B	IRQ1	AN017
66		PD0	POE4#		IRQ0	AN016
67		P47			IRQ15-DS	AN007
68		P46			IRQ14-DS	AN006
69		P45			IRQ13-DS	AN005
70		P44			IRQ12-DS	AN004
71		P43			IRQ11-DS	AN003
72		P42			IRQ10-DS	AN002
73		P41			IRQ9-DS	AN001
74	VREFL0	PJ7				
75		P40			IRQ8-DS	AN000

Table 1.7 List of Pin and Pin Functions (80-Pin LFQFP) (4/4)

Pin No. 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
76	VREFH0	PJ6				
77	AVCC0					
78		P07			IRQ15	ADTRG0#
79	AVSS0					
80		P05			IRQ13	DA1

Note 1. This pin function is not provided for products with no sub-clock oscillator.
Note 2. This pin function is not provided for products with a sub-clock oscillator.
Note 3. This pin function is not available in products with no sub-clock oscillator.

1.6.4 64-Pin LFQFP

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (1/3)

Pin No. 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
1		P03			IRQ11	DA0
2	VCL					
3	MD/FINED	PN6				
4	XCIN*1	PH7*2				
5	XCOUT*1	PH6*2				
6	RES#					
7	XTAL	P37			IRQ4	
8	VSS					
9	EXTAL	P36			IRQ5	
10	VCC					
11		P35			NMI	
12		P32	MTIOC0C/TMO3/ RTCIC2*3/RTCOUT*3/ POE0#/POE10#	TXD6/SMOSI6/ SSDA6/CTX0-A	IRQ2-DS	
13		P31	MTIOC4D/TMCI2/ RTCIC1*3	CTS1#/RTS1#/SS1#	IRQ1-DS	
14		P30	MTIOC4B/TMRI3/ RTCIC0*3/POE8#	RXD1/SMISO1/ SSCL1	IRQ0-DS	COMP3
15		P27	MTIOC2B/TMCI3	SCK1	IRQ7	CVREFC3
16		P26	MTIOC2A/TMO1	TXD1/SMOSI1/ SSDA1/CTS3#/ RTS3#/SS3#	IRQ6	CMPC30
17		P17	MTIOC3A/MTIOC3B/ TMO1/POE8#/ MTIOC4B	SCK1/TXD3/SMOSI3/ SSDA3/MISOA-C/ SDA2	IRQ7	COMP2
18		P16	MTIOC3C/MTIOC3D/ TMO2/RTCOUT*3	TXD1/SMOSI1/ SSDA1/RXD3/ SMISO3/SSCL3/ MOSIA-C/SCL2	IRQ6	ADTRG0#
19		P15	MTIOC0B/MTCLKB/ TMCI2	RXD1/SMISO1/ SSCL1/SCK3/ CRX0-C	IRQ5	CMPC20
20		P14	MTIOC3A/MTCLKA/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0-C	IRQ4	CVREFC2
21		PH3	MTIOC4D/TMCI0			
22		PH2	MTIOC4C/TMRI0/ TOC1		IRQ1	
23		PH1	MTIOC3D/TMO0/TIC1		IRQ0	ADST0
24		PH0	MTIOC3B/CACREF			ADTRG0#
25		P55	MTIOC4D/MTIOC4A/ TMO3	CRX0-D	IRQ10	
26		P54	MTIOC4B/TMCI1	CTX0-D	IRQ4	
27	UB	PC7	MTIOC3A/MTCLKB/ TMO2/CACREF/TOC0	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ TXD010-C/ SMOSI010-C/ SSDA010-C/MISOA-A	IRQ14	

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (2/3)

Pin No. 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
28		PC6	MTIOC3C/MTCLKA/ TMCI2/TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ RXD010-C/ SMISO010-C/ SSCL010-C/MOSIA-A	IRQ13	
29		PC5	MTIOC3B/MTCLKD/ TMRI2/MTIOC0C	SCK8/SCK10/ SCK010-C/ RSPCKA-A/PMC0	IRQ5	
30		PC4	MTIOC3D/MTCLKC/ TMCI1/POE0#/ MTIOC0A	SCK5/CTS8#/RTS8#/ SS8#/SS10#/CTS10#/ RTS10#/CTS010#-B/ RTS010#-B/ SS010#-B/DE010-B/ SSLA0-A/PMC0	IRQ12	
31		PC3	MTIOC4D	TXD5/SMOSI5/ SSDA5/PMC0	IRQ11	
32		PC2	MTIOC4B	RXD5/SMISO5/ SSCL5/TXDB011-A/ SSLA3-A	IRQ10	
33		PB7	MTIOC3B	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ TXD011-B/ SMOSI011-B/ SSDA011-B	IRQ15	
34		PB6	MTIOC3D	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ RXD011-B/ SMISO011-B/ SSCL011-B	IRQ6	
35		PB5	MTIOC2A/MTIOC1B/ TMRI1/POE4#/TOC2	SCK9/SCK11/ SCK011-B	IRQ13	
36		PB3	MTIOC0A/MTIOC4A/ TMO0/POE11#/TIC2	SCK4/SCK6/PMC0	IRQ3	
37		PB1	MTIOC0C/MTIOC4C/ TMCI0	TXD4/SMOSI4/ SSDA4/TXD6/ SMOSI6/SSDA6	IRQ4-DS	COMP1
38	VCC					
39		PB0	MTIC5W/MTIOC3D	RXD4/SMISO4/ SSCL4/RXD6/ SMISO6/SSCL6/ RSPCKA-C	IRQ12	
40	VSS					
41		PA6	MTIC5V/MTCLKB/ TMCI3/POE10#/ MTIOC3D/MTIOC6B	CTS5#/RTS5#/SS5#/ CTS12#/RTS12#/ SS12#/MOSIA-B	IRQ14	
42		PA4	MTIC5U/MTCLKA/ TMRI0/MTIOC4C/ MTIOC7C	TXD5/SMOSI5/ SSDA5/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12/ SSLA0-B	IRQ5-DS	CVREFC1/ ADST0
43		PA3	MTIOC0D/MTCLKD/ MTIC5V/MTIOC4D	RXD5/SMISO5/ SSCL5	IRQ6-DS	CMPC10
44		PA1	MTIOC0B/MTCLKC/ MTIOC7B/MTIOC3B	SCK5/SCK12/ SSLA2-B	IRQ11	ADTRG0#

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (3/3)

Pin No. 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, RTC, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, D/A, CMPC
45		PA0	MTIOC4A/CACREF/ MTIOC6D	SSLA1-B	IRQ0	
46		PE5	MTIOC4C/MTIOC2B		IRQ5	AN013/COMP0
47		PE4	MTIOC4D/MTIOC1A/ MTIOC4A/MTIOC7D		IRQ12	AN012
48		PE3	MTIOC4B/POE8#/ MTIOC1B/TOC3	CTS12#/RTS12#/ SS12#	IRQ11	AN011
49		PE2	MTIOC4A/MTIOC7A/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12	IRQ7-DS	AN010/ CVREFC0
50		PE1	MTIOC4C/MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	IRQ9	AN009/ CMPC00
51		PE0	MTIOC3D	SCK12	IRQ8	AN008
52		P47			IRQ15-DS	AN007
53		P46			IRQ14-DS	AN006
54		P45			IRQ13-DS	AN005
55		P44			IRQ12-DS	AN004
56		P43			IRQ11-DS	AN003
57		P42			IRQ10-DS	AN002
58		P41			IRQ9-DS	AN001
59	VREFL0	PJ7				
60		P40			IRQ8-DS	AN000
61	VREFH0	PJ6				
62	AVCC0					
63		P07			IRQ15	ADTRG0#
64	AVSS0					

Note 1. This pin function is not provided for products with no sub-clock oscillator.
Note 2. This pin function is not provided for products with a sub-clock oscillator.
Note 3. This pin function is not available in products with no sub-clock oscillator.

1.6.5 48-Pin LFQFP

Table 1.9 List of Pin and Pin Functions (48-Pin LFQFP) (1/2)

Pin No. 48-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, CMPC
1	VCL					
2	MD/FINED	PN6				
3	RES#					
4	XTAL	P37			IRQ4	
5	VSS					
6	EXTAL	P36			IRQ5	
7	VCC					
8		P35			NMI	
9		P31	MTIOC4D/TMCI2	CTS1#/RTS1#/SS1#	IRQ1-DS	
10		P30	MTIOC4B/POE8#	RXD1/SMISO1/ SSCL1	IRQ0-DS	COMP3
11		P27	MTIOC2B	SCK1	IRQ7	CVREFC3
12		P26	MTIOC2A/TMO1	TXD1/SMOSI1/ SSDA1/CTS3#/ RTS3#/SS3#	IRQ6	CMPC30
13		P17	MTIOC3A/MTIOC3B/ TMO1/POE8#/ MTIOC4B	SCK1/TXD3/SMOSI3/ SSDA3/MISOA-C/ SDA2	IRQ7	COMP2
14		P16	MTIOC3C/MTIOC3D/ TMO2	TXD1/SMOSI1/ SSDA1/RXD3/ SMISO3/SSCL3/ MOSIA-C/SCL2	IRQ6	ADTRG0#
15		P15	MTIOC0B/MTCLKB/ TMCI2	RXD1/SMISO1/ SSCL1/SCK3/ CRX0-C	IRQ5	CMPC20
16		P14	MTIOC3A/MTCLKA/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0-C	IRQ4	CVREFC2
17		PH3	MTIOC4D/TMCI0			
18		PH2	MTIOC4C/TMRI0/ TOC1		IRQ1	
19		PH1	MTIOC3D/TMO0/TIC1		IRQ0	ADST0
20		PH0	MTIOC3B/CACREF			ADTRG0#
21	UB	PC7	MTIOC3A/MTCLKB/ TMO2/CACREF/TOC0	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ TXD010-C/ SMOSI010-C/ SSDA010-C/MISOA-A	IRQ14	
22		PC6	MTIOC3C/MTCLKA/ TMCI2/TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ RXD010-C/ SMISO010-C/ SSCL010-C/MOSIA-A	IRQ13	
23		PC5	MTIOC3B/MTCLKD/ TMRI2/MTIOC0C	SCK8/SCK10/ SCK010-C/ RSPCKA-A/PMC0	IRQ5	

Table 1.9 List of Pin and Pin Functions (48-Pin LFQFP) (2/2)

Pin No. 48-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TMR, POE, CAC, CMTW)	Communication (SCI, RSCI, RSPI, RIIC, CANFD, REMC)	Interrupt (IRQ, NMI)	A/D, CMPC
24		PC4	MTIOC3D/MTCLKC/ TMCI1/POE0#/ MTIOC0A	SCK5/CTS8#/RTS8#/ SS8#/SS10#/CTS10#/ RTS10#/CTS010#-B/ RTS010#-B/ SS010#-B/DE010-B/ SSLA0-A/PMC0	IRQ12	
25		PB5	MTIOC2A/MTIOC1B/ TMRI1/POE4#/TOC2		IRQ13	
26		PB3	MTIOC0A/MTIOC4A/ TMO0/POE11#/TIC2	SCK4/SCK6/PMC0	IRQ3	
27		PB1	MTIOC0C/MTIOC4C/ TMCI0	TXD4/SMOSI4/ SSDA4/TXD6/ SMOSI6/SSDA6	IRQ4-DS	COMP1
28	VCC					
29		PB0	MTIC5W/MTIOC3D	RXD4/SMISO4/ SSCL4/RXD6/ SMISO6/SSCL6/ RSPCKA-C	IRQ12	
30	VSS					
31		PA6	MTIC5V/MTCLKB/ POE10#/MTIOC3D	CTS5#/RTS5#/SS5#/ CTS12#/RTS12#/ SS12#/MOSIA-B	IRQ14	
32		PA4	MTIC5U/MTCLKA/ TMRI0/MTIOC4C/ MTIOC7C	TXD5/SMOSI5/ SSDA5/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12/ SSLA0-B	IRQ5-DS	CVREFC1/ ADST0
33		PA3	MTIOC0D/MTCLKD/ MTIC5V/MTIOC4D	RXD5/SMISO5/ SSCL5	IRQ6-DS	CMPC10
34		PA1	MTIOC0B/MTCLKC/ MTIOC7B/MTIOC3B	SCK5/SCK12/ SSLA2-B	IRQ11	ADTRG0#
35		PE4	MTIOC4D/MTIOC1A/ MTIOC4A/MTIOC7D		IRQ12	AN012
36		PE3	MTIOC4B/POE8#/ MTIOC1B/TOC3	CTS12#/RTS12#/ SS12#	IRQ11	AN011
37		PE2	MTIOC4A/MTIOC7A/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12	IRQ7-DS	AN010/ CVREFC0
38		PE1	MTIOC4C/MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	IRQ9	AN009/ CMPC00
39		P47			IRQ15-DS	AN007
40		P46			IRQ14-DS	AN006
41		P45			IRQ13-DS	AN005
42		P42			IRQ10-DS	AN002
43		P41			IRQ9-DS	AN001
44	VREFL0	PJ7				
45		P40			IRQ8-DS	AN000
46	VREFH0	PJ6				
47	AVCC0					
48	AVSS0					

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

2.2 Register Set of the CPU

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

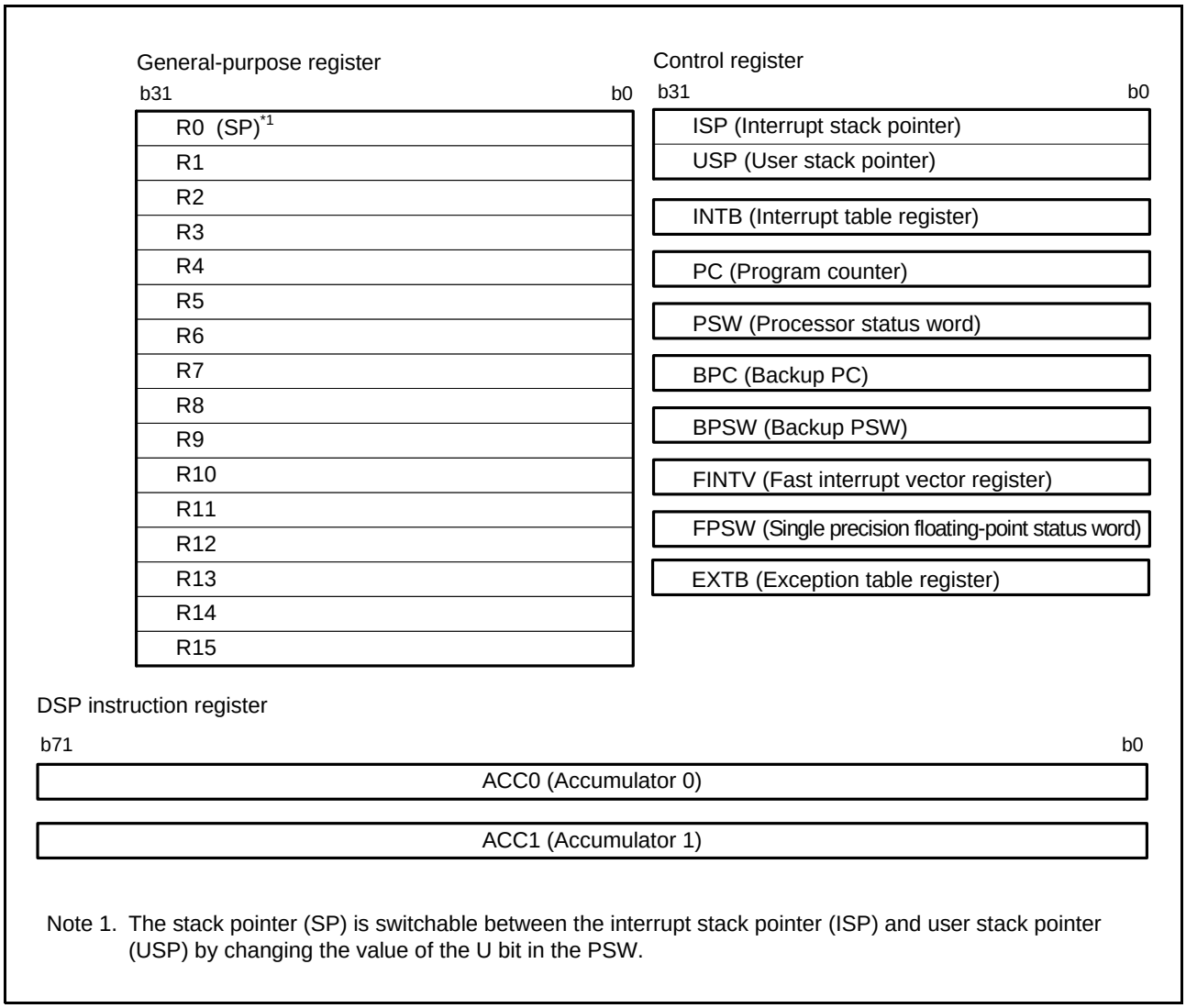


Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

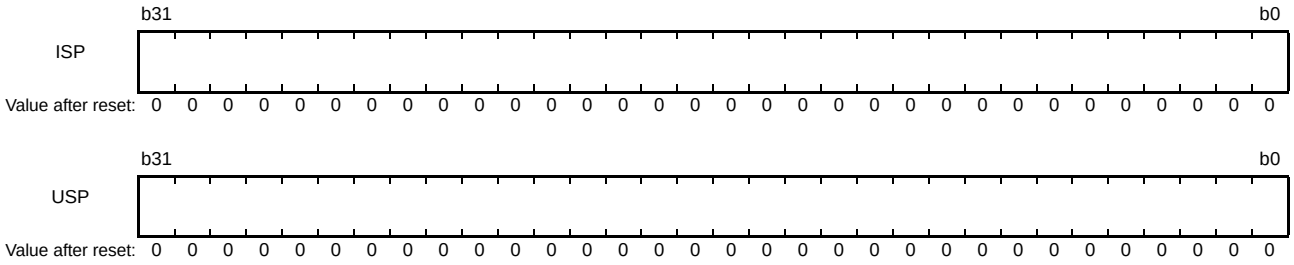
This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (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.

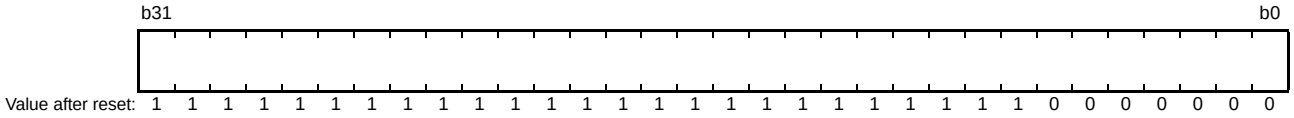
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Single-precision floating-point status word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

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



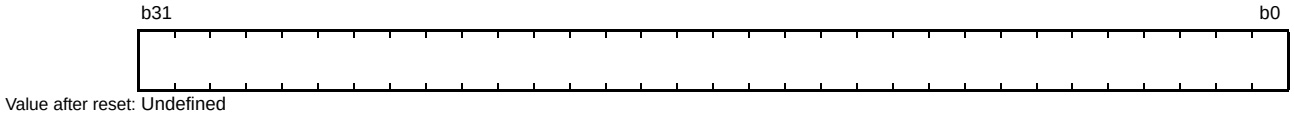
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2.2 Exception Table Register (EXTB)



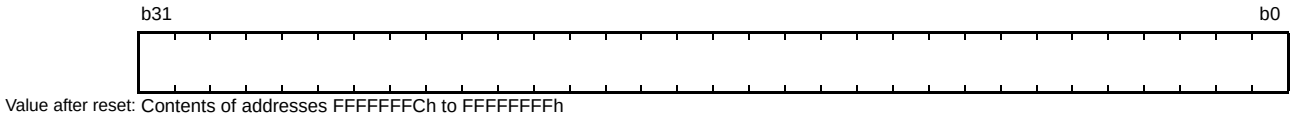
The exception table register (EXTB) specifies the address where the exception vector table starts.

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.

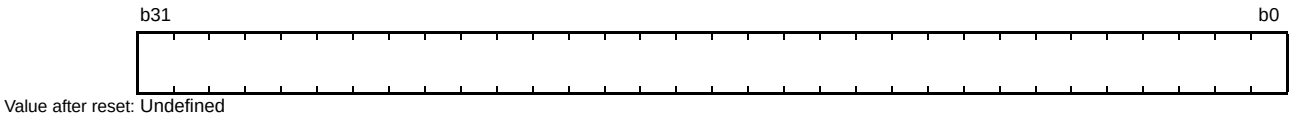
PM Bit (Processor Mode Select)

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.

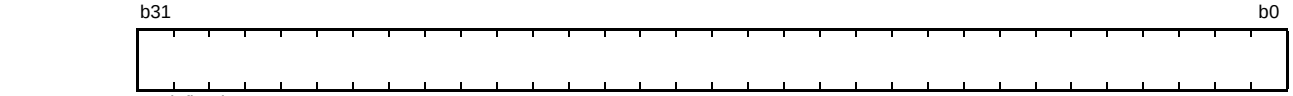
2.2.2.6 Backup PC (BPC)



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.

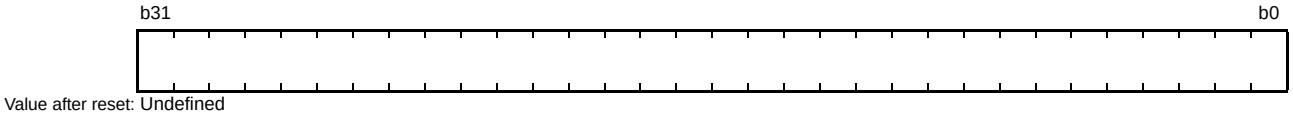
2.2.2.7 Backup PSW (BPSW)



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)

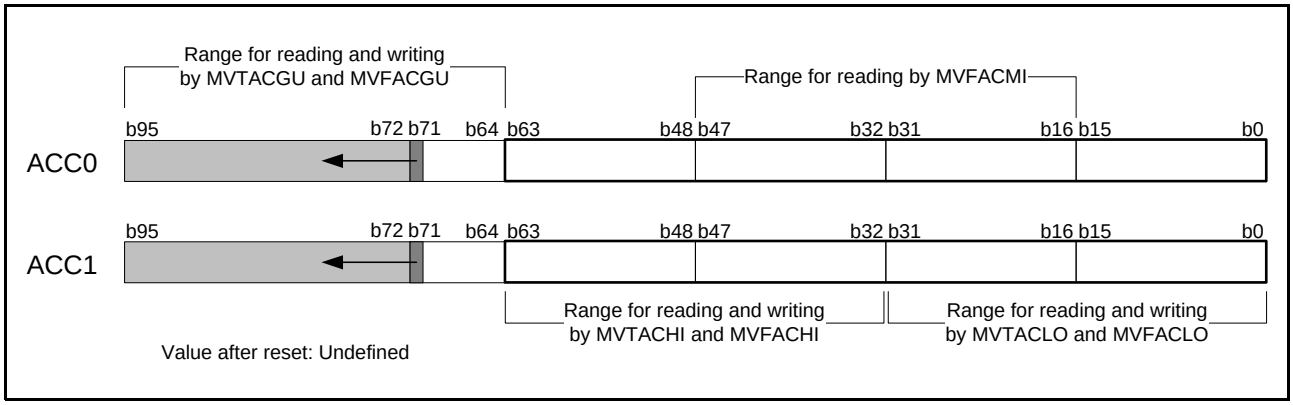
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

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.

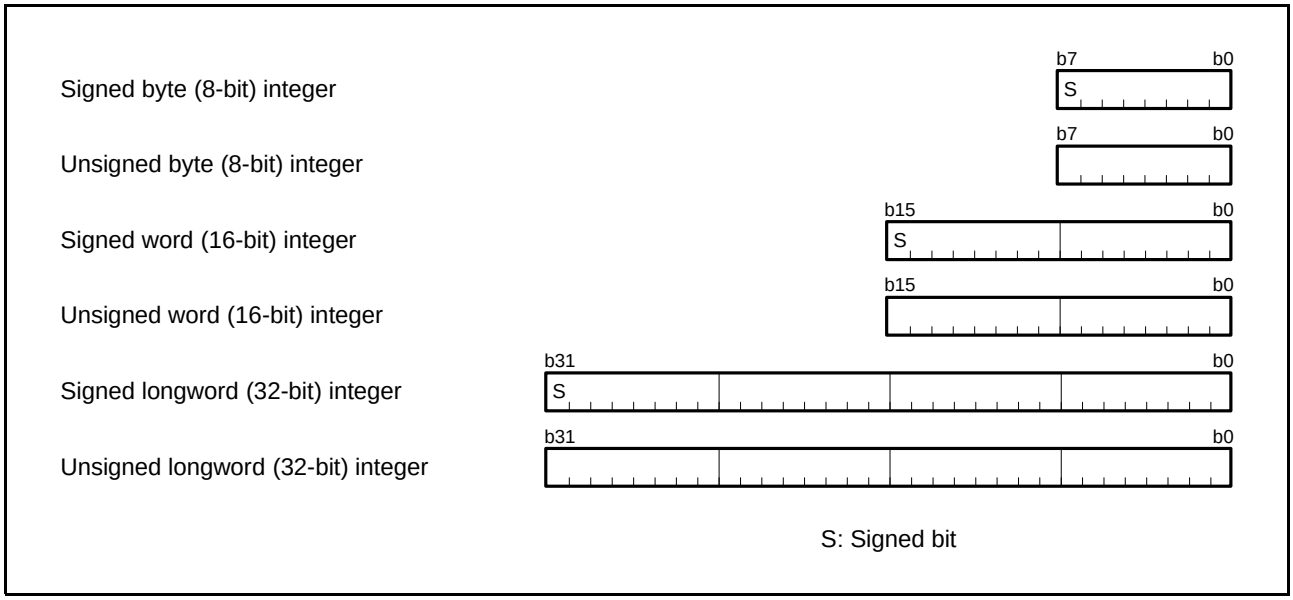


Figure 2.2 Integer

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.

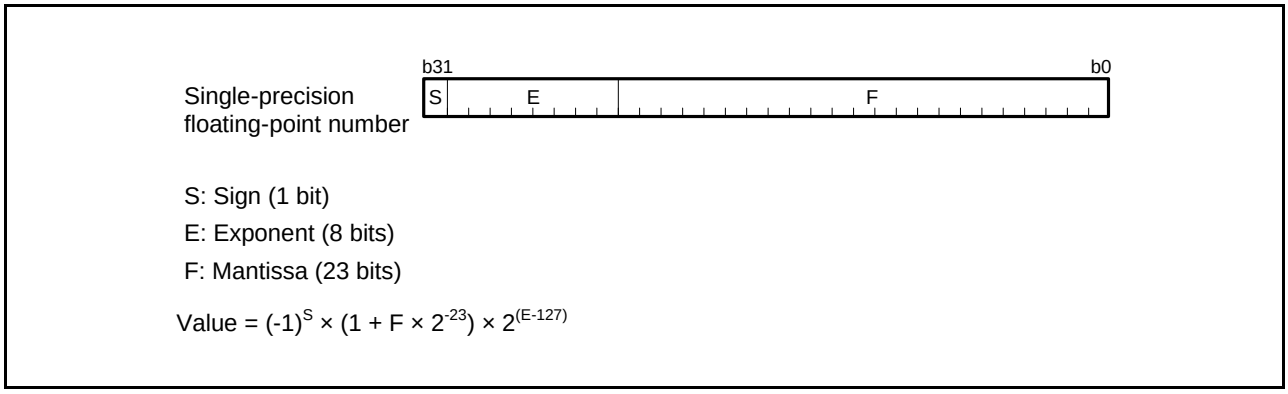


Figure 2.3 Single-Precision Floating-Point Number

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.

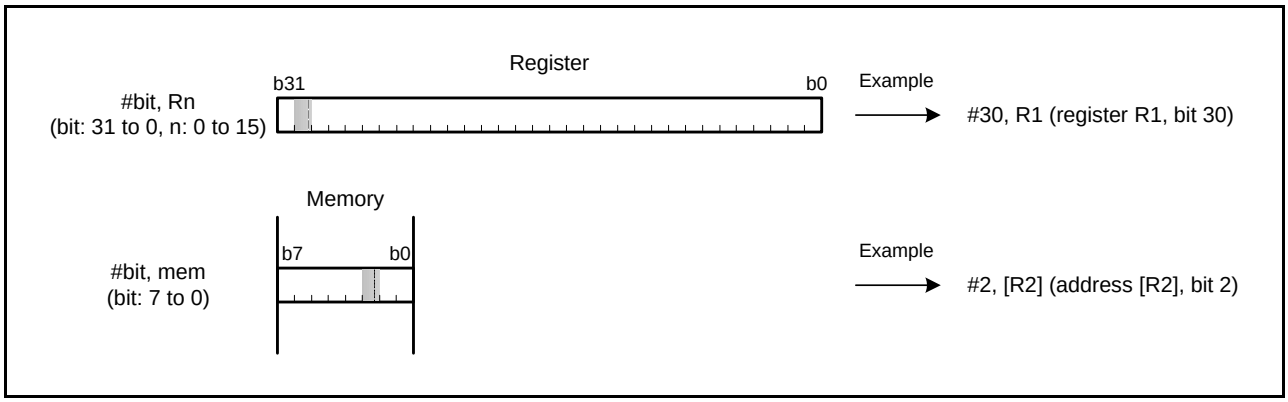


Figure 2.4 Bit

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.

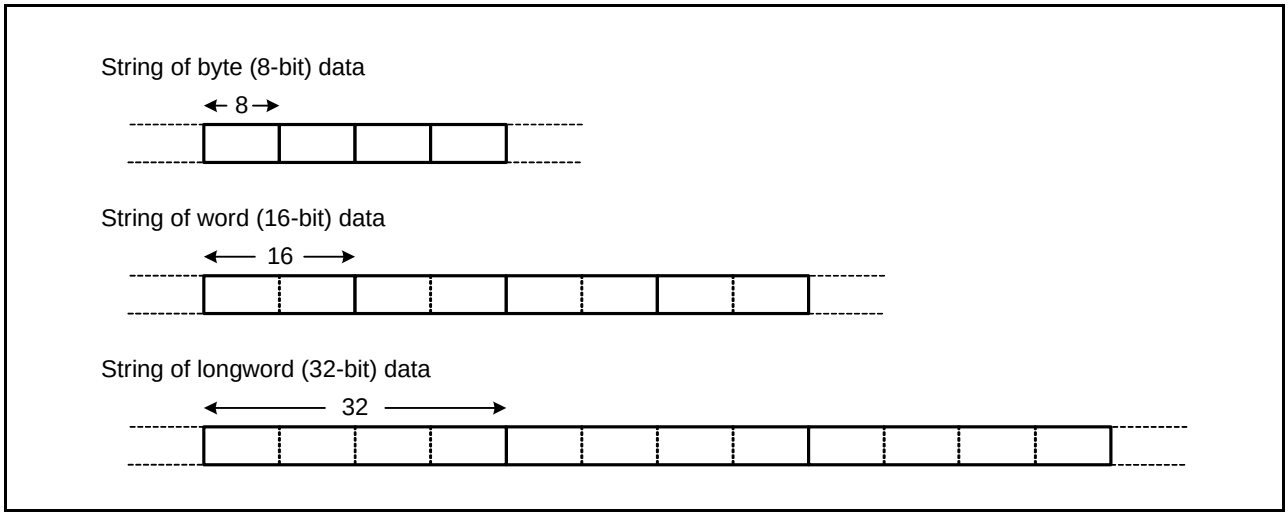


Figure 2.5 String

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.5.2 Access to I/O Registers

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

2.5.3 Notes on Access to I/O Registers

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

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

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

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

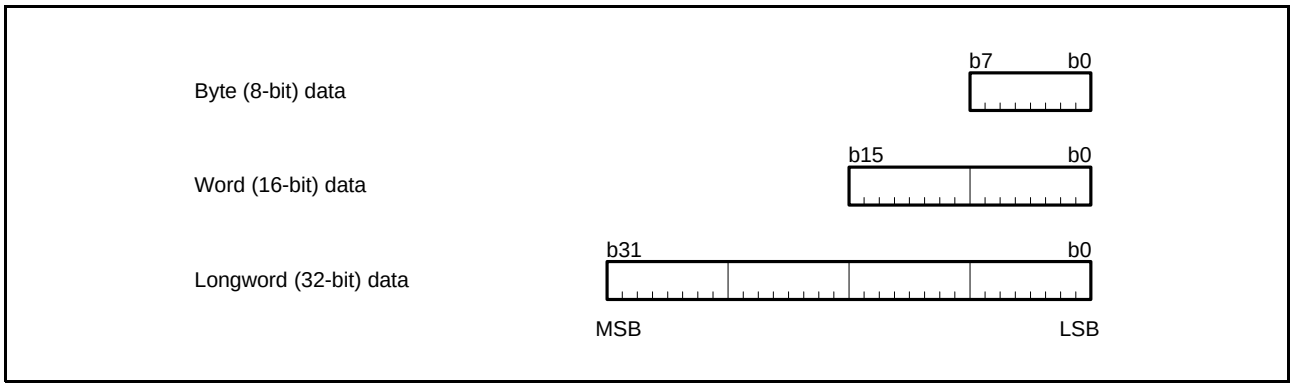


Figure 2.6 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

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

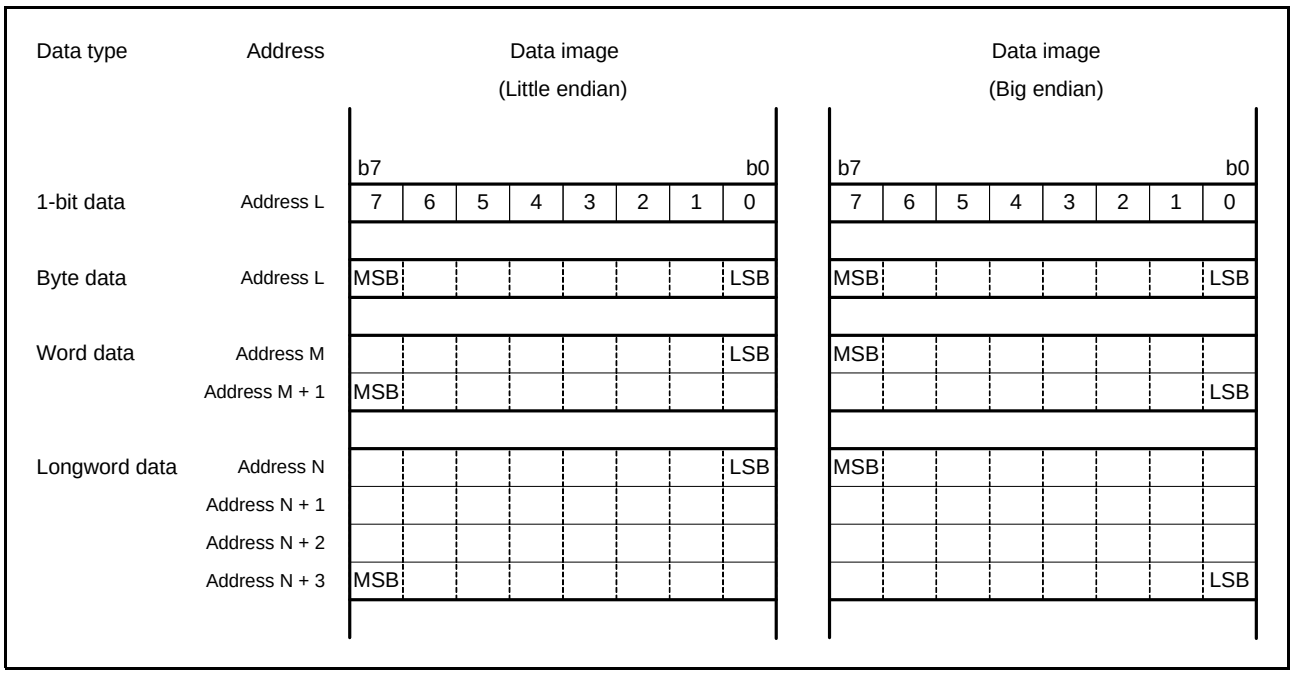


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

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

2.6 Vector Table

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

2.6.1 Exception Vector Table

In the exception vector table, the individual vectors for the privileged instruction exception, access exception, undefined instruction exception, 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.

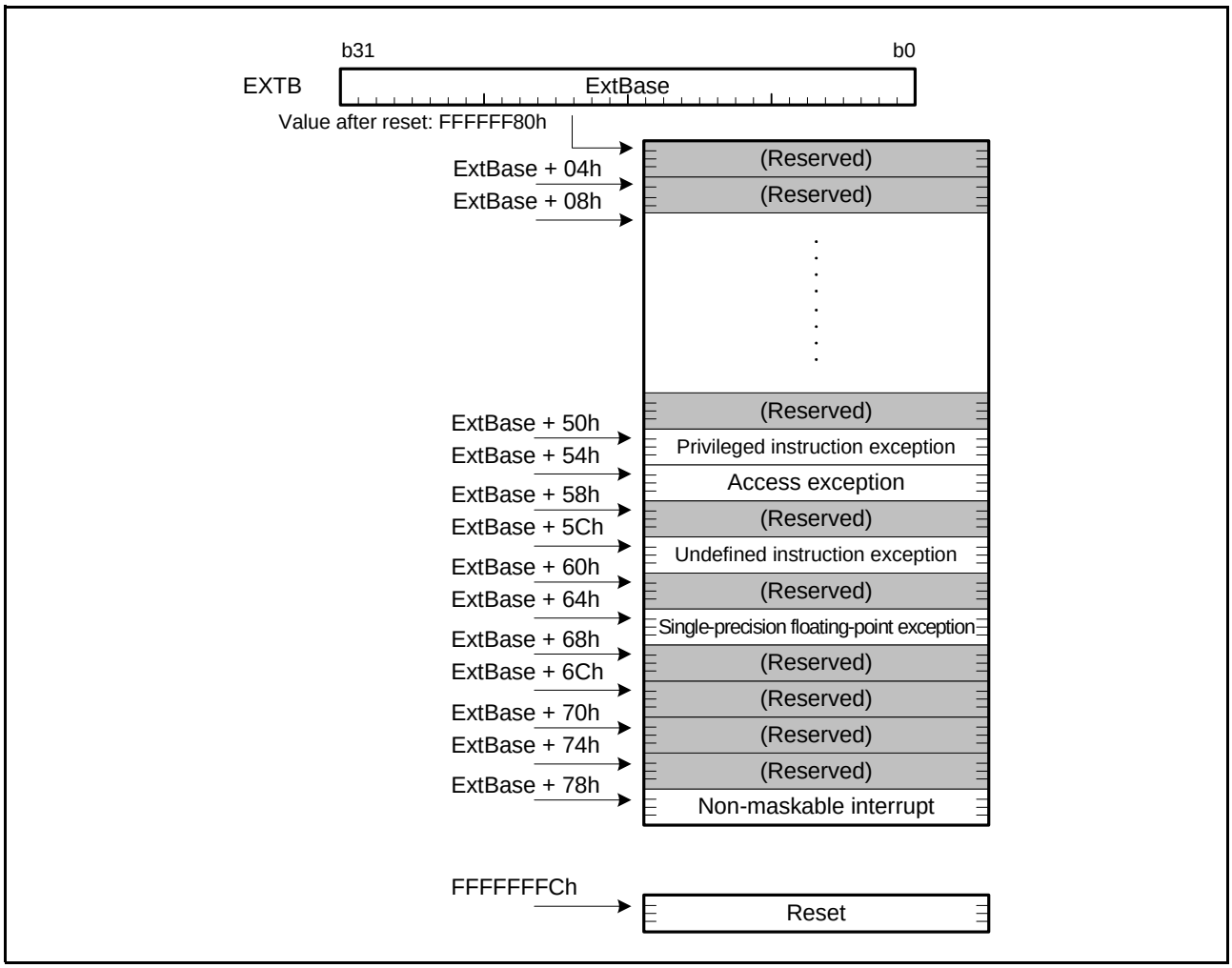


Figure 2.8 Exception Vector Table

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 14.3.1, Interrupt Vector Table.

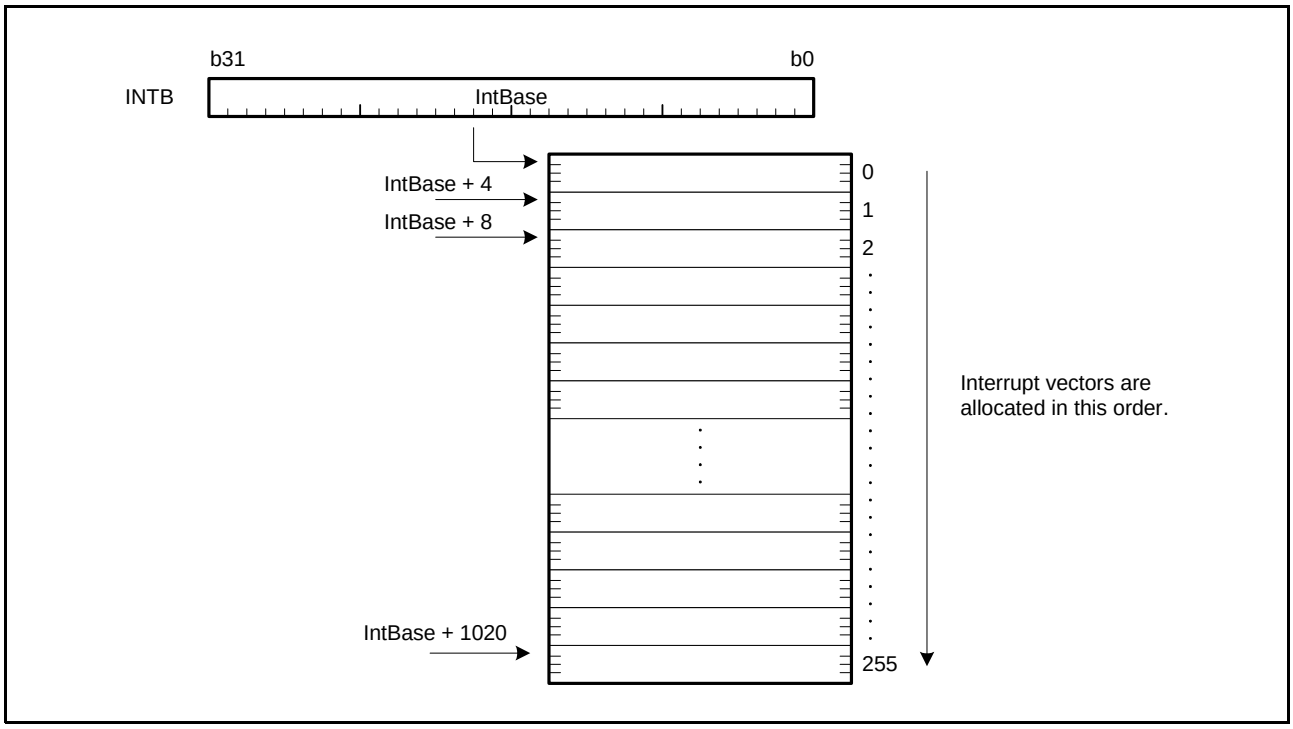


Figure 2.9 Interrupt Vector Table

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.

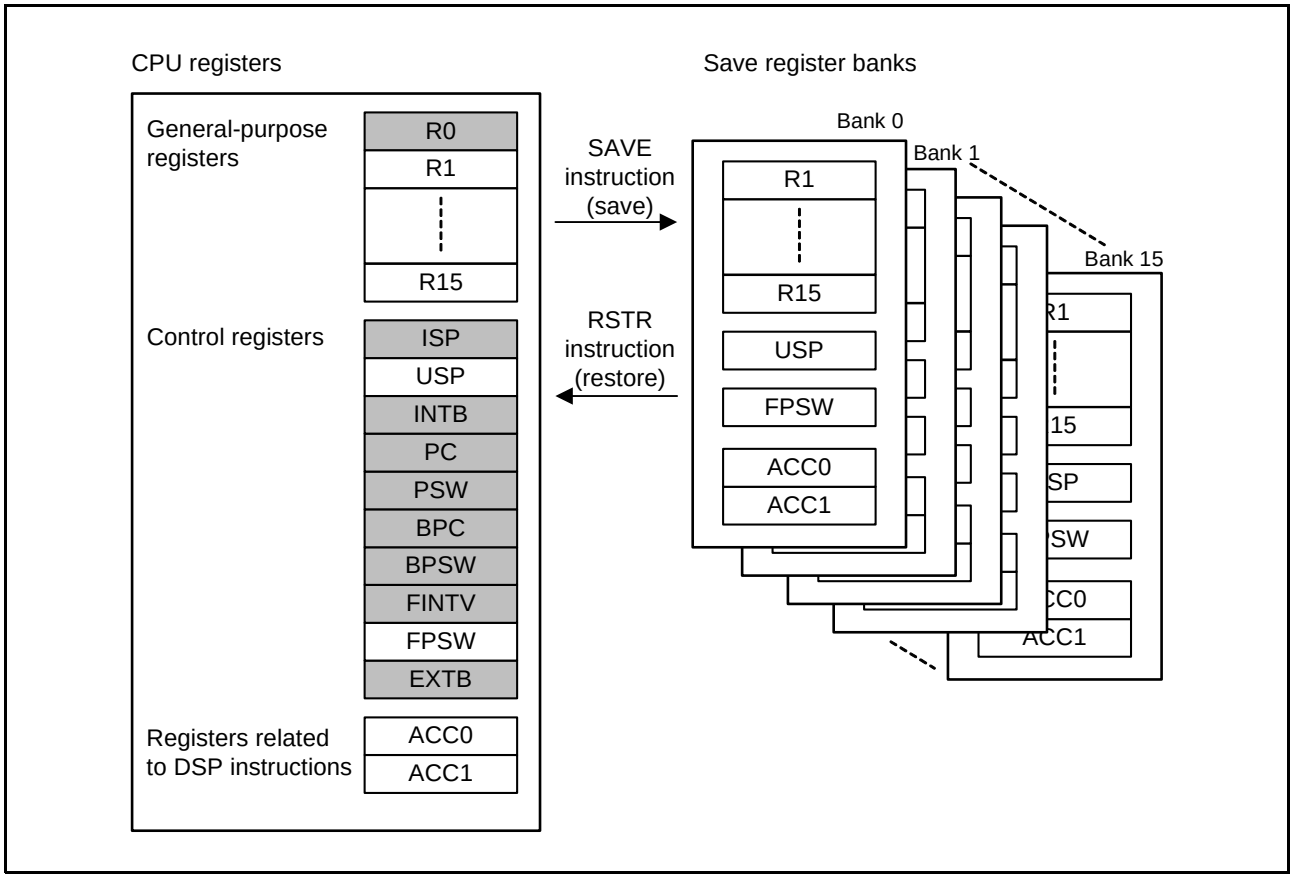


Figure 2.10 Save Register Banks

2.8 Operation of Instructions

2.8.1 Restrictions on RMPA and String-Manipulation Instructions

2.8.1.1 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

2.8.1.2 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.

2.8.1.3 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.

2.9 Numbers of Cycles

2.9.1 Instruction and Numbers of Cycles

Table 2.13 to Table 2.21 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 2.13 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 2.14 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 2.15 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 2.16 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 2.17 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 2.18 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 2.19 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 2.20 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 2.21 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

2.9.2 Numbers of Cycles for Response to Interrupts

Table 2.22 lists numbers of cycles taken by processing for response to interrupts.

Table 2.22 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 2.22 will be applicable when access to memory from the CPU is processed with no waiting. This MCU has a RAM and code flash memory that allow no-wait access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in code flash memory and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.
For information on the numbers of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13 to Table 2.21.
The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 13.3.1, Acceptance Timing and Saved PC Value.

2.10 Usage Note

2.10.1 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 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, *2	UB*3	Operating Mode	ROME	EXBE
High	—	Single-chip mode	1 (On-chip ROM enabled)	0 (External bus disabled)
Low	Low	Boot mode (SCI interface)		
	High	User boot mode		
Low → High*4	Low	Boot mode (FINE interface)		

Note 1. Transition between operating modes is in progress during the waiting time after release from the RES# pin reset, power-on reset time, or LVD0 reset time. Accordingly, do not change the input level on the MD pin during these times. For details on the waiting time after release from the RES# pin reset, power-on reset time, and LVD0 reset time, see section 45, Electrical Characteristics.
Note 2. After boot-up, the MD pin can be used as a general port pin (PN6) in any operating mode.
Note 3. The PC7 pin, which is multiplexed on the same pin as the UB pin function, may also be used as a general port pin.
Note 4. After release from 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, user boot mode
1 (On-chip ROM enabled)	0 (external bus disabled)	
0 (On-chip ROM disabled)*1	1 (external bus enabled)	On-chip ROM disabled extended mode
1 (On-chip ROM enabled)	1 (external bus enabled)	On-chip ROM enabled extended mode

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

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

Table 3.3 Selection of Endian

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

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

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

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

3.2.2 Mode Status Register (MDSR)

Address(es): 0008 0002h

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

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

Note 1. Depends on the operating mode at startup.

3.2.3 System Control Register 0 (SYSCR0)

Address(es): 0008 0006h

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

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

Note 1. Write data is not retained.

ROME Bit (On-Chip ROM Enable)

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

EXBE Bit (External Bus Enable)

The EXBE bit enables or disables the external bus.
Do not write 0 to this bit while a program is running from an external address space. Write 0 to this bit after access to the external bus is completed. Furthermore, when an external address space is included in the range of transfer by the bus masters other than the CPU (DMAC and DTC), 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 21, Multi-Function Pin Controller (MPC).

3.2.4 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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
b7 to b1	—	Reserved	These bits are read as 1. 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

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 45, Electrical Characteristics.

3.2.5 Voltage Level Setting Register (VOLSR)

Address(es): 0008 C295h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RICVLS	RIIC Operating Voltage Setting	0: VCC ≥ 4.5 V 1: VCC < 4.5 V	R/W

The VOLSR register specifies the power-supply voltage when the RIIC is in use.

RICVLS Bit (RIIC Operating Voltage Setting)

The RICVLS bit controls the slew rate for the RIIC. Set the bit according to the VCC voltage. Set this bit before releasing the RIIC from the module-stop state.

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 21, 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 21, 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 44, 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 (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 44, Flash Memory (FLASH).
The chip starts up in boot mode (FINE interface) when both the MD and UB pins are set to the low level at the time of release from the reset state and then the MD pin is switched to the high level within 20 to 100 ms.

3.3.6 User Boot Mode

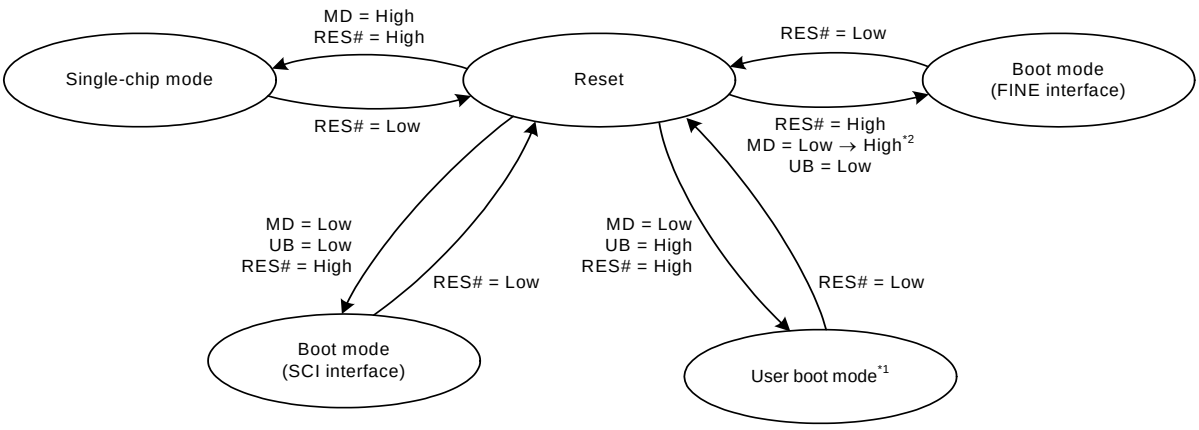
In user boot mode, the flash memory rewriting program (user boot program) created by the user operates. The chip starts up in a state equivalent to single-chip mode after release from the reset state.
After programming the prescribed values for UB code A and the UB code B, the chip starts up in user boot mode if the low level is on the MD pin and the high level is on the UB pin at the time of release from the reset state. In user boot mode, the reset vector is fetched from address FF7F FFFCh. For UB code A and UB code B, see section 7, Option-Setting Memory (OFSM).
After the chip has started up in user boot mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.

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

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins

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



Note 1. To start the MCU in user boot mode, settings for the UB code A and UB code B in the option setting memory are required along with the levels set on the mode-setting pins.
Note 2. 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.

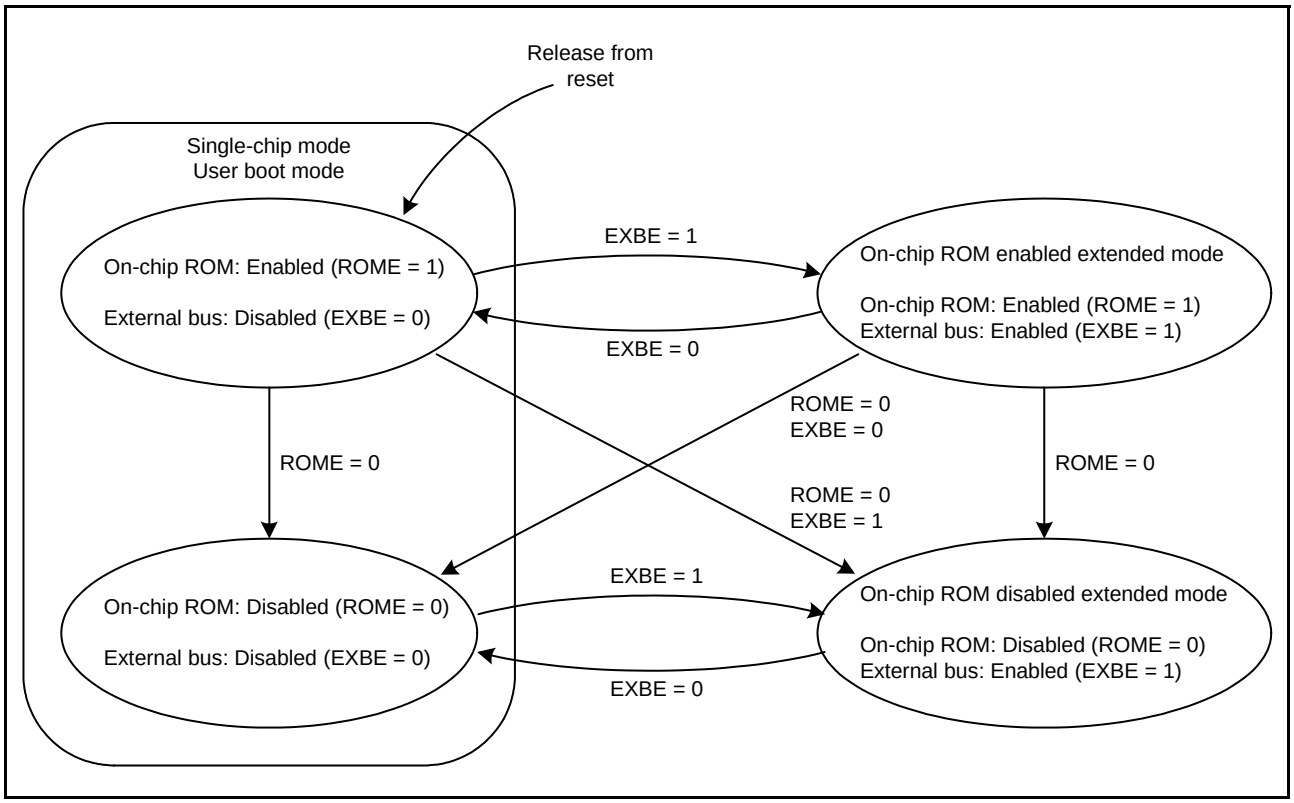
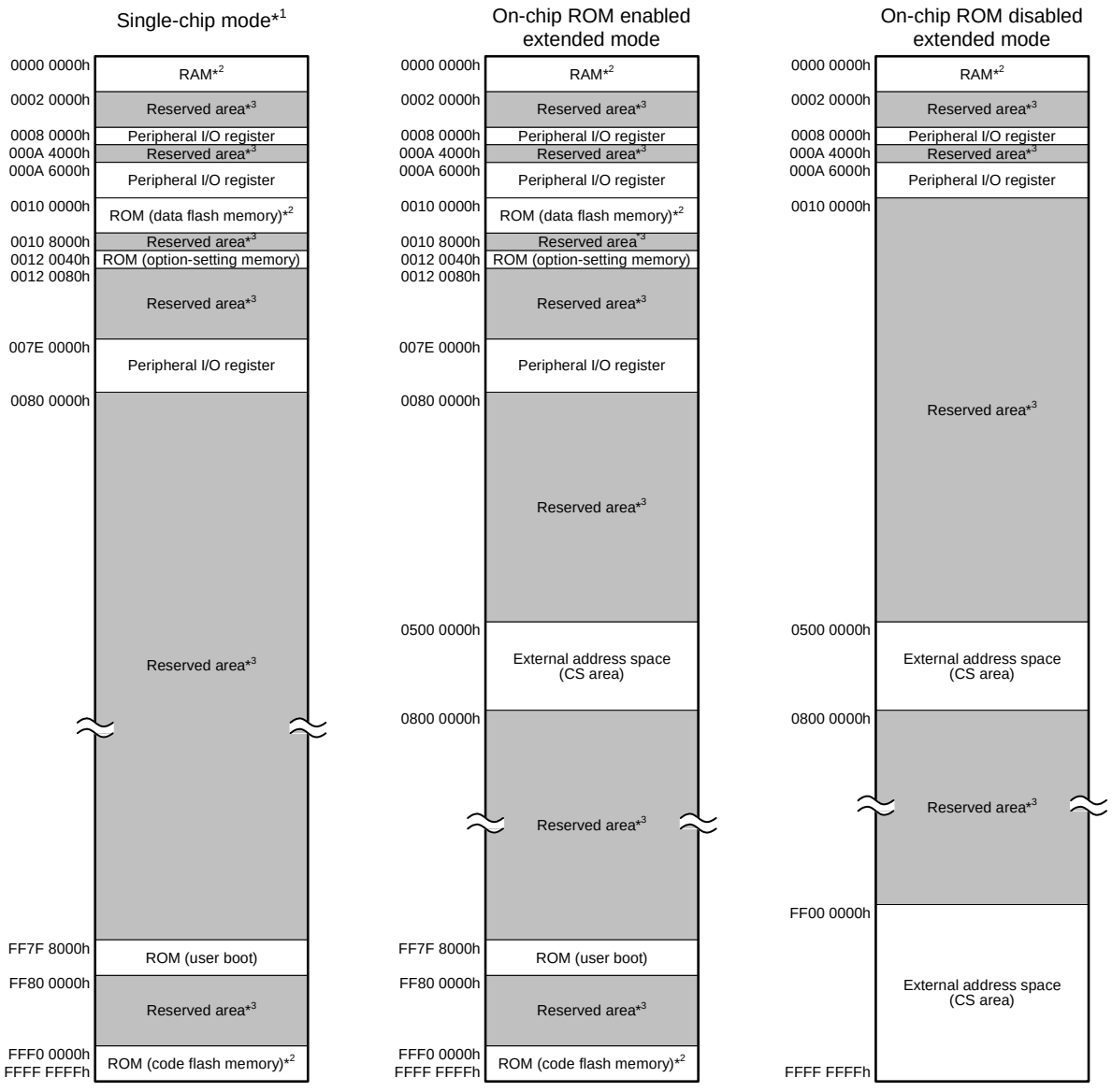


Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes

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. 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 memory map in boot mode and user boot mode is the same as that in single-chip mode.
Note 2. 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**
1 Mbyte	FFF0 0000h to FFFF FFFFh	32 Kbytes	0010 0000h to 0010 7FFFh	128 Kbytes	0000 0000h to 0001 FFFFh
512 Kbytes	FFF8 0000h to FFFF FFFFh				

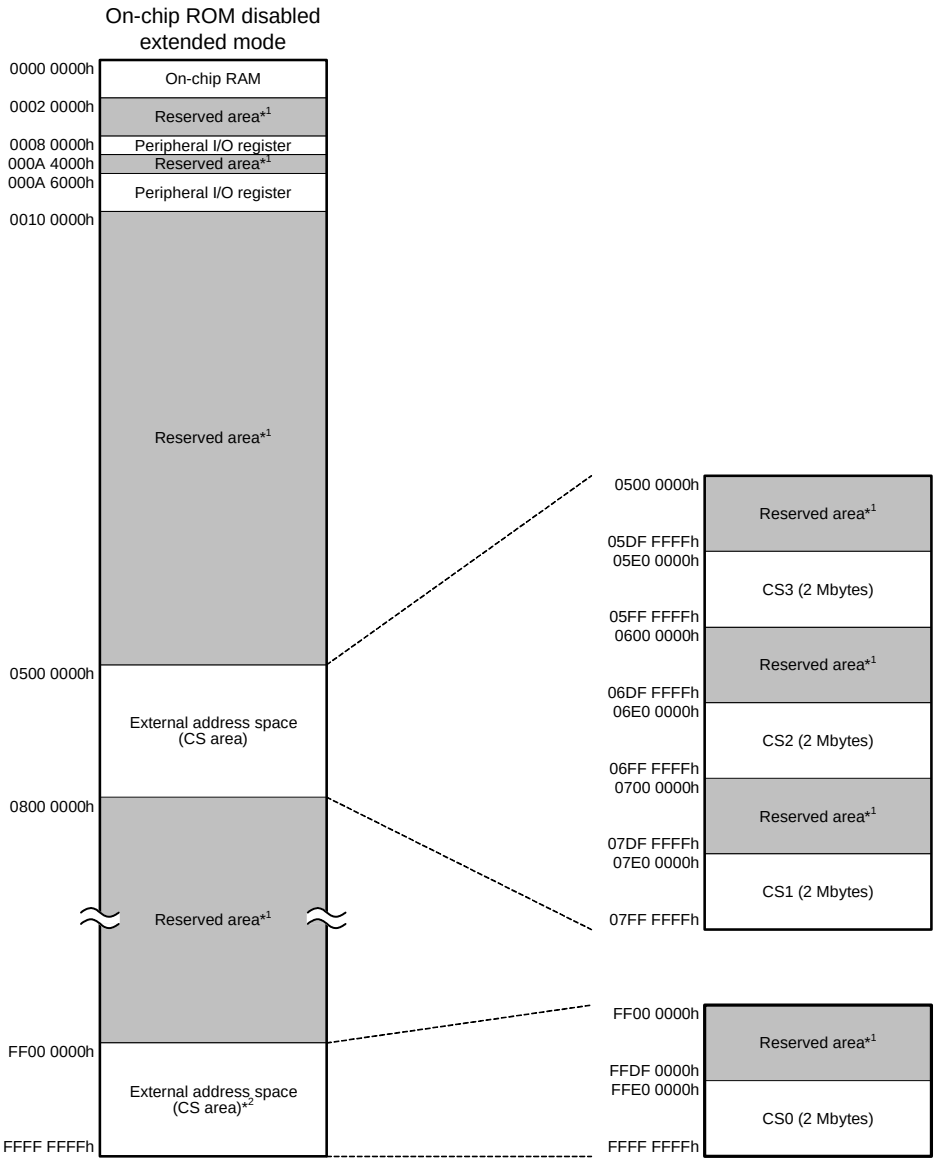
Note 3. Reserved areas should not be accessed.

Figure 4.1 Memory Map in Each Operating Mode

4.2 External Address Space

The external address space is divided into four CS areas (CS0 to CS3), each corresponding to the CSn# signal output from a CSn# (n = 0 to 3) pin.

Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS3) 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 memory map for the addresses above 0800 0000h is as shown in Figure 4.1.

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 / 43)

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 0002h	SYSTEM	Mode Status Register	MDSR	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 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 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 12
0008 1200h	RAM	RAM Operating Mode Control Register	RAMMODE	8	8	2 ICLK		section 43
0008 1201h	RAM	RAM Error Status Register	RAMSTS	8	8	2 ICLK		section 43
0008 1204h	RAM	RAM Protection Register	RAMPRCR	8	8	2 ICLK		section 43
0008 1208h	RAM	RAM Error Address Capture Register	RAMECAD	32	32	2 ICLK		section 43
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK		section 15
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK		section 15
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK		section 15
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK		section 15
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK		section 15
0008 2000h	DMAC0	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2004h	DMAC0	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17

Table 5.1 List of I/O Registers (Address Order) (2 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 2008h	DMAC0	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 200Ch	DMAC0	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2010h	DMAC0	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2013h	DMAC0	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2014h	DMAC0	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 2018h	DMAC0	DMA Offset Register	DMOFR	32	32	2 ICLK		section 17
0008 201Ch	DMAC0	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 201Dh	DMAC0	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 201Eh	DMAC0	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 201Fh	DMAC0	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2040h	DMAC1	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2044h	DMAC1	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2048h	DMAC1	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 204Ch	DMAC1	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2050h	DMAC1	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2053h	DMAC1	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2054h	DMAC1	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 205Ch	DMAC1	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 205Dh	DMAC1	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 205Eh	DMAC1	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 205Fh	DMAC1	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2080h	DMAC2	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2084h	DMAC2	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2088h	DMAC2	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 208Ch	DMAC2	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2090h	DMAC2	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2093h	DMAC2	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2094h	DMAC2	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 209Ch	DMAC2	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 209Dh	DMAC2	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 209Eh	DMAC2	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 209Fh	DMAC2	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 20C0h	DMAC3	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 20C4h	DMAC3	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 20C8h	DMAC3	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 20CCh	DMAC3	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 20D0h	DMAC3	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 20D3h	DMAC3	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 20D4h	DMAC3	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 20DCh	DMAC3	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 20DDh	DMAC3	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 20DEh	DMAC3	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 20DFh	DMAC3	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2100h	DMAC4	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2104h	DMAC4	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2108h	DMAC4	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 210Ch	DMAC4	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2110h	DMAC4	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2113h	DMAC4	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2114h	DMAC4	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17

Table 5.1 List of I/O Registers (Address Order) (3 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 211Ch	DMAC4	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 211Dh	DMAC4	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 211Eh	DMAC4	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 211Fh	DMAC4	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2140h	DMAC5	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2144h	DMAC5	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2148h	DMAC5	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 214Ch	DMAC5	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2150h	DMAC5	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2153h	DMAC5	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2154h	DMAC5	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 215Ch	DMAC5	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 215Dh	DMAC5	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 215Eh	DMAC5	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 215Fh	DMAC5	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2180h	DMAC6	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2184h	DMAC6	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2188h	DMAC6	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 218Ch	DMAC6	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2190h	DMAC6	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2193h	DMAC6	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2194h	DMAC6	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 219Ch	DMAC6	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 219Dh	DMAC6	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 219Eh	DMAC6	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 219Fh	DMAC6	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 21C0h	DMAC7	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 21C4h	DMAC7	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 21C8h	DMAC7	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 21CCh	DMAC7	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 21D0h	DMAC7	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 21D3h	DMAC7	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 21D4h	DMAC7	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 21DCh	DMAC7	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 21DDh	DMAC7	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 21DEh	DMAC7	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 21DFh	DMAC7	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2200h	DMAC	DMAC Module Start Register	DMAST	8	8	2 ICLK		section 17
0008 2204h	DMAC	DMAC74 Interrupt Status Monitor Register	DMIST	8	8	2 ICLK		section 17
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK		section 18
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK		section 18
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK		section 18
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK		section 18
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK		section 18
0008 2410h	DTC	DTC Index Table Base Register	DTCIBR	32	32	2 ICLK		section 18
0008 2414h	DTC	DTC Operation Register	DTCOR	8	8	2 ICLK		section 18
0008 2416h	DTC	DTC Sequence Transfer Enable Register	DTCSQE	16	16	2 ICLK		section 18
0008 2418h	DTC	DTC Address Displacement Register	DTCDISP	32	32	2 ICLK		section 18
0008 3002h	BSC	CS0 Mode Register	CS0MOD	16	16	1, 2 BCLK		section 15
0008 3004h	BSC	CS0 Wait Control Register 1	CS0WCR1	32	32	1, 2 BCLK		section 15

Table 5.1 List of I/O Registers (Address Order) (4 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 3008h	BSC	CS0 Wait Control Register 2	CS0WCR2	32	32	1, 2 BCLK		section 15
0008 3012h	BSC	CS1 Mode Register	CS1MOD	16	16	1, 2 BCLK		section 15
0008 3014h	BSC	CS1 Wait Control Register 1	CS1WCR1	32	32	1, 2 BCLK		section 15
0008 3018h	BSC	CS1 Wait Control Register 2	CS1WCR2	32	32	1, 2 BCLK		section 15
0008 3022h	BSC	CS2 Mode Register	CS2MOD	16	16	1, 2 BCLK		section 15
0008 3024h	BSC	CS2 Wait Control Register 1	CS2WCR1	32	32	1, 2 BCLK		section 15
0008 3028h	BSC	CS2 Wait Control Register 2	CS2WCR2	32	32	1, 2 BCLK		section 15
0008 3032h	BSC	CS3 Mode Register	CS3MOD	16	16	1, 2 BCLK		section 15
0008 3034h	BSC	CS3 Wait Control Register 1	CS3WCR1	32	32	1, 2 BCLK		section 15
0008 3038h	BSC	CS3 Wait Control Register 2	CS3WCR2	32	32	1, 2 BCLK		section 15
0008 3802h	BSC	CS0 Control Register	CS0CR	16	16	1, 2 BCLK		section 15
0008 380Ah	BSC	CS0 Recovery Cycle Register	CS0REC	16	16	1, 2 BCLK		section 15
0008 3812h	BSC	CS1 Control Register	CS1CR	16	16	1, 2 BCLK		section 15
0008 381Ah	BSC	CS1 Recovery Cycle Register	CS1REC	16	16	1, 2 BCLK		section 15
0008 3822h	BSC	CS2 Control Register	CS2CR	16	16	1, 2 BCLK		section 15
0008 382Ah	BSC	CS2 Recovery Cycle Register	CS2REC	16	16	1, 2 BCLK		section 15
0008 3832h	BSC	CS3 Control Register	CS3CR	16	16	1, 2 BCLK		section 15
0008 383Ah	BSC	CS3 Recovery Cycle Register	CS3REC	16	16	1, 2 BCLK		section 15
0008 3880h	BSC	CS Recovery Cycle Insertion Enable Register	CSRECEN	16	16	1, 2 BCLK		section 15
0008 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK		section 16
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK		section 16
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK		section 16
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK		section 16
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK		section 16
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK		section 16
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK		section 16
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK		section 16
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK		section 16
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK		section 16
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK		section 16
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK		section 16
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK		section 16
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK		section 16
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK		section 16
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK		section 16
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK		section 16
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK		section 16
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK		section 16
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK		section 16
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK		section 16
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK		section 16
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK		section 16
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK		section 16
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK		section 16
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK		section 16
0008 7010h to 0008 70FFh	ICU	Interrupt Request Register 016 to Interrupt Request Register 255	IR016 to IR255	8	8	2 ICLK		section 14
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 14
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 14

Table 5.1 List of I/O Registers (Address Order) (5 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	8	8	2 ICLK		section 14
0008 72E1h	ICU	Software Interrupt 2 Generation Register	SWINT2R	8	8	2 ICLK		section 14
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK		section 14
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 14
0008 7400h	ICU	DMAC Trigger Select Register 0	DMRSR0	8	8	2 ICLK		section 14
0008 7404h	ICU	DMAC Trigger Select Register 1	DMRSR1	8	8	2 ICLK		section 14
0008 7408h	ICU	DMAC Trigger Select Register 2	DMRSR2	8	8	2 ICLK		section 14
0008 740Ch	ICU	DMAC Trigger Select Register 3	DMRSR3	8	8	2 ICLK		section 14
0008 7410h	ICU	DMAC Trigger Select Register 4	DMRSR4	8	8	2 ICLK		section 14
0008 7414h	ICU	DMAC Trigger Select Register 5	DMRSR5	8	8	2 ICLK		section 14
0008 7418h	ICU	DMAC Trigger Select Register 6	DMRSR6	8	8	2 ICLK		section 14
0008 741Ch	ICU	DMAC Trigger Select Register 7	DMRSR7	8	8	2 ICLK		section 14
0008 7500h to 0008 750Fh	ICU	IRQ Control Register 0 to IRQ Control Register 15	IRQCR0 to IRQCR15	8	8	2 ICLK		section 14
0008 7520h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		section 14
0008 7521h	ICU	IRQ Pin Digital Filter Enable Register 1	IRQFLTE1	8	8	2 ICLK		section 14
0008 7528h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK		section 14
0008 752Ah	ICU	IRQ Pin Digital Filter Setting Register 1	IRQFLTC1	16	16	2 ICLK		section 14
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK		section 14
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK		section 14
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK		section 14
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK		section 14
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK		section 14
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK		section 14
0008 7630h	ICU	Group BL0 Interrupt Request Register	GRPBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7634h	ICU	Group BL1 Interrupt Request Register	GRPBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7638h	ICU	Group BL2 Interrupt Request Register	GRPBL2	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7670h	ICU	Group BL0 Interrupt Request Enable Register	GENBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7674h	ICU	Group BL1 Interrupt Request Enable Register	GENBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7678h	ICU	Group BL2 Interrupt Request Enable Register	GENBL2	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7700h	ICU	Software Configurable Interrupt B Request Register 0	PIBR0	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7701h	ICU	Software Configurable Interrupt B Request Register 1	PIBR1	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7705h	ICU	Software Configurable Interrupt B Request Register 5	PIBR5	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7706h	ICU	Software Configurable Interrupt B Request Register 6	PIBR6	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7708h	ICU	Software Configurable Interrupt B Request Register 8	PIBR8	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7709h	ICU	Software Configurable Interrupt B Request Register 9	PIBR9	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 770Ah	ICU	Software Configurable Interrupt B Request Register A	PIBRA	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 770Ch	ICU	Software Configurable Interrupt B Request Register C	PIBRC	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 770Dh	ICU	Software Configurable Interrupt B Request Register D	PIBRD	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7780h	ICU	Software Configurable Interrupt B Source Select Register X128	SLIBXR128	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7781h	ICU	Software Configurable Interrupt B Source Select Register X129	SLIBXR129	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (6 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 7782h	ICU	Software Configurable Interrupt B Source Select Register X130	SLIBXR130	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7783h	ICU	Software Configurable Interrupt B Source Select Register X131	SLIBXR131	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7784h	ICU	Software Configurable Interrupt B Source Select Register X132	SLIBXR132	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7785h	ICU	Software Configurable Interrupt B Source Select Register X133	SLIBXR133	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7786h	ICU	Software Configurable Interrupt B Source Select Register X134	SLIBXR134	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7787h	ICU	Software Configurable Interrupt B Source Select Register X135	SLIBXR135	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7788h	ICU	Software Configurable Interrupt B Source Select Register X136	SLIBXR136	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7789h	ICU	Software Configurable Interrupt B Source Select Register X137	SLIBXR137	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 778Ah	ICU	Software Configurable Interrupt B Source Select Register X138	SLIBXR138	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 778Bh	ICU	Software Configurable Interrupt B Source Select Register X139	SLIBXR139	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 778Ch	ICU	Software Configurable Interrupt B Source Select Register X140	SLIBXR140	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 778Dh	ICU	Software Configurable Interrupt B Source Select Register X141	SLIBXR141	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 778Eh	ICU	Software Configurable Interrupt B Source Select Register X142	SLIBXR142	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 778Fh	ICU	Software Configurable Interrupt B Source Select Register X143	SLIBXR143	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7790h	ICU	Software Configurable Interrupt B Source Select Register 144	SLIBR144	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7791h	ICU	Software Configurable Interrupt B Source Select Register 145	SLIBR145	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7792h	ICU	Software Configurable Interrupt B Source Select Register 146	SLIBR146	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7793h	ICU	Software Configurable Interrupt B Source Select Register 147	SLIBR147	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7794h	ICU	Software Configurable Interrupt B Source Select Register 148	SLIBR148	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7795h	ICU	Software Configurable Interrupt B Source Select Register 149	SLIBR149	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7796h	ICU	Software Configurable Interrupt B Source Select Register 150	SLIBR150	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7797h	ICU	Software Configurable Interrupt B Source Select Register 151	SLIBR151	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7798h	ICU	Software Configurable Interrupt B Source Select Register 152	SLIBR152	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7799h	ICU	Software Configurable Interrupt B Source Select Register 153	SLIBR153	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 779Ah	ICU	Software Configurable Interrupt B Source Select Register 154	SLIBR154	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 779Bh	ICU	Software Configurable Interrupt B Source Select Register 155	SLIBR155	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 779Ch	ICU	Software Configurable Interrupt B Source Select Register 156	SLIBR156	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 779Dh	ICU	Software Configurable Interrupt B Source Select Register 157	SLIBR157	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 779Eh	ICU	Software Configurable Interrupt B Source Select Register 158	SLIBR158	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 779Fh	ICU	Software Configurable Interrupt B Source Select Register 159	SLIBR159	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A0h	ICU	Software Configurable Interrupt B Source Select Register 160	SLIBR160	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (7 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 77A1h	ICU	Software Configurable Interrupt B Source Select Register 161	SLIBR161	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A2h	ICU	Software Configurable Interrupt B Source Select Register 162	SLIBR162	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A3h	ICU	Software Configurable Interrupt B Source Select Register 163	SLIBR163	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A4h	ICU	Software Configurable Interrupt B Source Select Register 164	SLIBR164	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A5h	ICU	Software Configurable Interrupt B Source Select Register 165	SLIBR165	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A6h	ICU	Software Configurable Interrupt B Source Select Register 166	SLIBR166	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A7h	ICU	Software Configurable Interrupt B Source Select Register 167	SLIBR167	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A8h	ICU	Software Configurable Interrupt B Source Select Register 168	SLIBR168	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77A9h	ICU	Software Configurable Interrupt B Source Select Register 169	SLIBR169	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77AAh	ICU	Software Configurable Interrupt B Source Select Register 170	SLIBR170	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77ABh	ICU	Software Configurable Interrupt B Source Select Register 171	SLIBR171	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77ACh	ICU	Software Configurable Interrupt B Source Select Register 172	SLIBR172	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77ADh	ICU	Software Configurable Interrupt B Source Select Register 173	SLIBR173	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77AEh	ICU	Software Configurable Interrupt B Source Select Register 174	SLIBR174	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77AFh	ICU	Software Configurable Interrupt B Source Select Register 175	SLIBR175	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B0h	ICU	Software Configurable Interrupt B Source Select Register 176	SLIBR176	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B1h	ICU	Software Configurable Interrupt B Source Select Register 177	SLIBR177	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B2h	ICU	Software Configurable Interrupt B Source Select Register 178	SLIBR178	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B3h	ICU	Software Configurable Interrupt B Source Select Register 179	SLIBR179	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B4h	ICU	Software Configurable Interrupt B Source Select Register 180	SLIBR180	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B5h	ICU	Software Configurable Interrupt B Source Select Register 181	SLIBR181	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B6h	ICU	Software Configurable Interrupt B Source Select Register 182	SLIBR182	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B7h	ICU	Software Configurable Interrupt B Source Select Register 183	SLIBR183	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B8h	ICU	Software Configurable Interrupt B Source Select Register 184	SLIBR184	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77B9h	ICU	Software Configurable Interrupt B Source Select Register 185	SLIBR185	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77BAh	ICU	Software Configurable Interrupt B Source Select Register 186	SLIBR186	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77BBh	ICU	Software Configurable Interrupt B Source Select Register 187	SLIBR187	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77BCh	ICU	Software Configurable Interrupt B Source Select Register 188	SLIBR188	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77BDh	ICU	Software Configurable Interrupt B Source Select Register 189	SLIBR189	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77BEh	ICU	Software Configurable Interrupt B Source Select Register 190	SLIBR190	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77BFh	ICU	Software Configurable Interrupt B Source Select Register 191	SLIBR191	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (8 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 77C0h	ICU	Software Configurable Interrupt B Source Select Register 192	SLIBR192	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C1h	ICU	Software Configurable Interrupt B Source Select Register 193	SLIBR193	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C2h	ICU	Software Configurable Interrupt B Source Select Register 194	SLIBR194	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C3h	ICU	Software Configurable Interrupt B Source Select Register 195	SLIBR195	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C4h	ICU	Software Configurable Interrupt B Source Select Register 196	SLIBR196	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C5h	ICU	Software Configurable Interrupt B Source Select Register 197	SLIBR197	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C6h	ICU	Software Configurable Interrupt B Source Select Register 198	SLIBR198	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C7h	ICU	Software Configurable Interrupt B Source Select Register 199	SLIBR199	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C8h	ICU	Software Configurable Interrupt B Source Select Register 200	SLIBR200	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77C9h	ICU	Software Configurable Interrupt B Source Select Register 201	SLIBR201	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77CAh	ICU	Software Configurable Interrupt B Source Select Register 202	SLIBR202	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77CBh	ICU	Software Configurable Interrupt B Source Select Register 203	SLIBR203	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77CCh	ICU	Software Configurable Interrupt B Source Select Register 204	SLIBR204	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77CDh	ICU	Software Configurable Interrupt B Source Select Register 205	SLIBR205	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77CEh	ICU	Software Configurable Interrupt B Source Select Register 206	SLIBR206	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 77CFh	ICU	Software Configurable Interrupt B Source Select Register 207	SLIBR207	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7830h	ICU	Group AL0 Interrupt Request Register	GRPAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7870h	ICU	Group AL0 Interrupt Request Enable Register	GENAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7900h	ICU	Software Configurable Interrupt A Request Register 0	PIAR0	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7901h	ICU	Software Configurable Interrupt A Request Register 1	PIAR1	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7902h	ICU	Software Configurable Interrupt A Request Register 2	PIAR2	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7903h	ICU	Software Configurable Interrupt A Request Register 3	PIAR3	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7904h	ICU	Software Configurable Interrupt A Request Register 4	PIAR4	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7905h	ICU	Software Configurable Interrupt A Request Register 5	PIAR5	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Bh	ICU	Software Configurable Interrupt A Request Register B	PIARB	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Ch	ICU	Software Configurable Interrupt A Request Register C	PIARC	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D0h	ICU	Software Configurable Interrupt A Source Select Register 208	SLIAR208	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D1h	ICU	Software Configurable Interrupt A Source Select Register 209	SLIAR209	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D2h	ICU	Software Configurable Interrupt A Source Select Register 210	SLIAR210	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D3h	ICU	Software Configurable Interrupt A Source Select Register 211	SLIAR211	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D4h	ICU	Software Configurable Interrupt A Source Select Register 212	SLIAR212	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D5h	ICU	Software Configurable Interrupt A Source Select Register 213	SLIAR213	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (9 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 79D6h	ICU	Software Configurable Interrupt A Source Select Register 214	SLIAR214	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D7h	ICU	Software Configurable Interrupt A Source Select Register 215	SLIAR215	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D8h	ICU	Software Configurable Interrupt A Source Select Register 216	SLIAR216	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D9h	ICU	Software Configurable Interrupt A Source Select Register 217	SLIAR217	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DAh	ICU	Software Configurable Interrupt A Source Select Register 218	SLIAR218	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DBh	ICU	Software Configurable Interrupt A Source Select Register 219	SLIAR219	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DCh	ICU	Software Configurable Interrupt A Source Select Register 220	SLIAR220	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DDh	ICU	Software Configurable Interrupt A Source Select Register 221	SLIAR221	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DEh	ICU	Software Configurable Interrupt A Source Select Register 222	SLIAR222	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DFh	ICU	Software Configurable Interrupt A Source Select Register 223	SLIAR223	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E0h	ICU	Software Configurable Interrupt A Source Select Register 224	SLIAR224	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E1h	ICU	Software Configurable Interrupt A Source Select Register 225	SLIAR225	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E2h	ICU	Software Configurable Interrupt A Source Select Register 226	SLIAR226	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E3h	ICU	Software Configurable Interrupt A Source Select Register 227	SLIAR227	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E4h	ICU	Software Configurable Interrupt A Source Select Register 228	SLIAR228	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E5h	ICU	Software Configurable Interrupt A Source Select Register 229	SLIAR229	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E6h	ICU	Software Configurable Interrupt A Source Select Register 230	SLIAR230	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E7h	ICU	Software Configurable Interrupt A Source Select Register 231	SLIAR231	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E8h	ICU	Software Configurable Interrupt A Source Select Register 232	SLIAR232	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E9h	ICU	Software Configurable Interrupt A Source Select Register 233	SLIAR233	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EAh	ICU	Software Configurable Interrupt A Source Select Register 234	SLIAR234	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EBh	ICU	Software Configurable Interrupt A Source Select Register 235	SLIAR235	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79ECh	ICU	Software Configurable Interrupt A Source Select Register 236	SLIAR236	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EDh	ICU	Software Configurable Interrupt A Source Select Register 237	SLIAR237	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EEh	ICU	Software Configurable Interrupt A Source Select Register 238	SLIAR238	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EFh	ICU	Software Configurable Interrupt A Source Select Register 239	SLIAR239	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F0h	ICU	Software Configurable Interrupt A Source Select Register 240	SLIAR240	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F1h	ICU	Software Configurable Interrupt A Source Select Register 241	SLIAR241	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F2h	ICU	Software Configurable Interrupt A Source Select Register 242	SLIAR242	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F3h	ICU	Software Configurable Interrupt A Source Select Register 243	SLIAR243	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F4h	ICU	Software Configurable Interrupt A Source Select Register 244	SLIAR244	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (10 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 79F5h	ICU	Software Configurable Interrupt A Source Select Register 245	SLIAR245	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F6h	ICU	Software Configurable Interrupt A Source Select Register 246	SLIAR246	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F7h	ICU	Software Configurable Interrupt A Source Select Register 247	SLIAR247	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F8h	ICU	Software Configurable Interrupt A Source Select Register 248	SLIAR248	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F9h	ICU	Software Configurable Interrupt A Source Select Register 249	SLIAR249	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FAh	ICU	Software Configurable Interrupt A Source Select Register 250	SLIAR250	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FBh	ICU	Software Configurable Interrupt A Source Select Register 251	SLIAR251	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FCh	ICU	Software Configurable Interrupt A Source Select Register 252	SLIAR252	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FDh	ICU	Software Configurable Interrupt A Source Select Register 253	SLIAR253	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FEh	ICU	Software Configurable Interrupt A Source Select Register 254	SLIAR254	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FFh	ICU	Software Configurable Interrupt A Source Select Register 255	SLIAR255	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7A00h	ICU	Software Configurable Interrupt Source Select Register Write Protect Register	SLIPRCR	8	8	2 ICLK to 1 PCLKA/B	2 ICLK	section 14
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	section 29
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 29
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK	section 29
0008 8040h	DA	D/A Data Register 0	DADR0	16	16	2, 3 PCLKB	2 ICLK	section 39
0008 8042h	DA	D/A Data Register 1	DADR1	16	16	2, 3 PCLKB	2 ICLK	section 39
0008 8044h	DA	D/A Control Register	DACR	8	8	2, 3 PCLKB	2 ICLK	section 39
0008 8045h	DA	Data Register Format Select Register	DADPR	8	8	2, 3 PCLKB	2 ICLK	section 39
0008 8046h	DA	D/A A/D Synchronous Start Control Register	DAADSCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0008 8049h	DA	D/A Destination Select Register	DADSELR	8	8	2, 3 PCLKB	2 ICLK	section 39
0008 8084h	S12AD	A/D Reference Voltage Control Register	ADVREFCR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (11 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8204h	TMR01	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8206h	TMR01	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8208h	TMR01	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 8209h	TMR1	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 820Ah	TMR01	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 820Ch	TMR0	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 820Dh	TMR1	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8214h	TMR23	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8216h	TMR23	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8218h	TMR23	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 8219h	TMR3	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 821Ah	TMR23	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 24
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 821Ch	TMR2	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 821Dh	TMR3	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 24
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 8284h	CRC	CRC Data Input Register	CRCDIR	32	8, 32	2, 3 PCLKB	2 ICLK	section 35
0008 8288h	CRC	CRC Data Output Register	CRCDOR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 35
0008 8300h	RIIC0	I²C-bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8301h	RIIC0	I²C-bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8302h	RIIC0	I²C-bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8303h	RIIC0	I²C-bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8304h	RIIC0	I²C-bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8305h	RIIC0	I²C-bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8306h	RIIC0	I²C-bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8307h	RIIC0	I²C-bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8308h	RIIC0	I²C-bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8309h	RIIC0	I²C-bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 32

Table 5.1 List of I/O Registers (Address Order) (12 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8310h	RIIC0	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8311h	RIIC0	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8312h	RIIC0	I²C-bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8313h	RIIC0	I²C-bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8340h	RIIC2	I²C-bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8341h	RIIC2	I²C-bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8342h	RIIC2	I²C-bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8343h	RIIC2	I²C-bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8344h	RIIC2	I²C-bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8345h	RIIC2	I²C-bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8346h	RIIC2	I²C-bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8347h	RIIC2	I²C-bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8348h	RIIC2	I²C-bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8349h	RIIC2	I²C-bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 834Ah	RIIC2	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 834Bh	RIIC2	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 834Ch	RIIC2	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 834Dh	RIIC2	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 834Eh	RIIC2	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 834Fh	RIIC2	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8350h	RIIC2	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8351h	RIIC2	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8352h	RIIC2	I²C-bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8353h	RIIC2	I²C-bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9006h	S12AD	A/D Channel Select Register A1	ADANSA1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 900Ah	S12AD	A/D-Converted Value Addition/Average Function Channel Select Register 1	ADADS1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9012h	S12AD	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9016h	S12AD	A/D Channel Select Register B1	ADANSB1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 901Ah	S12AD	A/D Temperature Sensor Data Register	ADTSDR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 38

Table 5.1 List of I/O Registers (Address Order) (13 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9030h	S12AD	A/D Data Register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9032h	S12AD	A/D Data Register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9034h	S12AD	A/D Data Register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9036h	S12AD	A/D Data Register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9038h	S12AD	A/D Data Register 12	ADDR12	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 903Ah	S12AD	A/D Data Register 13	ADDR13	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 903Ch	S12AD	A/D Data Register 14	ADDR14	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 903Eh	S12AD	A/D Data Register 15	ADDR15	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9040h	S12AD	A/D Data Register 16	ADDR16	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9042h	S12AD	A/D Data Register 17	ADDR17	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9044h	S12AD	A/D Data Register 18	ADDR18	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9046h	S12AD	A/D Data Register 19	ADDR19	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9048h	S12AD	A/D Data Register 20	ADDR20	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 904Ah	S12AD	A/D Data Register 21	ADDR21	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 904Ch	S12AD	A/D Data Register 22	ADDR22	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 904Eh	S12AD	A/D Data Register 23	ADDR23	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 907Dh	S12AD	A/D Event Link Control Register	ADELCCR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9086h	S12AD	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 908Ch	S12AD	A/D Comparison Function Window A/B Status Monitoring Register	ADWINMON	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 9090h	S12AD	A/D Comparison Function Control Register	ADCMPCR	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9092h	S12AD	A/D Comparison Function Window A Extended Input Select Register	ADCMPANSER	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 9093h	S12AD	A/D Comparison Function Window A Extended Input Comparison Condition Setting Register	ADCMPLER	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 9094h	S12AD	A/D Comparison Function Window A Channel Select Register 0	ADCMPANSR0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9096h	S12AD	A/D Comparison Function Window A Channel Select Register 1	ADCMPANSR1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 9098h	S12AD	A/D Comparison Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 909Ah	S12AD	A/D Comparison Function Window A Comparison Condition Setting Register 1	ADCMPLR1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 909Ch	S12AD	A/D Comparison Function Window A Lower Level Setting Register	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 909Eh	S12AD	A/D Comparison Function Window A Upper Level Setting Register	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 90A0h	S12AD	A/D Comparison Function Window A Channel Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 90A2h	S12AD	A/D Comparison Function Window A Channel Status Register 1	ADCMPSR1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 90A4h	S12AD	A/D Comparison Function Window A Extended Input Channel Status Register	ADCMPSER	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90A6h	S12AD	A/D Comparison Function Window B Channel Select Register	ADCMPBNSR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90A8h	S12AD	A/D Comparison Function Window B Lower Level Setting Register	ADWINLLB	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 90AAh	S12AD	A/D Comparison Function Window B Upper Level Setting Register	ADWINULB	16	16	2, 3 PCLKB	2 ICLK	section 38

Table 5.1 List of I/O Registers (Address Order) (14 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 90ACh	S12AD	A/D Comparison Function Window B Channel Status Register	ADCMPBSR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90D4h	S12AD	A/D Channel Select Register C0	ADANSC0	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 90D6h	S12AD	A/D Channel Select Register C1	ADANSC1	16	16	2, 3 PCLKB	2 ICLK	section 38
0008 90D8h	S12AD	A/D Group C Extended Input Control Register	ADGCEXCR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90D9h	S12AD	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90DDh	S12AD	A/D Sampling State Register L	ADSSTRL	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90DEh	S12AD	A/D Sampling State Register T	ADSSTRT	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90DFh	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E7h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E8h	S12AD	A/D Sampling State Register 8	ADSSTR8	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90E9h	S12AD	A/D Sampling State Register 9	ADSSTR9	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90EAh	S12AD	A/D Sampling State Register 10	ADSSTR10	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90EBh	S12AD	A/D Sampling State Register 11	ADSSTR11	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90ECh	S12AD	A/D Sampling State Register 12	ADSSTR12	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90EDh	S12AD	A/D Sampling State Register 13	ADSSTR13	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90EEh	S12AD	A/D Sampling State Register 14	ADSSTR14	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 90EFh	S12AD	A/D Sampling State Register 15	ADSSTR15	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C0h	S12AD	A/D Channel Conversion Order Setting Register 0	ADSCS0	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C1h	S12AD	A/D Channel Conversion Order Setting Register 1	ADSCS1	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C2h	S12AD	A/D Channel Conversion Order Setting Register 2	ADSCS2	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C3h	S12AD	A/D Channel Conversion Order Setting Register 3	ADSCS3	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C4h	S12AD	A/D Channel Conversion Order Setting Register 4	ADSCS4	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C5h	S12AD	A/D Channel Conversion Order Setting Register 5	ADSCS5	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C6h	S12AD	A/D Channel Conversion Order Setting Register 6	ADSCS6	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C7h	S12AD	A/D Channel Conversion Order Setting Register 7	ADSCS7	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C8h	S12AD	A/D Channel Conversion Order Setting Register 8	ADSCS8	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91C9h	S12AD	A/D Channel Conversion Order Setting Register 9	ADSCS9	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91CAh	S12AD	A/D Channel Conversion Order Setting Register 10	ADSCS10	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91CBh	S12AD	A/D Channel Conversion Order Setting Register 11	ADSCS11	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91CCh	S12AD	A/D Channel Conversion Order Setting Register 12	ADSCS12	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91CDh	S12AD	A/D Channel Conversion Order Setting Register 13	ADSCS13	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91CEh	S12AD	A/D Channel Conversion Order Setting Register 14	ADSCS14	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91CFh	S12AD	A/D Channel Conversion Order Setting Register 15	ADSCS15	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D0h	S12AD	A/D Channel Conversion Order Setting Register 16	ADSCS16	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D1h	S12AD	A/D Channel Conversion Order Setting Register 17	ADSCS17	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D2h	S12AD	A/D Channel Conversion Order Setting Register 18	ADSCS18	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D3h	S12AD	A/D Channel Conversion Order Setting Register 19	ADSCS19	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D4h	S12AD	A/D Channel Conversion Order Setting Register 20	ADSCS20	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D5h	S12AD	A/D Channel Conversion Order Setting Register 21	ADSCS21	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D6h	S12AD	A/D Channel Conversion Order Setting Register 22	ADSCS22	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91D7h	S12AD	A/D Channel Conversion Order Setting Register 23	ADSCS23	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 91E2h	S12AD	A/D Internal Reference Voltage Monitoring Circuit Enable Register	ADVMONCR	8	8	2, 3 PCLKB	2 ICLK	section 38

Table 5.1 List of I/O Registers (Address Order) (15 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 91E4h	S12AD	A/D Internal Reference Voltage Monitoring Circuit Output Enable Register	ADVMONO	8	8	2, 3 PCLKB	2 ICLK	section 38
0008 A000h	SCI0	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A000h	SMCI0	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A001h	SCI0	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A002h	SCI0	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A002h	SMCI0	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A003h	SCI0	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A004h	SCI0	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A004h	SMCI0	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A005h	SCI0	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A006h	SCI0	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A006h	SMCI0	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A007h	SCI0	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A008h	SCI0	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A009h	SCI0	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Ah	SCI0	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Bh	SCI0	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Ch	SCI0	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Dh	SCI0	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Eh	SCI0	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Fh	SCI0	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A00Eh	SCI0	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 30
0008 A010h	SCI0	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A011h	SCI0	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A010h	SCI0	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 30
0008 A012h	SCI0	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A013h	SCI0	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A01Ah	SCI0	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A01Bh	SCI0	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A01Ah	SCI0	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 30
0008 A01Ch	SCI0	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A01Dh	SCI0	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A020h	SMCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A022h	SMCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A024h	SMCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (16 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A033h	SCI1	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A03Ah	SCI1	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A03Bh	SCI1	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A03Ah	SCI1	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A03Ch	SCI1	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A03Dh	SCI1	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A040h	SCI2	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A040h	SMCI2	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A041h	SCI2	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A042h	SCI2	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A042h	SMCI2	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A043h	SCI2	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A044h	SCI2	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A044h	SMCI2	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A045h	SCI2	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A046h	SCI2	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A046h	SMCI2	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A047h	SCI2	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A048h	SCI2	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A049h	SCI2	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Ah	SCI2	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Bh	SCI2	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Ch	SCI2	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Dh	SCI2	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Eh	SCI2	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Fh	SCI2	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A04Eh	SCI2	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A050h	SCI2	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A051h	SCI2	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A050h	SCI2	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A052h	SCI2	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A053h	SCI2	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A05Ah	SCI2	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A05Bh	SCI2	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A05Ah	SCI2	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A05Ch	SCI2	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A05Dh	SCI2	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A060h	SCI3	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A060h	SMCI3	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A061h	SCI3	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A062h	SCI3	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A062h	SMCI3	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (17 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A063h	SCI3	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A064h	SCI3	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A064h	SMCI3	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A065h	SCI3	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A066h	SCI3	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A066h	SMCI3	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A067h	SCI3	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A068h	SCI3	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A069h	SCI3	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Ah	SCI3	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Bh	SCI3	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Ch	SCI3	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Dh	SCI3	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Eh	SCI3	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Fh	SCI3	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A06Eh	SCI3	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A070h	SCI3	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A071h	SCI3	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A070h	SCI3	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A072h	SCI3	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A073h	SCI3	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A07Ah	SCI3	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A07Bh	SCI3	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A07Ah	SCI3	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A07Ch	SCI3	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A07Dh	SCI3	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A080h	SCI4	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A080h	SMCI4	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A081h	SCI4	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A082h	SCI4	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A082h	SMCI4	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A083h	SCI4	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A084h	SCI4	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A084h	SMCI4	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A085h	SCI4	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A086h	SCI4	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A086h	SMCI4	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A087h	SCI4	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A088h	SCI4	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A089h	SCI4	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Ah	SCI4	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Bh	SCI4	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Ch	SCI4	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Dh	SCI4	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Eh	SCI4	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Fh	SCI4	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A08Eh	SCI4	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A090h	SCI4	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A091h	SCI4	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A090h	SCI4	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (18 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A092h	SCI4	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A093h	SCI4	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A09Ah	SCI4	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A09Bh	SCI4	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A09Ah	SCI4	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A09Ch	SCI4	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A09Dh	SCI4	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A0h	SMCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A2h	SMCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A4h	SMCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0B3h	SCI5	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0BAh	SCI5	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0BBh	SCI5	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0BAh	SCI5	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0BCh	SCI5	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0BDh	SCI5	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C0h	SMCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C2h	SMCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C4h	SMCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C6h	SCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C6h	SMCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (19 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0C9h	SCI6	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CAh	SCI6	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CBh	SCI6	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CCh	SCI6	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CEh	SCI6	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CFh	SCI6	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0CEh	SCI6	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0D0h	SCI6	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0D1h	SCI6	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0D0h	SCI6	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0D2h	SCI6	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0D3h	SCI6	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0DAh	SCI6	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0DBh	SCI6	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0DAh	SCI6	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0DCh	SCI6	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0DDh	SCI6	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E0h	SCI7	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E0h	SMCI7	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E1h	SCI7	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E2h	SCI7	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E2h	SMCI7	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E3h	SCI7	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E4h	SCI7	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E4h	SMCI7	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E5h	SCI7	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E6h	SCI7	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E6h	SMCI7	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E7h	SCI7	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E8h	SCI7	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0E9h	SCI7	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0EAh	SCI7	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0EBh	SCI7	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0ECh	SCI7	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0EDh	SCI7	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0EEh	SCI7	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0EFh	SCI7	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0EEh	SCI7	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0F0h	SCI7	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0F1h	SCI7	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0F0h	SCI7	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0F2h	SCI7	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0F3h	SCI7	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0FAh	SCI7	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0FBh	SCI7	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0FAh	SCI7	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A0FCh	SCI7	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A0FDh	SCI7	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (20 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A100h	SCI8	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A100h	SMCI8	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A101h	SCI8	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A102h	SCI8	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A102h	SMCI8	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A103h	SCI8	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A104h	SCI8	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A104h	SMCI8	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A105h	SCI8	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A106h	SCI8	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A106h	SMCI8	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A107h	SCI8	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A108h	SCI8	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A109h	SCI8	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Ah	SCI8	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Bh	SCI8	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Ch	SCI8	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Dh	SCI8	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Eh	SCI8	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Fh	SCI8	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A10Eh	SCI8	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A110h	SCI8	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A111h	SCI8	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A110h	SCI8	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A112h	SCI8	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A113h	SCI8	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A11Ah	SCI8	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A11Bh	SCI8	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A11Ah	SCI8	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A11Ch	SCI8	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A11Dh	SCI8	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A120h	SCI9	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A120h	SMCI9	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A121h	SCI9	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A122h	SCI9	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A122h	SMCI9	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A123h	SCI9	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A124h	SCI9	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A124h	SMCI9	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A125h	SCI9	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A126h	SCI9	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A126h	SMCI9	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A127h	SCI9	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A128h	SCI9	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A129h	SCI9	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A12Ah	SCI9	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A12Bh	SCI9	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A12Ch	SCI9	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A12Dh	SCI9	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A12Eh	SCI9	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (21 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A12Fh	SCI9	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A12Eh	SCI9	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A130h	SCI9	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A131h	SCI9	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A130h	SCI9	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A132h	SCI9	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A133h	SCI9	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A13Ah	SCI9	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A13Bh	SCI9	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A13Ah	SCI9	Comparison Data Register	CDR	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 A13Ch	SCI9	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 A13Dh	SCI9	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 30
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 19
0008 B101h	ELC	Event Link Setting Register 0	ELSR0	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Ch	ELC	Event Link Setting Register 11	ELSR11	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Eh	ELC	Event Link Setting Register 13	ELSR13	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B111h	ELC	Event Link Setting Register 16	ELSR16	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2, 3 PCLKB	2 ICLK	section 19

Table 5.1 List of I/O Registers (Address Order) (22 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Eh	ELC	Event Link Setting Register 30	ELSR30	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Fh	ELC	Event Link Setting Register 31	ELSR31	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B130h	ELC	Event Link Setting Register 32	ELSR32	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B13Eh	ELC	Event Link Option Setting Register E	ELOPE	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Eh	ELC	Event Link Setting Register 56	ELSR56	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B300h	SMCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B302h	SMCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B304h	SMCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B309h	SCI12	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Ah	SCI12	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Bh	SCI12	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Ch	SCI12	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	8, 16	4, 5 PCLKB	2 ICLK	section 30
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B321h	SCI12	Control Register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B322h	SCI12	Control Register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B323h	SCI12	Control Register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B324h	SCI12	Control Register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B325h	SCI12	Port Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B327h	SCI12	Status Register	STR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (23 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C006h	PORT6	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C008h	PORT8	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Fh	PORTF	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C011h	PORTH	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C013h	PORTK	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C014h	PORTL	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C016h	PORTN	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C026h	PORT6	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C028h	PORT8	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Fh	PORTF	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C031h	PORTH	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C033h	PORTK	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C034h	PORTL	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C036h	PORTN	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C040h	PORT0	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (24 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C041h	PORT1	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C042h	PORT2	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C043h	PORT3	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C044h	PORT4	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C045h	PORT5	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C046h	PORT6	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C047h	PORT7	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C048h	PORT8	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C049h	PORT9	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C04Ah	PORTA	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C04Bh	PORTB	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C04Ch	PORTC	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C04Dh	PORTD	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C04Eh	PORTE	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C04Fh	PORTF	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C051h	PORTH	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C052h	PORTJ	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C053h	PORTK	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C054h	PORTL	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C056h	PORTN	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C066h	PORT6	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C068h	PORT8	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Fh	PORTF	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C071h	PORTH	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C073h	PORTK	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C074h	PORTL	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C076h	PORTN	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C080h	PORT0	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C081h	PORT0	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C082h	PORT1	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C083h	PORT1	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C084h	PORT2	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C085h	PORT2	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C086h	PORT3	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C087h	PORT3	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C088h	PORT4	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (25 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C089h	PORT4	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Ah	PORT5	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Bh	PORT5	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Ch	PORT6	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Dh	PORT6	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Eh	PORT7	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Fh	PORT7	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C090h	PORT8	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C091h	PORT8	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C092h	PORT9	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C094h	PORTA	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C095h	PORTA	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C096h	PORTB	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C097h	PORTB	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C098h	PORTC	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C099h	PORTC	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Ah	PORTD	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Bh	PORTD	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Ch	PORTE	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Dh	PORTE	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Fh	PORTF	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A2h	PORTH	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A3h	PORTH	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A4h	PORTJ	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A5h	PORTJ	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A6h	PORTK	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A7h	PORTK	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0A8h	PORTL	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0ADh	PORTN	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C0h	PORT0	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C1h	PORT1	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C2h	PORT2	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C3h	PORT3	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C4h	PORT4	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C5h	PORT5	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C6h	PORT6	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C7h	PORT7	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C8h	PORT8	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C9h	PORT9	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CAh	PORTA	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CBh	PORTB	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CCh	PORTC	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CDh	PORTD	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CEh	PORTE	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CFh	PORTF	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0D1h	PORTH	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0D2h	PORTJ	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0D3h	PORTK	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0D4h	PORTL	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0D6h	PORTN	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (26 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C0E0h	PORT0	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E3h	PORT3	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E5h	PORT5	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E6h	PORT6	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E7h	PORT7	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E8h	PORT8	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0ECh	PORTC	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EEh	PORTE	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EFh	PORTF	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0F1h	PORTH	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0F2h	PORTJ	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0F3h	PORTK	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0F4h	PORTL	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0F6h	PORTN	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C100h	MPC	CS Output Enable Register	PFCSE	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C102h	MPC	CS Output Pin Select Register 0	PFCSS0	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C104h	MPC	Address Output Enable Register 0	PFAOE0	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C105h	MPC	Address Output Enable Register 1	PFAOE1	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C106h	MPC	External Bus Control Register 0	PFBCR0	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C107h	MPC	External Bus Control Register 1	PFBCR1	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C108h	MPC	External Bus Control Register 2	PFBCR2	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C109h	MPC	External Bus Control Register 3	PFBCR3	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C140h	MPC	P00 Pin Function Control Register	P00PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C141h	MPC	P01 Pin Function Control Register	P01PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C142h	MPC	P02 Pin Function Control Register	P02PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C143h	MPC	P03 Pin Function Control Register	P03PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C145h	MPC	P05 Pin Function Control Register	P05PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C14Ah	MPC	P12 Pin Function Control Register	P12PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C14Bh	MPC	P13 Pin Function Control Register	P13PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C14Ch	MPC	P14 Pin Function Control Register	P14PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C14Dh	MPC	P15 Pin Function Control Register	P15PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C14Eh	MPC	P16 Pin Function Control Register	P16PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C14Fh	MPC	P17 Pin Function Control Register	P17PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2, 3 PCLKB	2 ICLK	section 21

Table 5.1 List of I/O Registers (Address Order) (27 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Bh	MPC	P33 Pin Function Control Register	P33PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Ch	MPC	P34 Pin Function Control Register	P34PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Eh	MPC	P36 Pin Function Control Register	P36PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Fh	MPC	P37 Pin Function Control Register	P37PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C168h	MPC	P50 Pin Function Control Register	P50PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C169h	MPC	P51 Pin Function Control Register	P51PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Ah	MPC	P52 Pin Function Control Register	P52PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Bh	MPC	P53 Pin Function Control Register	P53PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Ch	MPC	P54 Pin Function Control Register	P54PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Dh	MPC	P55 Pin Function Control Register	P55PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Eh	MPC	P56 Pin Function Control Register	P56PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C170h	MPC	P60 Pin Function Control Register	P60PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C171h	MPC	P61 Pin Function Control Register	P61PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C172h	MPC	P62 Pin Function Control Register	P62PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C173h	MPC	P63 Pin Function Control Register	P63PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C174h	MPC	P64 Pin Function Control Register	P64PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C175h	MPC	P65 Pin Function Control Register	P65PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C176h	MPC	P66 Pin Function Control Register	P66PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C177h	MPC	P67 Pin Function Control Register	P67PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C178h	MPC	P70 Pin Function Control Register	P70PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C179h	MPC	P71 Pin Function Control Register	P71PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Ah	MPC	P72 Pin Function Control Register	P72PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Bh	MPC	P73 Pin Function Control Register	P73PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Ch	MPC	P74 Pin Function Control Register	P74PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Dh	MPC	P75 Pin Function Control Register	P75PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Eh	MPC	P76 Pin Function Control Register	P76PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Fh	MPC	P77 Pin Function Control Register	P77PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C180h	MPC	P80 Pin Function Control Register	P80PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C181h	MPC	P81 Pin Function Control Register	P81PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C182h	MPC	P82 Pin Function Control Register	P82PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C183h	MPC	P83 Pin Function Control Register	P83PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C186h	MPC	P86 Pin Function Control Register	P86PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C187h	MPC	P87 Pin Function Control Register	P87PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C188h	MPC	P90 Pin Function Control Register	P90PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C189h	MPC	P91 Pin Function Control Register	P91PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Ah	MPC	P92 Pin Function Control Register	P92PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Bh	MPC	P93 Pin Function Control Register	P93PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C190h	MPC	PA0 Pin Function Control Register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C191h	MPC	PA1 Pin Function Control Register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21

Table 5.1 List of I/O Registers (Address Order) (28 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C196h	MPC	PA6 Pin Function Control Register	PA6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C197h	MPC	PA7 Pin Function Control Register	PA7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A0h	MPC	PC0 Pin Function Control Register	PC0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A1h	MPC	PC1 Pin Function Control Register	PC1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A2h	MPC	PC2 Pin Function Control Register	PC2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A3h	MPC	PC3 Pin Function Control Register	PC3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A4h	MPC	PC4 Pin Function Control Register	PC4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A5h	MPC	PC5 Pin Function Control Register	PC5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A6h	MPC	PC6 Pin Function Control Register	PC6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A7h	MPC	PC7 Pin Function Control Register	PC7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A8h	MPC	PD0 Pin Function Control Register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A9h	MPC	PD1 Pin Function Control Register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1AAh	MPC	PD2 Pin Function Control Register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1AFh	MPC	PD7 Pin Function Control Register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B5h	MPC	PE5 Pin Function Control Register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B6h	MPC	PE6 Pin Function Control Register	PE6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B7h	MPC	PE7 Pin Function Control Register	PE7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1BDh	MPC	PF5 Pin Function Control Register	PF5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1C8h	MPC	PH0 Pin Function Control Register	PH0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1C9h	MPC	PH1 Pin Function Control Register	PH1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1CAh	MPC	PH2 Pin Function Control Register	PH2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1CBh	MPC	PH3 Pin Function Control Register	PH3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1D1h	MPC	PJ1 Pin Function Control Register	PJ1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1D3h	MPC	PJ3 Pin Function Control Register	PJ3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1D5h	MPC	PJ5 Pin Function Control Register	PJ5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1DAh	MPC	PK2 Pin Function Control Register	PK2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1DBh	MPC	PK3 Pin Function Control Register	PK3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1DCh	MPC	PK4 Pin Function Control Register	PK4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1DDh	MPC	PK5 Pin Function Control Register	PK5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C280h	SYSTEM	Deep Standby Control Register	DPSBYCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 11

Table 5.1 List of I/O Registers (Address Order) (29 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
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 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 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 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 C292h	SYSTEM	Sub-Clock Oscillator Forced Oscillation Control Register	SOFCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9
0008 C293h	SYSTEM	Main Clock Oscillator Function 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 C295h	SYSTEM	Voltage Level Setting Register	VOLSR	8	8	4, 5 PCLKB	2, 3 ICLK	section 3
0008 C296h	FLASH	Flash P/E Protect Register	FWEPROR	8	8	4, 5 PCLKB	2, 3 ICLK	section 44
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 C2A0h to 0008 C2BFh	SYSTEM	Deep Standby Backup Register 0 to Deep Standby Backup Register 31	DPSBKR0 to DPSBKR31	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C400h	RTC	64-Hz Counter	R64CNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C402h	RTC	Second Counter	RSECCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C402h	RTC	Binary Counter 0	BCNT0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C404h	RTC	Minute Counter	RMINCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C404h	RTC	Binary Counter 1	BCNT1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C406h	RTC	Hour Counter	RHRCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C406h	RTC	Binary Counter 2	BCNT2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C408h	RTC	Day-of-Week Counter	RWKCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C408h	RTC	Binary Counter 3	BCNT3	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C40Ah	RTC	Date Counter	RDAYCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C40Ch	RTC	Month Counter	RMONCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C40Eh	RTC	Year Counter	RYRCNT	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 C410h	RTC	Second Alarm Register	RSECAR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C410h	RTC	Binary Counter 0 Alarm Register	BCNT0AR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C412h	RTC	Minute Alarm Register	RMINAR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C412h	RTC	Binary Counter 1 Alarm Register	BCNT1AR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C414h	RTC	Hour Alarm Register	RHRAR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C416h	RTC	Day-of-Week Alarm Register	RWKAR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C416h	RTC	Binary Counter 3 Alarm Register	BCNT3AR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C418h	RTC	Date Alarm Register	RDAYAR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C418h	RTC	Binary Counter 0 Alarm Enable Register	BCNT0AER	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C41Ah	RTC	Month Alarm Register	RMONAR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C41Ah	RTC	Binary Counter 1 Alarm Enable Register	BCNT1AER	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C41Ch	RTC	Year Alarm Register	RYRAR	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 C41Ch	RTC	Binary Counter 2 Alarm Enable Register	BCNT2AER	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 C41Eh	RTC	Year Alarm Enable Register	RYRAREN	8	8	2, 3 PCLKB	2 ICLK	section 27

Table 5.1 List of I/O Registers (Address Order) (30 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C41Eh	RTC	Binary Counter 3 Alarm Enable Register	BCNT3AER	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C422h	RTC	RTC Control Register 1	RCR1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C424h	RTC	RTC Control Register 2	RCR2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C426h	RTC	RTC Control Register 3	RCR3	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C428h	RTC	RTC Control Register 4	RCR4	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C42Eh	RTC	Time Error Adjustment Register	RADJ	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C440h	RTC	Time Capture Control Register 0	RTCCR0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C442h	RTC	Time Capture Control Register 1	RTCCR1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C444h	RTC	Time Capture Control Register 2	RTCCR2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C452h	RTC	Second Capture Register 0	RSECCP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C452h	RTC	BCNT0 Capture Register 0	BCNT0CP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C454h	RTC	Minute Capture Register 0	RMINCP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C454h	RTC	BCNT1 Capture Register 0	BCNT1CP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C456h	RTC	Hour Capture Register 0	RHRCP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C456h	RTC	BCNT2 Capture Register 0	BCNT2CP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C45Ah	RTC	Date Capture Register 0	RDAYCP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C45Ah	RTC	BCNT3 Capture Register 0	BCNT3CP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C45Ch	RTC	Month Capture Register 0	RMONCP0	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C462h	RTC	Second Capture Register 1	RSECCP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C462h	RTC	BCNT0 Capture Register 1	BCNT0CP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C464h	RTC	Minute Capture Register 1	RMINCP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C464h	RTC	BCNT1 Capture Register 1	BCNT1CP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C466h	RTC	Hour Capture Register 1	RHRCP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C466h	RTC	BCNT2 Capture Register 1	BCNT2CP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C46Ah	RTC	Date Capture Register 1	RDAYCP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C46Ah	RTC	BCNT3 Capture Register 1	BCNT3CP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C46Ch	RTC	Month Capture Register 1	RMONCP1	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C472h	RTC	Second Capture Register 2	RSECCP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C472h	RTC	BCNT0 Capture Register 2	BCNT0CP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C474h	RTC	Minute Capture Register 2	RMINCP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C474h	RTC	BCNT1 Capture Register 2	BCNT1CP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C476h	RTC	Hour Capture Register 2	RHRCP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C476h	RTC	BCNT2 Capture Register 2	BCNT2CP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C47Ah	RTC	Date Capture Register 2	RDAYCP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C47Ah	RTC	BCNT3 Capture Register 2	BCNT3CP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C47Ch	RTC	Month Capture Register 2	RMONCP2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 C4C0h	POE3	Input Level Control/Status Register 1	ICSR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4C2h	POE3	Output Level Control/Status Register 1	OCSR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4C4h	POE3	Input Level Control/Status Register 2	ICSR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4C6h	POE3	Output Level Control/Status Register 2	OCSR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4C8h	POE3	Input Level Control/Status Register 3	ICSR3	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4CAh	POE3	Software Port Output Enable Register	SPOER	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C4CBh	POE3	Port Output Enable Control Register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C4CCh	POE3	Port Output Enable Control Register 2	POECR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4D0h	POE3	Port Output Enable Control Register 4	POECR4	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4D2h	POE3	Port Output Enable Control Register 5	POECR5	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4D6h	POE3	Input Level Control/Status Register 4	ICSR4	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4D8h	POE3	Input Level Control/Status Register 5	ICSR5	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4DAh	POE3	Active Level Setting Register 1	ALR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0008 C4DCh	POE3	Input Level Control/Status Register 6	ICSR6	16	16	2, 3 PCLKB	2 ICLK	section 23

Table 5.1 List of I/O Registers (Address Order) (31 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C4E4h	POE3	MTU0 Pin Select Register 1	M0SELR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C4E5h	POE3	MTU0 Pin Select Register 2	M0SELR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C4E6h	POE3	MTU3 Pin Select Register	M3SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C4E7h	POE3	MTU4 Pin Select Register 1	M4SELR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C4E8h	POE3	MTU4 Pin Select Register 2	M4SELR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 4200h	CMTW0	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 4204h	CMTW0	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 4208h	CMTW0	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 4210h	CMTW0	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4214h	CMTW0	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4218h	CMTW0	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 421Ch	CMTW0	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4220h	CMTW0	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4224h	CMTW0	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4280h	CMTW1	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 4284h	CMTW1	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 4288h	CMTW1	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 4290h	CMTW1	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4294h	CMTW1	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 4298h	CMTW1	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 429Ch	CMTW1	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 42A0h	CMTW1	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 42A4h	CMTW1	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	section 26
000A 0580h	DOC	DOC Control Register	DOCR	8	8	2, 3 PCLKB	2 ICLK	section 42
000A 0584h	DOC	DOC Status Register	DOSR	8	8	2, 3 PCLKB	2 ICLK	section 42
000A 0588h	DOC	DOC Status Clear Register	DOSCR	8	8	2, 3 PCLKB	2 ICLK	section 42
000A 058Ch	DOC	DOC Data Input Register	DODIR	32	16, 32	2, 3 PCLKB	2 ICLK	section 42
000A 0590h	DOC	DOC Data Setting Register 0	DODSR0	32	16, 32	2, 3 PCLKB	2 ICLK	section 42
000A 0594h	DOC	DOC Data Setting Register 1	DODSR1	32	16, 32	2, 3 PCLKB	2 ICLK	section 42
000A 0B00h	REMC0	Function Select Register 0	REMCON0	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B01h	REMC0	Function Select Register 1	REMCON1	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B02h	REMC0	Status Register	REMSTS	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B03h	REMC0	Interrupt Control Register	REMINT	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B05h	REMC0	Compare Control Register	REMCPC	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B06h	REMC0	Compare Value Setting Register	REMCPD	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0B08h	REMC0	Header Pattern Minimum Width Setting Register	HDPMIN	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0B0Ah	REMC0	Header Pattern Maximum Width Setting Register	HDPMAX	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0B0Ch	REMC0	Data ‘0’ Pattern Minimum Width Setting Register	D0PMIN	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B0Dh	REMC0	Data ‘0’ Pattern Maximum Width Setting Register	D0PMAX	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B0Eh	REMC0	Data ‘1’ Pattern Minimum Width Setting Register	D1PMIN	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B0Fh	REMC0	Data ‘1’ Pattern Maximum Width Setting Register	D1PMAX	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B10h	REMC0	Special Data Pattern Minimum Width Setting Register	SDPMIN	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0B12h	REMC0	Special Data Pattern Maximum Width Setting Register	SDPMAX	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0B14h	REMC0	Pattern End Setting Register	REMPE	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0B17h	REMC0	Receive Bit Count Register	REMRBIT	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B18h	REMC0	Receive Data 0 Register	REMDAT0	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B19h	REMC0	Receive Data 1 Register	REMDAT1	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B1Ah	REMC0	Receive Data 2 Register	REMDAT2	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B1Bh	REMC0	Receive Data 3 Register	REMDAT3	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B1Ch	REMC0	Receive Data 4 Register	REMDAT4	8	8	2, 3 PCLKB	2 ICLK	section 36

Table 5.1 List of I/O Registers (Address Order) (32 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0B1Dh	REMC0	Receive Data 5 Register	REMDAT5	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B1Eh	REMC0	Receive Data 6 Register	REMDAT6	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B1Fh	REMC0	Receive Data 7 Register	REMDAT7	8	8	2, 3 PCLKB	2 ICLK	section 36
000A 0B20h	REMC0	Measurement Result Register	REMTIM	16	16	2, 3 PCLKB	2 ICLK	section 36
000A 0C80h	CMPC0	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0C84h	CMPC0	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0C88h	CMPC0	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0C8Ch	CMPC0	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0C90h	CMPC0	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CA0h	CMPC1	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CA4h	CMPC1	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CA8h	CMPC1	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CACh	CMPC1	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CB0h	CMPC1	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CC0h	CMPC2	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CC4h	CMPC2	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CC8h	CMPC2	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CCCh	CMPC2	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CD0h	CMPC2	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CE0h	CMPC3	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CE4h	CMPC3	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CE8h	CMPC3	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CECh	CMPC3	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 0CF0h	CMPC3	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 41
000A 8000h	CANFD0	Nominal Bit Rate Configuration Register	NBCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8004h	CANFD0	Channel Control Register	CHCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8008h	CANFD0	Channel Status Register	CHSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 800Ch	CANFD0	Channel Error Status Register	CHESR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8014h	CANFD	Global Configuration Register	GCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8018h	CANFD	Global Control Register	GCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 801Ch	CANFD	Global Status Register	GSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8020h	CANFD	Global Error Status Register	GESR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8024h	CANFD	Timestamp Counter Register	TSCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8028h	CANFD	Acceptance Filter List Control Register	AFCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 802Ch	CANFD	Acceptance Filter List Configuration Register	AFCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8030h	CANFD	Receive Message Buffer Configuration Register	RMCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8034h	CANFD	Receive Message Buffer New Data Register	RMNDR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8038h	CANFD	Receive Message Buffer Interrupt Enable Register	RMIER	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 803Ch	CANFD	Receive FIFO 0 Configuration Register	RFCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8040h	CANFD	Receive FIFO 1 Configuration Register	RFCR1	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8044h	CANFD	Receive FIFO 0 Status Register	RFSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8048h	CANFD	Receive FIFO 1 Status Register	RFSR1	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 804Ch	CANFD	Receive FIFO 0 Pointer Control Register	RFPCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8050h	CANFD	Receive FIFO 1 Pointer Control Register	RFPCR1	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8054h	CANFD	Common FIFO 0 Configuration Register	CFCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8058h	CANFD	Common FIFO 0 Status Register	CFSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 805Ch	CANFD	Common FIFO 0 Pointer Control Register	CFPCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8060h	CANFD	FIFO Empty Status Register	FESR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8064h	CANFD	FIFO Full Status Register	FFSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8068h	CANFD	FIFO Message Lost Status Register	FMLSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33

Table 5.1 List of I/O Registers (Address Order) (33 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 806Ch	CANFD	Receive FIFO Interrupt Status Register	RFISR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8070h	CANFD	Transmit Message Buffer 0 Control Register	TMCR0	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8071h	CANFD	Transmit Message Buffer 1 Control Register	TMCR1	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8072h	CANFD	Transmit Message Buffer 2 Control Register	TMCR2	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8073h	CANFD	Transmit Message Buffer 3 Control Register	TMCR3	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8074h	CANFD	Transmit Message Buffer 0 Status Register	TMSR0	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8075h	CANFD	Transmit Message Buffer 1 Status Register	TMSR1	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8076h	CANFD	Transmit Message Buffer 2 Status Register	TMSR2	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8077h	CANFD	Transmit Message Buffer 3 Status Register	TMSR3	8	8	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8078h	CANFD	Transmit Message Buffer Transmission Request Status Register 0	TMTRSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 807Ch	CANFD	Transmit Message Buffer Transmission Abort Request Status Register 0	TMARSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8080h	CANFD	Transmit Message Buffer Transmission Completion Status Register 0	TMTCSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8084h	CANFD	Transmit Message Buffer Transmission Abort Status Register 0	TMTASR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8088h	CANFD	Transmit Message Buffer Interrupt Enable Register 0	TMIER0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 808Ch	CANFD0	Transmit Queue 0 Configuration Register	TQCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8090h	CANFD0	Transmit Queue 0 Status Register	TQSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8094h	CANFD0	Transmit Queue 0 Pointer Control Register	TQPCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8098h	CANFD0	Transmission History Configuration Register	THCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 809Ch	CANFD0	Transmission History Status Register	THSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80A0h	CANFD0	Transmission History Pointer Control Register	THPCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80A4h	CANFD	Transmit Interrupt Status Register	TISR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80A8h	CANFD	Global Test Mode Configuration Register	GTMCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80ACh	CANFD	Global Test Mode Enable Register	GTMER	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80B0h	CANFD	Global CAN FD Configuration Register	GFDCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80B8h	CANFD	Global Test Mode Lock Key Register	GTMLKR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80C0h	CANFD	Acceptance Filter List Ignore Entry Setting Register	AFIGSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80C4h	CANFD	Acceptance Filter List Ignore Entry Enable Register	AFIGER	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80C8h	CANFD	DMA Transfer Control Register	DTCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80CCh	CANFD	DMA Transfer Status Register	DTSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 80D8h	CANFD	Global Reset Control Register	GRCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8100h	CANFD0	Data Bit Rate Configuration Register	DBCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8104h	CANFD0	CAN FD Configuration Register	FDCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8108h	CANFD0	CAN FD Control Register	FDCTR	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 810Ch	CANFD0	CAN FD Status Register	FDSTS	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8110h	CANFD0	CAN FD CRC Register	FDCRC	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8120h to 000A 812Ch	CANFD	Acceptance Filter List 0	AFL0	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8130h to 000A 813Ch	CANFD	Acceptance Filter List 1	AFL1	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8140h to 000A 814Ch	CANFD	Acceptance Filter List 2	AFL2	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8150h to 000A 815Ch	CANFD	Acceptance Filter List 3	AFL3	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8160h to 000A 816Ch	CANFD	Acceptance Filter List 4	AFL4	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8170h to 000A 817Ch	CANFD	Acceptance Filter List 5	AFL5	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8180h to 000A 818Ch	CANFD	Acceptance Filter List 6	AFL6	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8190h to 000A 819Ch	CANFD	Acceptance Filter List 7	AFL7	128	32	3, 4 PCLKB	1, 2 ICLK	section 33

Table 5.1 List of I/O Registers (Address Order) (34 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 81A0h to 000A 81ACh	CANFD	Acceptance Filter List 8	AFL8	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 81B0h to 000A 81BCh	CANFD	Acceptance Filter List 9	AFL9	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 81C0h to 000A 81CCh	CANFD	Acceptance Filter List 10	AFL10	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 81D0h to 000A 81DCh	CANFD	Acceptance Filter List 11	AFL11	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 81E0h to 000A 81ECh	CANFD	Acceptance Filter List 12	AFL12	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 81F0h to 000A 81FCh	CANFD	Acceptance Filter List 13	AFL13	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8200h to 000A 820Ch	CANFD	Acceptance Filter List 14	AFL14	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8210h to 000A 821Ch	CANFD	Acceptance Filter List 15	AFL15	128	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8280h to 000A 837Ch	CANFD	RAM Test Page Access Register 0 to RAM Test Page Access Register 63	RTPAR0 to RTPAR63	32	32	2, 3 PCLKB	1, 2 ICLK	section 33
000A 8520h to 000A 856Bh	CANFD	Receive FIFO 0	RFB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 856Ch to 000A 85B7h	CANFD	Receive FIFO 1	RFB1	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 85B8h to 000A 8603h	CANFD	Common FIFO	CFB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8604h to 000A 864Fh	CANFD	Transmit Message Buffer 0	TMB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8650h to 000A 869Bh	CANFD	Transmit Message Buffer 1	TMB1	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 869Ch to 000A 86E7h	CANFD	Transmit Message Buffer 2	TMB2	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 86E8h to 000A 8733h	CANFD	Transmit Message Buffer 3	TMB3	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8740h	CANFD0	Transmission History Access Register 0	THACR0	32	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8744h	CANFD0	Transmission History Access Register 1	THACR1	32	32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8920h to 000A 896Bh	CANFD	Receive Message Buffer 0	RMB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 896Ch to 000A 89B7h	CANFD	Receive Message Buffer 1	RMB1	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 89B8h to 000A 8A03h	CANFD	Receive Message Buffer 2	RMB2	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8A04h to 000A 8A4Fh	CANFD	Receive Message Buffer 3	RMB3	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8A50h to 000A 8A9Bh	CANFD	Receive Message Buffer 4	RMB4	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8A9Ch to 000A 8AE7h	CANFD	Receive Message Buffer 5	RMB5	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8AE8h to 000A 8B33h	CANFD	Receive Message Buffer 6	RMB6	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8B34h to 000A 8B7Fh	CANFD	Receive Message Buffer 7	RMB7	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8D20h to 000A 8D6Bh	CANFD	Receive Message Buffer 8	RMB8	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8D6Ch to 000A 8DB7h	CANFD	Receive Message Buffer 9	RMB9	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8DB8h to 000A 8E03h	CANFD	Receive Message Buffer 10	RMB10	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8E04h to 000A 8E4Fh	CANFD	Receive Message Buffer 11	RMB11	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8E50h to 000A 8E9Bh	CANFD	Receive Message Buffer 12	RMB12	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8E9Ch to 000A 8EE7h	CANFD	Receive Message Buffer 13	RMB13	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33

Table 5.1 List of I/O Registers (Address Order) (35 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 8EE8h to 000A 8F33h	CANFD	Receive Message Buffer 14	RMB14	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 8F34h to 000A 8F7Fh	CANFD	Receive Message Buffer 15	RMB15	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9120h to 000A 916Bh	CANFD	Receive Message Buffer 16	RMB16	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 916Ch to 000A 91B7h	CANFD	Receive Message Buffer 17	RMB17	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 91B8h to 000A 9203h	CANFD	Receive Message Buffer 18	RMB18	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9204h to 000A 924Fh	CANFD	Receive Message Buffer 19	RMB19	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9250h to 000A 929Bh	CANFD	Receive Message Buffer 20	RMB20	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 929Ch to 000A 92E7h	CANFD	Receive Message Buffer 21	RMB21	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 92E8h to 000A 9333h	CANFD	Receive Message Buffer 22	RMB22	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9334h to 000A 937Fh	CANFD	Receive Message Buffer 23	RMB23	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9520h to 000A 956Bh	CANFD	Receive Message Buffer 24	RMB24	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 956Ch to 000A 95B7h	CANFD	Receive Message Buffer 25	RMB25	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 95B8h to 000A 9603h	CANFD	Receive Message Buffer 26	RMB26	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9604h to 000A 964Fh	CANFD	Receive Message Buffer 27	RMB27	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9650h to 000A 969Bh	CANFD	Receive Message Buffer 28	RMB28	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 969Ch to 000A 96E7h	CANFD	Receive Message Buffer 29	RMB29	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 96E8h to 000A 9733h	CANFD	Receive Message Buffer 30	RMB30	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000A 9734h to 000A 977Fh	CANFD	Receive Message Buffer 31	RMB31	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 33
000C 1200h	MTU3	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1201h	MTU4	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1202h	MTU3	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1203h	MTU4	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1204h	MTU3	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1205h	MTU3	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1206h	MTU4	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1207h	MTU4	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1208h	MTU3	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1209h	MTU4	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 120Ah	MTU	Timer Output Master Enable Register A	TOERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 120Dh	MTU	Timer Gate Control Register A	TGCRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 120Eh	MTU	Timer Output Control Register 1A	TOCR1A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 120Fh	MTU	Timer Output Control Register 2A	TOCR2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1210h	MTU3	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1212h	MTU4	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1214h	MTU	Timer Period Data Register A	TCDRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1216h	MTU	Timer Dead Time Data Register A	TDDRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1218h	MTU3	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 121Ah	MTU3	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 121Ch	MTU4	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (36 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 121Eh	MTU4	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1220h	MTU	Timer Subcounter A	TCNTSA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1222h	MTU	Timer Period Buffer Register A	TCBRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1224h	MTU3	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1226h	MTU3	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1228h	MTU4	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 122Ah	MTU4	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 122Ch	MTU3	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 122Dh	MTU4	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1230h	MTU	Timer Interrupt Skipping Set Register 1A	TITCR1A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1231h	MTU	Timer Interrupt Skipping Counter 1A	TITCNT1A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1232h	MTU	Timer Buffer Transfer Set Register A	TBTERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1234h	MTU	Timer Dead Time Enable Register A	TDERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1236h	MTU	Timer Output Level Buffer Register A	TOLBRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1238h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1239h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 123Ah	MTU	Timer Interrupt Skipping Mode Register A	TITMRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 123Bh	MTU	Timer Interrupt Skipping Set Register 2A	TITCR2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 123Ch	MTU	Timer Interrupt Skipping Counter 2A	TITCNT2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1240h	MTU4	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1244h	MTU4	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1246h	MTU4	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1248h	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 124Ah	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 124Ch	MTU3	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 124Dh	MTU4	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1260h	MTU	Timer Waveform Control Register A	TWCRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1270h	MTU	Timer Mode Register 2A	TMDR2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1272h	MTU3	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1274h	MTU4	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1276h	MTU4	Timer General Register F	TGRF	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1280h	MTU	Timer Start Register A	TSTRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1281h	MTU	Timer Synchronous Register A	TSYRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1282h	MTU	Timer Counter Synchronous Start Register	TCSYSTR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1284h	MTU	Timer Read/Write Enable Register A	TRWERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1290h	MTU0	Noise Filter Control Register 0	NFCR0	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1291h	MTU1	Noise Filter Control Register 1	NFCR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1292h	MTU2	Noise Filter Control Register 2	NFCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1293h	MTU3	Noise Filter Control Register 3	NFCR3	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1294h	MTU4	Noise Filter Control Register 4	NFCR4	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1298h	MTU8	Noise Filter Control Register 8	NFCR8	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1299h	MTU0	Noise Filter Control Register C	NFCRC	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1300h	MTU0	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1301h	MTU0	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1302h	MTU0	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1303h	MTU0	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1304h	MTU0	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (37 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1306h	MTU0	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1308h	MTU0	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 130Ah	MTU0	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 130Ch	MTU0	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 130Eh	MTU0	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1320h	MTU0	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1322h	MTU0	Timer General Register F	TGRF	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1324h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1326h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1328h	MTU0	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1380h	MTU1	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1381h	MTU1	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1382h	MTU1	Timer I/O Control Register	TIOR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1384h	MTU1	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1385h	MTU1	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1386h	MTU1	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1388h	MTU1	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 138Ah	MTU1	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1390h	MTU1	Timer Input Capture Control Register	TICCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1391h	MTU1	Timer Mode Register 3	TMDR3	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1394h	MTU1	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 13A0h	MTU1	Timer Longword Counter	TCNTLW	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 13A4h	MTU1	Timer Longword General Register A	TGRALW	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 13A8h	MTU1	Timer Longword General Register B	TGRBLW	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1400h	MTU2	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1401h	MTU2	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1402h	MTU2	Timer I/O Control Register	TIOR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1404h	MTU2	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1405h	MTU2	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1406h	MTU2	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1408h	MTU2	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 140Ah	MTU2	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 140Ch	MTU2	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1600h	MTU8	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1601h	MTU8	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1602h	MTU8	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1603h	MTU8	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1604h	MTU8	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1606h	MTU8	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1608h	MTU8	Timer Counter	TCNT	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 160Ch	MTU8	Timer General Register A	TGRA	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1610h	MTU8	Timer General Register B	TGRB	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1614h	MTU8	Timer General Register C	TGRC	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1618h	MTU8	Timer General Register D	TGRD	32	32	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A00h	MTU6	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A01h	MTU7	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A02h	MTU6	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A03h	MTU7	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A04h	MTU6	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A05h	MTU6	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (38 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1A06h	MTU7	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A07h	MTU7	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A08h	MTU6	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A09h	MTU7	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A0Ah	MTU	Timer Output Master Enable Register B	TOERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A0Eh	MTU	Timer Output Control Register 1B	TOCR1B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A0Fh	MTU	Timer Output Control Register 2B	TOCR2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A10h	MTU6	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A12h	MTU7	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A14h	MTU	Timer Period Data Register B	TCDRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A16h	MTU	Timer Dead Time Data Register B	TDDRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A18h	MTU6	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A1Ah	MTU6	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A1Ch	MTU7	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A1Eh	MTU7	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A20h	MTU	Timer Subcounter B	TCNTSB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A22h	MTU	Timer Period Buffer Register B	TCBRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A24h	MTU6	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A26h	MTU6	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A28h	MTU7	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A2Ah	MTU7	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A2Ch	MTU6	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A2Dh	MTU7	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A30h	MTU	Timer Interrupt Skipping Set Register 1B	TITCR1B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A31h	MTU	Timer Interrupt Skipping Counter 1B	TITCNT1B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A32h	MTU	Timer Buffer Transfer Set Register B	TBTERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A34h	MTU	Timer Dead Time Enable Register B	TDERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A36h	MTU	Timer Output Level Buffer Register B	TOLBRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A38h	MTU6	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A39h	MTU7	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A3Ah	MTU	Timer Interrupt Skipping Mode Register B	TITMRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A3Bh	MTU	Timer Interrupt Skipping Set Register 2B	TITCR2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A3Ch	MTU	Timer Interrupt Skipping Counter 2B	TITCNT2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A40h	MTU7	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A44h	MTU7	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A46h	MTU7	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A48h	MTU7	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A4Ah	MTU7	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A4Ch	MTU6	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A4Dh	MTU7	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A50h	MTU6	Timer Synchronous Clear Register	TSYCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A60h	MTU	Timer Waveform Control Register B	TWCRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A70h	MTU	Timer Mode Register 2B	TMDR2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A72h	MTU6	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A74h	MTU7	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A76h	MTU7	Timer General Register F	TGRF	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A80h	MTU	Timer Start Register B	TSTRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (39 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1A81h	MTU	Timer Synchronous Register B	TSYRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A84h	MTU	Timer Read/Write Enable Register B	TRWERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A93h	MTU6	Noise Filter Control Register 6	NFCR6	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A94h	MTU7	Noise Filter Control Register 7	NFCR7	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1A95h	MTU5	Noise Filter Control Register 5	NFCR5	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C80h	MTU5	Timer Counter U	TCNTU	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C82h	MTU5	Timer General Register U	TGRU	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C84h	MTU5	Timer Control Register U	TCRU	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C85h	MTU5	Timer Control Register 2	TCR2U	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C86h	MTU5	Timer I/O Control Register U	TIORU	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C90h	MTU5	Timer Counter V	TCNTV	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C92h	MTU5	Timer General Register V	TGRV	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C94h	MTU5	Timer Control Register V	TCRV	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C95h	MTU5	Timer Control Register 2	TCR2V	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1C96h	MTU5	Timer I/O Control Register V	TIORV	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CA0h	MTU5	Timer Counter W	TCNTW	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CA2h	MTU5	Timer General Register W	TGRW	16	16	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CA4h	MTU5	Timer Control Register W	TCRW	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CA5h	MTU5	Timer Control Register 2	TCR2W	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CA6h	MTU5	Timer I/O Control Register W	TIORW	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CB2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CB4h	MTU5	Timer Start Register	TSTR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000C 1CB6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	4, 5 PCLKA	1, 2 ICLK	section 22
000D 0000h	SCI10	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0000h	SMCI10	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0001h	SCI10	Bit Rate Register	BRR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0002h	SCI10	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0002h	SMCI10	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0003h	SCI10	Transmit Data Register	TDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0004h	SCI10	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0004h	SMCI10	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0004h	SCI10	Serial Status Register	SSRFIFO	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0005h	SCI10	Receive Data Register	RDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0006h	SCI10	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0006h	SMCI10	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0007h	SCI10	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0008h	SCI10	Noise Filter Setting Register	SNFR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0009h	SCI10	I^2C Mode Register 1	SIMR1	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Ah	SCI10	I^2C Mode Register 2	SIMR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Bh	SCI10	I^2C Mode Register 3	SIMR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Ch	SCI10	I^2C Status Register	SISR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Dh	SCI10	SPI Mode Register	SPMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Eh	SCI10	Transmit Data Register H	TDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Fh	SCI10	Transmit Data Register L	TDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Eh	SCI10	Transmit Data Register HL	TDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 000Eh	SCI10	Transmit FIFO Data Register	FTDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Fh	SCI10	Transmit FIFO Data Register	FTDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 000Eh	SCI10	Transmit FIFO Data Register	FTDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0010h	SCI10	Receive Data Register H	RDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0011h	SCI10	Receive Data Register L	RDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (40 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000D 0010h	SCI10	Receive Data Register HL	RDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0010h	SCI10	Receive FIFO Data Register	FRDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0011h	SCI10	Receive FIFO Data Register	FRDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0010h	SCI10	Receive FIFO Data Register	FRDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0012h	SCI10	Modulation Duty Register	MDDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0013h	SCI10	Data Comparison Control Register	DCCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0014h	SCI10	FIFO Control Register	FCR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0015h	SCI10	FIFO Control Register	FCR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0014h	SCI10	FIFO Control Register	FCR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0016h	SCI10	FIFO Data Count Register	FDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0017h	SCI10	FIFO Data Count Register	FDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0016h	SCI10	FIFO Data Count Register	FDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0018h	SCI10	Line Status Register	LSR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0019h	SCI10	Line Status Register	LSR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0018h	SCI10	Line Status Register	LSR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 001Ah	SCI10	Comparison Data Register	CDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 001Bh	SCI10	Comparison Data Register	CDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 001Ah	SCI10	Comparison Data Register	CDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 001Ch	SCI10	Serial Port Register	SPTR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 001Dh	SCI10	Transmit/Receive Timing Select Register	TMGR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0020h	SCI11	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0020h	SMCI11	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0021h	SCI11	Bit Rate Register	BRR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0022h	SCI11	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0022h	SMCI11	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0023h	SCI11	Transmit Data Register	TDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0024h	SCI11	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0024h	SMCI11	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0024h	SCI11	Serial Status Register	SSRFIFO	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0025h	SCI11	Receive Data Register	RDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0026h	SCI11	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0026h	SMCI11	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0027h	SCI11	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0028h	SCI11	Noise Filter Setting Register	SNFR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0029h	SCI11	I^2C Mode Register 1	SIMR1	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Ah	SCI11	I^2C Mode Register 2	SIMR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Bh	SCI11	I^2C Mode Register 3	SIMR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Ch	SCI11	I^2C Status Register	SISR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Dh	SCI11	SPI Mode Register	SPMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Eh	SCI11	Transmit Data Register H	TDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Fh	SCI11	Transmit Data Register L	TDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Eh	SCI11	Transmit Data Register HL	TDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 002Eh	SCI11	Transmit FIFO Data Register	FTDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Fh	SCI11	Transmit FIFO Data Register	FTDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 002Eh	SCI11	Transmit FIFO Data Register	FTDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0030h	SCI11	Receive Data Register H	RDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0031h	SCI11	Receive Data Register L	RDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0030h	SCI11	Receive Data Register HL	RDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0030h	SCI11	Receive FIFO Data Register	FRDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0031h	SCI11	Receive FIFO Data Register	FRDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (41 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000D 0030h	SCI11	Receive FIFO Data Register	FRDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0032h	SCI11	Modulation Duty Register	MDDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0033h	SCI11	Data Comparison Control Register	DCCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0034h	SCI11	FIFO Control Register	FCR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0035h	SCI11	FIFO Control Register	FCR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0034h	SCI11	FIFO Control Register	FCR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0036h	SCI11	FIFO Data Count Register	FDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0037h	SCI11	FIFO Data Count Register	FDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0036h	SCI11	FIFO Data Count Register	FDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 0038h	SCI11	Line Status Register	LSR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0039h	SCI11	Line Status Register	LSR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0038h	SCI11	Line Status Register	LSR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 003Ah	SCI11	Comparison Data Register	CDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 003Bh	SCI11	Comparison Data Register	CDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 003Ah	SCI11	Comparison Data Register	CDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 30
000D 003Ch	SCI11	Serial Port Register	SPTR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 003Dh	SCI11	Transmit/Receive Timing Select Register	TMGR	8	8	3, 4 PCLKA	1, 2 ICLK	section 30
000D 0100h	RSPI0	RSPI Control Register	SPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0101h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0102h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0103h	RSPI0	RSPI Status Register	SPSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0104h	RSPI0	RSPI Data Register	SPDR	32	8, 16, 32	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0108h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0109h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 010Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 010Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 010Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 010Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 010Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 010Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0110h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0112h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0114h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0116h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0118h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 011Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 011Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 011Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0120h	RSPI0	RSPI Data Control Register 2	SPDCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000D 0121h	RSPI0	RSPI Control Register 3	SPCR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 34
000E 2000h	RSCI10	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 31
000E 2004h	RSCI10	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 31
000E 2008h	RSCI10	Control Register 0	SCR0	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 200Ch	RSCI10	Control Register 1	SCR1	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2010h	RSCI10	Control Register 2	SCR2	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2014h	RSCI10	Control Register 3	SCR3	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2018h	RSCI10	Control Register 4	SCR4	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 201Eh	RSCI10	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKA	2 ICLK	section 31

Table 5.1 List of I/O Registers (Address Order) (42 / 43)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E 2020h	RSCI10	I²C Mode Register	SIMR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2024h	RSCI10	FIFO Control Register	FCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 202Ch	RSCI10	Manchester Mode Control Register	MMCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2030h	RSCI10	DE Signal Control Register	DECR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2034h	RSCI10	Extended Serial Mode Control Register 0	XCR0	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2038h	RSCI10	Extended Serial Mode Control Register 1	XCR1	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 203Ch	RSCI10	Extended Serial Mode Control Register 2	XCR2	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2048h	RSCI10	Status Register	SSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 204Ch	RSCI10	I²C Status Register	SISR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2050h	RSCI10	Receive FIFO Status Register	RFSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2054h	RSCI10	Transmit FIFO Status Register	TFSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2058h	RSCI10	Manchester Mode Status Register	MMSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 205Ch	RSCI10	Extended Serial Mode Status Register 0	XSR0	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2060h	RSCI10	Extended Serial Mode Status Register 1	XSR1	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2068h	RSCI10	Status Clear Register	SSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 206Ch	RSCI10	I²C Status Clear Register	SISCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2070h	RSCI10	Receive FIFO Status Clear Register	RFSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2074h	RSCI10	Manchester Mode Status Clear Register	MMSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2078h	RSCI10	Extended Serial Mode Status Clear Register	XSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2080h	RSCI11	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 31
000E 2084h	RSCI11	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 31
000E 2088h	RSCI11	Control Register 0	SCR0	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 208Ch	RSCI11	Control Register 1	SCR1	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2090h	RSCI11	Control Register 2	SCR2	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2094h	RSCI11	Control Register 3	SCR3	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 2098h	RSCI11	Control Register 4	SCR4	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 209Eh	RSCI11	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKA	2 ICLK	section 31
000E 20A0h	RSCI11	I²C Mode Register	SIMR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20A4h	RSCI11	FIFO Control Register	FCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20ACh	RSCI11	Manchester Mode Control Register	MMCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20B0h	RSCI11	DE Signal Control Register	DECR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20B4h	RSCI11	Extended Serial Mode Control Register 0	XCR0	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20B8h	RSCI11	Extended Serial Mode Control Register 1	XCR1	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20BCh	RSCI11	Extended Serial Mode Control Register 2	XCR2	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20C8h	RSCI11	Status Register	SSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20CCh	RSCI11	I²C Status Register	SISR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20D0h	RSCI11	Receive FIFO Status Register	RFSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20D4h	RSCI11	Transmit FIFO Status Register	TFSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20D8h	RSCI11	Manchester Mode Status Register	MMSR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20DCh	RSCI11	Extended Serial Mode Status Register 0	XSR0	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20E0h	RSCI11	Extended Serial Mode Status Register 1	XSR1	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20E8h	RSCI11	Status Clear Register	SSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20ECh	RSCI11	I²C Status Clear Register	SISCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20F0h	RSCI11	Receive FIFO Status Clear Register	RFSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20F4h	RSCI11	Manchester Mode Status Clear Register	MMSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E 20F8h	RSCI11	Extended Serial Mode Status Clear Register	XSCR	32	32	2, 3 PCLKA	2 ICLK	section 31
000E D000h	CANFD	ECC Control/Status Register	ECCSR	32	32	2, 3 PCLKA	1, 2 ICLK	section 33
000E D004h	CANFD	ECC Test Mode Register	ECTMR	16	16	2, 3 PCLKA	1, 2 ICLK	section 33
000E D00Ch	CANFD	ECC Decoder Test Data Register	ECTDR	32	32	2, 3 PCLKA	1, 2 ICLK	section 33

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E D010h	CANFD	ECC Error Address Register	ECEAR	32	32	2, 3 PCLKA	1, 2 ICLK	section 33
0012 0040h	OFSM	Serial Programmer Command Control Register	SPCC	32	32	8 FCLK		section 7
0012 0048h	OFSM	TM Enable Flag Register	TMEF	32	32	8 FCLK		section 7
0012 0050h	OFSM	OCD/Serial Programmer ID Setting Register	OSIS	128	32	8 FCLK		section 7
0012 0060h	OFSM	TM Identification Data Register	TMINF	32	32	8 FCLK		section 7
0012 0064h	OFSM	Endian Select Register	MDE	32	32	8 FCLK		section 7
0012 0068h	OFSM	Option Function Select Register 0	OFS0	32	32	8 FCLK		section 7
0012 006Ch	OFSM	Option Function Select Register 1	OFS1	32	32	8 FCLK		section 7
0012 007Ch	OFSM	ROM Code Protection Register	ROMCODE	32	32	8 FCLK		section 7
007F B174h	FLASH	Unique ID Register 0	UIDR0	32	32	3 to 5 FCLK	3, 4 ICLK	section 44
007F B17Ch	TEMPS	Temperature Sensor Calibration Data Register	TSCDR	32	32	3 to 5 FCLK	3, 4 ICLK	section 40
007F B1E4h	FLASH	Unique ID Register 1	UIDR1	32	32	3 to 5 FCLK	3, 4 ICLK	section 44
007F B1E8h	FLASH	Unique ID Register 2	UIDR2	32	32	3 to 5 FCLK	3, 4 ICLK	section 44
007F E010h	FLASH	Flash Access Status Register	FASTAT	8	8	2 to 4 FCLK	2, 3 ICLK	section 44
007F E014h	FLASH	Flash Access Error Interrupt Enable Register	FAEINT	8	8	2 to 4 FCLK	2, 3 ICLK	section 44
007F E018h	FLASH	Flash Ready Interrupt Enable Register	FRDYIE	8	8	2 to 4 FCLK	2, 3 ICLK	section 44
007F E030h	FLASH	FACI Command Processing Start Address Register	FSADDR	32	32	2 to 4 FCLK	2, 3 ICLK	section 44
007F E034h	FLASH	FACI Command Processing End Address Register	FEADDR	32	32	2 to 4 FCLK	2, 3 ICLK	section 44
007F E080h	FLASH	Flash Status Register	FSTATR	32	32	2 to 4 FCLK	2, 3 ICLK	section 44
007F E084h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 to 4 FCLK	2, 3 ICLK	section 44
007F E088h	FLASH	Flash Protection Register	FPROTR	16	16	2 to 4 FCLK	2, 3 ICLK	section 44
007F E08Ch	FLASH	Flash Sequencer Set-Up Initialization Register	FSUINITR	16	16	2 to 4 FCLK	2, 3 ICLK	section 44
007F E090h	FLASH	Lock Bit Status Register	FLKSTAT	8	8	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0A0h	FLASH	FACI Command Register	FCMDR	16	16	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0C0h	FLASH	Flash P/E Status Register	FPESTAT	16	16	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0D0h	FLASH	Data Flash Blank Check Control Register	FBCCNT	8	8	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0D4h	FLASH	Data Flash Blank Check Status Register	FBCSTAT	8	8	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0D8h	FLASH	Data Flash Programming Start Address Register	FPSADDR	32	32	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0E0h	FLASH	Flash Sequencer Processing Switching Register	FCPSR	16	16	2 to 4 FCLK	2, 3 ICLK	section 44
007F E0E4h	FLASH	Flash Sequencer Processing Clock Frequency Notification Register	FPCKAR	16	16	2 to 4 FCLK	2, 3 ICLK	section 44

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 45, 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)	—	✓	—	—	—	—	—	—	—
Sub clock oscillator forced oscillation bit (SOFCR.SOFE)	—	✓	—	—	—	—	—	—	—
Voltage-monitoring 0 reset detect flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—	—	—
Voltage level setting register (VOLSR)	✓	✓	✓	—	—	—	—	—	—
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 DPSIER2, DPSIFR0 to DPSIFR2, DPSIEGR0 to DPSIEGR2)*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
Operating mode*3	✓	✓	✓	—	—	—	—	—	—
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 27, Realtime Clock (RTCC).

Note 2. Of the registers related to the low-power-consumption function, the DPSBKRy register is not initialized by any reset. For details, refer to section 11, Low Power Consumption.

Note 3. 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 13, 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

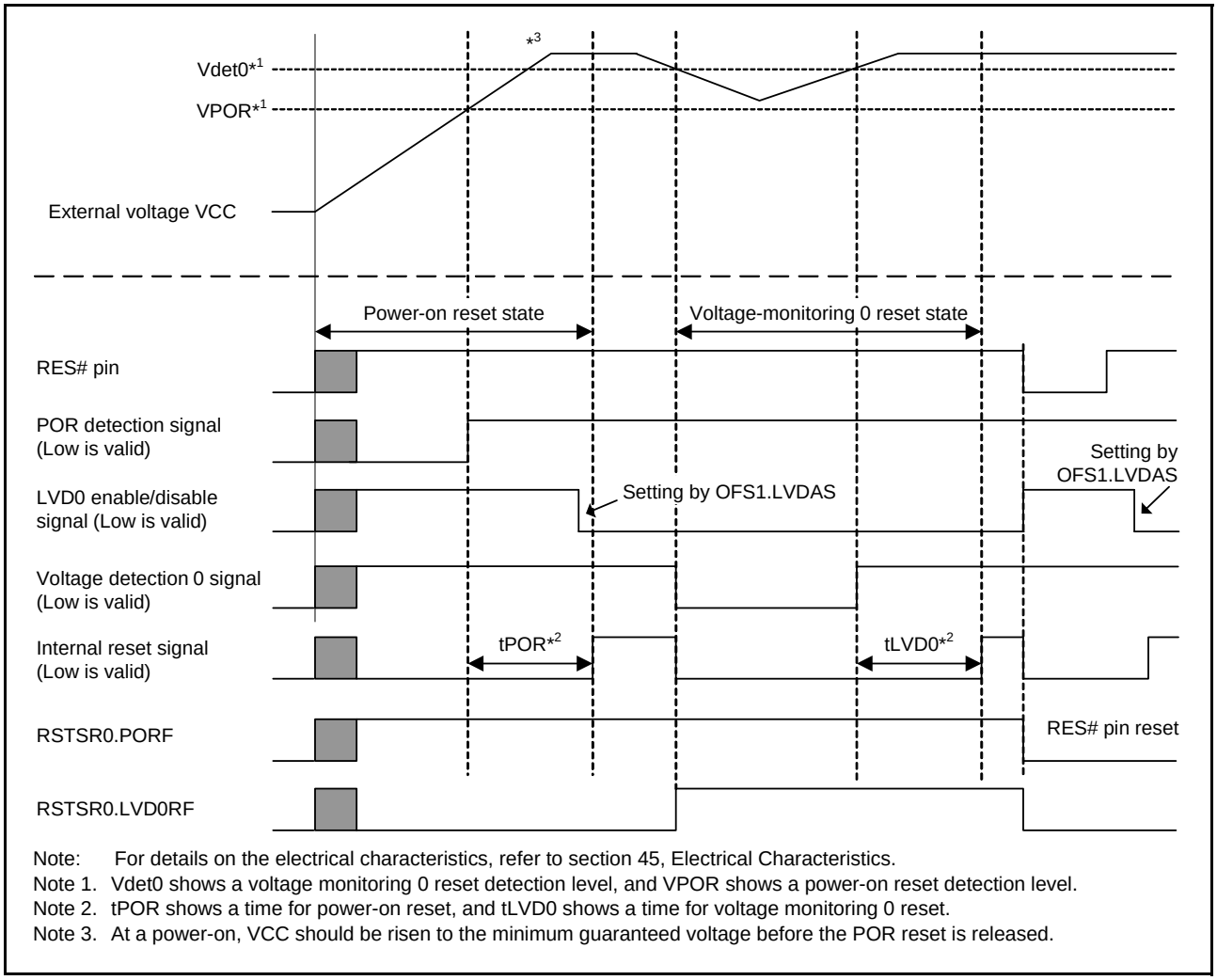
6.3 Operation

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 45, 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 45, 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

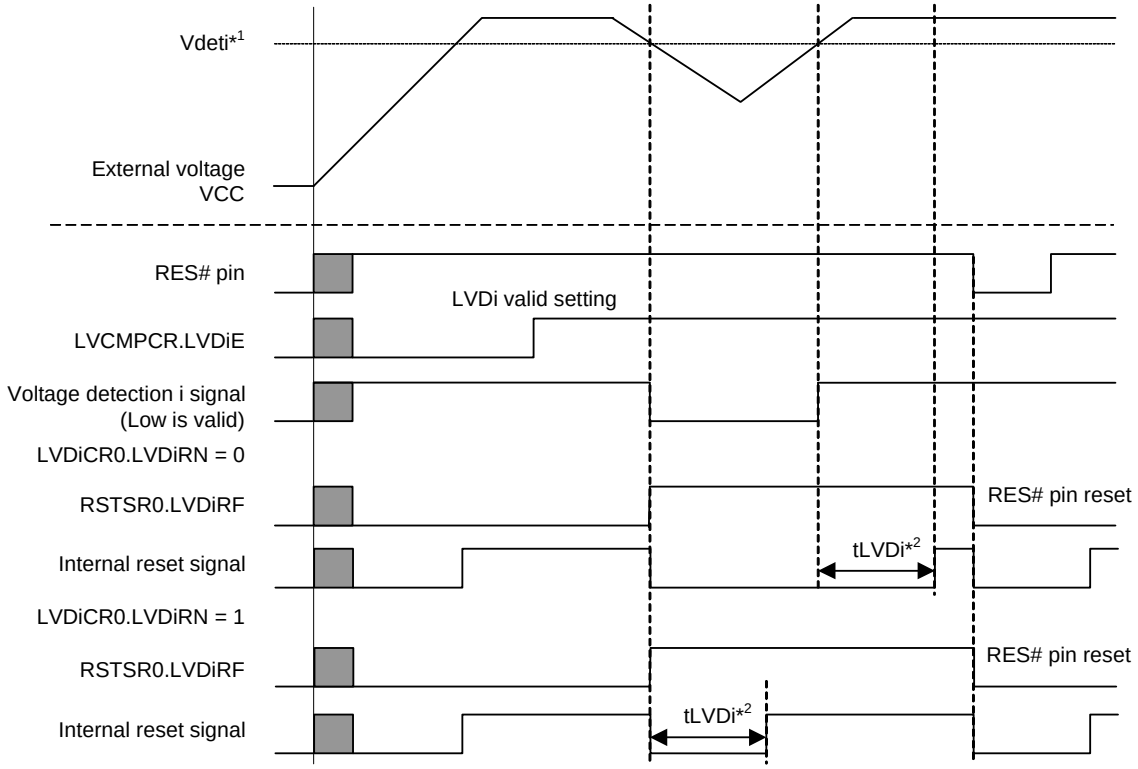
The voltage monitoring 1 reset and voltage monitoring 2 reset are internal resets generated by the voltage monitoring circuit.

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 45, 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 29, 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 28, 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.

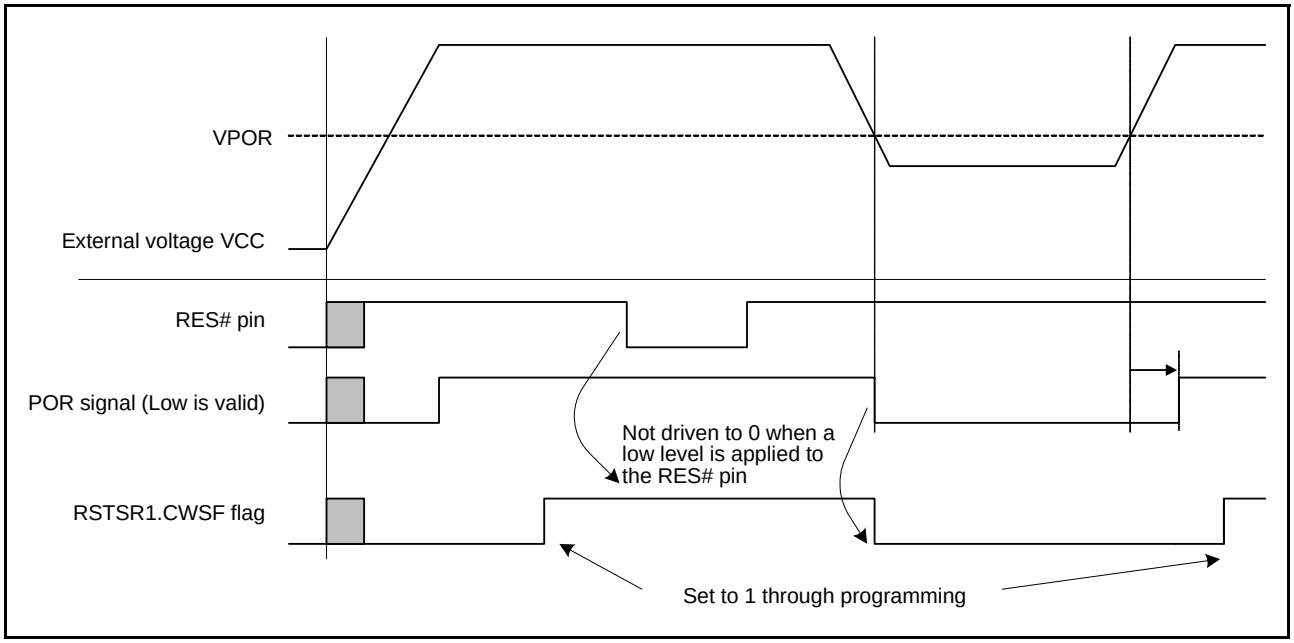


Figure 6.3 Example of Cold/Warm Start Determination Operation

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.

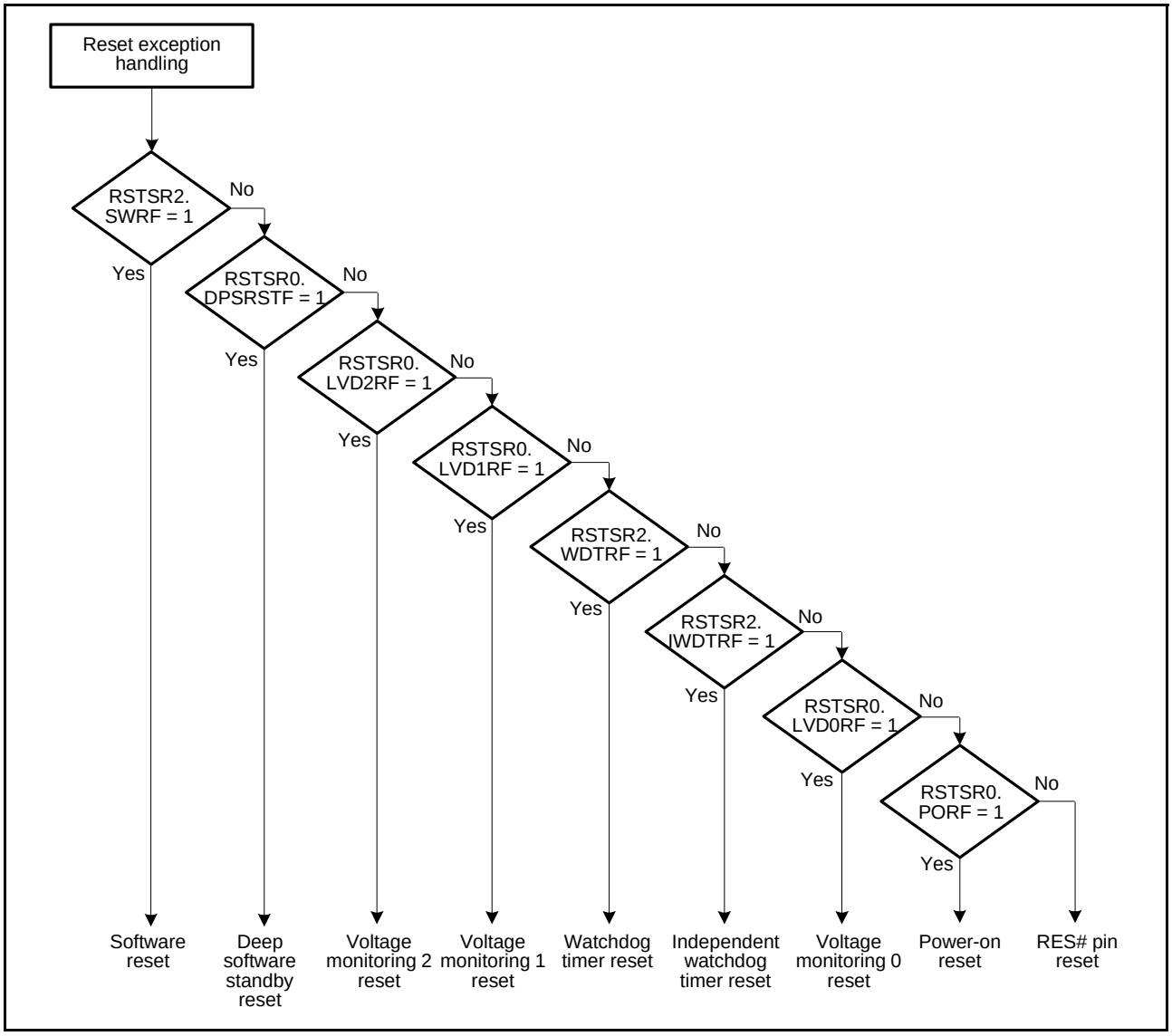


Figure 6.4 Example of Reset Generation Source Determination Flow

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)
- UB code A
- UB code B
- ROM Code Protection Register (ROMCODE)

The option-setting memory 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 7.6, Setting the Option-Setting Memory.
Figure 7.1 shows the option-setting memory area.

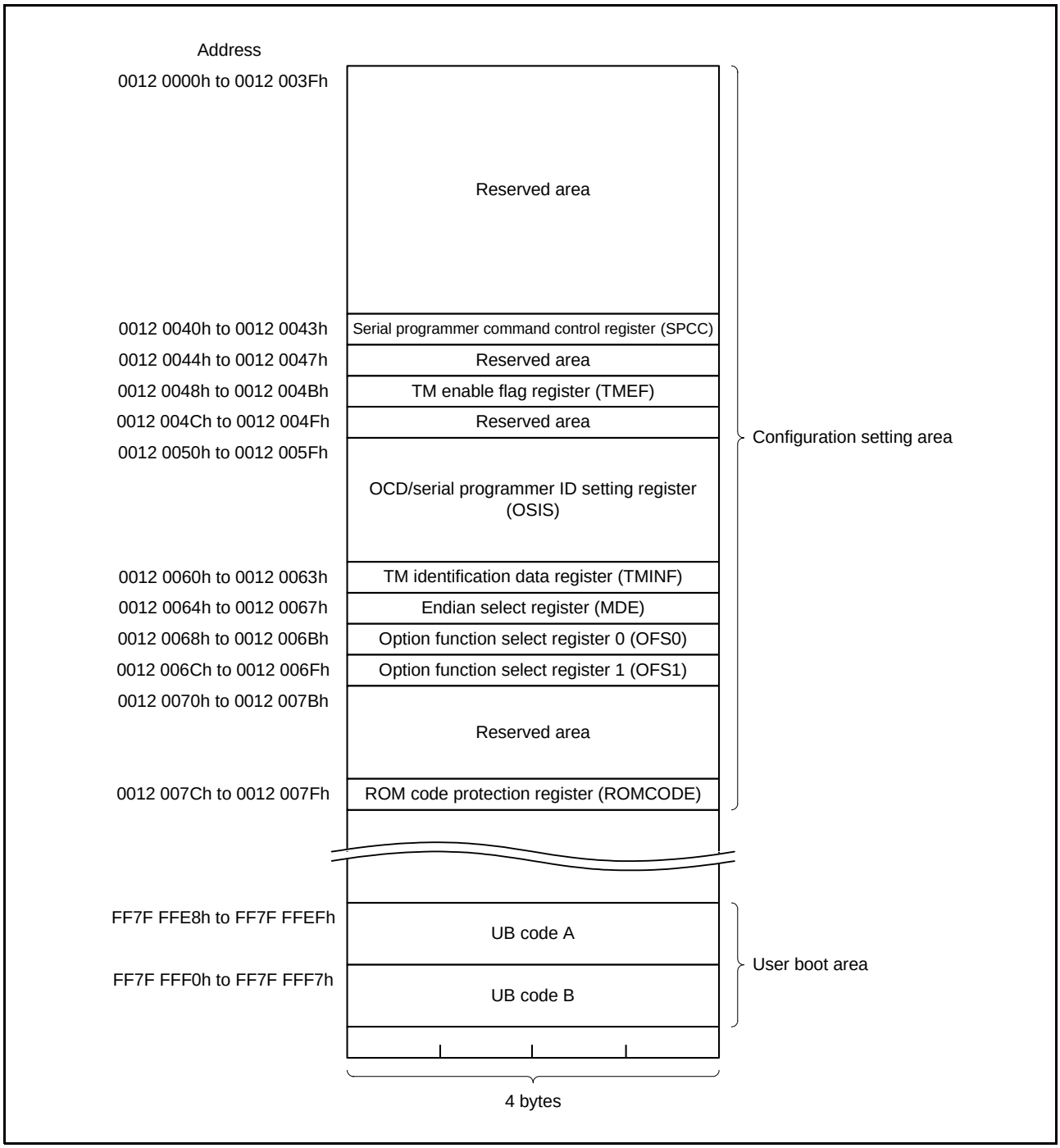


Figure 7.1 Option-Setting Memory Area

7.2 Register Descriptions

7.2.1 Serial Programmer Command Control Register (SPCC)

Address(es): OFSM.SPCC 0012 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDPR	WRPR	SEPR	—	SPE	—	—	IDE	—	—	—	—	—	—	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
b23 to b18	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b24	IDE	ID Code Protection Enable	0: ID code protection is enabled after a reset.*2 1: ID code protection is disabled after a reset.	R
b26, b25	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R
b27	SPE	Serial Programmer Connection Enable	0: Connection of a serial programmer is prohibited after a reset. 1: Connection of a serial programmer is permitted after a reset.	R
b28	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R
b29	SEPR	Block Erasure Command Protect	0: Execution of block erasure commands is prohibited after a reset. 1: Execution of block erasure commands is permitted after a reset.	R
b30	WRPR	Programming Command Protect	0: Execution of programming commands is prohibited after a reset. 1: Execution of programming commands is permitted after a reset.	R
b31	RDPR	Read Command Protect	0: Execution of read commands is prohibited after a reset. 1: Execution of read commands is permitted after a reset.	R

Note 1. 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 ID code protection is to be used, set the RDPR, SEPR, and WRPR bits to 0.

This register is used to enable or disable serial ID code protection, and to permit or prohibit the connection of an on-chip debugger or the connection of a serial programmer, or the execution of block erasure commands, programming commands, or read commands.

OCDE Bit (On-Chip Debugger Connection Enable)

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

IDE Bit (ID Code Protection Enable)

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

SPE Bit (Serial Programmer Connection Enable)

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

SEPR Bit (Block Erasure Command Protect)

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

WRPR Bit (Programming Command Protect)

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

RDPR Bit (Read Command Protect)

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

7.2.2 OCD/Serial Programmer ID Setting Register (OSIS)

This register is used 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.
Enabling ID code protection through the serial programmer requires setting of the IDE, SPE, RDPR, WRPR, and SEPR bits in the SPCC register, in addition to setting of this register.
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
0012 0050h to 0012 0053h	ID Code 4	ID Code 3	ID Code 2	ID Code 1/Control Code
0012 0054h to 0012 0057h	ID Code 8	ID Code 7	ID Code 6	ID Code 5
0012 0058h to 0012 005Bh	ID Code 12	ID Code 11	ID Code 10	ID Code 9
0012 005Ch to 0012 005Fh	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 7.5, Settings of the Option-Setting Memory and Reading, Programming, and Erasure.

7.2.3 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 0012 0068h

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

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

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

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

Note 1. 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 setting in the OFS0 register is ignored in user boot mode, and this register functions similarly when it is set to FFFF FFFFh.

IWDTSTRT Bit (IWDT Start Mode Select)

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

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

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

For details, refer to section 29, 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 29, 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 29, 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 29, 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 29, 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 28, 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 28, 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 28, 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 28, 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 28, Watchdog Timer (WDTA).

7.2.4 Option Function Select Register 1 (OFS1)

Address(es): OFSM.OFS1 0012 006Ch

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

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: Reserved 0 1: Reserved 1 0: Selects 2.83 V 1 1: Selects 4.22 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.

The setting in the OFS1 register is ignored in user boot mode, and this register functions similarly when it is set to FFFF FFFFh.

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

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

LVDAS Bit (Voltage Detection 0 Circuit Start)

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

HOCOEN Bit (HOCO Oscillation Enable)

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

7.2.5 Endian Select Register (MDE)

Address(es): OFSM.MDE 0012 0064h

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

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

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

Note 1. 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.

MDE[2:0] Bits (Endian Select)

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

7.2.6 TM Enable Flag Register (TMEF)

Address(es): OFSM.TMEF 0012 0048h

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

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

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

Note 1. 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 for blocks 8 and 9 in the user area.
To enable the TM function, refer to section 44.9.38, TM Setting Command. When the TMEF[2:0] bits are rewritten with the TM function enabled, rewriting the bits is ignored.
To disable the TM function, refer to section 44.9.37, Configuration Clearing Command.

TMEF[2:0] Bits (TM Enable)

These bits enable or disable the TM function for blocks 8 and 9 in the user area.

7.2.7 TM Identification Data Register (TMINF)

Address(es): OFSM.TMINF 0012 0060h

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

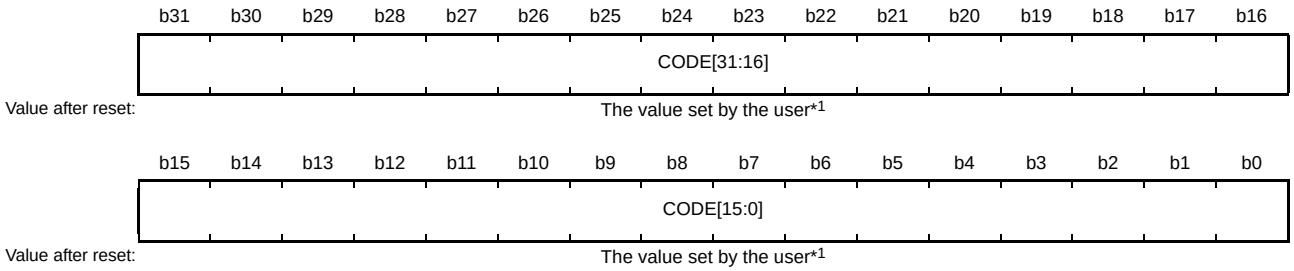
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. 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 44.9.37, Configuration Clearing Command.

7.2.8 ROM Code Protection Register (ROMCODE)

Address(es): OFSM.ROMCODE 0012 007Ch



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 ROM code 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 UB Codes

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

- UB code A is 5573 6572h and 426F 6F74h.
- UB code B is FFFF FF00h and 0008 C040h.
- The low level is being input on the MD pin.
- The high level is being input on the UB pin.

7.3.1 UB Code A

UB code A consists of two 32-bit words. Set UB code A to 5573 6572h and 426F 6F74h. Do not set any values other than these values in user boot mode. Set UB code A to FFFF FFFFh and FFFF FFFFh in any boot modes other than user boot mode.

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

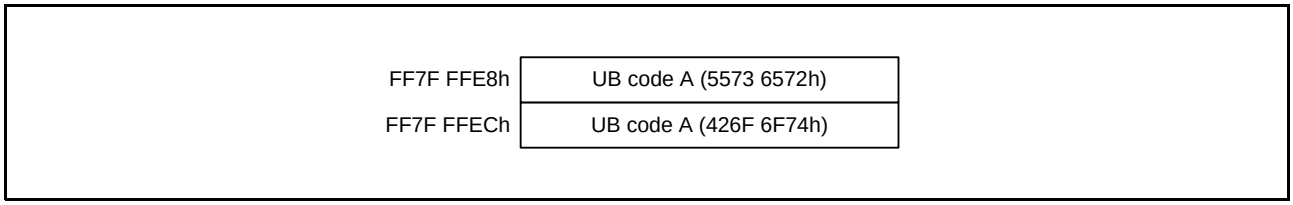


Figure 7.2 UB Code A Structure

7.3.2 UB Code B

UB code B consists of two words, i.e. 32 bits. Set UB code B to FFFF FF00h and 0008 C040h. Do not set any values other than these values in user boot mode. Set UB code B to FFFF FFFFh and FFFF FFFFh in any boot modes other than user boot mode.

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

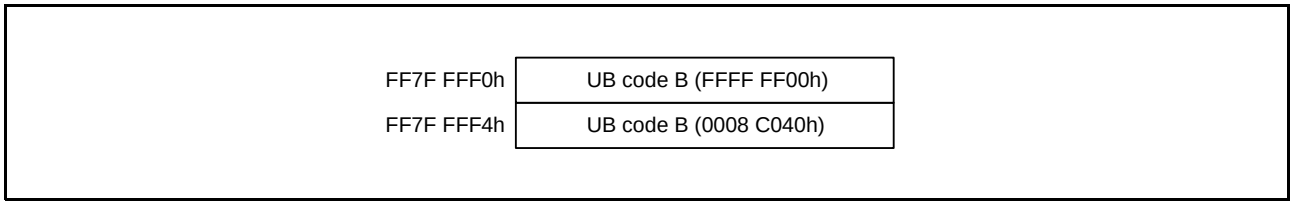


Figure 7.3 UB Code B Structure

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

Table 7.2 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, FINE Interface)		Self-Programming		Parallel Programmer	
		Programming	Erasure	Programming	Erasure	Programming	Erasure
ROM code protection register (ROMCODE)	0000 0000h	✓*2	✓*2	✓*3	×	×	×
	0000 0001h	✓*2	✓*2	✓*3	×	×*4	✓*5, *6
	Other than above	✓*2	✓*2	✓*3	×	✓*5	✓*5
UB code A and UB code B	0000 0000h	✓*2	✓*2	×	×	×	×
	0000 0001h	✓*2	✓*2	×	×	✓*5	✓*5
	Other than above	✓*2	✓*2	×	×	✓*5	✓*5
Option-setting memory other than above*1	0000 0000h	✓*2	✓*2	✓*3	×	×	×
	0000 0001h	✓*2	✓*2	✓*3	×	✓*5	✓*5, *6
	Other than above	✓*2	✓*2	✓*3	×	✓*5	✓*5

✓: Possible

×: Not possible

Note 1. The option-setting memory other than the ROM code protection register, UB code A, and UB code B includes the following registers.
- Serial programmer command control register (SPCC)
- OCD/serial programmer ID setting register (OSIS)
- Endian select register (MDE)
- Option function select register 0 (OFS0)
- Option function select register 1 (OFS1)
- TM enable flag register (TMEF)
- TM identification data register (TMINF)

Note 2. The commands for boot mode (for the SCI and FINE interfaces) are used for programming or erasure. For details, refer to section 44.7, Boot Mode.

Note 3. The configuration setting command is used for programming. For how to use the configuration setting command, refer to section 44.6.6.10, Configuration Set Command.

Note 4. The ROM code cannot be programmed when the ROM code protection is set.

Note 5. The parallel programmer is used for programming and erasure. For details, refer to the manual of the parallel programmer you are using.

Note 6. It is erasable when the code flash memory is blank.

7.5 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	SPCC. IDE	OSIS (Control Code)	OSIS (ID code 2 to 16)	SPCC. RDPR	SPCC. WRPR	SPCC. SEPR	Connection of a Serial Programmer	Reading, Programming and Erasure after the Connection of a Serial Programmer
1	0	x	Any value	Any value	x	x	x	Connection prohibited	—
2	1	0	Other than 45h		0	0	0	Judgment on control codes and ID codes*1	Reading permitted, programming permitted, erasure permitted
3			45h					Judgment on control codes and ID codes*2	Reading permitted, programming permitted, erasure permitted
4	1	1	Any value		0	0	0	Connection permitted	Reading prohibited, programming prohibited, erasure prohibited
5					1	0	0	Connection permitted	Reading permitted, programming prohibited, erasure prohibited
6					0	1	0	Connection permitted	Reading prohibited, programming permitted, erasure prohibited
7					1	1	0	Connection permitted	Reading permitted, programming permitted, erasure prohibited
8					0	0	1	Connection permitted	Reading prohibited, programming prohibited, erasure permitted
9					1	0	1	Connection permitted	Reading permitted, programming prohibited, erasure permitted
10					0	1	1	Connection permitted	Reading prohibited, programming permitted, erasure permitted
11					1	1	1	Connection permitted	Reading permitted, programming permitted, erasure permitted

x: Don't care

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

Note 2. This determines whether the control code or ID code sent by the serial programmer matches the control code or ID code set in the OSIS register. When the codes match, connection is permitted; if not, connection is not possible. However, when results of judgment do not match 3 times in a row, the all data in the flash memory will be 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. IDE	SPCC. OCDE	OSIS (ID Code 1)	OSIS (ID Code 2 to 16)	SPCC. RDPR	SPCC. WRPR	SPCC. SEPR	Connection to an OCD
1	x	x	1	Any value	Any value	x	x	x	ID codes matched: Connection to an OCD is permitted ID codes unmatched: Waiting for input of ID code
2	x	x	0	—	—	x	x	x	Connection to an OCD is prohibited (this is independent of the state of ID code matching).

x: Don't care

7.6 Setting the Option-Setting Memory

7.6.1 Allocation of Data in the Option-Setting Memory

Data for programming in the option-setting memory should be allocated to the addresses shown in Figure 7.1.
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 1EFFFFFFh in the serial programmer command control register (SPCC)
.ORG 000120040H
.LWORD 01EFFFFFFH

Setting the following ID codes in the OCD/serial programmer ID setting register (OSIS)
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 000120050H
.LWORD 0040302FFH, 008070605H, 00C0B0A09H, 0100F0E0DH

Setting EF67BA5Dh in the option function select register 0 (OFS0)
.ORG 000120068H
.LWORD 0EF67BA5DH

Setting FFFFFEFAh in the option function select register 1 (OFS1)
.ORG 00012006CH
.LWORD 0FFFFFEFAH

Setting FFFFFFF8h in the endian select register (MDE)
.ORG 000120064H
.LWORD 0FFFFFFF8H

Setting UB codes A and B
.ORG 0FF7FFFE8H
.LWORD 055736572H, 0426F6F74H
.LWORD 0FFFFFF00H, 00008C040H

7.7 Usage Note

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

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

8. Voltage Detection Circuit (LVDA)

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

8.1 Overview

For voltage detection 0, the detection voltage is selectable from among two 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 five 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 two different levels by using OFS1.VDSEL[1:0] bits	Selectable from among five different levels by using LVDLVLR.LVD1LVL[3:0] bits	Selectable from among five 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 Vdet1 crossings	Available Output of event signals on detection of Vdet2 crossings

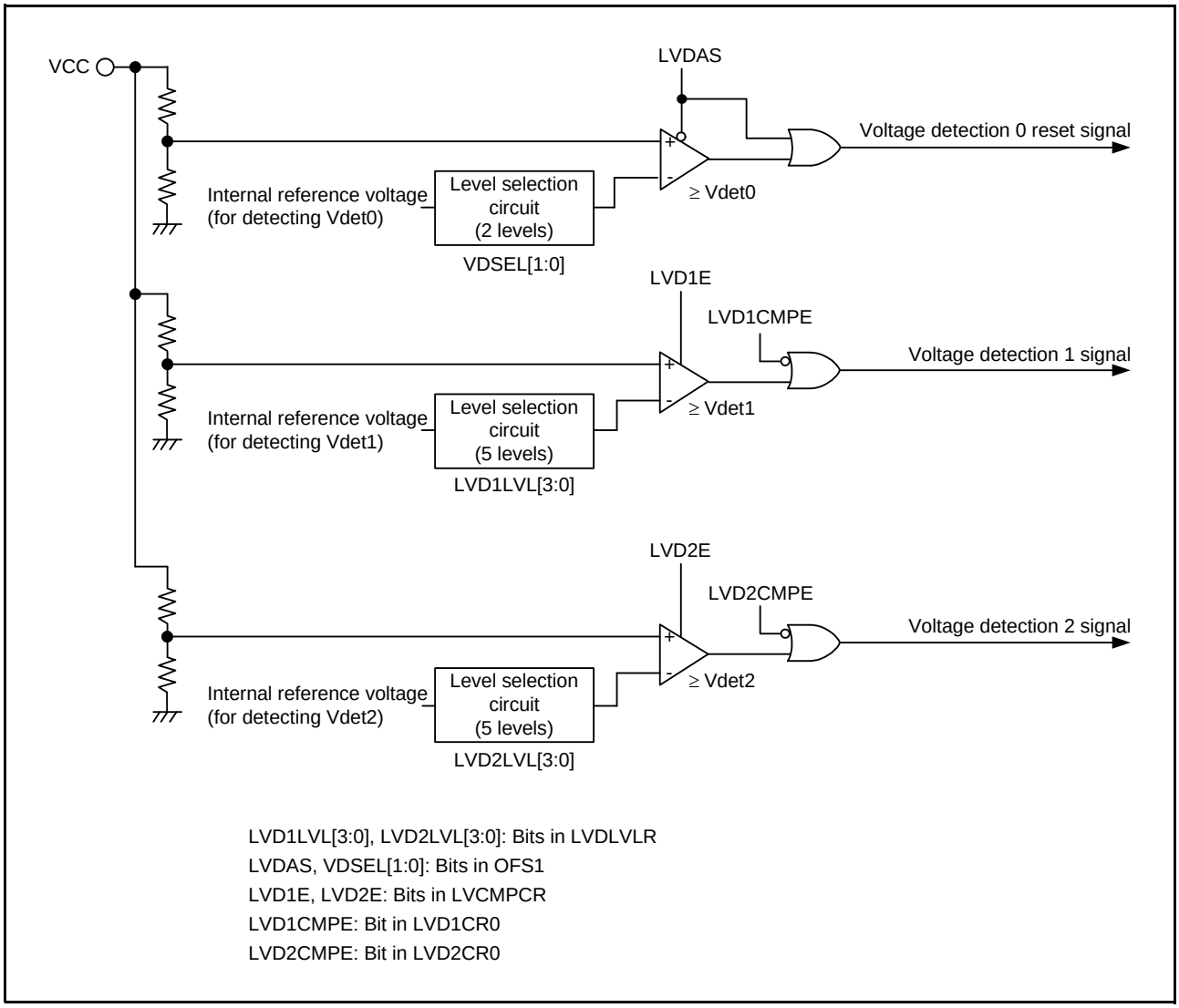


Figure 8.1 Block Diagram of Voltage Detection Circuit

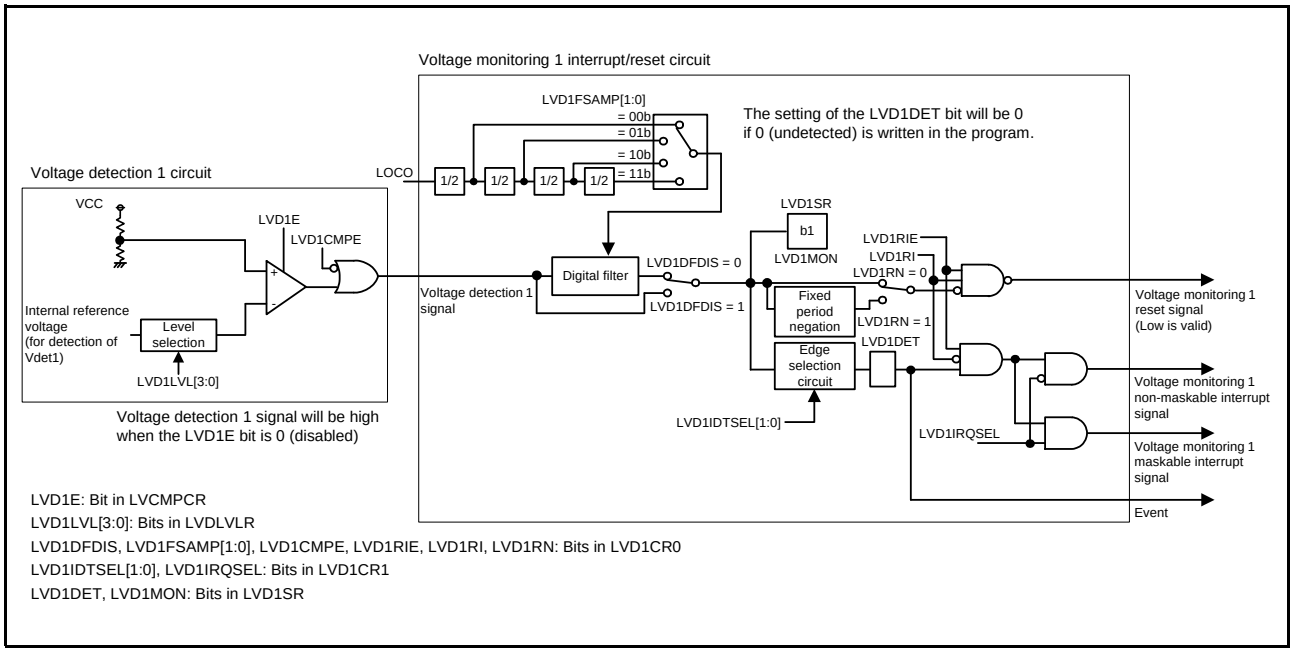


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

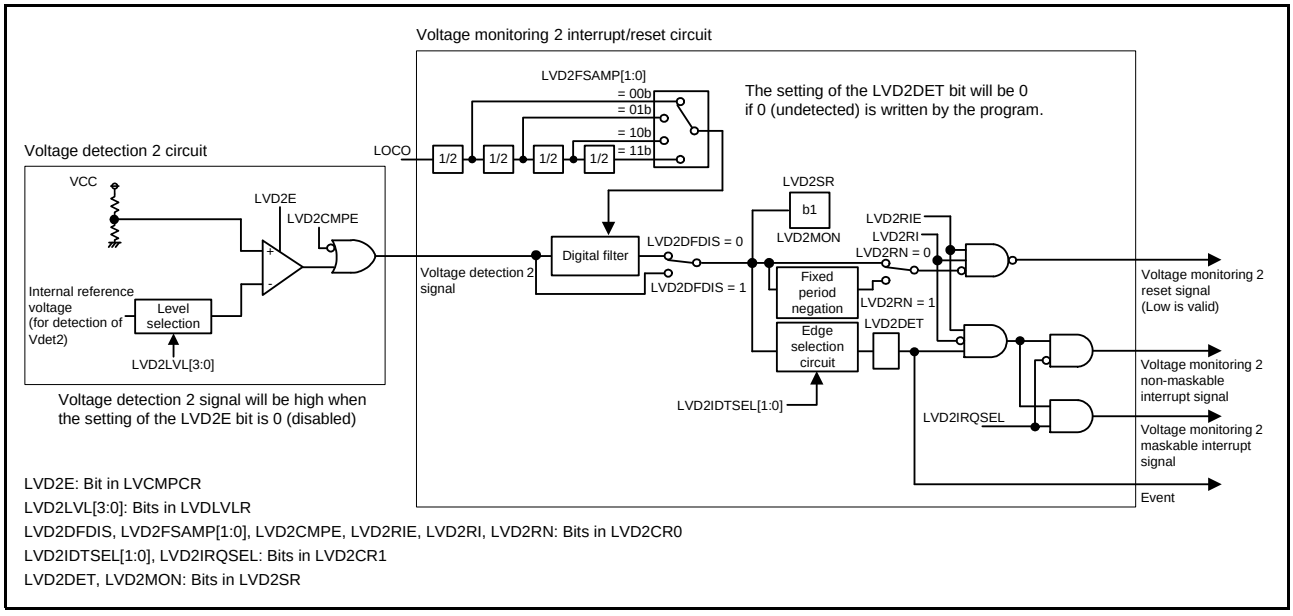


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

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	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2 Enable	0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1.

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.

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:	0	1	1	0	0	1	1	0

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 0 1 0 0: 4.57 V (Vdet1_0) 0 1 0 1: 4.47 V (Vdet1_1) 0 1 1 0: 4.32 V (Vdet1_2) 1 0 1 0: 2.93 V (Vdet1_3) 1 0 1 1: 2.88 V (Vdet1_4) 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 0 1 0 0: 4.57 V (Vdet2_0) 0 1 0 1: 4.47 V (Vdet2_1) 0 1 1 0: 4.32 V (Vdet2_2) 1 0 1 0: 2.93 V (Vdet2_3) 1 0 1 1: 2.88 V (Vdet2_4) Settings other than above are prohibited.	R/W

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

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

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

x: Undefined

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

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

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

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

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

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

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

LVD1RI Bit (Voltage Monitoring 1 Circuit Mode Select)

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

LVD1RN Bit (Voltage Monitoring 1 Reset Negate Select)

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

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

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

x: Undefined

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

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

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

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

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

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

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

LVD2RI Bit (Voltage Monitoring 2 Circuit Mode Select)

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

LVD2RN Bit (Voltage Monitoring 2 Reset Negate Select)

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

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

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

Table 8.2 Procedures for Setting up Monitoring against Vdet1

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

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), see section 45, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.3.3 Monitoring Vdet2

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

Table 8.3 Procedures for Setting up Monitoring against Vdet2

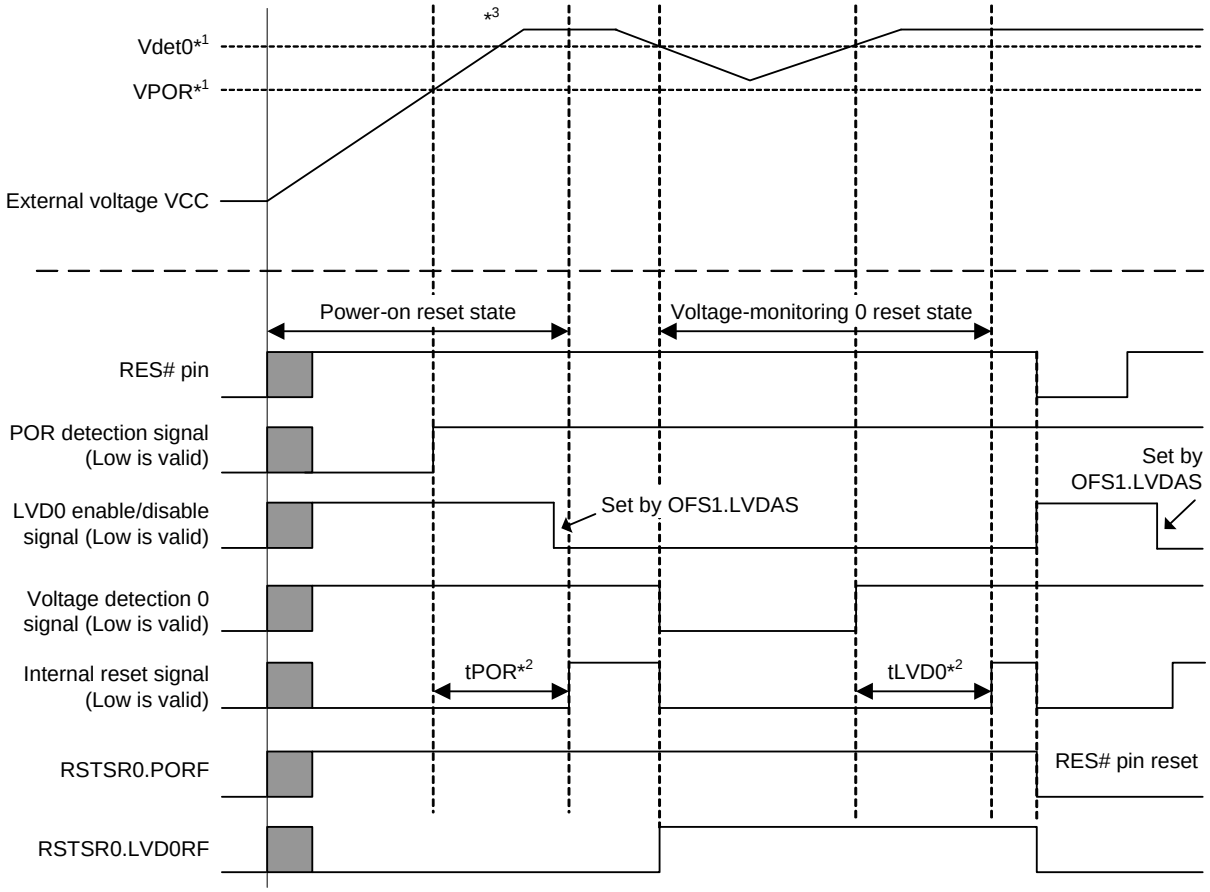
Step		Monitoring the Results of Comparison by Voltage Monitoring 2
Setting the voltage detection 2 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.
	2	Set LVCMPCR.LVD2E = 1 (enabling the voltage detection 2 circuit).
	3	Wait for at least td(E-A) (LVD operation stabilization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

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), see section 45, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.4 Reset from Voltage Monitor 0

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

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



Note: For details of the electrical characteristics, see section 45, 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, see Figure 6.2 in section 6, Resets. Furthermore, if you intend to use the voltage monitoring 1 circuit in software standby or deep software standby mode, make settings for the voltage monitoring 1 circuit according to the following procedures.

(1) Setting in software standby mode

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

(2) Settings in deep software standby mode

- Disable the digital filter (LVD1DFDIS = 1).
- Enable voltage monitoring 1 interrupts (LVD1RI = 0). If the voltage monitoring 1 reset is enabled (LVD1RI = 1), a transition to deep software standby mode will not be possible, and the transition will be to software standby mode instead.

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

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), see section 45, 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.

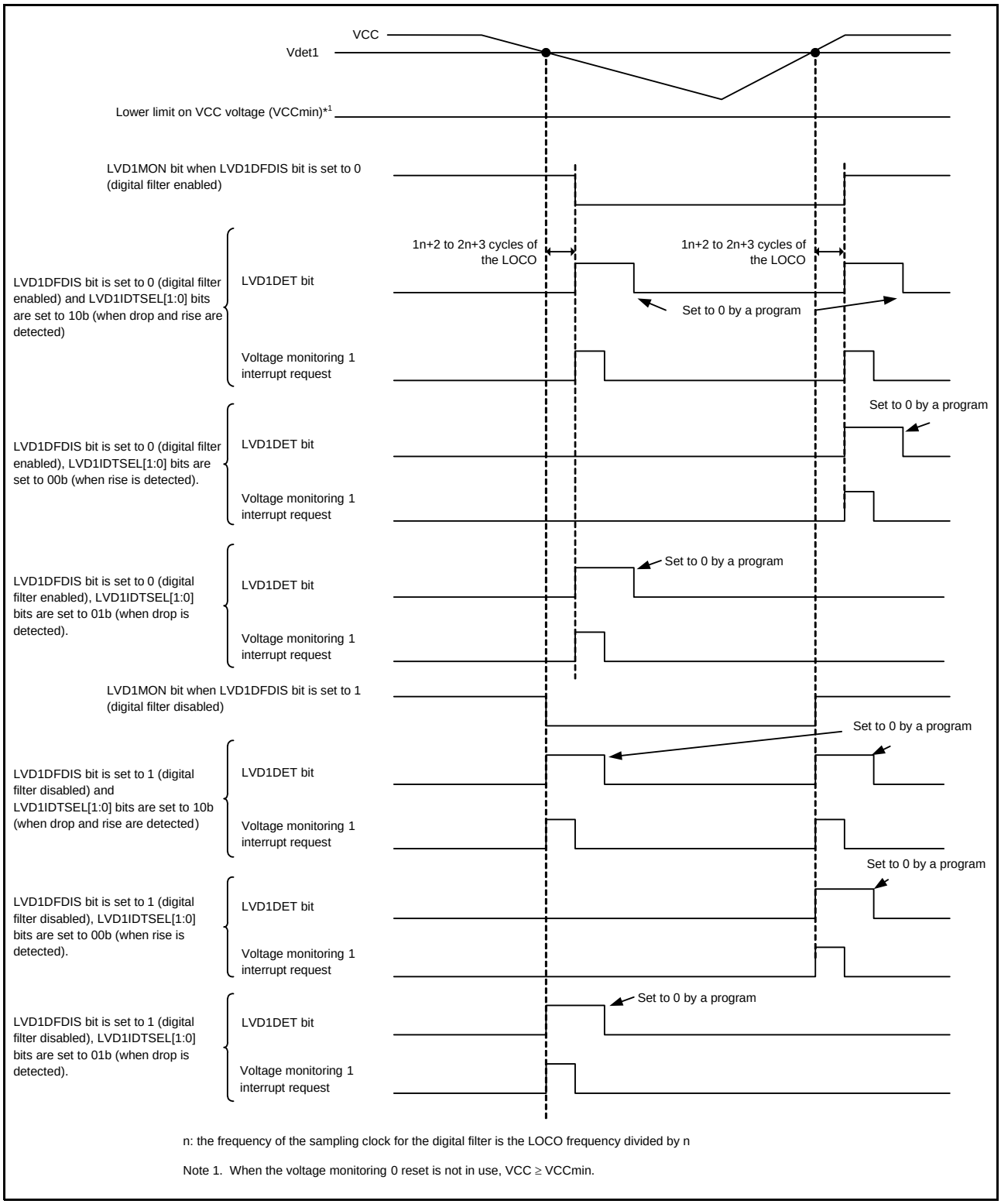
Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Stops

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, see Figure 6.2 in section 6, Resets. Furthermore, if you intend to use the voltage monitoring 2 circuit in software standby or deep software standby mode, make settings for the voltage monitoring 2 circuit according to the following procedures.

(1) Setting in software standby mode

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

(2) Settings in deep software standby mode

- Disable the digital filter (LVD2DFDIS = 1).
- Enable voltage monitoring 2 interrupts (LVD2RI = 0). If the voltage monitoring 2 reset is enabled (LVD2RI = 1), a transition to deep software standby mode will not be possible, and the transition will be to software standby mode instead.

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

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), see section 45, 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.

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

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

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

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

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

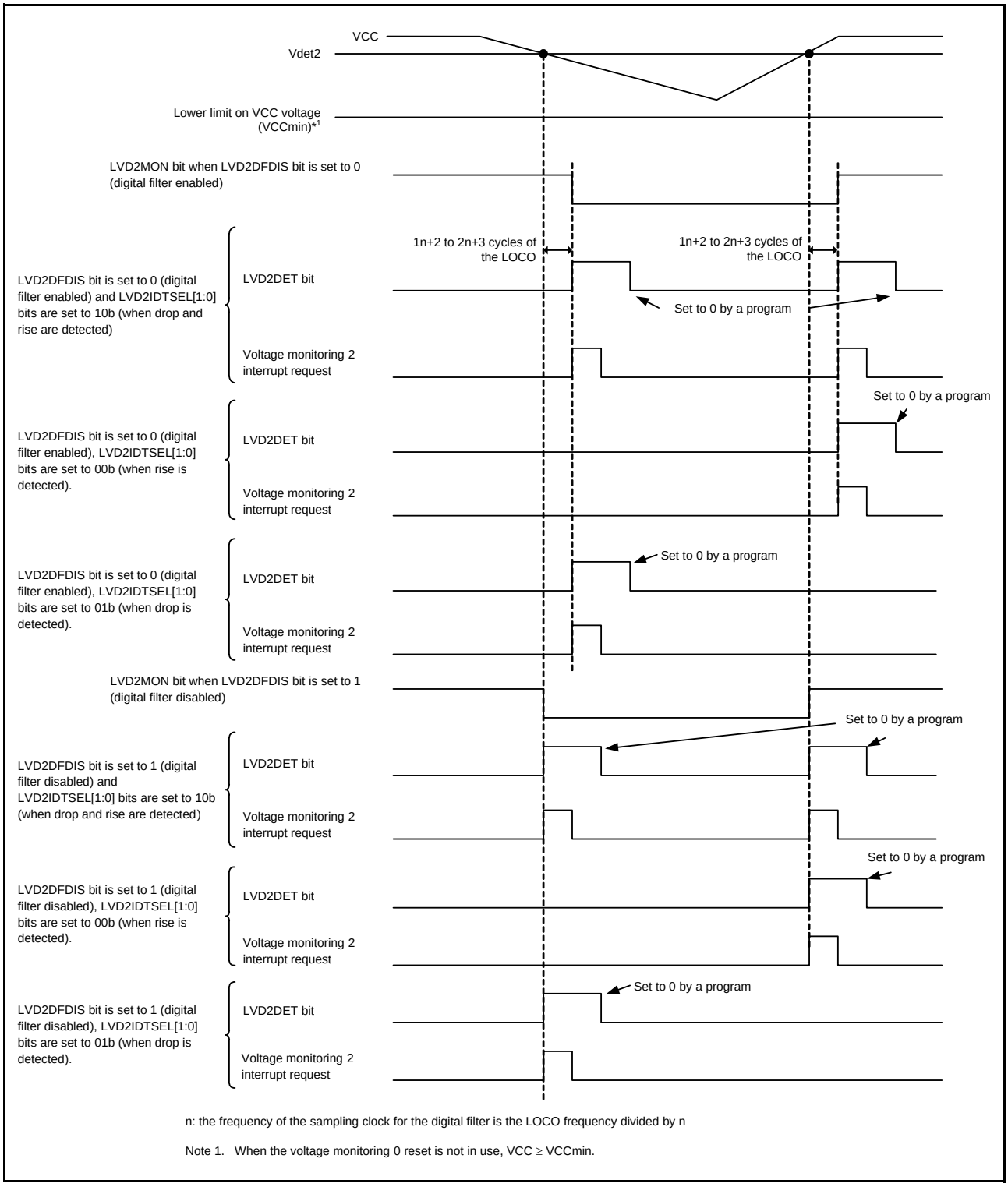


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

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)

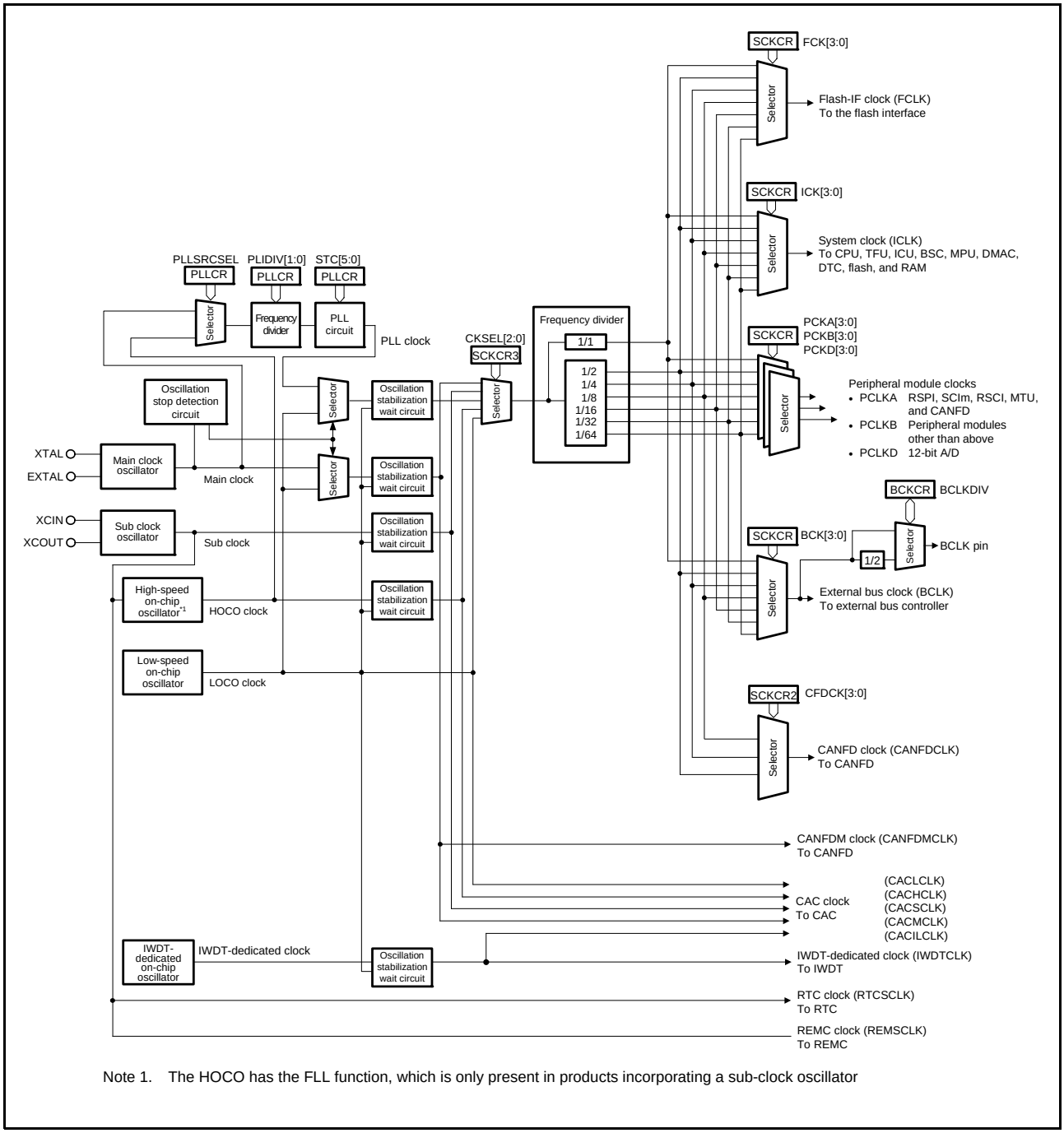
Item	Specification
Use	• Generates the system clock (ICLK) to be supplied to the CPU, TFU, DMAC, DTC, code flash memory, and RAM. • Generates the peripheral module clock (PCLKA) to be supplied to the RSPI, SCIm, RSCI, MTU, and CANFD. • Generates the peripheral module clock (PCLKB) to be supplied to peripheral modules. • Generates the peripheral module clocks (for analog conversion) (PCLKD) 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 CAC clock (CACCLK) to be supplied to the CAC. • Generates the CANFD clock (CANFDCLK) to be supplied to the CANFD. • Generates the CANFD main clock (CANFDMCLK) to be supplied to the CANFD. • Generates the RTC sub-clock (RTCSCLK) to be supplied to the RTC. • Generates the REMC sub-clock (REMSCLK) to be supplied to the REMC. • Generates the IWDT-dedicated clock (IWDTCLK) to be supplied to the IWDT.
Operating frequency*1	• ICLK: 120 MHz (max) • PCLKA: 120 MHz (max) • PCLKB: 60 MHz (max) • PCLKD: 8 MHz to 60 MHz (for conversion with 12-bit A/D converter) • 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: 60 MHz (max) • BCLK pin output: 40 MHz (max) • CACCLK: Same as the clock from respective oscillators. • CANFDCLK: 60 MHz (max) • CANFDMCLK: 24 MHz (max) • RTCSCLK: 32.768 kHz • REMSCLK: 32.768 kHz • IWDTCLK: 120 kHz
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
PLL frequency synthesizer	• Input clock source: Main clock, HOCO*2 • 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 • FLL function (only present in products incorporating a sub-clock oscillator)
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 240 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 120 kHz
Control of output on the BCLK pin	• BCLK clock output or high output is selectable • BCLK or BCLK/2 is selectable

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

Item	Specification
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 clock-frequency settings: ICLK ≥ BCLK, PCLKA ≥ PCLKB
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:PCLKD = N:1 or 1:N;
PCLKB:PCLKD = 1:1, 2:1, 4:1 or 1:2
Restrictions on clock-frequency settings when the CAN FD module is to be used: PCLKA:PCLKB = 2:1, PCLKB ≥ CANFDCLK, PCLKB ≥ CANFDMCLK

Note 2. When using the HOCO as the input clock source for the PLL, select the multiplication ratio of the PLL so that the HOCO clock oscillation frequency (min./max.) is in the range of 120 to 240 MHz.



Note 1. The HOCO has the FLL function, which is only present in products incorporating a sub-clock oscillator

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	
BCLK	Output	This pin is used to supply external devices with the external bus clock (BCLK).
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator.
XCOUT	Output	

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	—	—	—	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	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	This MCU does not have PCLKC. However, set these bits to 0001b.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select	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	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	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
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	PSTOP1	BCLK Pin Output Control*2	0: BCLK pin output is enabled. 1: BCLK pin output is disabled. (Fixed high)	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select*1	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	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. Do not make a setting such that the ICLK runs at a lower frequency than the external bus clock.
Note 2. 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.

The SCKCR register should not be modified in the following cases:

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

9.2.2 System Clock Control Register 2 (SCKCR2)

Address(es): 0008 0024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDCK[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	1	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
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	CFDCK[3:0]	CANFD Clock (CANFDCLK) Select	b15 b12 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 Settings other than above are prohibited.	R/W

The SCKCR2 register should not be modified in the following cases:

- 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

9.2.3 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

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

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

The SCKCR3 register should not be modified in the following cases:

- 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, and PCLKD), flash-IF clock (FCLK), external bus clock (BCLK), and CANFD clock (CANFDCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, the sub-clock oscillator, and the PLL circuit. Transitions to clock sources which are not in operation are prohibited.

9.2.4 PLL Control Register (PLLCR)

Address(es): 0008 0028h

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

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	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

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

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

These bits select the frequency division ratio of the PLL clock source.
Set these bits so that the frequency of PLL input signal is within the range of 8 MHz to 24 MHz.

PLLSRCSEL Bit (PLL Clock Source Select)

This bit selects the clock source for the PLL.

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

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

9.2.5 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

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

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

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

The PLL clock source is selectable as the main clock oscillator and HOCO.

Selecting the main clock oscillator as the PLL clock source with the PLLCR.PLLSRCSEL bit requires setting the main clock oscillator wait control register (MOSCWTCR).

After the setting of the PLLEN bit has been changed to make the PLL run, only start using the PLL clock after confirming that the OSCOVFSR.PLOVF flag has been set to 1.

That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation by the PLLEN bit. The following notes apply when selecting the main clock oscillator as the PLL clock source.

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

Writing of 1 to the PLLEN bit (stopping the PLL) is prohibited while the PLL clock is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

9.2.6 External Bus Clock Control Register (BCKCR)

Address(es): 0008 0030h

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

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

The BCKCR register should not be modified in the following cases:

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

BCLKDIV Bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.
Either the BCLK clock with the frequency selected by the BCK[3:0] bits in SCKCR or the BCLK clock divided by 2 can be selected. To control external bus control signals at the falling edge of the BCLK pin, set this bit to 1.

9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

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

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Main clock oscillator is operating. 1: Main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

The main clock oscillator is operated or stopped by the MOSTP bit. The main clock oscillator can be started by setting the MOSTP bit to operating.

To make the main clock run, 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.

For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

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

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited in either of the following cases:

- The main clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- 0 (PLL operation) is selected by the PLL stop control bit in PLL control register 2 (PLLCR2.PLLEN), and the main clock oscillator is selected by the PLL clock source select bit in the PLL control register (PLLCR.PLLSRCSEL).

9.2.8 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h

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

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 oscillator forced oscillation bit (SOFCR.SOFE) 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 SOFCR.SOFE 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 or the operating clock of the remote control signal receiver (REMC), set the SOFCR.SOFE bit to 1 (forcibly making the sub-clock oscillator run) and the SOSCCR.SOSTP bit to 0 (making the sub-clock oscillator run). When the sub-clock is not to be used as the source to drive counting by the realtime clock or the operating clock of the REMC but is to be used as the system clock, set the SOSCCR.SOSTP bit to 0 (making the sub-clock oscillator run).

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.
- If a reset other than the power-on reset is generated while the sub-clock oscillator is operating with the SOFCR.SOFE bit set to 1 (forcibly making the sub-clock oscillator run) and the SOSCCR.SOSTP bit set to 0

(making the sub-clock oscillator run), the sub-clock oscillator continues to oscillate but the SOSTP bit is initialized to 1 (stopping the sub-clock oscillator). After release from the reset state, set the SOSTP bit to 0.

Writing of 1 to the SOSTP bit (stopping the sub-clock oscillator) is prohibited while the sub-clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

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

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

LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO after the LOCO clock oscillation stabilization waiting time (tLOCOWT) has elapsed.

That is, a fixed time for stabilization of oscillation is required for oscillation to become stable after setting LOCO operation with the LCSTP bit. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby or deep software standby.
- When a transition to software standby or deep software standby is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited while the LOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited if detection of oscillation stopping is enabled by the oscillation stop detection enable bit in the oscillation stop detection control register (OSTDCR.OSTDE).

Since the LOCO clock is used to measure the waiting time for other oscillators, the LOCO clock oscillates while the waiting time for other oscillators is being measured, regardless of the setting of LCSTP bit. Therefore, the LOCO clock may be unintentionally supplied even if the LCSTP bit is set to be stopped.

9.2.10 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

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

Bit	Symbol	Bit Name	Description	R/W
b0	ILCSTP	IWDT-Dedicated On-Chip Oscillator Stop	0: IWDT-dedicated on-chip oscillator is operating. 1: IWDT-dedicated on-chip oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the IWDT start mode select bit in the option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator operating) to 1 (IWDT-dedicated on-chip oscillator stopped) while ILOCOCR is valid.

ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed to make the IWDT-dedicated on-chip oscillator run, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before starting to use the oscillator.

When a transition to software standby or deep software standby mode is to follow the setting to start the IWDT-dedicated on-chip oscillator, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before executing the WAIT instruction.

9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

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

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

Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

After the setting of the HCSTP bit has been changed to make the HOCO run, confirm that the OSCOVFSR.HCOVF flag has been set to 1 before starting to use the oscillator.

A fixed time for stabilization is required for oscillation to become stable after setting HOCO operation with the HCSTP bit. Furthermore, a fixed time is required for oscillation to actually stop after the setting to stop operation. Accordingly, take note of the following when starting and stopping operation with the HCSTP bit.

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

Writing of 1 to the HCSTP bit (stopping the HOCO) is prohibited while the HOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), or the HOCO is selected as the clock source for the PLL by the PLLCR.PLLSRCSEL bit, and the PLL is selected by the SCKCR3.CKSEL[2:0] bits.

9.2.12 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2)

Address(es): 0008 0037h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HCFRQ[1:0]	HOCO Frequency Setting	b1 b0 0 0: 16 MHz 0 1: 18 MHz 1 0: 20 MHz Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to the HOCOCR2 register is prohibited when the HOCOCR.HCSTP bit is 0 (making the HOCO run).

9.2.13 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 9.2 shows the flow of setting FLL after release from the reset state or deep software standby mode. Figure 9.3 shows the flow of setting FLL before transition to and after release from the software standby mode.

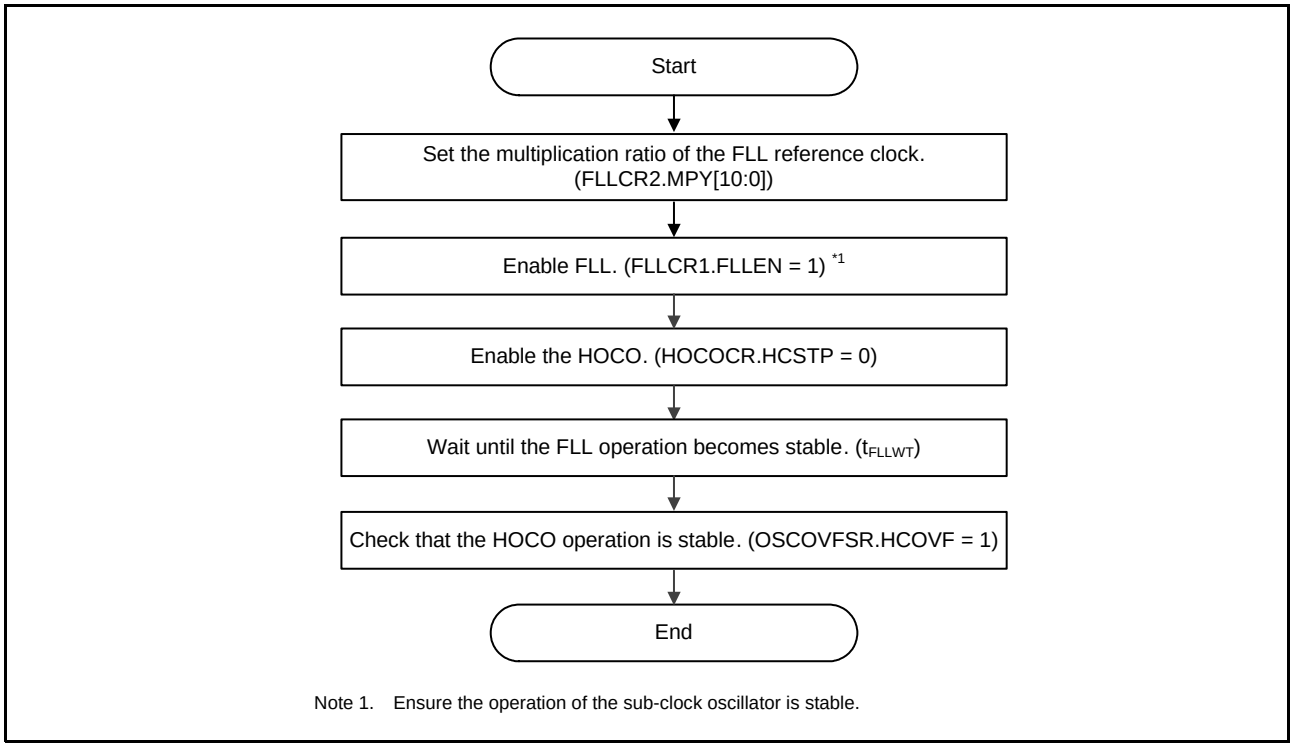
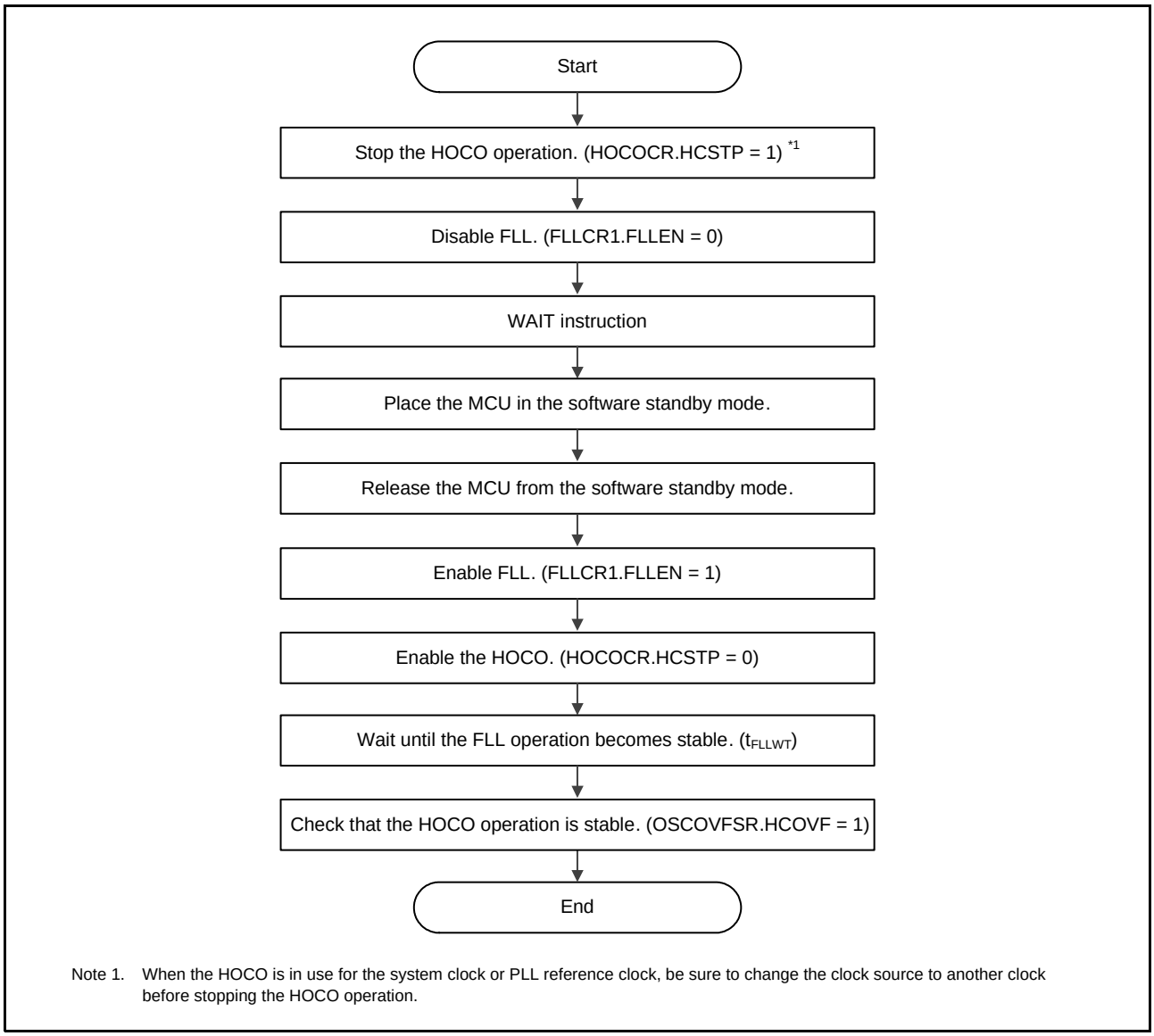


Figure 9.2 Flow of Setting FLL after Release from the Reset State or Deep Software Standby Mode



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 9.3 Flow of Setting FLL before Transition to and after Release from the Software Standby Mode

9.2.14 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).

9.2.15 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 1: Oscillation of the main clock is stable so the clock is available for use as the system clock.	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.*3 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 so the clock is available for use as the IWDT-dedicated clock.	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. 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.
Note 4. The SOOVF flag does not reflect control of the sub-clock oscillator by the SOFCR.SOFE bit. Accordingly, if the SOSCCR.SOSTP bit is set to 1 while the value of the SOFCR.SOFE bit is 1, the sub-clock oscillator continues to oscillate but the value of the SOOVF flag becomes 0.

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.

9.2.16 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

The 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).

9.2.17 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. When a resonator is selected as the source for the main clock oscillator, apply a reset to clear the OSTDF flag.

[Setting condition]

- The main clock oscillation is stopped with the OSTDCR.OSTDE being 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKCR3.CKSEL[2:0] bits are neither 010b nor 100b.

9.2.18 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	MSTS[7:0]							
Value after reset:	0	1	0	1	0	0	1	1

The MOSCWTCR register 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

9.2.19 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

The SOSCWTCR register 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

9.2.20 Sub-Clock Oscillator Forced Oscillation Control Register (SOFCR)

Address(es): 0008 C292h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SOFE	Sub-Clock Oscillator Forced Oscillation	0: Oscillator is not controlled by this bit. 1: The sub-clock oscillator is forcedly oscillated.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SOFCR register is used to control forced oscillation of the sub-clock oscillator.

SOFE Bit (Sub-Clock Oscillator Forced Oscillation)

This bit controls forced oscillation of the sub-clock oscillator. Setting this bit to 1 allows the continued use of the sub-clock oscillator while the chip is on software standby or deep software standby.

For a further detail on control over the sub-clock oscillator, see section 9.2.8, Sub-Clock Oscillator Control Register (SOSCCR).

9.2.21 Main Clock Oscillator Function Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV2[1:0]		—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	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

The MOFCR register selects the driving ability of the main clock oscillator, and also selects the oscillator or an external clock signal.

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, see section 45, Electrical Characteristics.

Do not change the value of the HOCOPCNT bit 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]).

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 9.4 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 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.

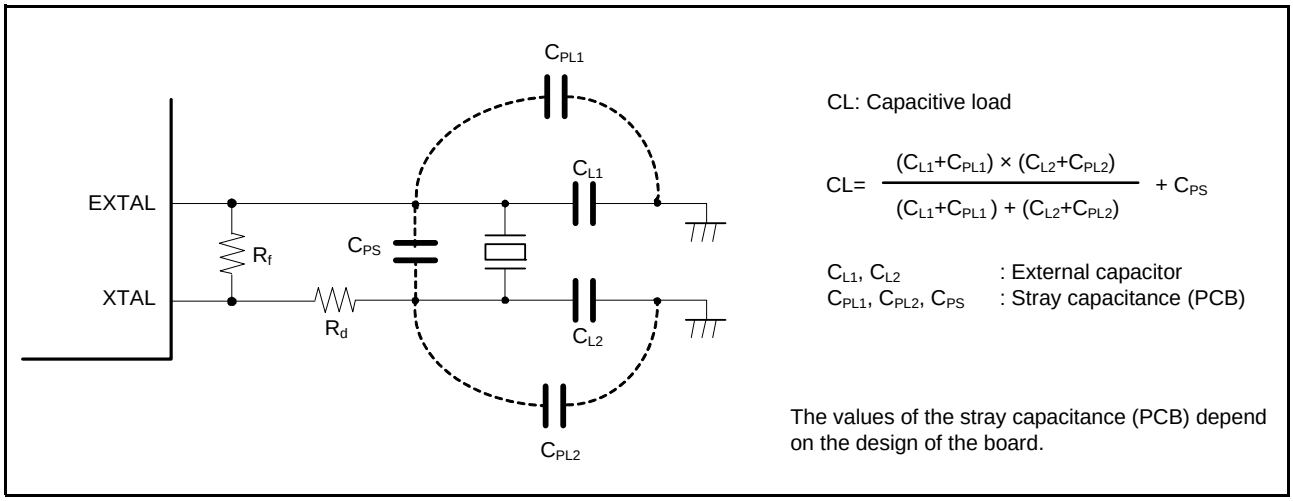


Figure 9.4 Example of Crystal Connection

Figure 9.5 shows an equivalent circuit of the crystal resonator.

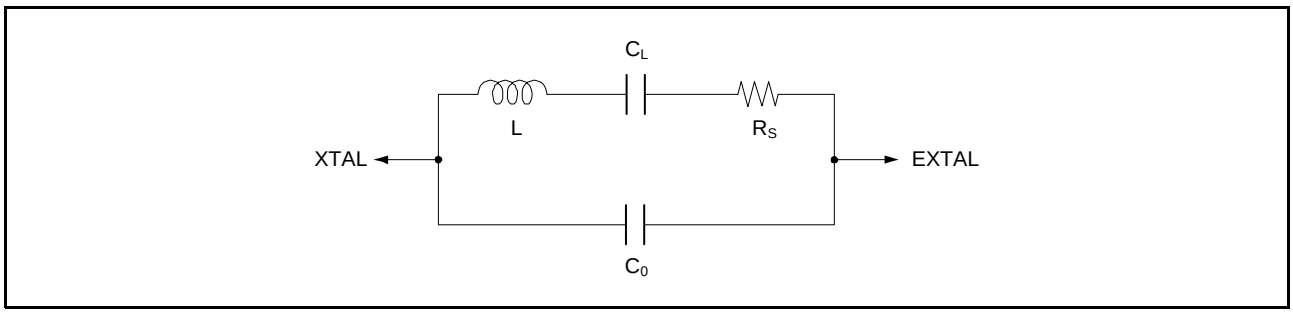


Figure 9.5 Equivalent Circuit of Crystal Resonator

9.3.2 External Clock Input

Figure 9.6 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.

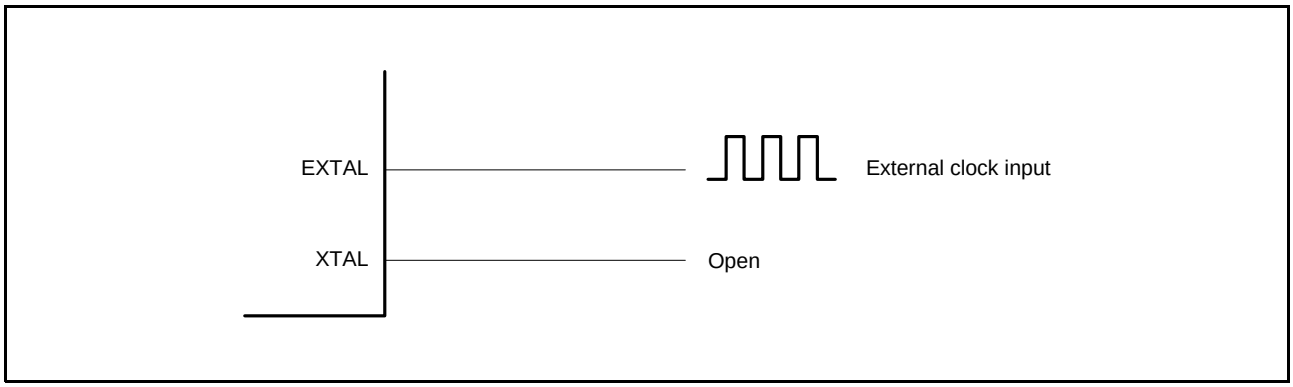


Figure 9.6 Example of Connection of External Clock

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).

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 9.7. A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator, 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.

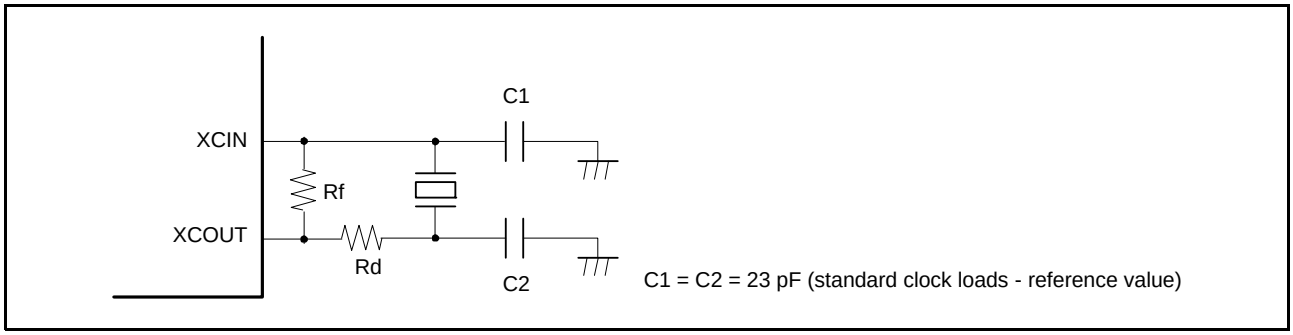


Figure 9.7 Connection Example of 32.768-kHz Crystal Resonator

Figure 9.8 shows an equivalent circuit for the 32.768-kHz crystal resonator. Use a crystal resonator that has the characteristics listed in Table 9.3.

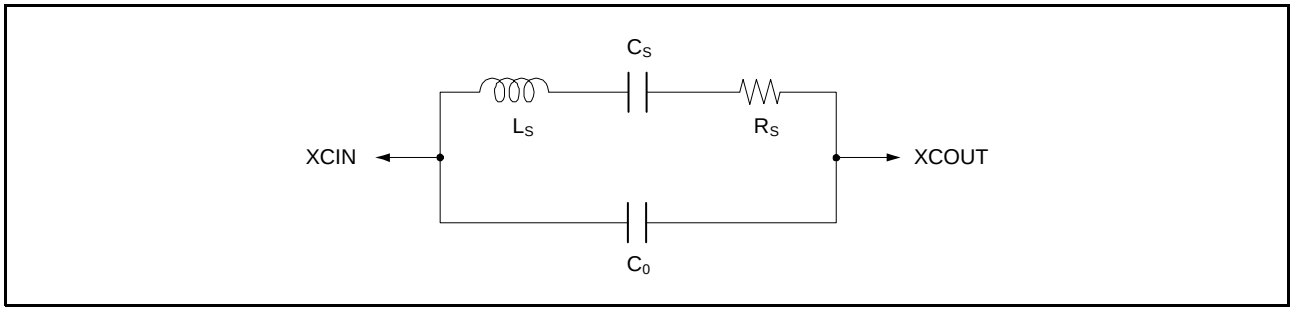


Figure 9.8 Equivalent Circuit for Crystal Resonator

Table 9.3 Crystal Resonator Characteristics (Reference Values)

Frequency (kHz)	32.768 (Standard Clock Loads)
R_S max (kΩ)	60

Standard clock loads: 12.5 pF

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 9.9.
In addition, if an oscillator for the sub-clock is not connected, set the SOSCCR.SOSTP bit to 1 (stopping the sub-clock oscillator) and the sub-clock oscillator forced oscillation bit (SOFCR.SOFE) to 0 (no control over the oscillator by this bit).

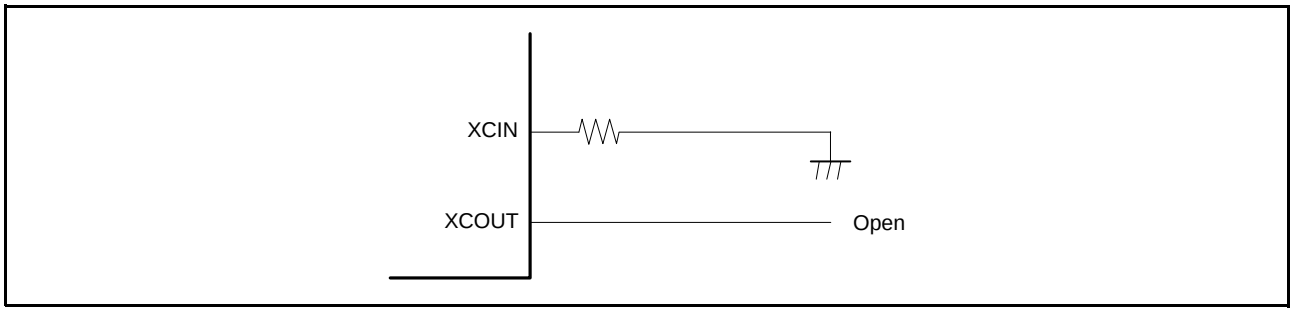


Figure 9.9 Pin Handling when Sub-Clock is Not Used

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, see section 22, Multi-Function Timer Pulse Unit 3 (MTU3a) and section 23, 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 45.52, 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). Switching to the LOCO clock is carried out by setting the OSTDF flag.

After a reset is released, activate the main clock oscillator, and then, set the SCKCR3.CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation settling time has elapsed. If a resonator is selected as the source of the main clock oscillator, apply a reset to clear the OSTDF flag.

When the external clock input is selected as the source for the main clock oscillator, clearing the OSTDF flag by the software switches the clock source back to the main clock or PLL clock.

However, the OSTDF flag cannot be cleared if the main clock oscillator is selected with the SCKCR3.CKSEL [2:0] bits, or if the PLL clock is selected in a state that the main clock oscillator is set for a clock source. After detecting the oscillation stop, in order to set the clock source back to the main clock or the PLL clock, change the setting of the SCKCR3.CKSEL[2:0] bits to the setting other than the main clock oscillator and the PLL clock with the main clock oscillator as the clock source, and then, clear the OSTDF flag by the software. Subsequently, confirm that the OSTDF flag is not set again, and then, set the SCKCR3.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), and CANFD main clock (CANFDMCLK), which are provided as the system clock sources. 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 9.10 shows an example of a flowchart for initialization of the oscillation stop detection function.

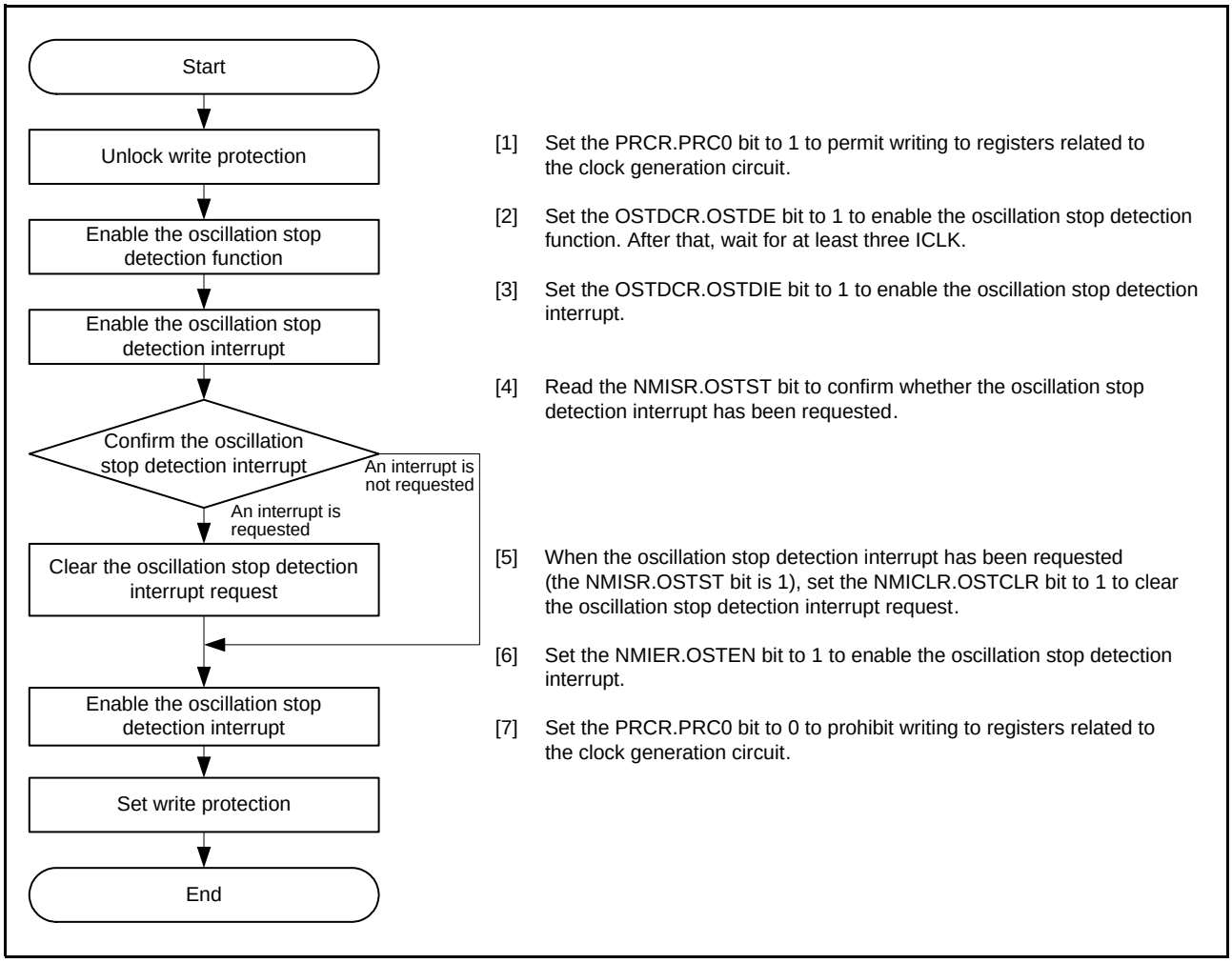
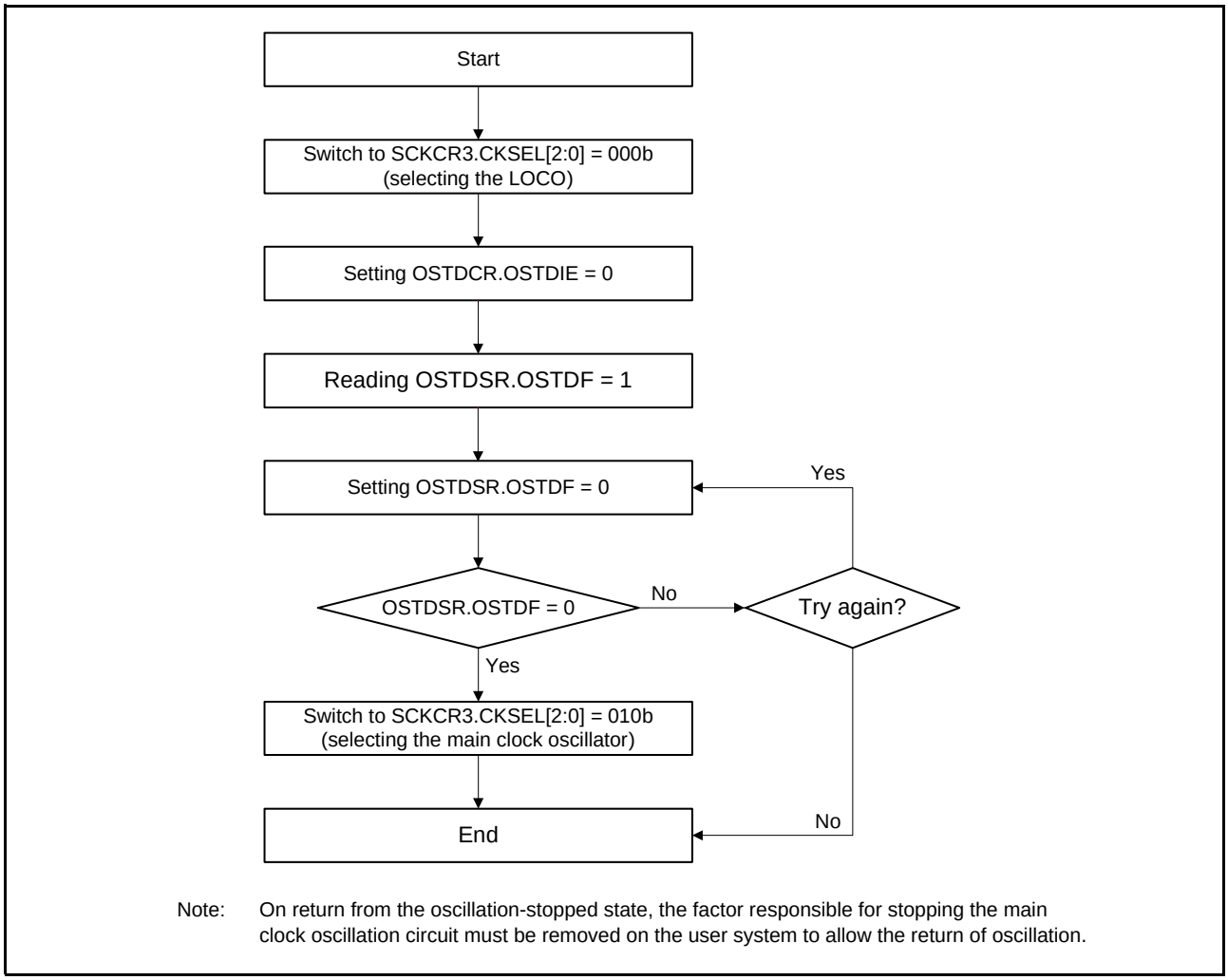


Figure 9.10 Flowchart Example for Initialization of Oscillation Stop Detection Function



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 9.11 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 POE3 sets the OSTST high-impedance flag in input level control/status register 6 (ICSR6.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLKB before writing to this ICSR6.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE). Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured.

If the oscillation stop detection interrupt is to be used as a non-maskable interrupt, since non-maskable interrupts are disabled in the initial state after a reset release, set the corresponding bit in the NMIER register to 1 by software to enable non-maskable interrupts. If it is to be used as a maskable interrupt, do not change the value of the NMIER register from the value after a reset. For details, see section 14, Interrupt Controller (ICUF).

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, and dedicated clock for the IWDT. 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], and PCKD[3:0] bits in SCKCR, the BCLKDIV bit in BCKCR, the CFDCK[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 9.4 Internal Clocks and Supply Destination Modules

	Type of Internal Clock	Clock Name	Supply Destination Module
1	System clock	ICLK	CPU, TFU, code flash memory, RAM, ICU, BSC, DMAC, DTC, MPU
2	Peripheral module clocks	PCLKA	RSPI, SCIm, RSCI, MTU, CANFD
		PCLKB	TMR, CMT, CMTW, RTC, WDT, IWDT, POE3, SCIk, SCIh, RIIC, CANFD, REMC, S12AD, R12DA, CMPC, CRC, DOC, CAC, I/O ports, MPC, ICU, ELC, temperature sensor
		PCLKD	S12AD
3	Flash-IF clock	FCLK	Data flash memory, code flash memory (P/E)
4	External bus clock	BCLK	BSC, I/O ports
5	CANFD clock	CANFDCLK	CANFD
6	CANFD main clock	CANFDMCLK	CANFD
7	CAC clocks	CACMCLK (Main clock)	CAC
		CACSCLK (Sub-clock)	
		CACHCLK (HOCO clock)	
		CACLCLK (LOCO clock)	
		CACILCLK (IWDT-dedicated clock)	
8	RTC clock	RTCSCLK (Sub-clock)	RTC
9	REMC clock	REMSCLK (Sub-clock)	REMC
10	IWDT-dedicated clock	IWDTCLK	IWDT

9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, ICU, BSC, MPU, DMAC, DTC, code flash memory, and RAM.
The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, 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], 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. It is also output externally from the BCLK pin to the external bus. When the external bus is enabled, P53 that is function-multiplexed with the BCLK pin cannot be used as an I/O port.
BCLK can be output from the BCLK pin by setting the SCKCR.PSTOP1 bit to 0 and setting the external bus enable bit in the system control register 0 (SYSCR0.EXBE) to 1. Make sure that modification of the SYSCR0.EXBE bit to 1 must always be performed while the PSTOP1 bit in SCKCR is 1.
When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the BCLK pin.
The BCLK frequency is specified by the BCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.
A frequency higher than the system clock (ICLK) should not be set for the BCLK. If such a frequency is set, the clock frequency will be the same as the ICLK.

9.7.5 CANFD Clock

The CANFD clock (CANFDCLK) is used as the operating clock for the CANFD.
Use the SCKCR2.CFDCK[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0], or PLLCR.PLIDIV[1:0] bits to set the CANFDCLK frequency.

9.7.6 CANFD Main Clock

The CANFD main clock (CANFDMCLK) is an operating clock for the CANFD.
CANFDMCLK is generated by the main clock oscillator.

9.7.7 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.

9.7.8 RTC Clock

The RTC clock (RTCSCLK) is the operating clock for the RTC.
RTCSCLK is generated by the sub-clock oscillator.

9.7.9 REMC Clock

The REMC clock (REMSCLK) is the operating clock for the REMC module.
REMSCLK is generated by the sub-clock oscillator.

9.7.10 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.

9.8 Clock Source Switching

In this MCU, the clock signal from the LOCO, which oscillates during release from the reset state, is used to start the fetching of CPU instructions after the internal reset time (tRESWT) has elapsed. After that, set up the clock to which the CPU will be switched while it is still driven by the LOCO, and read the oscillation stabilization flag register to confirm that oscillation of the given clock signal is stable before switching to the selected clock source.

(1) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: main clock) after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Set the frequency multiplication factor in the PLLCR register (the initial setting for the PLL clock source selects the main clock oscillator).
5. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
6. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
7. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
8. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(2) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: HOCO) after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Set the frequency multiplication factor and set the HOCO clock as the PLL clock source by writing to the PLLCR register.
4. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
5. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
6. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
7. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(3) Example of procedure for settings to switch the system clock source from the LOCO to the main clock after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Confirm that the main clock has become stable by reading the MOOVF flag in the OSCOVFSR register.
5. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
6. Change the clock signal from the LOCO clock to the main clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(4) Example of procedure for settings to switch the system clock source from the LOCO to the HOCO after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Confirm that the HOCO clock has become stable by reading the HCOVF flag in the OSCOVFSR register.
4. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
5. Change the clock signal from the LOCO clock to the HOCO clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

9.9 Operations Linked by the ELC

9.9.1 Event Signal Output to the ELC

The clock generation circuit is capable of operation linked with another module set in advance when its interrupt request signal is used as an even signal by the event link controller (ELC) on detection of stopping of the main clock oscillation. The clock generation circuit outputs the event signal regardless of the setting of the corresponding oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE). For details, see section 19, Event Link Controller (ELC).

9.9.2 Clock Source Switching on Reception of the Event Signal from the ELC

The clock generation circuit is capable of switching the clock source to the low-speed on-chip oscillator in response to the event set in advance when the event specified in the ELSRn register of the ELC occurs.
While this function is in use, clock source switching on return from sleep mode cannot be used. For details, see section 11.2.6, 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), and external bus clock (BCLK) 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.
This MCU does not have PCLKC. However, set the SCKCR.PCKC[3:0] bits to 0001b.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKA and PCLKB. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The following relation between the frequencies of the system clock (ICLK) and external bus clock (BCLK) must apply.
ICLK ≥ BCLK
Also, the following relations between the frequencies of the peripheral module clocks must apply.
PCLKA ≥ PCLKB
PCLKB:PCLKD = 1:1, 2:1, 4:1, or 1:2

(3) The following relations between frequencies must apply if the CAN FD module is to be used.
PCLKA:PCLKB = 2:1
PCLKB ≥ CANFDCLK
PCLKB ≥ CANFDMCLK

(4) 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.

(5) If the clock frequency is to be changed by modifying the value of SCKCR, SCKCR2, SCKCR3, or BCKCR register, wait for writing of the value to the register to be complete and the new frequency to be stable before starting subsequent processing. For the procedure to confirm the completion of writing to I/O registers, see (2) Notes on writing to I/O registers, in section 5, I/O Registers.

9.10.2 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.10.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.12 to prevent electromagnetic induction from interfering with correct oscillation.

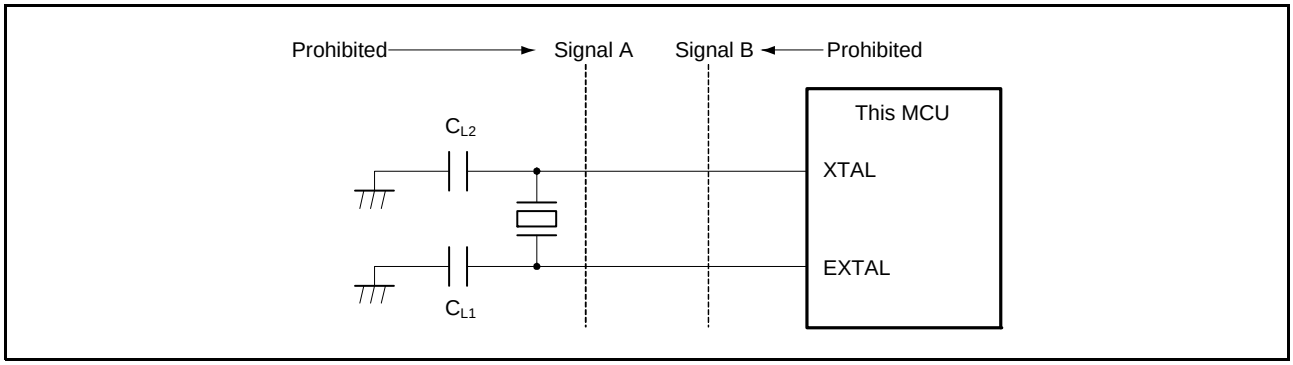


Figure 9.12 Notes on Board Design for Oscillation Circuit

9.10.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When they are used as the general ports, the main clock should be stopped (MOSCCR.MOSTP should be set to 1). 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 21.42, Register Settings.

9.10.5 Notes on Sub-Clock Oscillator

The sub-clock can be used as the system clock, as the source to drive counting by the realtime clock, or as both. Accordingly, take note of the following limitations and points for caution regarding the settings.

- To select the sub-clock as the system clock, set the SOSCCR.SOSTP bit to 0 (making the sub-clock oscillator run). On the other hand, to set the sub-clock as the source to drive counting by the realtime clock or the operating clock of the remote control signal receiver (REMC), set the SOFCR.SOFE bit to 1 (forcibly making the sub-clock oscillator run) and the SOSCCR.SOSTP bit to 0 (making the sub-clock oscillator run). Furthermore, when the sub-clock is to be used as the source to drive counting by the realtime clock or the operating clock of the REMC, 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 9.13 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 the operating clock of the REMC, or when the sub-clock is only to be used as the source to drive counting by the realtime clock*1 or the operating clock of the REMC. Make settings for the realtime clock in accord with section 27.3.2, Clock and Count Mode Setting Procedure.

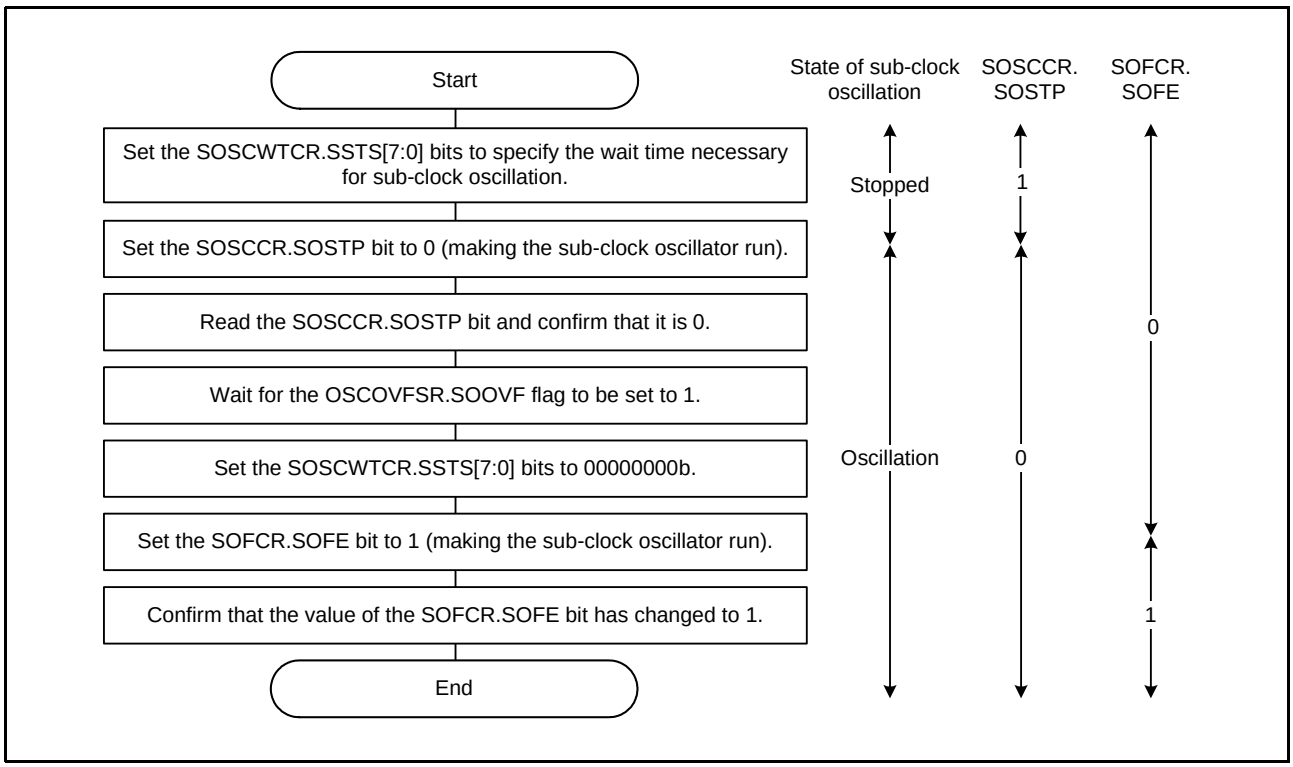


Figure 9.13 Example Flowchart of Initialization when the Sub-Clock is to be Used as the Source to Drive Counting by the Realtime Clock or the Operating Clock of the REMC

Note 1. If the realtime clock is not to be used, initialize the realtime clock in accord with section 27.6.7, Initialization Procedure When the Realtime Clock is Not to be Used.

- Make initial settings according to the example flowchart shown in Figure 9.14 when the sub-clock is only to be used as the system clock*2.

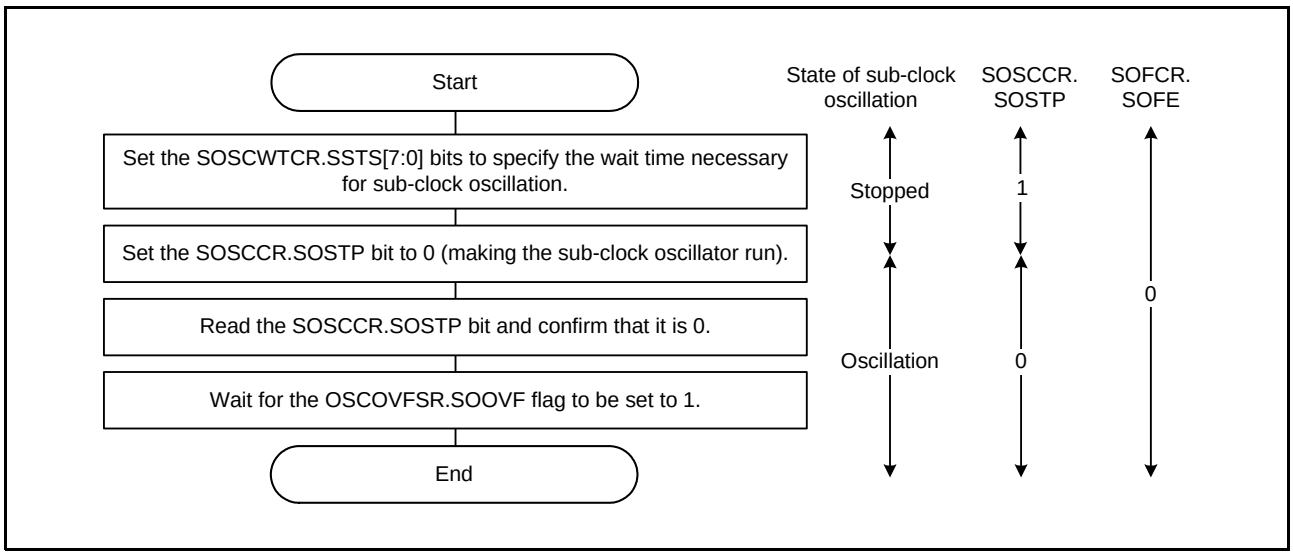


Figure 9.14 Example Flowchart for Initialization when the Sub-Clock is to be Used only as the System Clock

- When the sub-clock is used as the system clock, even if the SOFCR.SOFE 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.
- If the SOFCR.SOFE bit is modified after the SOSCCR.SOSTP bit is modified, or if the SOSCCR.SOSTP bit is modified after the SOFCR.SOFE bit is modified, make sure that first modified bit has been rewritten before the subsequent bit is modified.

Note 2. If the realtime clock is not to be used, initialize the realtime clock in accord with section 27.6.7, Initialization Procedure When the Realtime Clock is Not to be Used.

9.10.6 Point to Note regarding Products with No Sub-clock Oscillator

The realtime clock cannot be used in products with no sub-clock oscillator. In such products, the internal state of the realtime clock is undefined after power on. Accordingly, after the power is turned on, disable the realtime clock in accord with section 27.6.7, Initialization Procedure When the Realtime Clock is Not to be Used.

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.

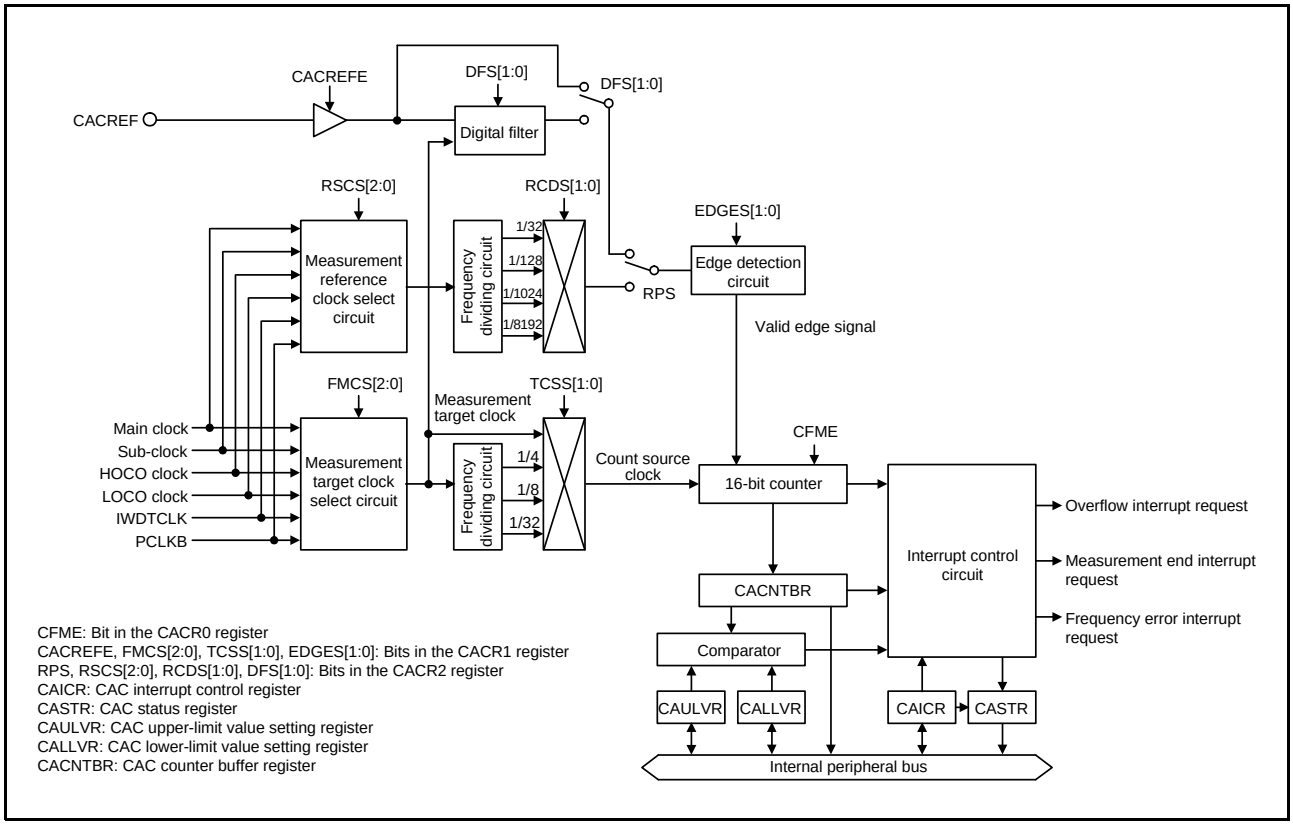


Figure 10.1 CAC Block Diagram

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

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: 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.

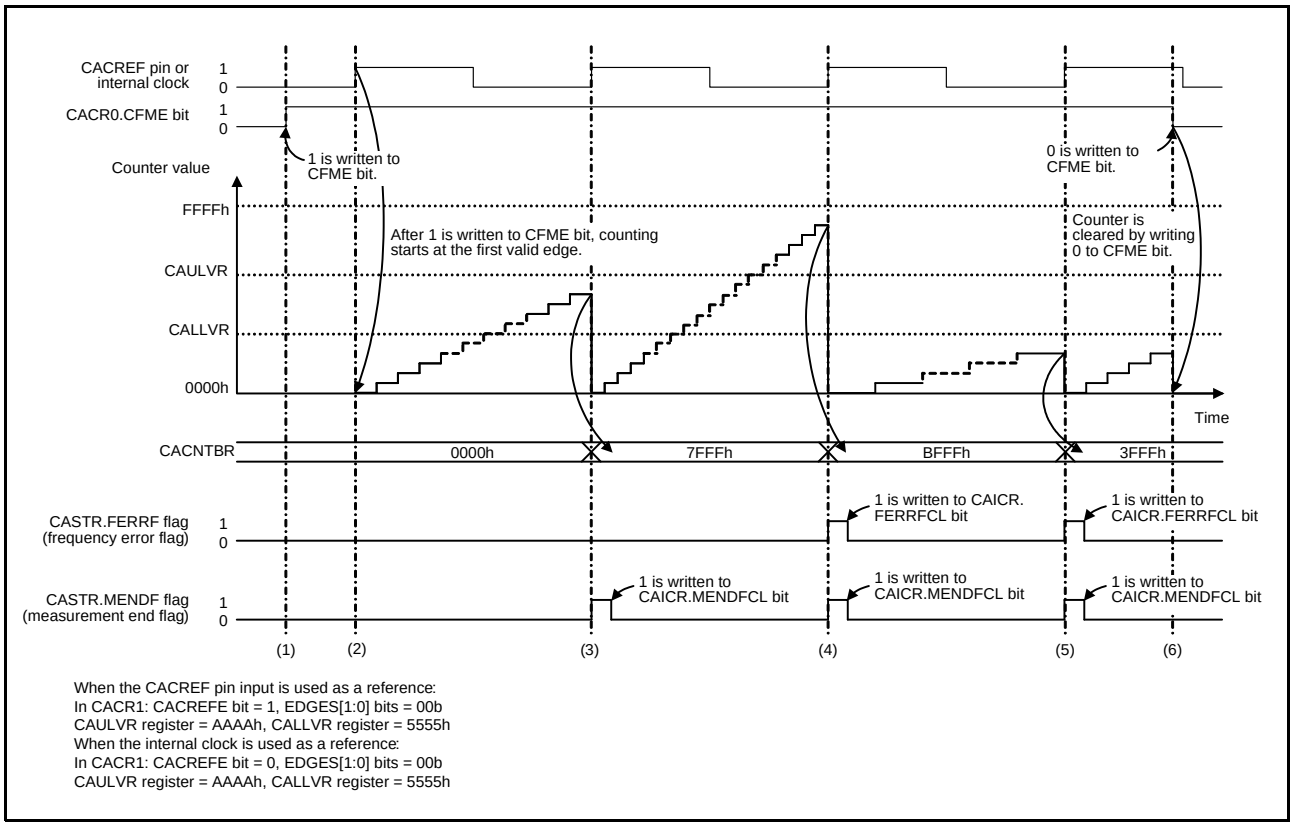


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(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, stopping modules, 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, 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
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

Note 1. For details, see section 9, Clock Generation Circuit.

Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Method of release other than reset	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after release*4	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (reset processing)
Main clock oscillator	Operating possible	Operating possible	Stopped	Stopped
Sub-clock oscillator	Operating possible	Operating possible	Operating possible*5	Operating possible*5
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*6	Operating possible*6	Operating possible*6	Stopped (Undefined)*6
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)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
Independent watchdog timer (IWDT)	Operating possible*6	Operating possible*6	Operating possible*6	Stopped (Undefined)*6
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible	Operating possible
Port output enable (POE)	Operating possible	Operating possible*7	Stopped (Retained)	Stopped (Undefined)
Remote control signal receiver (REMC)	Operating possible	Operating possible*8	Operating possible	Stopped (Undefined)
8-bit timer (unit 0, unit 1) (TMR)	Operating possible	Operating possible*9	Stopped (Retained)	Stopped (Undefined)
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible	Operating possible*10
Power-on reset circuit	Operating	Operating	Operating	Operating
Peripheral modules	Operating possible	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
I/O ports	Operating	Retained*11	Retained*12	Retained*12

“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.

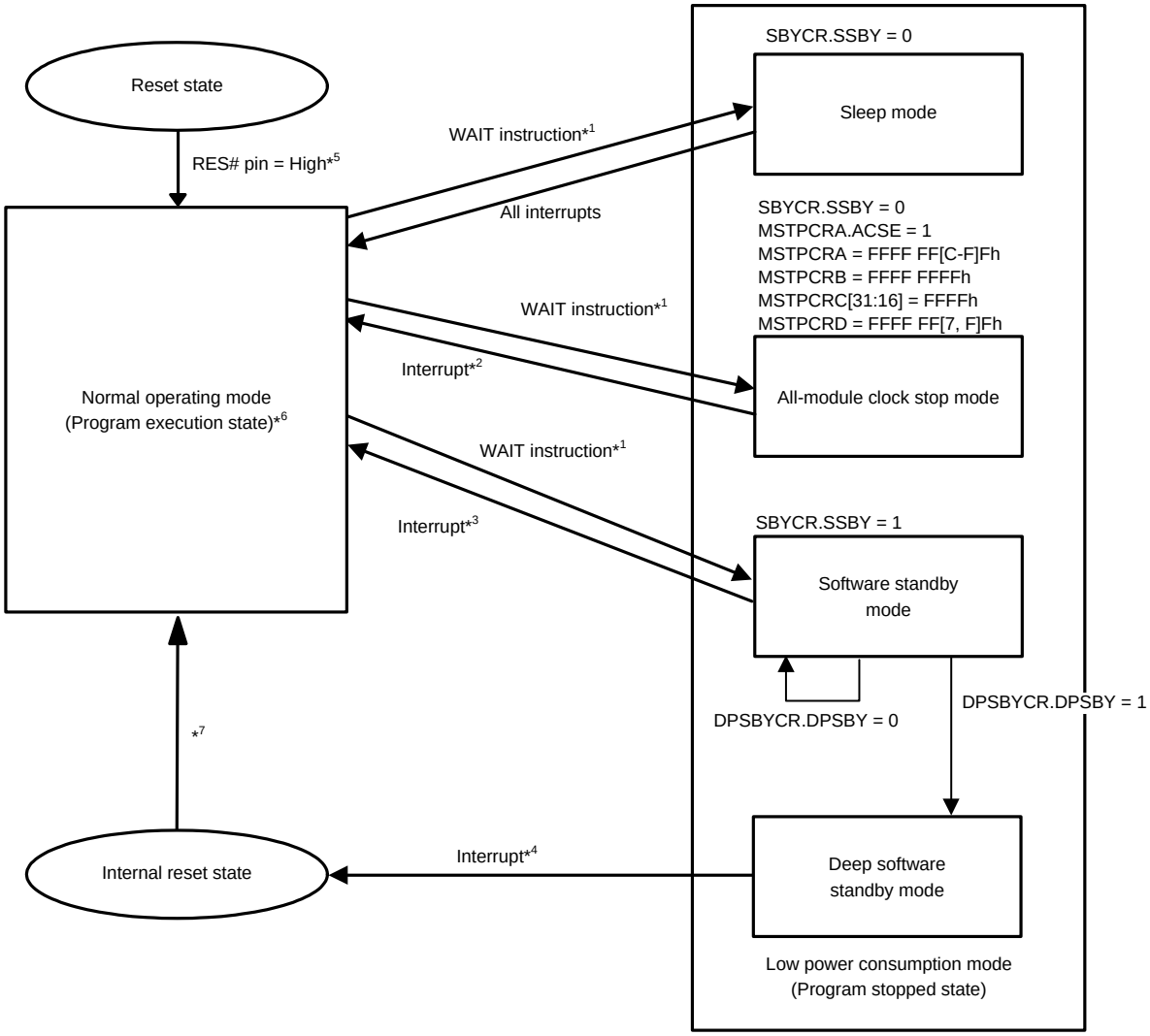
Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, REMC reception, voltage monitoring 1, voltage monitoring 2, and main-clock oscillation stop detection).
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, REMC reception, voltage monitoring 1, and voltage monitoring 2 interrupts).
Note 3. “Interrupts” here indicates a certain external pin interrupt source pin (the NMI or IRQ0-DS to IRQ15-DS) or any of peripheral interrupts (the RTC alarm, RTC periodic, 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 2) 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.
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. Whether to have the sub-clock oscillator operate or stop is selectable by the setting of the sub-clock oscillator forced oscillation bit in the sub-clock oscillator forced oscillation control register (SOFCR.SOFE).
Note 6. 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 7. 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 8. If PCLKB is selected as the operating clock for the REMC, whether to have the REMC operate or stop is selectable by the setting of the remote control signal receiver module stop bit in module stop control register D (MSTPCRD.MSTPD7).
Note 9. 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 10. 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 11. If pin P53 is being used for the BCLK signal, operation continues with as-is output of BCLK. While the 8-bit timer is operated, the related pins continue operation.

Note 12. Retention of levels or placement in the high-impedance state is selectable for the address bus and bus control signals (CS0# to CS3#, RD#, WR0#, WR1#, WR#, BC0#, BC1#, and ALE) by the output port enable bit (OPE) in the standby control register (SBYCR).



Note 1. When an interrupt that acts as a trigger for release is received during the transition to the program-stopped state after the execution of a WAIT instruction, the transition to the program stopped state does not proceed and interrupt exception processing is executed instead.

Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, REMC reception, voltage-monitoring 1, voltage-monitoring 2, and oscillator-stopped detection interrupts). However, the 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.
The REMC reception interrupt is only effective when the value of the remote control signal receiver module stop bit in module stop control register D (MSTPCRD.MSTPD7) is 0 if PCLKB is selected as the operating clock for the REMC.

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, REMC reception, voltage-monitoring 1, and voltage-monitoring 2 interrupts).

Note 4. “Interrupts” here indicates a certain external pin interrupt source pin (the NMI or IRQ0-DS to IRQ15-DS) or any of peripheral interrupts (the RTC alarm, RTC periodic, 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 2) is set to 1.

Note 5. The LOCO is the source of the operating clock following a transition from the reset state to normal operating mode.

Note 6. The transition to normal operating mode is made due to an interrupt from sleep mode, all-module clock-stop mode, or software standby mode. In the case of recovery from sleep mode, the clock source for subsequent use is selectable. For details, refer to section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). Also in the case of recovery from 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 CS3#, RD#, WR0#, WR1#, WR#, BC0#, BC1#, and ALE) in software standby mode or deep software standby mode, or to set the output to the high-impedance state.

SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.

When the SSBY bit is set to 1, the 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	—	—	—	—	MSTPA19	—	MSTPA17	—
Value after reset:	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA15	MSTPA14	—	—	—	—	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: MTU3 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b13 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	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	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPA17	12-bit A/D Converter (Unit 0) Module Stop	Target module: S12AD unit 0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	12-bit D/A Converter Module Stop	Target module: 12-bit D/A 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b23 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b24	MSTPA24	Module Stop A24	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b26, b25	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPA27	Module Stop A27	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W

Bit	Symbol	Bit Name	Description	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	Module Stop A29	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
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, see section 11.5.2, All-Module Clock Stop Mode.

Whether to stop the 8-bit timers or not can be selected by the MSTPA5 and MSTPA4 bits.

If PCLKB is selected as the operating clock for the REMC, whether to have the REMC operate or stop is selectable by the setting of the MSTPCRD.MSTPD7 bit.

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 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 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	—	—	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSTPB10	MSTPB9	—	—	MSTPB6	—	MSTPB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface 12 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
b8, b7	—	Reserved	These bits are read as 1. The write value should be 1.	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
b10	MSTPB10	Comparator C Module Stop	Target module: CMPC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b16 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b20 to b18	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: 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

Bit	Symbol	Bit Name	Description	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

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	—	—	—	—	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 0001 FFFFh) 0: RAM operating 1: RAM stopped	R/W
b15 to b1	—	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	CAC Module Stop*2	Target module: CAC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b23 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b24	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 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	MSTPD10	—	—	MSTPD7	—	—	—	MSTPD3	MSTPD2	—	—
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
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	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
b6 to b4	—	Reserved	These bits are 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
b9 to b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b10	MSTPD10	CANFD Module Stop*1	Target module: CANFD 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPD10 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 CANFD clock (CANFDCLK) and CANFD main clock (CANFDMCLK), and then execute a WAIT instruction.

11.2.6 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTCKEN	—	—	—	—	RSTCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RSTCKSEL[2:0]	Sleep Mode Return Clock Source Select	b2 b0 0 0 1: HOCO is selected 0 1 0: Main clock oscillator is selected Settings other than above are prohibited while the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching 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.

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.

11.2.7 Deep Standby Control Register (DPSBYCR)

Address(es): 0008 C280h

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: Release from deep software standby mode and cancellation of I/O port retention proceed simultaneously. 1: The I/O port state is retained even after release from deep software standby mode. Then, writing 0 to the IOKEEP bit cancels the I/O port retention.	R/W
b7	DPSBY	Deep Software Standby	SSBY b7 0 0: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 0 1: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 1 0: Transition to software standby mode is made after the WAIT instruction is executed 1 1: Transition to deep software standby mode is made after the WAIT instruction is executed	R/W

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

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 or IRQ0-DS to IRQ15-DS) or peripheral interrupts (the RTC alarm, RTC periodic, voltage monitoring 1, and voltage monitoring 2 interrupts). 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.

11.2.8 Deep Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): 0008 C282h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	0: Release from deep software standby mode by the IRQ0-DS pin is disabled 1: Release from deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Release from deep software standby mode by the IRQ1-DS pin is disabled 1: Release from deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Release from deep software standby mode by the IRQ2-DS pin is disabled 1: Release from deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Release from deep software standby mode by the IRQ3-DS pin is disabled 1: Release from deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Release from deep software standby mode by the IRQ4-DS pin is disabled 1: Release from deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Release from deep software standby mode by the IRQ5-DS pin is disabled 1: Release from deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Release from deep software standby mode by the IRQ6-DS pin is disabled 1: Release from deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Release from deep software standby mode by the IRQ7-DS pin is disabled 1: Release from deep software standby mode by the IRQ7-DS pin is enabled	R/W

Register DPSIER0 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be set to 0 before a transition to deep software standby mode.

11.2.9 Deep Standby Interrupt Enable Register 1 (DPSIER1)

Address(es): 0008 C283h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8E	IRQ8-DS Pin Enable	0: Release from deep software standby mode by the IRQ8-DS pin is disabled 1: Release from deep software standby mode by the IRQ8-DS pin is enabled	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	0: Release from deep software standby mode by the IRQ9-DS pin is disabled 1: Release from deep software standby mode by the IRQ9-DS pin is enabled	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	0: Release from deep software standby mode by the IRQ10-DS pin is disabled 1: Release from deep software standby mode by the IRQ10-DS pin is enabled	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	0: Release from deep software standby mode by the IRQ11-DS pin is disabled 1: Release from deep software standby mode by the IRQ11-DS pin is enabled	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	0: Release from deep software standby mode by the IRQ12-DS pin is disabled 1: Release from deep software standby mode by the IRQ12-DS pin is enabled	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	0: Release from deep software standby mode by the IRQ13-DS pin is disabled 1: Release from deep software standby mode by the IRQ13-DS pin is enabled	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	0: Release from deep software standby mode by the IRQ14-DS pin is disabled 1: Release from deep software standby mode by the IRQ14-DS pin is enabled	R/W
b7	DIRQ15E	IRQ15-DS Pin Enable	0: Release from deep software standby mode by the IRQ15-DS pin is disabled 1: Release from deep software standby mode by the IRQ15-DS pin is enabled	R/W

Register DPSIER1 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be set to 0 before a transition to deep software standby mode.

11.2.10 Deep Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): 0008 C284h

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

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

Register DPSIER2 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.
After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be set to 0 before a transition to deep software standby mode.

11.2.11 Deep Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): 0008 C286h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ0-DS pin 1: Request for release is being generated on the IRQ0-DS pin	R(/W) *1
b1	DIRQ1F	IRQ1-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ1-DS pin 1: Request for release is being generated on the IRQ1-DS pin	R(/W) *1
b2	DIRQ2F	IRQ2-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ2-DS pin 1: Request for release is being generated on the IRQ2-DS pin	R(/W) *1
b3	DIRQ3F	IRQ3-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ3-DS pin 1: Request for release is being generated on the IRQ3-DS pin	R(/W) *1
b4	DIRQ4F	IRQ4-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ4-DS pin 1: Request for release is being generated on the IRQ4-DS pin	R(/W) *1
b5	DIRQ5F	IRQ5-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ5-DS pin 1: Request for release is being generated on the IRQ5-DS pin	R(/W) *1
b6	DIRQ6F	IRQ6-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ6-DS pin 1: Request for release is being generated on the IRQ6-DS pin	R(/W) *1
b7	DIRQ7F	IRQ7-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ7-DS pin 1: Request for release is being generated on the IRQ7-DS pin	R(/W) *1

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

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

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

These flags indicate that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR0 register

[Clearing condition]

- 0 is written to these flags after confirming these flags are 1.

11.2.12 Deep Standby Interrupt Flag Register 1 (DPSIFR1)

Address(es): 0008 C287h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8F	IRQ8-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ8-DS pin 1: Request for release is being generated on the IRQ8-DS pin	R(/W) *1
b1	DIRQ9F	IRQ9-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ9-DS pin 1: Request for release is being generated on the IRQ9-DS pin	R(/W) *1
b2	DIRQ10F	IRQ10-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ10-DS pin 1: Request for release is being generated on the IRQ10-DS pin	R(/W) *1
b3	DIRQ11F	IRQ11-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ11-DS pin 1: Request for release is being generated on the IRQ11-DS pin	R(/W) *1
b4	DIRQ12F	IRQ12-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ12-DS pin 1: Request for release is being generated on the IRQ12-DS pin	R(/W) *1
b5	DIRQ13F	IRQ13-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ13-DS pin 1: Request for release is being generated on the IRQ13-DS pin	R(/W) *1
b6	DIRQ14F	IRQ14-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ14-DS pin 1: Request for release is being generated on the IRQ14-DS pin	R(/W) *1
b7	DIRQ15F	IRQ15-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ15-DS pin 1: Request for release is being generated on the IRQ15-DS pin	R(/W) *1

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

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

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

These flags indicate that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR1 register

[Clearing condition]

- 0 is written to these flags after confirming these flags are 1.

11.2.13 Deep Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): 0008 C288h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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
b7 to b5	—	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 DPSIEGR2 register is generated.
Each flag may be set to 1 when a request for release is generated in any mode (not even deep software standby mode) or when the setting of the DPSIER2 register is modified. Therefore, a transition to deep software standby mode should be made after the DPSIFR2 register is set to 00h.
To set the DPSIFR2 register to 00h after modifying the DPSIER2 register, wait for at least six PCLKB cycles, read the DPSIFR2 register, and then write 0 to the DPSIFR2 register. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER2 register.
The DPSIFR2 register is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

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

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

- A request for release is generated by the voltage monitoring m signal that is selected in the DPSIEGR2 register

[Clearing condition]

- 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.

11.2.14 Deep Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): 0008 C28Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Register DPSIEGR0 is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.15 Deep Standby Interrupt Edge Register 1 (DPSIEGR1)

Address(es): 0008 C28Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8EG	IRQ8-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ9EG	IRQ9-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ10EG	IRQ10-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ11EG	IRQ11-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ12EG	IRQ12-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ13EG	IRQ13-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ14EG	IRQ14-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ15EG	IRQ15-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

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

11.2.16 Deep Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): 0008 C28Ch

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

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

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

Address(es): 0008 C2A0h to 0008 C2BFh

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

x: Undefined

Register DPSBKRy is a 32-byte readable/writable register to store data during deep software standby mode. The value of this register is retained even in deep software standby mode where RAM data is not retained. DPSBKRy is not initialized, and the register value is undefined immediately after power-on.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, code flash memory, and RAM operate on the operating clock specified by the ICK[3:0] bits.
Peripheral modules operate on the operating clock specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits.
The data flash memory operates on the operating clock specified by the FCK[3:0] bits.
The external bus operates on the operating clock specified by the BCK[3:0] bits. For details, see section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to D, i = 31 to 0) in registers MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module is released from the module-stop state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.
After release from a reset, all modules except for the DMAC, DTC, and 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 Low Power Consumption Modes

11.5.1 Sleep Mode

11.5.1.1 Transition to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
When the WDT is used, the WDT stops counting when sleep mode is entered.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from sleep mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for that interrupt to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, see section 2, CPU.
Note 2. For details, see section 14.7.3, Selecting Interrupt Request Destination.
Note 3. For details, see section 14, Interrupt Controller (ICUF).

11.5.1.2 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, see section 14, Interrupt Controller (ICUF).
Note 2. For details, see section 2, CPU.

11.5.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used on return from sleep mode, the clock used after return needs to be set by the sleep mode return clock source switching register (RSTCKCR) and the wait control register needs to be set for each clock source. When the return interrupt is generated, after oscillation settling of the oscillator specified as the return clock, the clock source is automatically switched, and then operation returns from sleep mode. At this time, the registers related to clock source switching are automatically rewritten.
For details, see section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). For setting a waiting time for oscillation stabilization, see section 9.2.18, Main Clock Oscillator Wait Control Register (MOSCWTCR).

11.5.2 All-Module Clock Stop Mode

11.5.2.1 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 FF[7, F]Fh), 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, REMC*3, power-on reset circuit, voltage detection circuit at the end of the current bus cycle, and the chip enters all-module clock stop mode*4.

When the WDT is used, the WDT stops counting when all-module clock stop mode is entered.

Counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

To use all-module clock-stop mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*5 of the CPU to 0.
(2) Set the interrupt request destination*6 to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*7 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*5 of the CPU.
(4) Set the IERm.IENj bit*7 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*5 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. If PCLKB is selected as the operating clock for the REMC, whether to have the REMC operate or stop is selectable by the setting of the MSTPCRD.MSTPD7 bit.
Note 4. 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 5. For details, see section 2, CPU.
Note 6. For details, see section 14.7.3, Selecting Interrupt Request Destination.
Note 7. For details, see section 14, Interrupt Controller (ICUF).

11.5.2.2 Release from All-Module Clock Stop Mode

Release from all-module clock-stop mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), a peripheral interrupt (8-bit timer*1, RTC alarm, RTC periodic, IWDT*2, REMC reception*3, voltage monitoring 1, voltage monitoring 2, or oscillator-stopped detection interrupt), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset, and the transition to the normal program execution state proceeds via exception processing for the given interrupt or reset.
However, note that in cases where a maskable interrupt has been masked by the CPU (priority level*4 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*5 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. If PCLKB is selected as the operating clock for the REMC, whether to have the REMC operate or stop is selectable by the setting of the MSTPCRD.MSTPD7 bit.
Note 4. For details, see section 14, Interrupt Controller (ICUF).
Note 5. For details, see section 2, CPU.

11.5.3 Software Standby Mode

11.5.3.1 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, whether to have the sub-clock oscillator operate or stop is selectable. For details, see 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).

Do not allow the chip to enter software standby mode while the FLL function of the HOCO is enabled (FLLCR1.FLLEN = 1).

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, see section 2, CPU.
Note 2. For details, see section 14.7.3, Selecting Interrupt Request Destination.
Note 3. For details, see section 14, Interrupt Controller (ICUF).

11.5.3.2 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, REMC reception, 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 condition, but the return from software standby still follows the period for stabilization of oscillation.

- SOFCR.SOFE = 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, REMC reception, voltage monitoring 1, or voltage monitoring 2 interrupt is generated, each oscillator which was operating before a transition to software standby mode resumes oscillation. After the time for return from software standby has elapsed, the chip is released from software standby and starts interrupt exception processing.
The time for return after release from software standby is the oscillation stabilization waiting time plus the time required for operations by the software standby release sequencer.

tSBYi = tSBYOSCWT + tSBYSEQ

tSBYi (i = MC, EX, PC, PE, PH, SC, HO, LO): Time for return after release from software standby
tSBYOSCWT: Oscillation stabilization waiting time
tSBYSEQ: Time required for operations by the software standby release sequencer

For the oscillation stabilization waiting time to be used in calculating the time for return after release from software standby, use the greatest value of the oscillation stabilization waiting time of the oscillators which are to be started. For the oscillation stabilization waiting times of the oscillators, see section 45, 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.

11.5.3.3 Example of Software Standby Mode Application

Figure 11.2 shows an example where a transition to software standby mode is made at the falling edge of the IRQn pin, and release from software standby mode is initiated at the rising edge of the IRQn pin.

In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus a transition to software standby mode is made. After that, release from software standby mode is initiated at the rising edge of the IRQn pin.

To return from software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, see section 14, Interrupt Controller (ICUF).

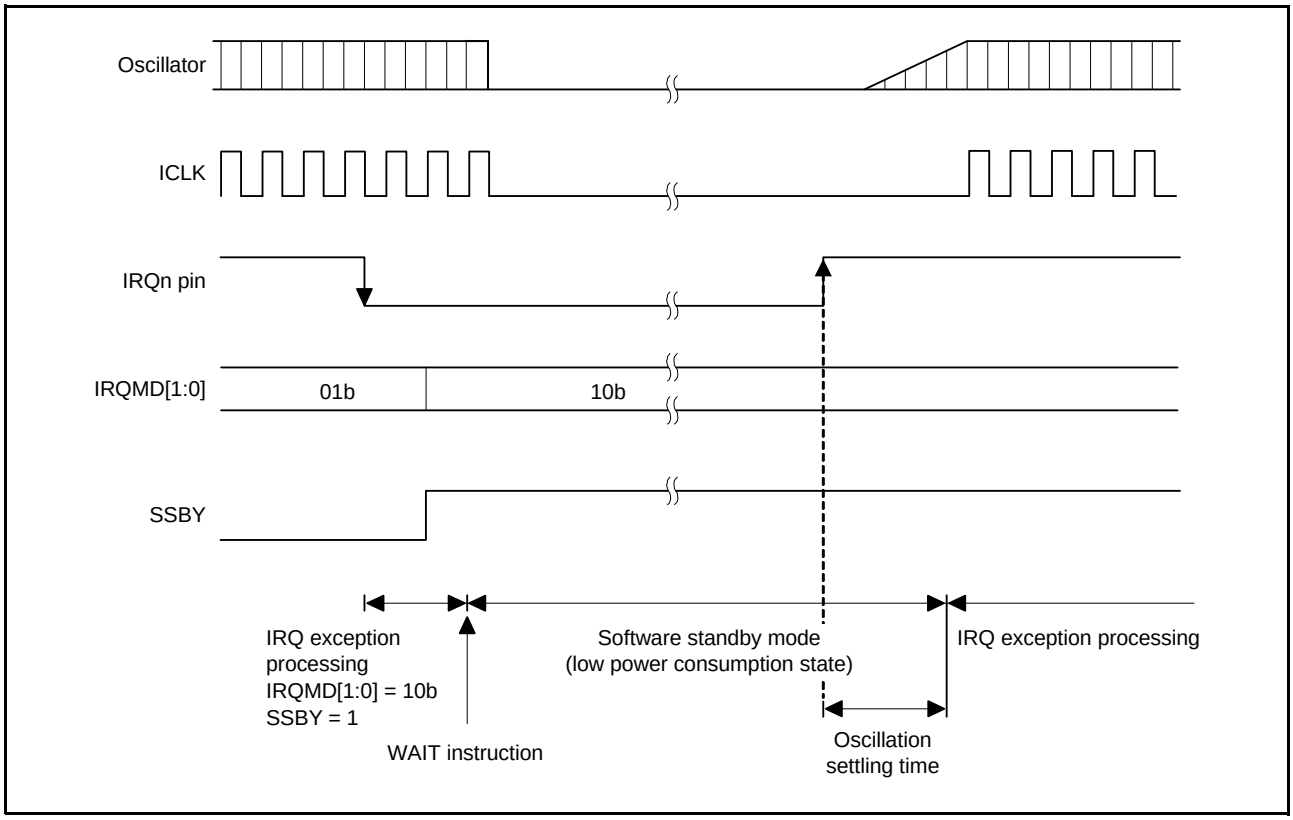


Figure 11.2 Example of Software Standby Mode Application

11.5.4 Deep Software Standby Mode

11.5.4.1 Transition to Deep Software Standby Mode

Executing a WAIT instruction while the SBYCR.SSBY bit is 1 causes a transition to software standby mode. After that, if the DPSBYCR.DPSBY bit is 1, the transition continues to deep software standby mode.*1
On deep software standby mode, the CPU, internal peripheral modules (except for the RTC alarm and RTC periodic), 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, whether to have the sub-clock oscillator operate or stop is selectable. For details, see Table 11.2, Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode. At this time, the contents of all the registers of the CPU and internal peripheral modules (except for the RTC alarm and RTC periodic) become undefined.
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 satisfied before the WAIT instruction is executed. For details, refer to section 11.5.3, Software Standby Mode.

11.5.4.2 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 or IRQ0-DS to IRQ15-DS), peripheral interrupts (the RTC alarm, RTC periodic, voltage monitoring 1 and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, or a voltage monitoring 0 reset.

(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 2) and DPSIFRn (n = 0 to 2). 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 2) registers. The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ15-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 45, 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.

11.5.4.3 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 clearing 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.

11.5.4.4 Example of Deep Software Standby Mode Application

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

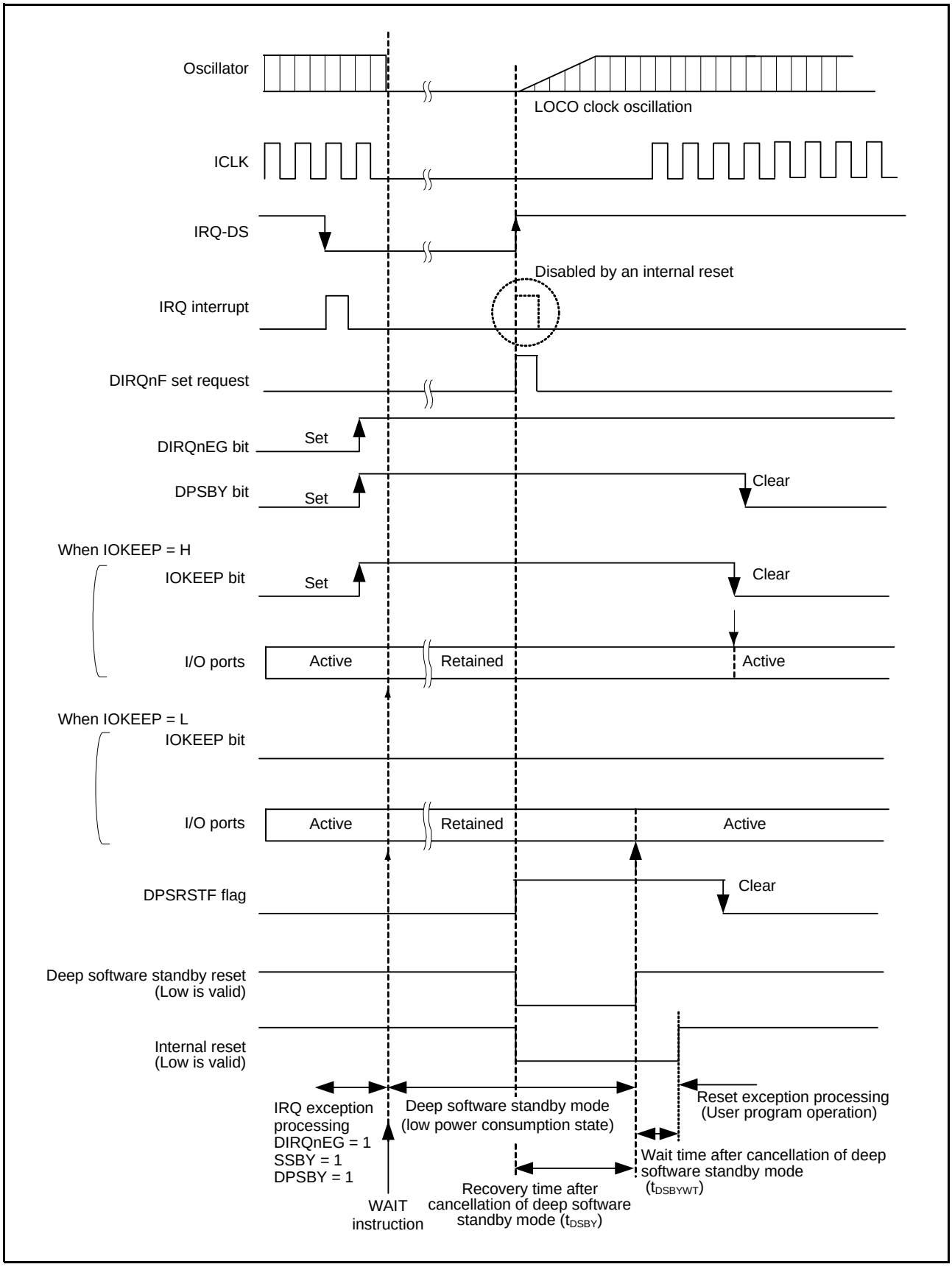


Figure 11.3 Example of Deep Software Standby Mode Application

11.5.4.5 Flowchart to Use Deep Software Standby Mode

Figure 11.4 shows an example of a flowchart to use deep software standby mode.
In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception 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.

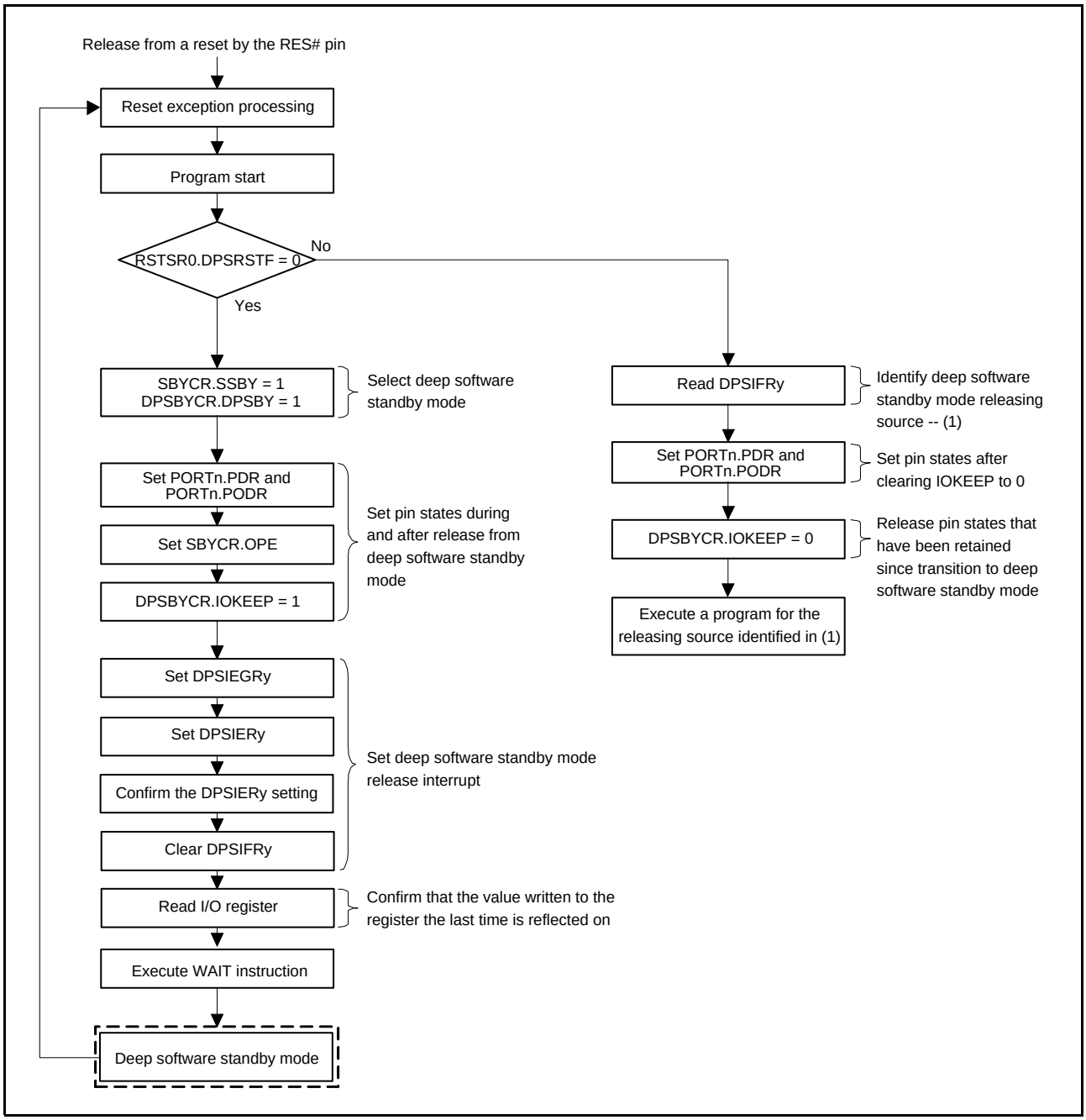


Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode

11.6 Usage Notes

11.6.1 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode.

11.6.2 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, see section 17, DMA Controller (DMACAa) and section 18, Data Transfer Controller (DTCb).

11.6.3 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.

11.6.4 Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD

Write accesses to registers MSTPCRA, MSTPCRB, MSTPCRC and MSTPCRD should be made only by the CPU.

11.6.5 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.

11.6.6 Timing of WAIT Instruction

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.

11.6.7 Rewriting the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMACA and DTC to rewrite any registers related to the WDT while the chip is in sleep mode.
According to the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. If that is the case, do not set up the DMAC and DTC to rewrite any registers related to the IWDT in sleep mode.
The RSTCKCR register can be set so that the clock source is switched on recovery from sleep mode. For this reason, rewriting the register while the chip is in sleep mode may lead to unintended operation, so do not allow rewriting of the RSTCKCR register in sleep mode.

12. 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 12.1 lists the association between the PRCR bits and the registers to be protected.

Table 12.1 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, FLLCR1, FLLCR2, OSTDCR, OSTDSR
PRC1	• Registers related to the operating modes: SYSCR0, SYSCR1, VOLSR • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, RSTCKCR, DPSBYCR, DPSIER0 to DPSIER2, DPSIFR0 to DPSIFR2, DPSIEGR0 to DPSIEGR2 • Registers related to clock generation circuit: MOSCWTCR, SOSCWTCR, MOFCR, SOFCR, HOCOPCR • Software reset register: SWRR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR

12.1 Register Descriptions

12.1.1 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, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note 1. Written values are not retained. These bits are read as 00h.

PRCi Bits (Protect Bit i) (i = 0, 1, 3)

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enables and disables writing to the corresponding registers to be protected, respectively.

13. Exception Handling

13.1 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 eight types of exceptions. The types of exception events are shown in Figure 13.1.
The occurrence of an exception causes the processor mode to shift to supervisor mode.

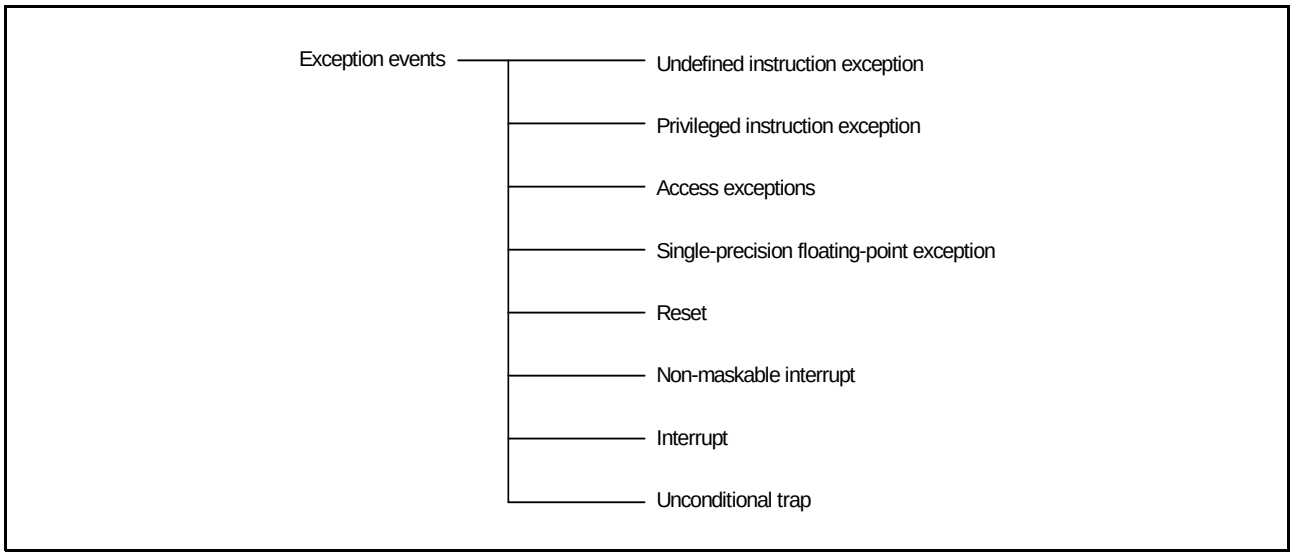


Figure 13.1 Types of Exception Events

13.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

13.1.2 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.

13.1.3 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.

13.1.4 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.

13.1.5 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

13.1.6 Non-Maskable Interrupt

The non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

13.1.7 Interrupt

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

13.1.8 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

13.2 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 13.2 shows the processing procedure when an exception other than a reset is accepted.

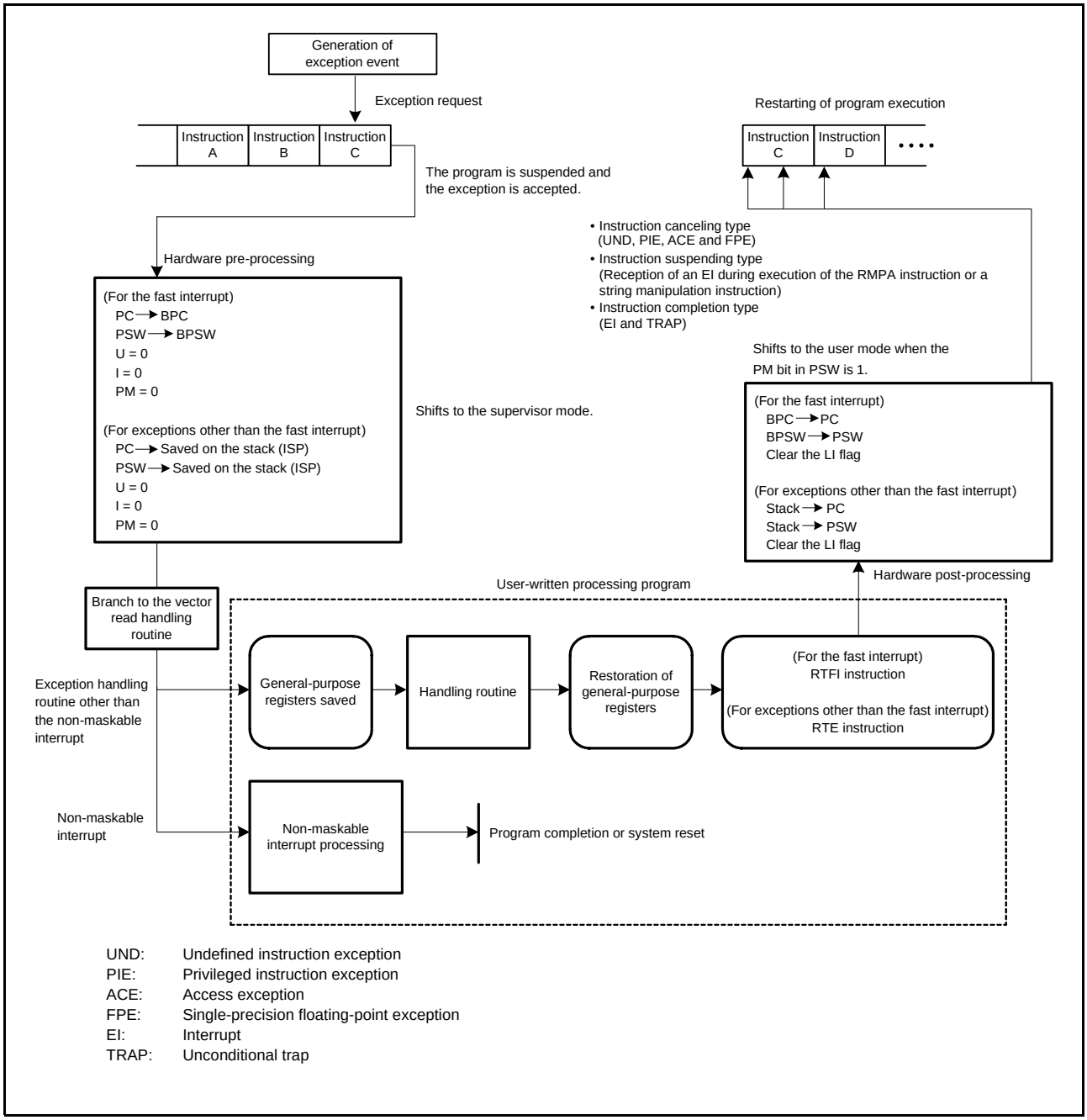


Figure 13.2 Outline of Exception Handling Procedure

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.

13.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

13.3.1 Acceptance Timing and Saved PC Value

Table 13.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 13.1 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
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

13.3.2 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 13.2. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Table.

Table 13.2 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
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

13.4 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

13.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

13.5.1 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.

13.5.2 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.

13.5.3 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.

13.5.4 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.

13.5.5 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

13.5.6 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from the value of EXTB + address 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.7 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.8 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.

13.6 Return from Exception Handling Routine

Executing the instruction listed in Table 13.3 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 13.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access 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

13.7 Priority of Exception Events

The priority of exception events is listed in Table 13.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 13.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Undefined instruction exception Privileged instruction exception
	6	Unconditional trap
	7	Operand access exception
Low	8	Single-precision floating-point exception

14. Interrupt Controller (ICUF)

14.1 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 14.1 lists the ICU specifications, and Figure 14.1 shows a block diagram of the interrupt controller.

Table 14.1 ICU Specifications (1/2)

Item		Description

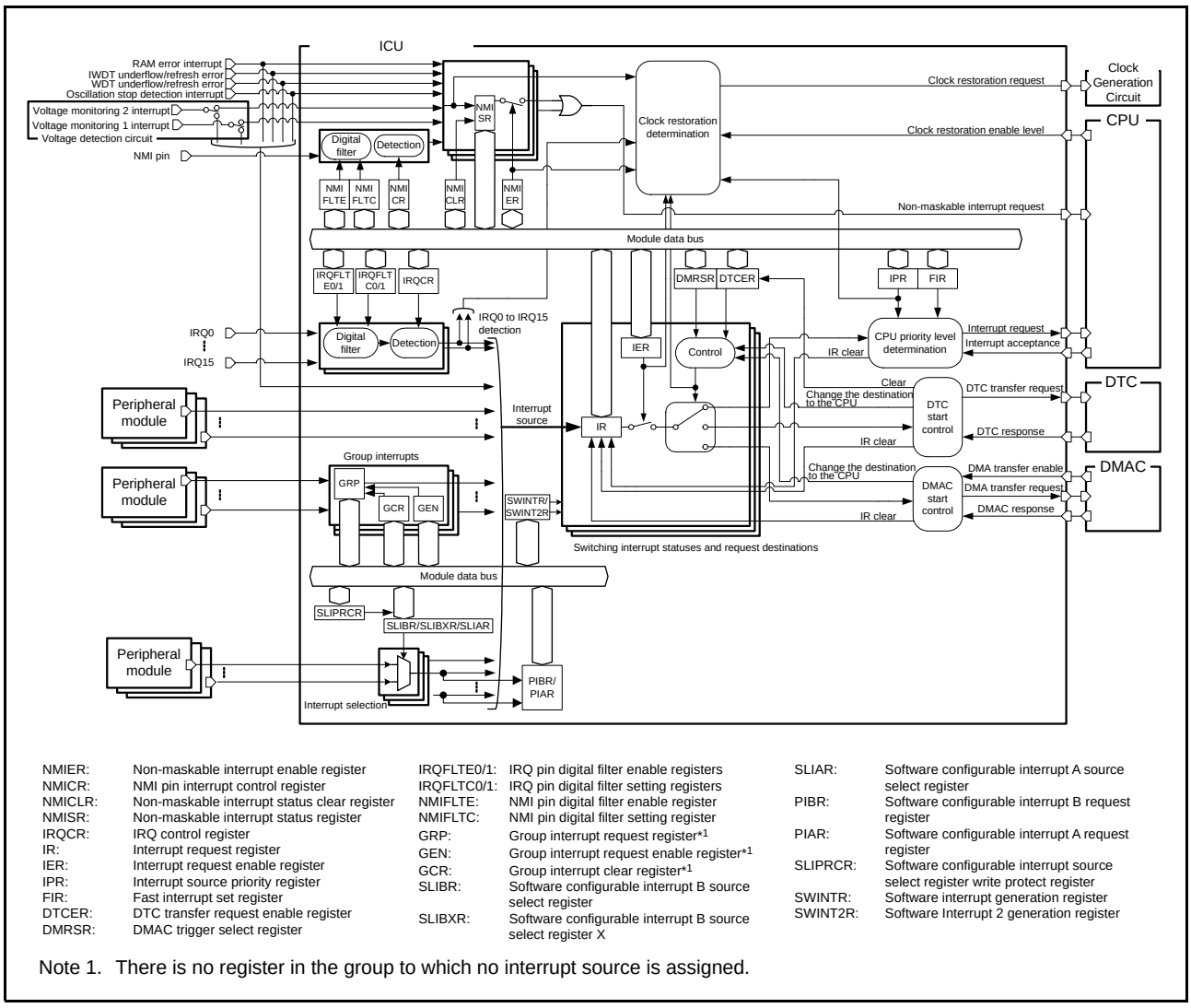
Interrupts	Peripheral interrupts	Interrupts from peripheral modules
		• Interrupt detection method: Edge detection/level detection (fixed for each interrupt source)
		• Group interrupt: Multiple interrupt sources are grouped together and treated as an interrupt source.*1
		  • Group IE0 interrupt: Interrupt 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
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.

Table 14.1 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, RTC alarm, RTC period, IWDT, 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, RTC alarm, RTC period, IWDT, 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, RTC alarm, RTC period).

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 14.5, 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 register value from the value after reset. To enable the voltage monitoring 1 interrupt, set the LVD1CR1.LVD1IRQSEL bit to 1, and to enable the voltage monitoring 2 interrupt, set the LVD2CR1.LVD2IRQSEL bit to 1.



NMIER:	Non-maskable interrupt enable register	IRQFLTE0/1:	IRQ pin digital filter enable registers	SLIAR:	Software configurable interrupt A source select register
NMICR:	NMI pin interrupt control register	IRQFLTC0/1:	IRQ pin digital filter setting registers	PIBR:	Software configurable interrupt B request register
NMICLR:	Non-maskable interrupt status clear register	NMIFLTE:	NMI pin digital filter enable register	PIAR:	Software configurable interrupt A request register
NMISR:	Non-maskable interrupt status register	NMIFLTC:	NMI pin digital filter setting register	SLIPRCR:	Software configurable interrupt source select register write protect register
IRQCR:	IRQ control register	GRP:	Group interrupt request register*1	SWINTR:	Software interrupt generation register
IR:	Interrupt request register	GEN:	Group interrupt request enable register*1	SWINT2R:	Software Interrupt 2 generation register
IER:	Interrupt request enable register	GCR:	Group interrupt clear register*1		
IPR:	Interrupt source priority register	SLIBR:	Software configurable interrupt B source select register		
FIR:	Fast interrupt set register	SLIBXR:	Software configurable interrupt B source select register X		
DTCER:	DTC transfer request enable register				
DMRSR:	DMAC trigger select register				

Note 1. There is no register in the group to which no interrupt source is assigned.

Figure 14.1 Block Diagram of the ICU

Table 14.2 lists I/O pins used for the ICU.

Table 14.2 ICU I/O Pins

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

14.2 Register Descriptions

14.2.1 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 14.5, 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 14.5, 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 (GRPBL0, GRPBL1, GRPBL2, GRPAL0). 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 14.4.4, 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 (GRPBL0, GRPBL1, GRPBL2, GRPAL0) is 1 (interrupt request is generated) while the ENj bit in the group interrupt request enable register (GENBL0, GENBL1, GENBL2, GENAL0) 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.

14.2.2 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 14.2.1, Interrupt Request Register n (IRn) (n = 016 to 255).

The IERm.IENj bit is provided for each interrupt vector number.

Refer to Table 14.5, 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 14.7.3.1, Interrupt Request Destination Setting Procedure for the procedure to set the IERm.IENj bit to select the interrupt request destination.

14.2.3 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 14.5, 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.

14.2.4 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 14.9, 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 14.5, 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 14.10.3, Exiting Software Standby Mode for details.

14.2.5 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.

14.2.6 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.

14.2.7 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 14.5, 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:

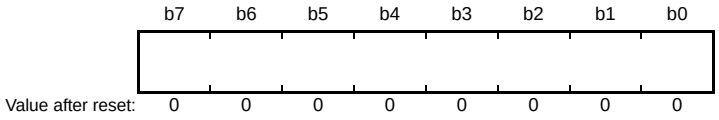
- 1 is written to the DTCE bit

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

14.2.8 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

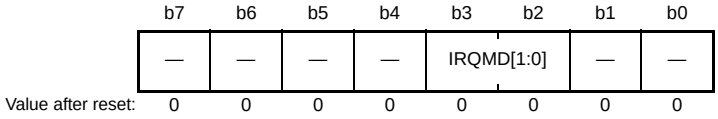


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	These bits set the interrupt vector number of the interrupt source as the DMAC trigger.	R/W

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 14.5, Interrupt Vector Table for interrupt vector numbers of interrupt sources.
Write the DMRSRm register while the DMACm.DMCNT.DTE bit is 0.

14.2.9 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): ICU.IRQCR0 0008 7500h to ICU.IRQCR15 0008 750Fh



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 14.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupts.

14.2.10 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 14.7.6, Digital Filter for details on the digital filter.

14.2.11 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 14.7.6, Digital Filter for details on the digital filter.

14.2.12 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 14.7.6, Digital Filter for details on the digital filter.

14.2.13 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 14.7.6, Digital Filter for details on the digital filter.

14.2.14 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RAMST	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	RAMST	RAM Error Interrupt Status Flag	0: RAM error interrupt is not requested. 1: RAM error interrupt is requested.	R
b7	—	Reserved	This bit is read as 0 and cannot be modified.	R

The NMISR register indicates whether a non-maskable interrupt request has occurred.
Each flag in the NMISR register is not affected by the setting of the corresponding bit in the NMIER register.
In the non-maskable interrupt handler, read the NMISR register to check if the other non-maskable interrupt has occurred. Confirm that all the status flags are 0 before exiting the interrupt handler.

NMIST Flag (NMI Status Flag)

The NMIST flag indicates whether an NMI pin interrupt request has been generated.
This flag is read-only. To set the NMIST flag to 0, set the NMICLR.NMICLR bit to 1.
This flag becomes 1 under the following condition:

- An edge set in the NMICR.NMIMD bit is input to the NMI pin

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates whether an oscillation stop detection interrupt request has been generated.
The OSTST flag is read-only. To set the OSTST flag to 0, set the NMICLR.OSTCLR bit to 1.
This flag becomes 1 under the following condition:

- An oscillation stop detection interrupt occurs

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates whether an WDT underflow/refresh error interrupt request is generated.
The WDTST flag is read-only. To set the WDTST flag to 0, set the NMICLR.WDTCLR bit to 1.
This flag becomes 1 under the following condition:

- A WDT underflow/refresh error interrupt occurs while the WDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates whether an IWDT underflow/refresh error interrupt request is generated.
The IWDTST flag is read-only. To set the IWDTST flag to 0, set the NMICLR.IWDTCLR bit to 1.
This flag becomes 1 under the following condition:

- An IWDT underflow/refresh error interrupt occurs while the IWDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

The LVD1ST flag indicates whether a voltage monitoring 1 interrupt request is generated.
The LVD1ST flag is read-only. To set the LVD1ST flag to 0, set the NMICLR.LVD1CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 1 interrupt occurs while the LVD1CR1.LVD1IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

The LVD2ST flag indicates whether a voltage monitoring 2 interrupt request is generated.
The LVD2ST flag is read-only. To set the LVD2ST flag to 0, set the NMICLR.LVD2CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 2 interrupt occurs while the LVD2CR1.LVD2IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD2CLR bit

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 error status flags of the RAM. Refer to section 43.3.2, 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.

14.2.15 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RAMEN	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	RAMEN	RAM Error Interrupt Enable	0: RAM error interrupt is disabled. 1: RAM error interrupt is enabled.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The NMIER register enables and disables generation of non-maskable interrupt requests. When each bit is 1, the corresponding interrupt source is used as a non-maskable interrupt.

NMIEN Bit (NMI Pin Interrupt Enable)

The NMIEN bit enables and disables using the NMI pin interrupt.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables and disables generation of a non-maskable interrupt by the oscillation stop detection interrupt. When the oscillation stop detection interrupt is used as a maskable interrupt, leave this bit set to 0.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

The WDTEN bit enables and disables generation of a non-maskable interrupt by the WDT underflow/refresh error interrupt.

When the WDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

The IWDTEN bit enables and disables generation of a non-maskable interrupt by the IWDT underflow/refresh error interrupt.

When the IWDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

The LVD1EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 1 interrupt.

When the voltage monitoring 1 interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

The LVD2EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 2 interrupt. When the voltage monitoring 2 interrupt is used as a maskable interrupt, leave this bit set to 0.

RAMEN Bit (RAM Error Interrupt Enable)

The RAMEN bit enables and disables generation of a non-maskable interrupt by the RAM error interrupt. When the RAM error interrupt is used as a maskable interrupt, leave this bit set to 0.

14.2.16 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.

14.2.17 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.

14.2.18 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 14.7.6, Digital Filter for details on the digital filter.

14.2.19 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 14.7.6, Digital Filter for details on the digital filter.

14.2.20 Group BL0/BL1/BL2 Interrupt Request Register (GRPBL0/GRPBL1/GRPBL2), Group AL0 Interrupt Request Register (GRPAL0)

Address(es): ICU.GRPBL0 0008 7630h, ICU.GRPBL1 0008 7634h, ICU.GRPBL2 0008 7638h, ICU.GRPAL0 0008 7830h

	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.

Registers GRPBL0, GRPBL1, and GRPBL2 contain the statuses of interrupt sources that are detected by level detection

and use PCLKB as the operating clock.
The GRPAL0 register contains 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 14.4.4, 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 BL0/BL1/BL2

This flag becomes 1 under the following condition:

- The GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENBL0/GENBL1/GENBL2.ENj bit is 1.

This flag becomes 0 under any of the following conditions:

- The GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 0 when the GENBL0/GENBL1/GENBL2.ENj bit is set to 0.

(2) Group AL0

This flag becomes 1 under the following condition:

- The GRPAL0.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENAL0.ENj bit is 1.

This flag becomes 0 under any of the following conditions:

- The GRPAL0.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPAL0.ISj flag becomes 0 when the GENAL0.ENj bit is set to 0.

14.2.21 Group BL0/BL1/BL2 Interrupt Request Enable Register (GENBL0/GENBL1/GENBL2), Group AL0 Interrupt Request Enable Register (GENAL0)

Address(es): ICU.GENBL0 0008 7670h, ICU.GENBL1 0008 7674h, ICU.GENBL2 0008 7678h, ICU.GENAL0 0008 7870h

	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 GENBL0/GENBL1/GENBL2, and GENAL0 control the ISj flag in registers GRPBL0/GRPBL1/GRPBL2, and GRPAL0, respectively.
Refer to section 14.4.4, 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 BL0/BL1/BL2

When a peripheral module interrupt request is generated while the corresponding GENBL0/GENBL1/GENBL2.ENj bit is 1, the GRPBL0/GRPBL1/GRPBL2.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.

(2) Group AL0

When a peripheral module interrupt request is generated while the corresponding GENAL0.ENj bit is 1, the GRPAL0.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.

14.2.22 Software Configurable Interrupt B Request Register k (PIBRk) (k = 0h, 1h, 5h, 6h, 8h to Ah, Ch, Dh)

Address(es): ICU.PIBR0 0008 7700h, ICU.PIBR1 0008 7701h, ICU.PIBR5 0008 7705h, ICU.PIBR6 0008 7706h, ICU.PIBR8 0008 7708h, ICU.PIBR9 0008 7709h, ICU.PIBRA 0008 770Ah, ICU.PIBRC 0008 770Ch, ICU.PIBRD 0008 770Dh

	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 14.3, 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

14.2.23 Software Configurable Interrupt A Request Register k (PIARk) (k = 0h to 5h, Bh, Ch)

Address(es): ICU.PIAR0 0008 7900h to ICU.PIAR5 0008 7905h, ICU.PIARB 0008 790Bh, ICU.PIARC 0008 790Ch

	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 14.4, 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:

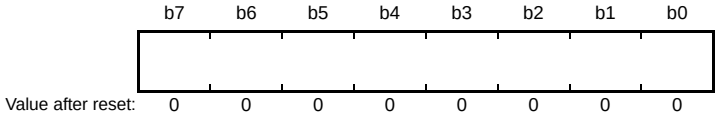
- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIARk.PIRj flag

14.2.24 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 14.3, 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.

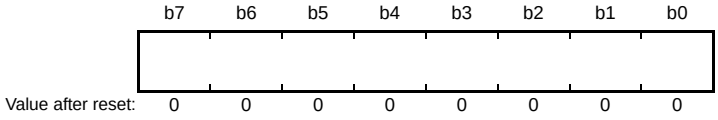
Do not assign the same interrupt source to multiple registers from registers SLIBXRn and SLIBRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 14.3, Interrupt Sources for Software Configurable Interrupt B for details on which sources can be used to start the DTC or DMAC.

Refer to section 14.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

14.2.25 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 14.3, 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 14.3, Interrupt Sources for Software Configurable Interrupt B for details on which sources can be used to start the DTC or DMAC.

Refer to section 14.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 14.3 Interrupt Sources for Software Configurable Interrupt B (1/2)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	—	None	No interrupt selected (initial value)	N/A	N/A	PIBR0.PIR0
1	Edge	CMT2	CMI2 (CMCOR compare match)	✓	✓	PIBR0.PIR1
2		CMT3	CMI3 (CMCOR compare match)	✓	✓	PIBR0.PIR2
3		TMR0	CMIA0 (TCORA compare match)	✓	N/A	PIBR0.PIR3
4			CMIB0 (TCORB compare match)	✓	N/A	PIBR0.PIR4
5			OVI0 (TCNT overflow)	N/A	N/A	PIBR0.PIR5
6		TMR1	CMIA1 (TCORA compare match)	✓	N/A	PIBR0.PIR6
7			CMIB1 (TCORB compare match)	✓	N/A	PIBR0.PIR7
8			OVI1 (TCNT overflow)	N/A	N/A	PIBR1.PIR0
9		TMR2	CMIA2 (TCORA compare match)	✓	N/A	PIBR1.PIR1
10			CMIB2 (TCORB compare match)	✓	N/A	PIBR1.PIR2
11			OVI2 (TCNT overflow)	N/A	N/A	PIBR1.PIR3
12		TMR3	CMIA3 (TCORA compare match)	✓	N/A	PIBR1.PIR4
13			CMIB3 (TCORB compare match)	✓	N/A	PIBR1.PIR5
14			OVI3 (TCNT overflow)	N/A	N/A	PIBR1.PIR6
15 to 40		Reserved	—	N/A	N/A	—
41		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 to 63		Reserved	—	N/A	N/A	—
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 to 78		Reserved	—	N/A	N/A	—
79		ELC	ELSR18I (ELC interrupt)	✓	✓	PIBR9.PIR7
80			ELSR19I (ELC interrupt)	✓	✓	PIBRA.PIR0
81 to 95		Reserved	—	N/A	N/A	—
96		CMPC0	CMPC0	✓	✓	PIBRC.PIR0
97		CMPC1	CMPC1	✓	✓	PIBRC.PIR1
98		CMPC2	CMPC2	✓	✓	PIBRC.PIR2
99		CMPC3	CMPC3	✓	✓	PIBRC.PIR3
100 to 103		Reserved	—	N/A	N/A	—
104		CANFD	RFDREQ0 (receive FIFO 0 DTC/DMA transfer request)	✓	✓	PIBRD.PIR0
105			RFDREQ1 (receive FIFO 1 DTC/DMA transfer request)	✓	✓	PIBRD.PIR1
106		CANFD0	CFDREQ0 (common FIFO 0 DTC/DMA transfer request)	✓	✓	PIBRD.PIR2
107 to 254	—	Reserved	—	N/A	N/A	—

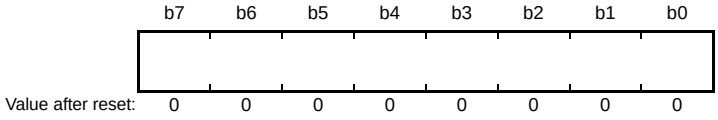
Table 14.3 Interrupt Sources for Software Configurable Interrupt B (2/2)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
255	—	Reserved/ none *1	—/No interrupt selected	N/A	N/A	—

Note 1. “Reserved” for the SLIBXRn register, and “none” for the SLIBRn register.

14.2.26 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 14.4, 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 14.4, Interrupt Sources for Software Configurable Interrupt A for details on which sources can be used to start the DTC or DMAC.

Refer to section 14.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 14.4 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		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 14.4 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 90		Reserved	—	N/A	N/A	—
91		RSPI0	SPCI0 (communication end)	N/A	N/A	PIARB.PIR3
92		Reserved	—	N/A	N/A	PIARB.PIR4
93		Reserved	—	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		CANFD	EC1EI (1-bit ECC error)	N/A	N/A	PIARC.PIR0
97			EC2EI (2-bit ECC error)	N/A	N/A	PIARC.PIR1
98			ECOVI (ECC overflow)	N/A	N/A	PIARC.PIR2
99 to 254		Reserved	—	N/A	N/A	—
255	—	None	No interrupt selected	N/A	N/A	—

14.2.27 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, and SLIARn 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 14.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

14.3 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.

14.3.1 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 14.5 lists details of the interrupt vectors. Details of the headings in Table 14.5 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 14.5 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
—	Reserved	34	0088h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	36	0090h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	37	0094h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RSPI0	SPRI0	38	0098h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN6	IPR038	DTCER038
	SPTI0	39	009Ch	Edge	✓	✓	✓	N/A	N/A	IER04.IEN7	IPR039	DTCER039
—	Reserved	40	00A0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	41	00A4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
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 14.5 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
—	Reserved	44	00B0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	45	00B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	46	00B8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	47	00BCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	48	00C0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	49	00C4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	50	00C8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	51	00CCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
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 14.5 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	—
—	Reserved	90	0168h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	91	016Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RTC	ALM	92	0170h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN5	IPR093	—
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	—
	Reserved	108	01B0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	Reserved	109	01B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	GROUPBL0	110	01B8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN6	IPR110	—
	GROUPBL1	111	01BCh	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN7	IPR111	—
	GROUPAL0	112	01C0h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN0	IPR112	—
	GROUPAL1	113	01C4h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN1	IPR113	—
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
—	Reserved	118	01D8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	119	01DCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
DMAC	DMAC0I	120	01E0h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN0	IPR120	DTCER120
	DMAC1I	121	01E4h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN1	IPR121	DTCER121
	DMAC2I	122	01E8h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN2	IPR122	DTCER122
	DMAC3I	123	01ECh	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN3	IPR123	DTCER123
	DMAC74I	124	01F0h	Level	✓	N/A	N/A	N/A	N/A	IER0F.IEN4	IPR124	—

Table 14.5 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	—
—	Reserved	126	01F8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	127	01FCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
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	0240h	Edge	✓	✓	✓	N/A	N/A	IER12.IEN0	IPR144	DTCER144
	INTB145	145	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 14.5 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 *4)	INTA208	208	0340h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN0	IPR208	DTCER208
	INTA209	209	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 14.5 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 *4)	INTA215	215	035Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN7	IPR215	DTCER215
	INTA216	216	0360h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN0	IPR216	DTCER216
	INTA217	217	0364h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN1	IPR217	DTCER217
	INTA218	218	0368h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN2	IPR218	DTCER218
	INTA219	219	036Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN3	IPR219	DTCER219
	INTA220	220	0370h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN4	IPR220	DTCER220
	INTA221	221	0374h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN5	IPR221	DTCER221
	INTA222	222	0378h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN6	IPR222	DTCER222
	INTA223	223	037Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN7	IPR223	DTCER223
	INTA224	224	0380h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN0	IPR224	DTCER224
	INTA225	225	0384h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN1	IPR225	DTCER225
	INTA226	226	0388h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN2	IPR226	DTCER226
	INTA227	227	038Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN3	IPR227	DTCER227
	INTA228	228	0390h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN4	IPR228	DTCER228
	INTA229	229	0394h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN5	IPR229	DTCER229
	INTA230	230	0398h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN6	IPR230	DTCER230
	INTA231	231	039Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN7	IPR231	DTCER231
	INTA232	232	03A0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN0	IPR232	DTCER232
	INTA233	233	03A4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN1	IPR233	DTCER233
	INTA234	234	03A8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN2	IPR234	DTCER234
	INTA235	235	03ACh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN3	IPR235	DTCER235
	INTA236	236	03B0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN4	IPR236	DTCER236
	INTA237	237	03B4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN5	IPR237	DTCER237
	INTA238	238	03B8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN6	IPR238	DTCER238
	INTA239	239	03BCh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN7	IPR239	DTCER239
	INTA240	240	03C0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN0	IPR240	DTCER240
	INTA241	241	03C4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN1	IPR241	DTCER241
	INTA242	242	03C8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN2	IPR242	DTCER242
	INTA243	243	03CCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN3	IPR243	DTCER243
	INTA244	244	03D0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN4	IPR244	DTCER244
	INTA245	245	03D4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN5	IPR245	DTCER245
	INTA246	246	03D8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN6	IPR246	DTCER246
	INTA247	247	03DCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN7	IPR247	DTCER247
	INTA248	248	03E0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN0	IPR248	DTCER248
	INTA249	249	03E4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN1	IPR249	DTCER249
	INTA250	250	03E8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN2	IPR250	DTCER250
	INTA251	251	03ECh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN3	IPR251	DTCER251
	INTA252	252	03F0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN4	IPR252	DTCER252
	INTA253	253	03F4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN5	IPR253	DTCER253
	INTA254	254	03F8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN6	IPR254	DTCER254
	INTA255	255	03FCh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN7	IPR255	DTCER255

Note: This table lists the interrupt vectors for the maximum specification. The interrupt vectors for individual products correspond to the functions listed in Table 1.2. For details, refer to Table 1.2, Comparison of Functions for Different Packages.

Note 1. For the group interrupt sources, refer to Table 14.7, 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 14.3, Interrupt Sources for Software Configurable Interrupt B.

Note that some interrupt sources cannot start the DTC or DMAC.

Note 4. For the software configurable interrupt A sources, refer to Table 14.4, Interrupt Sources for Software Configurable Interrupt A.
Note that some interrupt sources cannot start the DTC or DMAC.

14.3.2 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.

14.3.3 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.

14.4 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 14.2 shows types of interrupts.

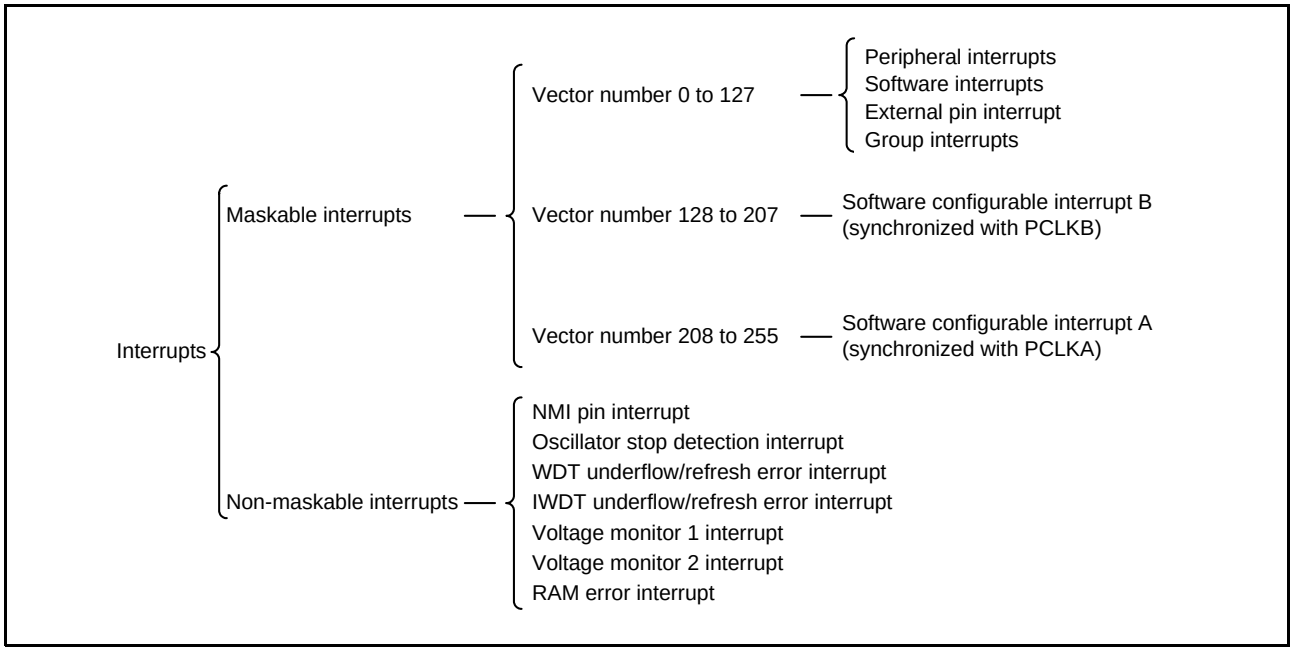


Figure 14.2 Types of Interrupts

14.4.1 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 14.4.5, Software Configurable Interrupts for details on software configurable interrupts.

14.4.2 Software Interrupts

When the SWINTR.SWINT bit and SWINT2R.SWINT2 bit are set to 1, the SWINT interrupt and SWINT2 interrupt occur, respectively.

14.4.3 External Pin Interrupt

An external pin interrupt is generated by signals input to the IRQi pin (i = 0 to 15). Refer to section 14.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupt.

14.4.4 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 14.6 lists types of group interrupts.

Table 14.6 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 14.3 shows the configuration of group interrupts.

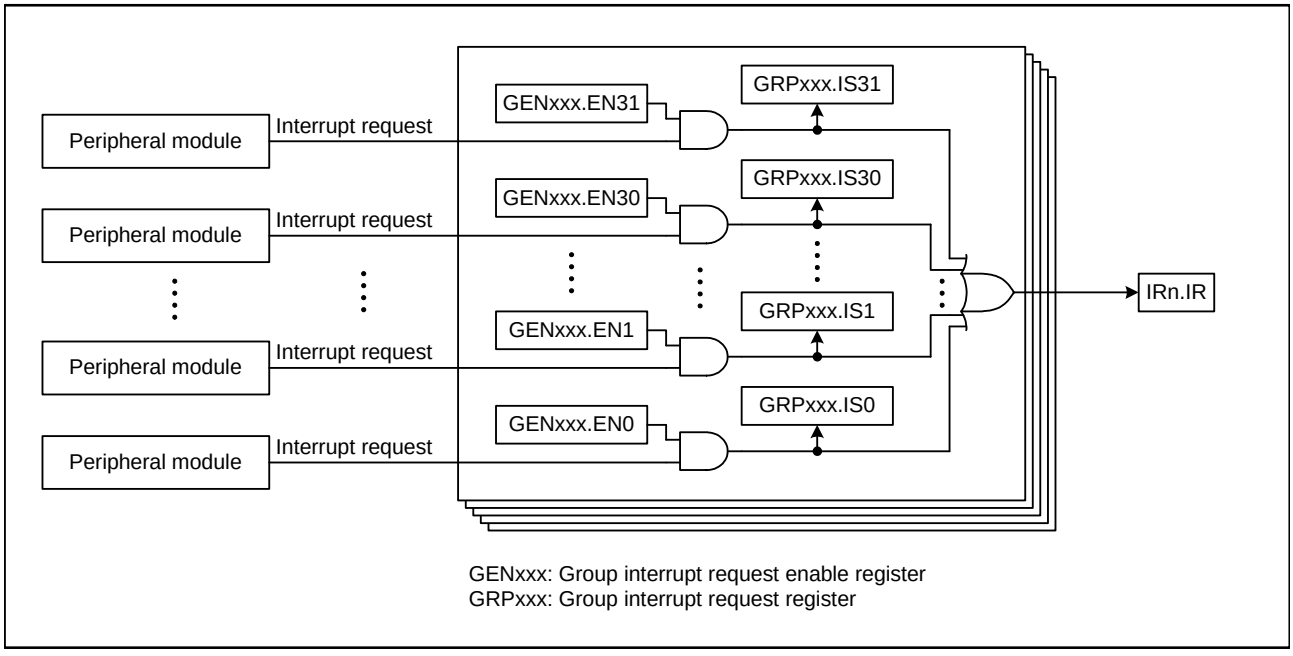


Figure 14.3 Group Interrupt Configuration (n = 17, 106, 107, 110 to 113)

Note 1. There is no register in the group to which no interrupt source is assigned.

(3) Group Interrupt Sources

Table 14.7 lists peripheral interrupt sources that are assigned to group interrupts.

Table 14.7 Group Interrupt Requests (1/2)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
IE0	0 to 31	Reserved	—	—	—	—	17
BE0	0 to 31	Reserved	—	—	—	—	106
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 14.7 Group Interrupt Requests (2/2)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
BL1	0 to 8	Reserved	—	—	—	—	111
	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 to 19	Reserved	—	—	—	—	
	20	S12AD	S12CMPAI (compare interrupt)	GENBL1.EN20	GRPBL1.IS20	—	
	21		S12CMPBI (compare interrupt)	GENBL1.EN21	GRPBL1.IS21	—	
	22, 23	Reserved	—	—	—	—	
	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 to 31	Reserved	—	—	—	—	
BL2	0	Reserved	—	—	—	—	107
	1	CANFD0	CHEI (channel error)	GENBL2.EN1	GRPBL2.IS1	—	
	2		CFRI (common FIFO receive)	GENBL2.EN2	GRPBL2.IS2	—	
	3	CANFD	GLEI (global error)	GENBL2.EN3	GRPBL2.IS3	—	
	4		RFRI (receive FIFO)	GENBL2.EN4	GRPBL2.IS4	—	
	5	CANFD0	CHTI (channel transmit)	GENBL2.EN5	GRPBL2.IS5	—	
	6	CANFD	RMRI (receive message buffer)	GENBL2.EN6	GRPBL2.IS6	—	
	7 to 31	Reserved	—	—	—	—	
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 to 23	Reserved	—	—	—	—	
	24	RSCI10	TEI (transmission end/condition generated)	GENAL0.EN24	GRPAL0.IS24	—	
	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, 31	Reserved	—	—	—	—	
AL1	0 to 31	Reserved	—	—	—	—	113

14.4.5 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 14.4 shows the software configurable interrupt configuration.

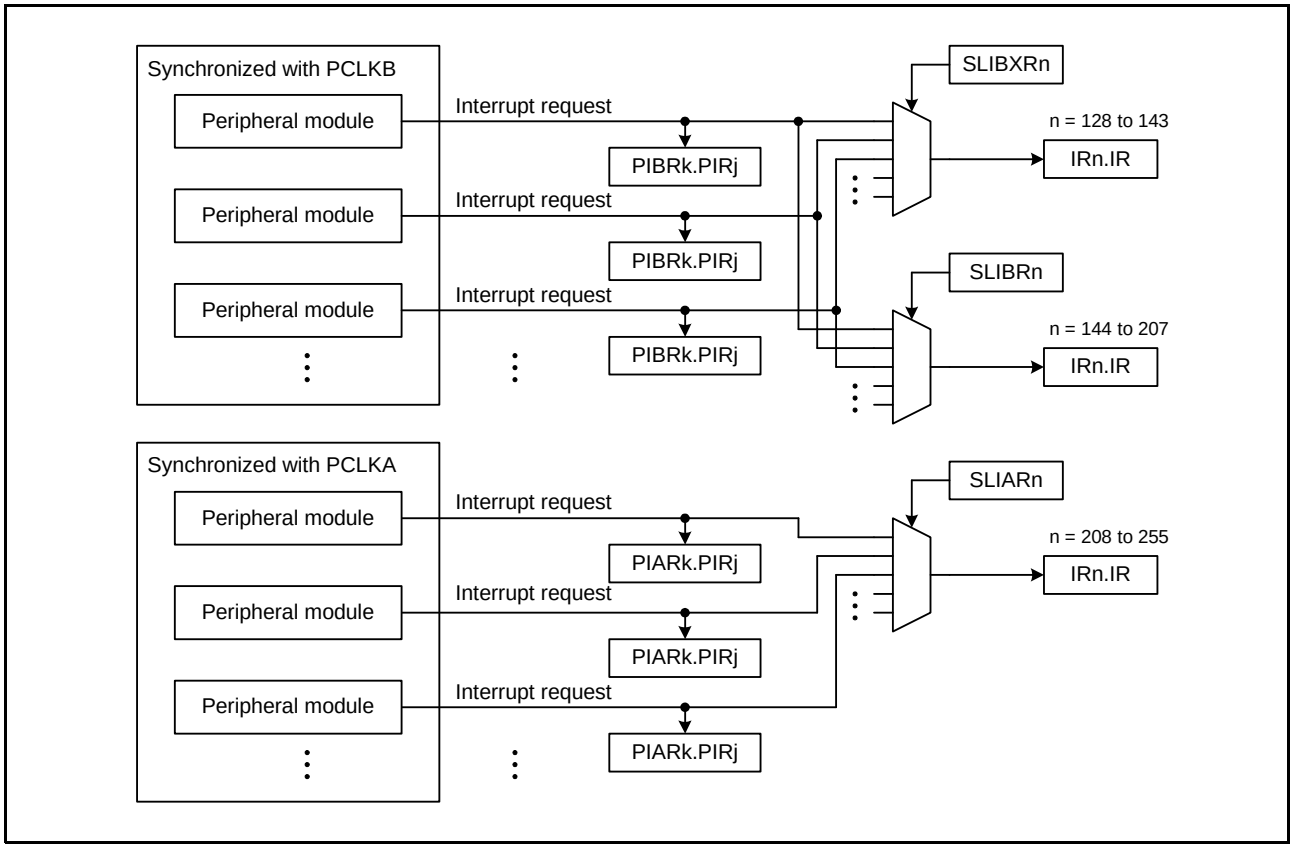


Figure 14.4 Software Configurable Interrupt Configuration

14.4.5.1 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 14.3, Interrupt Sources for Software Configurable Interrupt B for interrupt sources that can be assigned to software configurable interrupt B.

14.4.5.2 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 14.4, Interrupt Sources for Software Configurable Interrupt A for interrupt sources that can be assigned for software configurable interrupt A.

14.4.6 Non-Maskable Interrupts

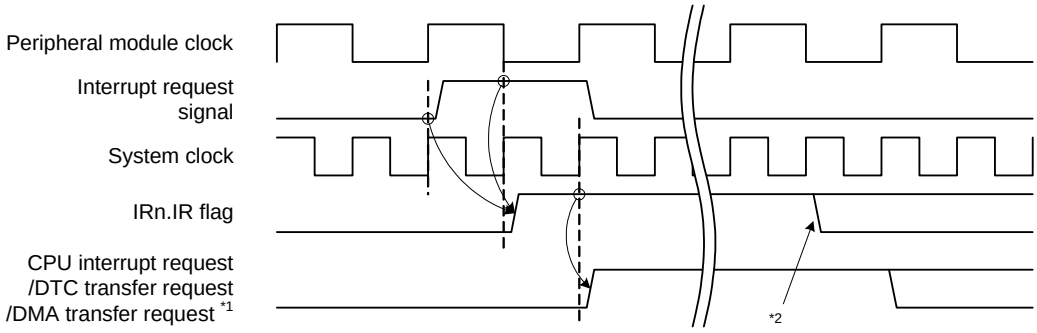
Non-maskable interrupts include the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error interrupt, IWDT underflow/refresh error interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and RAM error interrupt.
Non-maskable interrupts have the highest priority in all interrupts, including the fast interrupt, so are accepted regardless of the settings of the PSW.I bit (interrupt enable) and IPL[3:0] bits (processor interrupt priority level) in the CPU.
The NMISR register can be used to confirm whether a non-maskable interrupt is generated.
Only the CPU can be selected as the interrupt request destination of non-maskable interrupt. The DTC and DMAC cannot be selected.

14.5 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 14.5, 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 14.4.4, Group Interrupts for group interrupts. Refer to section 14.5.3, Group Interrupts Using Edge Detection and section 14.5.4, Group Interrupts Using Level Detection for interrupt requests of group interrupts.

14.5.1 Edge Detection

Figure 14.5 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 14.8, 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 14.7.3.1, 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 14.8, Operations When Starting the DTC/DMAC.

Figure 14.5 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 14.6 shows the interval for accepting consecutive interrupt request signals.

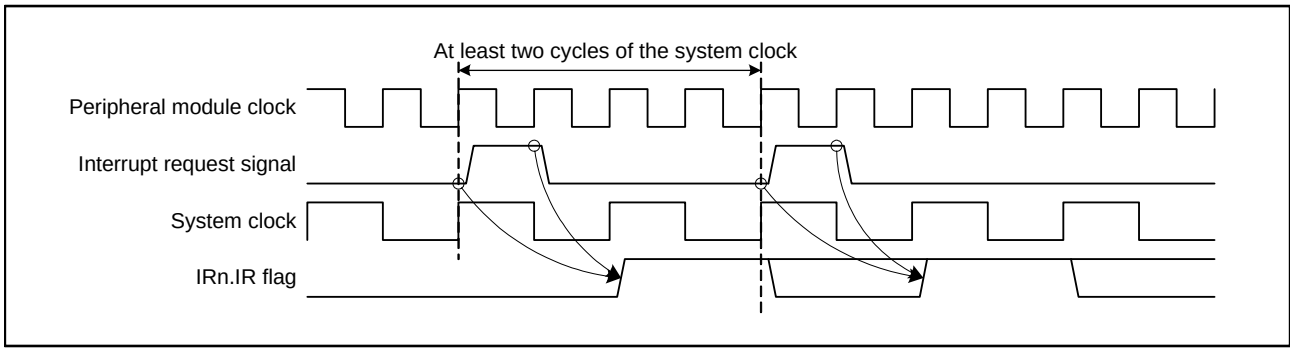


Figure 14.6 Accepting Consecutive Interrupt Signals (when the system clock frequency is lower than the peripheral clock frequency)

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, and RSPI, 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 14.7 shows the timing when the IRn.IR flag is set again.

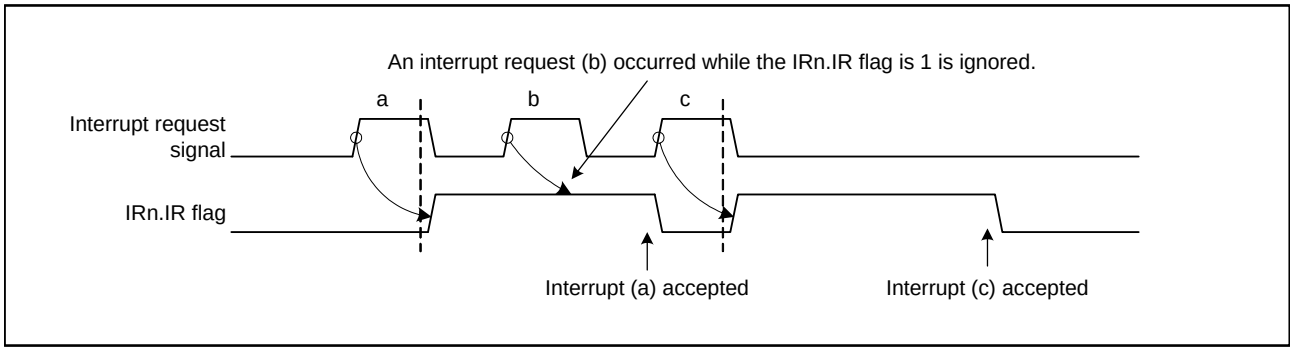


Figure 14.7 Timing to Set the IRn.IR Flag Again

(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.

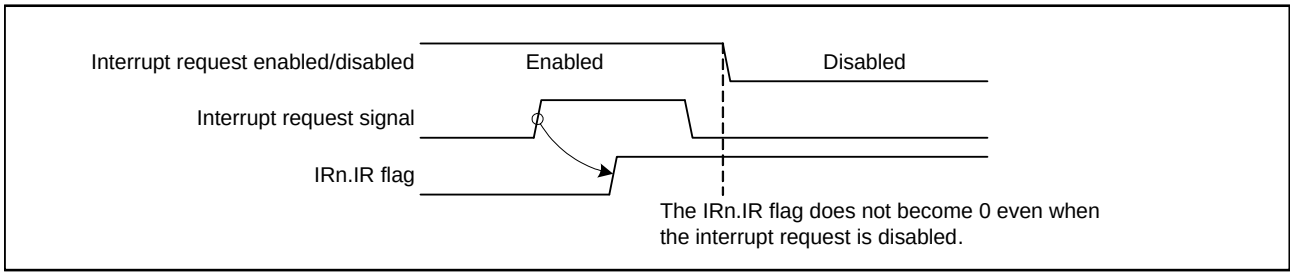


Figure 14.8 Relation Between Disabling the Interrupt Request and the IRn.IR Flag

14.5.2 Level Detection

Figure 14.9 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.

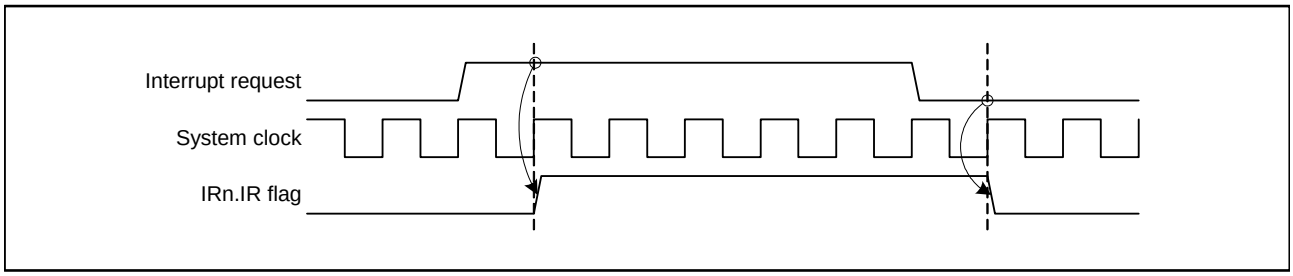


Figure 14.9 IRn.IR Flag Operation for Level Detection

Figure 14.10 shows an example of the procedure to handle interrupts for level detection.

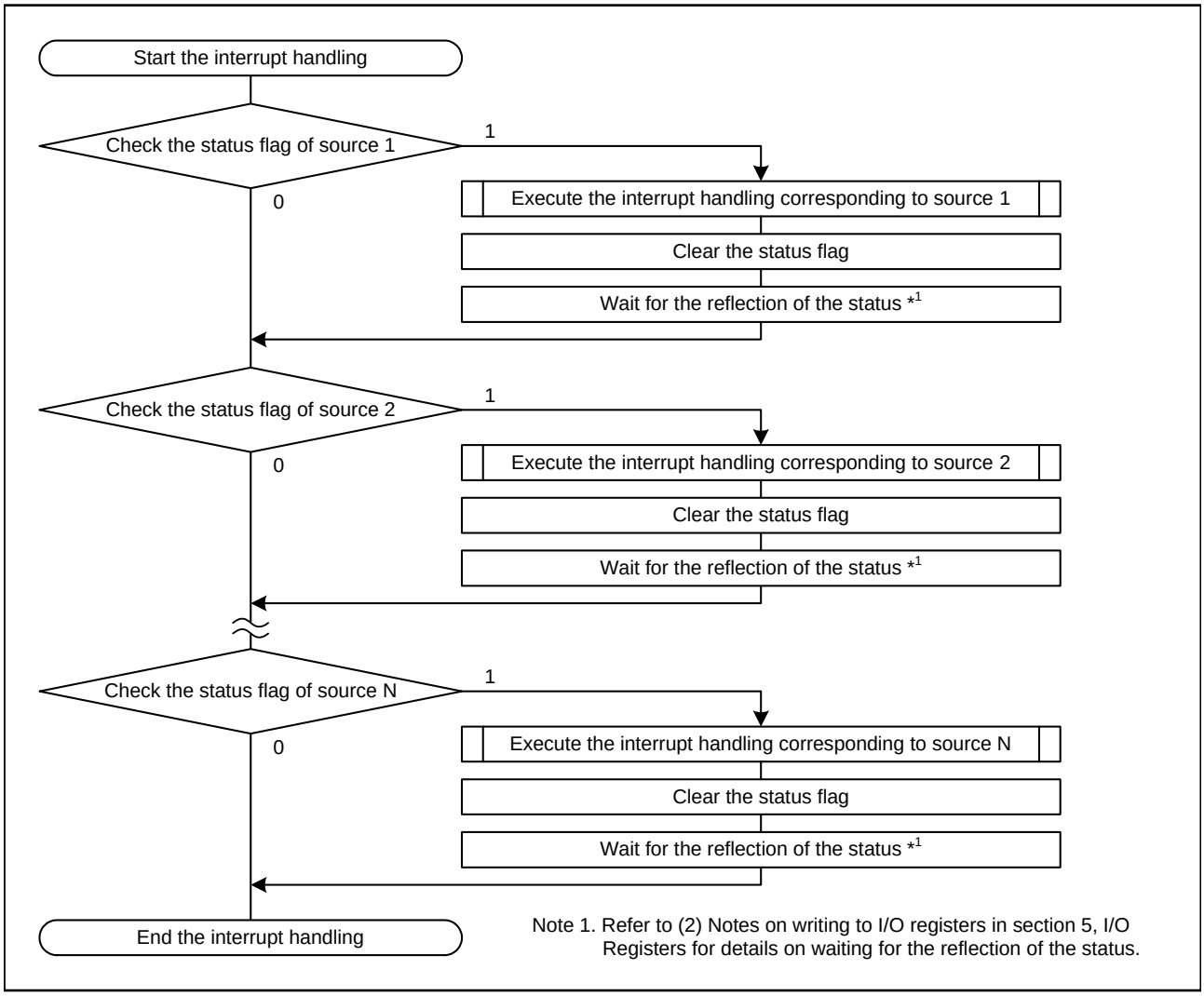


Figure 14.10 Example of Level Detection Interrupt Handling Procedure (N indicates the number of status flags)

14.5.3 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 14.11 and Figure 14.12 show operation examples of group interrupts using edge detection. Figure 14.13 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.

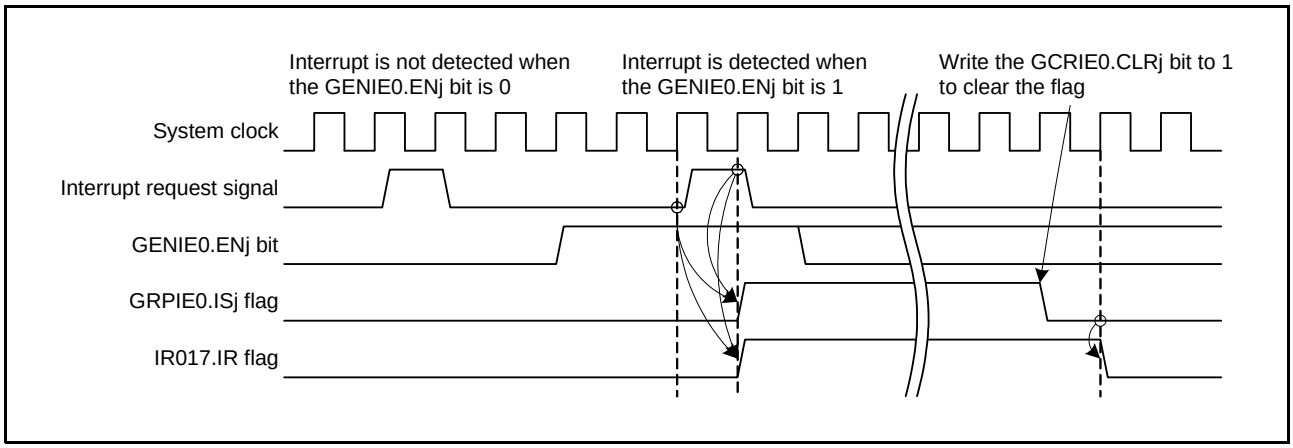


Figure 14.11 Example of Interrupt Request for Group Interrupt Using Edge Detection (Group IE0)

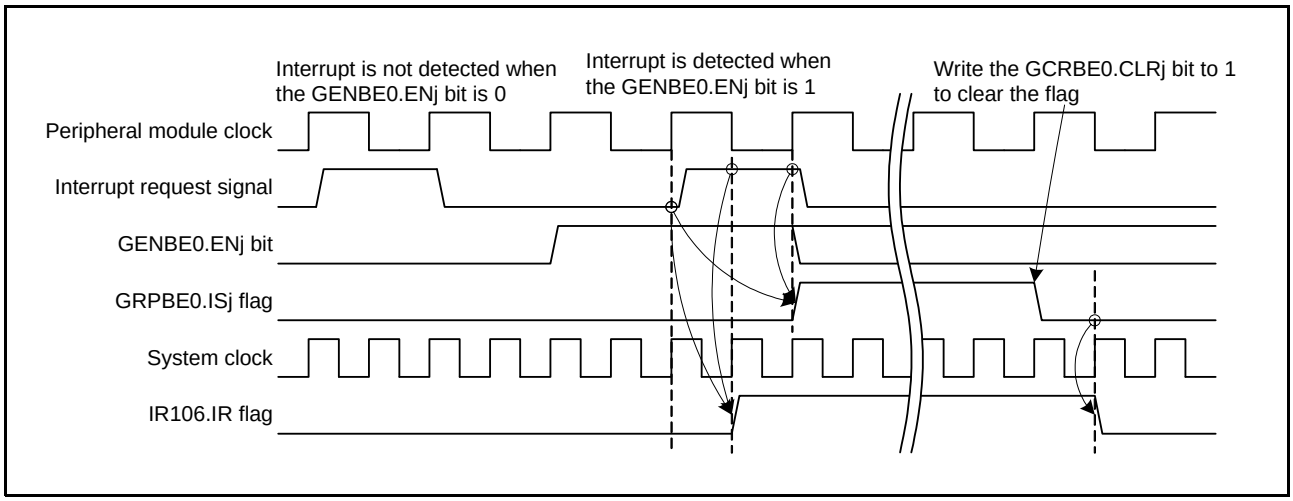


Figure 14.12 Example of Interrupt Request for Group Interrupt Using Edge Detection (Group BE0)

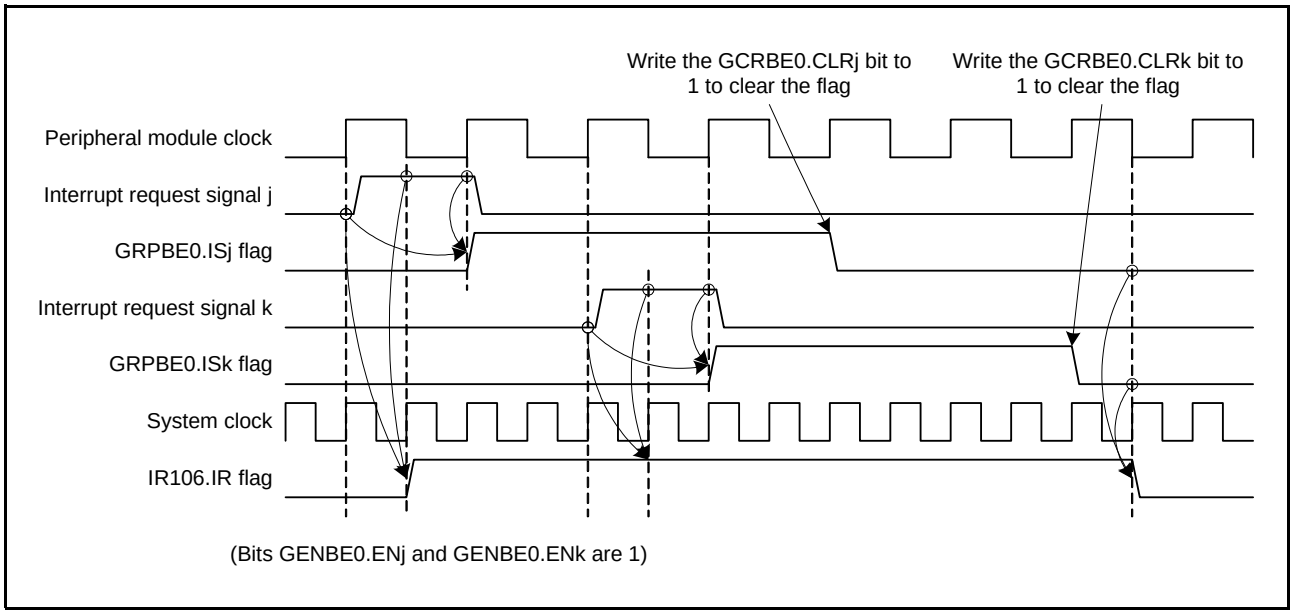


Figure 14.13 Example of Operation When Multiple Edge Detection Interrupt Requests Are Generated in the Same Group (Group BE0)

Figure 14.14 and Figure 14.15 show examples of the procedure to handle group interrupts using edge detection.

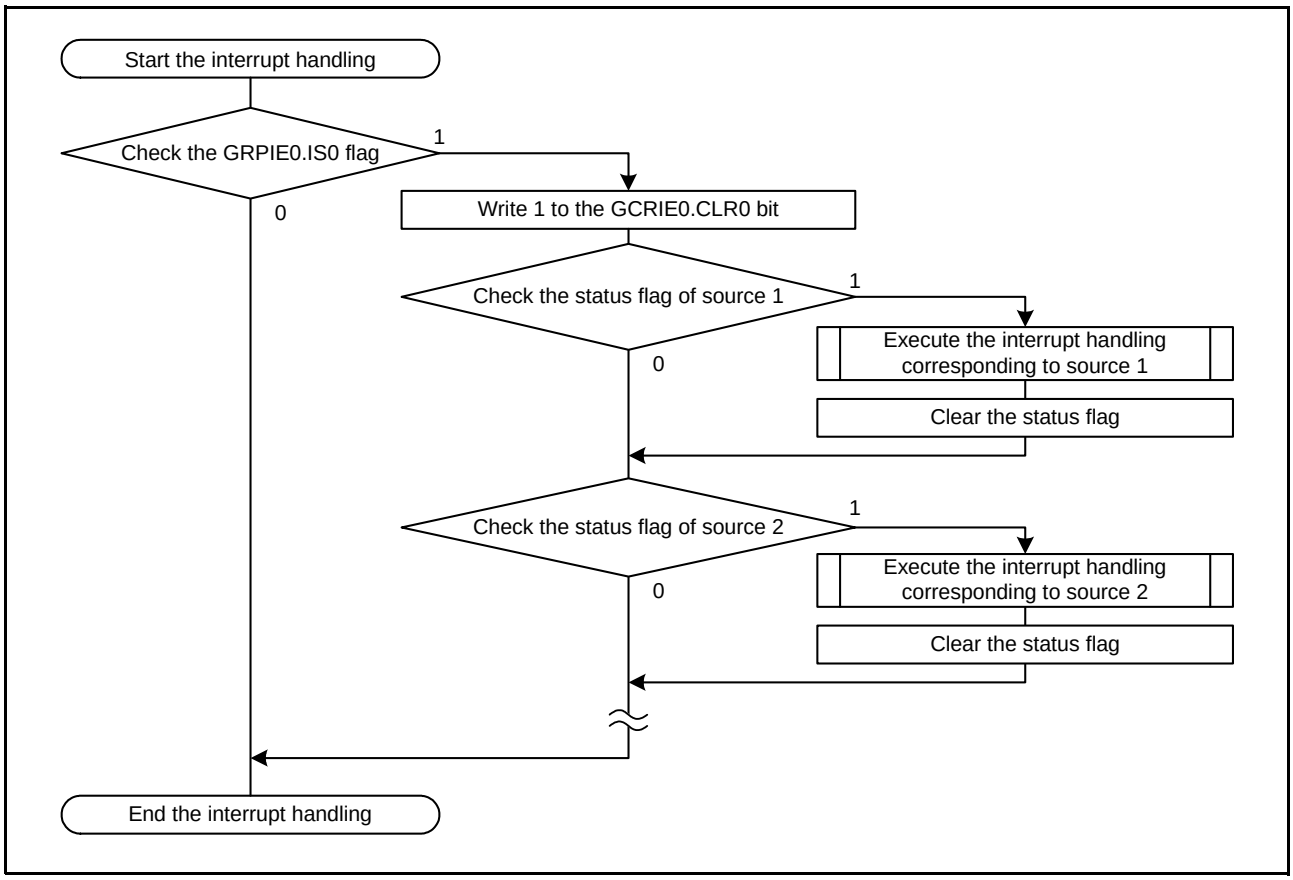


Figure 14.14 Example of Procedure to Handle Group Interrupts Using Edge Detection (Group IE0)

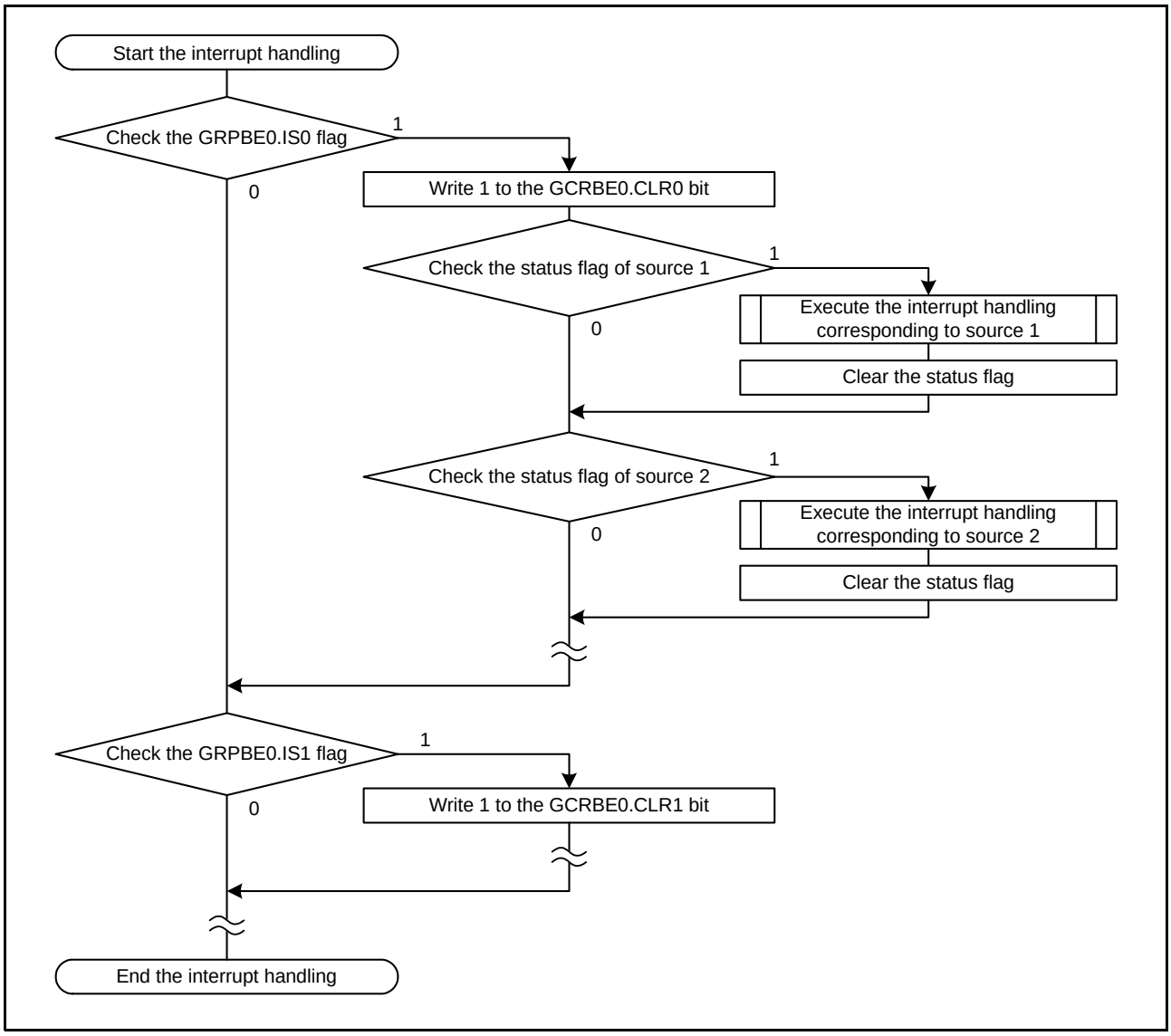


Figure 14.15 Example of Procedure to Handle Group Interrupts Using Edge Detection (Group BE0)

14.5.4 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 14.16 shows an operation example of group interrupts using edge detection. Figure 14.17 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.

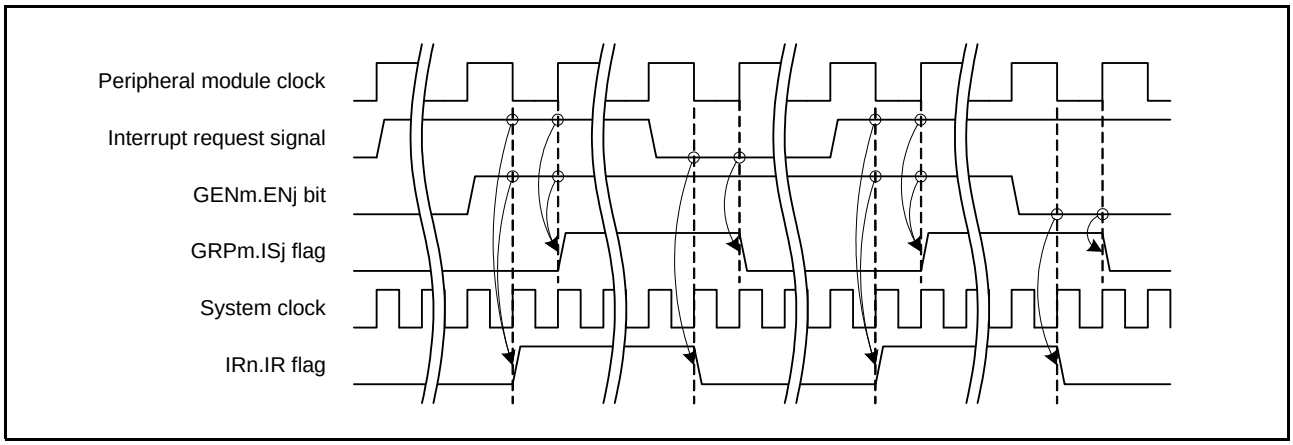


Figure 14.16 Operation Example of Group Interrupt Using Level Detection (m = BL0, BL1, BL2, AL0, AL1)

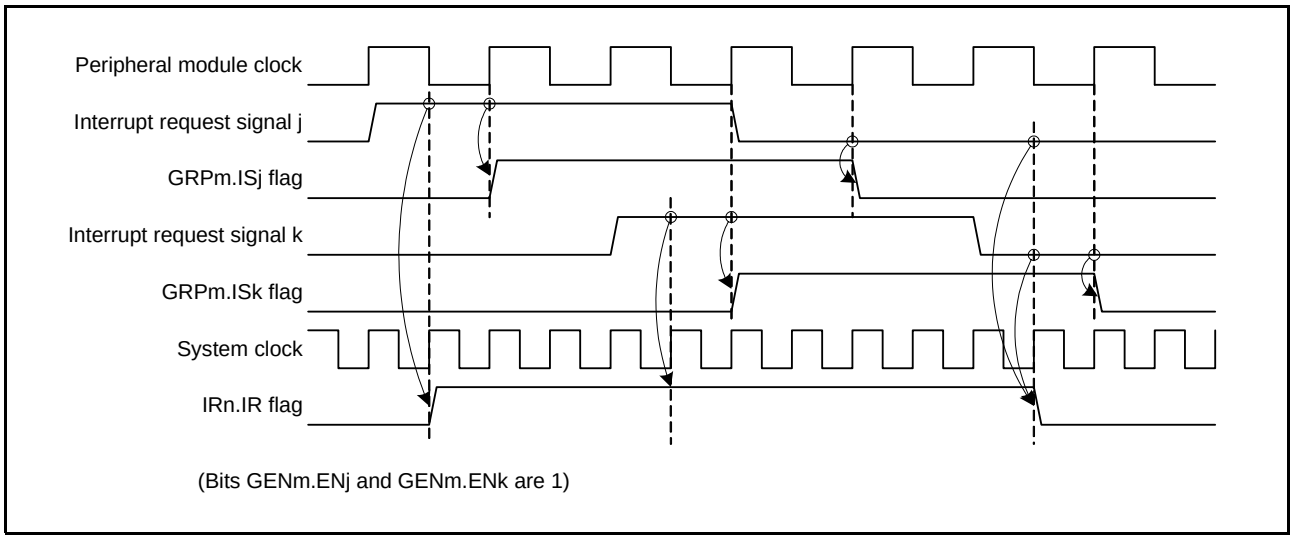


Figure 14.17 Operation Example When Multiple Interrupt Requests are Generated in the Same Group (m = BL0, BL1, BL2, AL0, AL1)

Figure 14.18 shows the procedure to handle group interrupts for level detection.

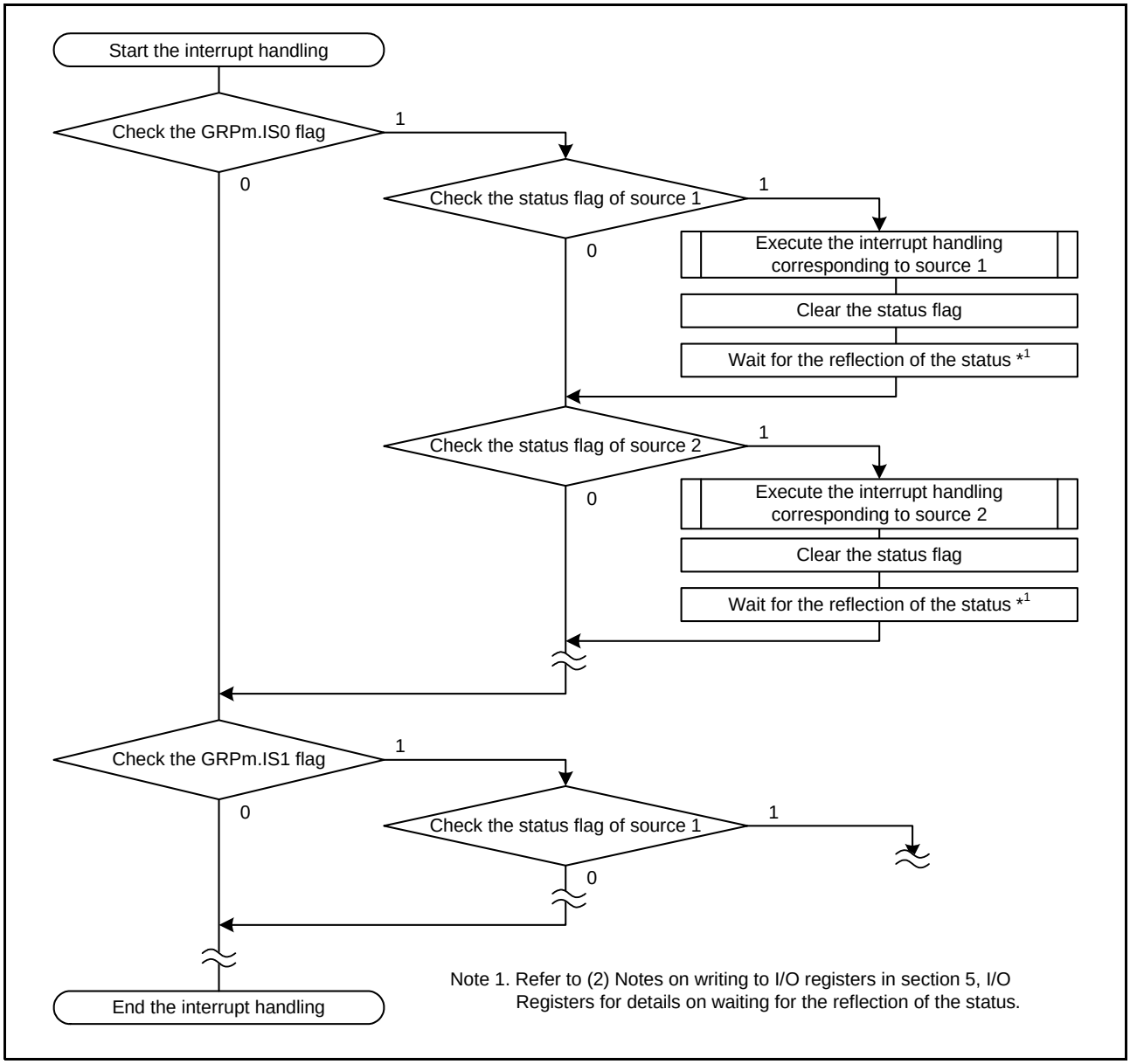


Figure 14.18 Example of Group Interrupt Handling Procedure for Level Detection (m = BL0, BL1, BL2, AL0, AL1)

14.5.5 Software Configurable Interrupts

Interrupt sources and interrupt requests for software configurable interrupts are detected by edge detection.
Figure 14.19 shows an operation example of the interrupt request and interrupt status flag for software configurable interrupts.

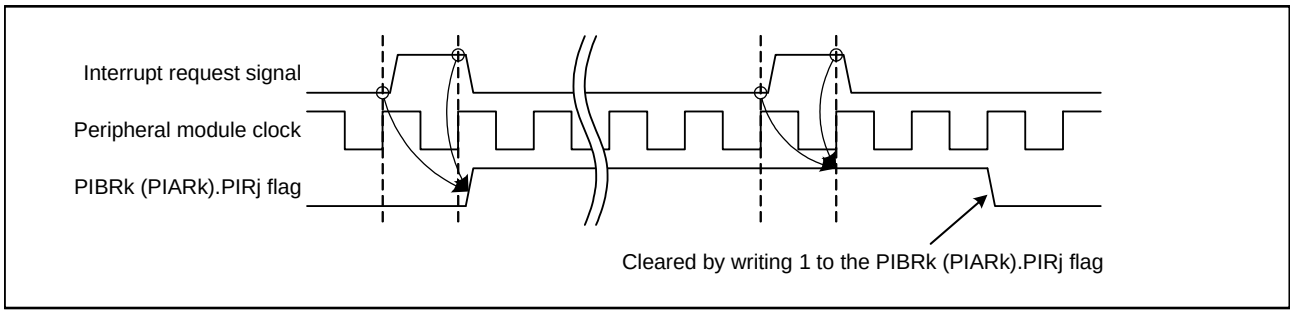


Figure 14.19 Operation Example of the Interrupt Request and Interrupt Status Flag for Software Configurable Interrupts

14.6 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 17, DMA Controller (DMACAa) for details on the DMAC channel priority.

14.7 Interrupt Setting Procedure

14.7.1 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.

14.7.2 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, or RSPI, 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.

14.7.3 Selecting Interrupt Request Destination

14.7.3.1 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 14.5, Interrupt Vector Table for details on the destinations. Do not select a destination that is not indicated as "✓" in Table 14.5.
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 17.3.7, Activating the DMAC in section 17, DMA Controller (DMACAa) for the procedure to set the DMAC.

(2) Setting Interrupt Sources as the DTC trigger

Perform the following setting while the IERm.IENj bit of the interrupt source that is selected as the DTC trigger is 0.

(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 18.5, DTC Setting Procedure in section 18, 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.

14.7.3.2 Operations When the DTC/DMAC Selected

Table 14.8 lists operations when the DTC or DMAC is set as an interrupt request destination.

Table 14.8 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 18.4, Chain Transfer Conditions in section 18, 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.

14.7.3.3 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 14.7.3.1, 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 14.7.3.1 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 14.7.3.1, 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 14.7.3.1 Interrupt Request Destination Setting Procedure.

14.7.4 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).

14.7.5 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 13, Exception Handling for details on the flow of non-maskable interrupt handling.
Excluding the RAMST flag, each flag in the NMISR register becomes 0 by writing 1 to the corresponding bit in the NMICLR register. To set the RAMST flag to 0, set the RAM.RAMSTS.RAMERR flag that has become 1 to 0.
Confirm that all flags in the NMISR register are 0 before exiting the interrupt handler of non-maskable interrupts.
Non-maskable interrupts, excluding the NMI pin interrupt, can be used as a maskable interrupt. When using as a maskable interrupt, do not change the NMIER register value from the value after reset. In addition, set the LVD1CR1.LVD1IRQSEL bit and LVD2CR1.LVD2IRQSEL bit to 1 when using voltage monitoring 1 interrupt and voltage monitoring 2 interrupt as a maskable interrupt.

14.7.6 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 14.7.4, Setting the External Pin Interrupt to set the associated registers. When using the digital filter for the NMI pin, refer to section 14.7.5, 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 14.20 shows an example of digital filter operation.

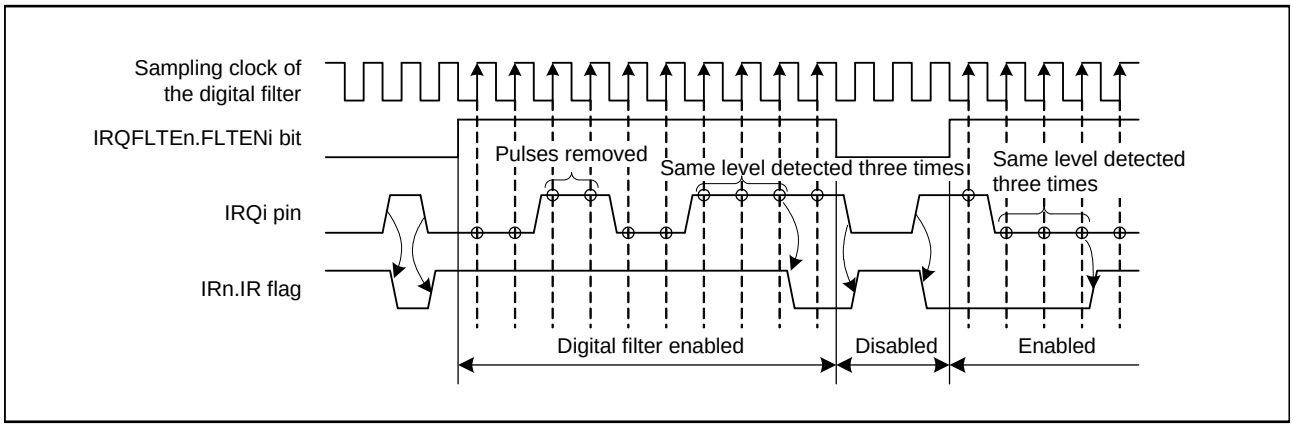


Figure 14.20 Digital Filter Operation Example (when the IRQCRi.IRQMD[1:0] bits are 00b (low level))

14.7.7 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 14.3, 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 14.4, Interrupt Sources for Software Configurable Interrupt A for details on interrupt source numbers that are assigned to software configurable interrupt A.
(4) Set the SLIPRCR.WPRC bit to 1.
(5) Confirm that the SLIPRCR.WPRC bit is 1.
(6) Select the interrupt request destination from the CPU, DTC, or DMAC. Refer to section 14.7.3.1, Interrupt Request Destination Setting Procedure for details on the setting procedure.
(7) Write 0 to the IRn.IR flag only when edge detection is selected (n = 128 to 255).
(8) Set the IERm.IENj bit to 1.

14.7.7.1 Polling for Software Configurable Interrupts

When polling an interrupt request by reading the PIBRk.PIRj (k = 0h, 1h, 5h, 6h, 8h to Ah, Ch, Dh) or PIARk.PIRj (k = 0h to 5h, Bh, Ch), 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.

14.8 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.

14.9 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 13, Exception Handling for details on the fast interrupt.

14.10 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 11.5.4, Deep Software Standby Mode for details on exiting deep software standby mode.

14.10.1 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 (GENBL0, GENBL1, GENBL2, GENAL0) (j = 0 to 31).

14.10.2 Exiting All-Module Clock Stop Mode

Non-maskable interrupts and interrupt sources that have a "✓" in the Exit from ACS column in Table 14.5, 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.

14.10.3 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 14.5, 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 14.7.6, Digital Filter for details on the procedure to set the digital filter.

14.11 Usage Notes

14.11.1 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.

14.11.2 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.

14.11.3 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.

15. Buses

15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned for each bus.

Table 15.1 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 (TFU, DTC, DMAC, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK)
	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 (DOC, REMC, CANFD and CMPC) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (MTU, RSPI and SCIi) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 5	• Connected to peripheral modules (RSCI and CANFD) • 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)

P/E: Programming/Erasure

BCLK (external-bus clock): 40 MHz (max.) The CSC (CS area controller) operates in synchronization with the BCLK.

BCLK pin output: The frequency is the same as the BCLK as default. 1/2 BCLK can be supplied by setting the BCLK pin output select bit (BCKCR.BCLKDIV) in the external bus clock control register. For details, see section 9, Clock Generation Circuit.

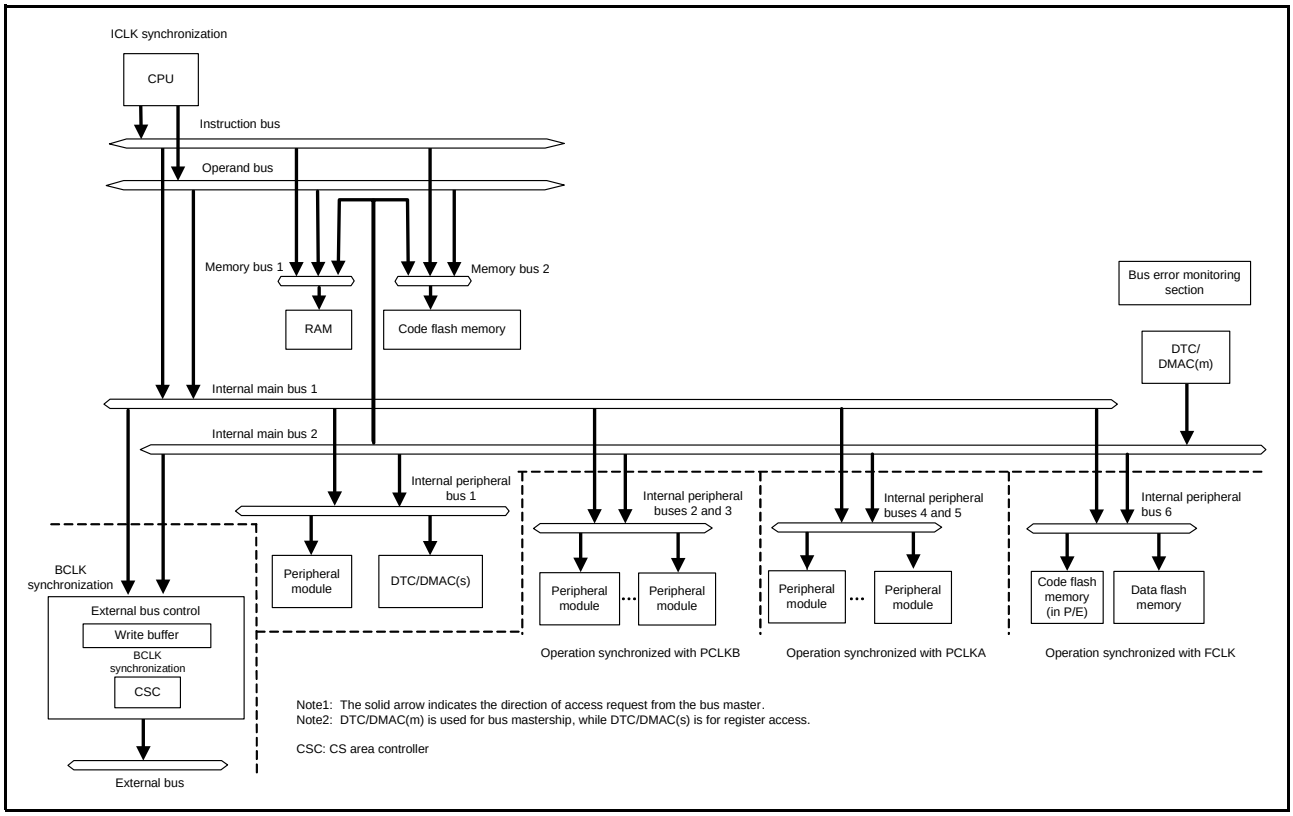


Figure 15.1 Bus Configuration

Table 15.2 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 0001 FFFFh	Memory bus 1		RAM	
0002 0000h to 0007 FFFFh			Reserved area	
0008 0000h to 0008 7FFFh	Internal peripheral bus 1		Peripheral I/O registers	
0008 8000h to 0009 FFFFh	Internal peripheral bus 2			
000A 0000h to 000B FFFFh	Internal peripheral bus 3			
000C 0000h to 000D FFFFh	Internal peripheral bus 4			
000E 0000h to 000F FFFFh	Internal peripheral bus 5			
0010 0000h to 007F FFFFh	Internal peripheral bus 6	Reserved area	Data flash memory, code flash memory (for programming only)	Reserved area
0080 0000h to 00FF FFFFh	Reserved area		Reserved area	
0100 0000h to 04FF FFFFh	External bus		Reserved area	
0500 0000h to 07FF FFFFh			External address space (CS1 to CS3)	
0800 0000h to 0FFF FFFFh			Reserved area	
1000 0000h to 7FFF FFFFh	Reserved area		Reserved area	
8000 0000h to FFDF FFFFh	Memory bus 2	Reserved area	Code flash memory (for reading only)	Reserved area
FFE0 0000h to FFFF FFFFh		External bus		External address space (CS0)

15.2 Description of Buses

15.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. The instruction bus is 64 bits while the operand bus is 32 bits.
Connection of the instruction and operand buses to RAM and code flash memory provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to code flash memory by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, 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 address space is possible.

15.2.2 Memory Buses

The memory buses consist of 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.

15.2.3 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 15.3.
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 15.3 Order of Priority for Bus Masters

Priority	Bus Master
High	DMAC
↑	DTC
Low	CPU

Note: The above applies when the priority order of the buses is fixed.

15.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 15.4.

Table 15.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	TFU, DTC, DMAC, 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	DOC, REMC, CANFD, CMPC
Internal peripheral bus 4	MTU, RSPI, and SCIi
Internal peripheral bus 5	RSCI, CANFD
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 15.2).

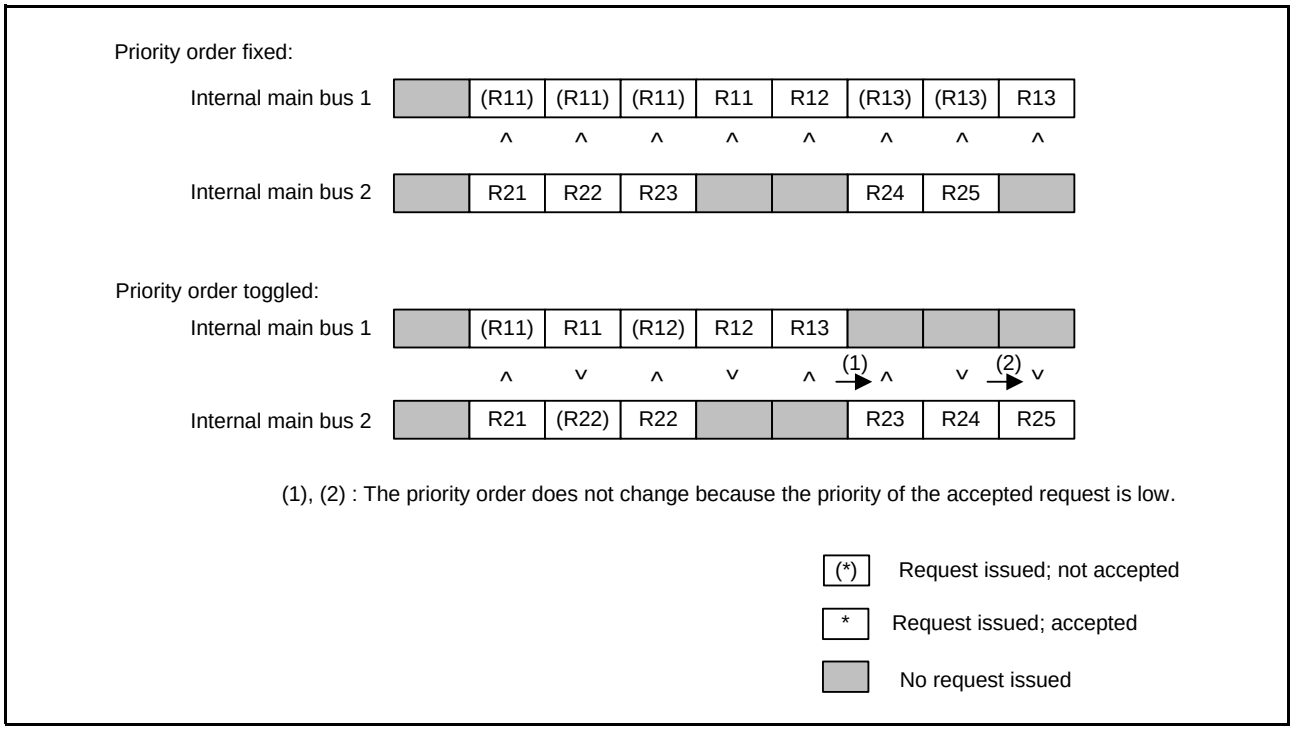


Figure 15.2 Priority Order between Internal Peripheral Bus Accesses

15.2.5 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 15.3).

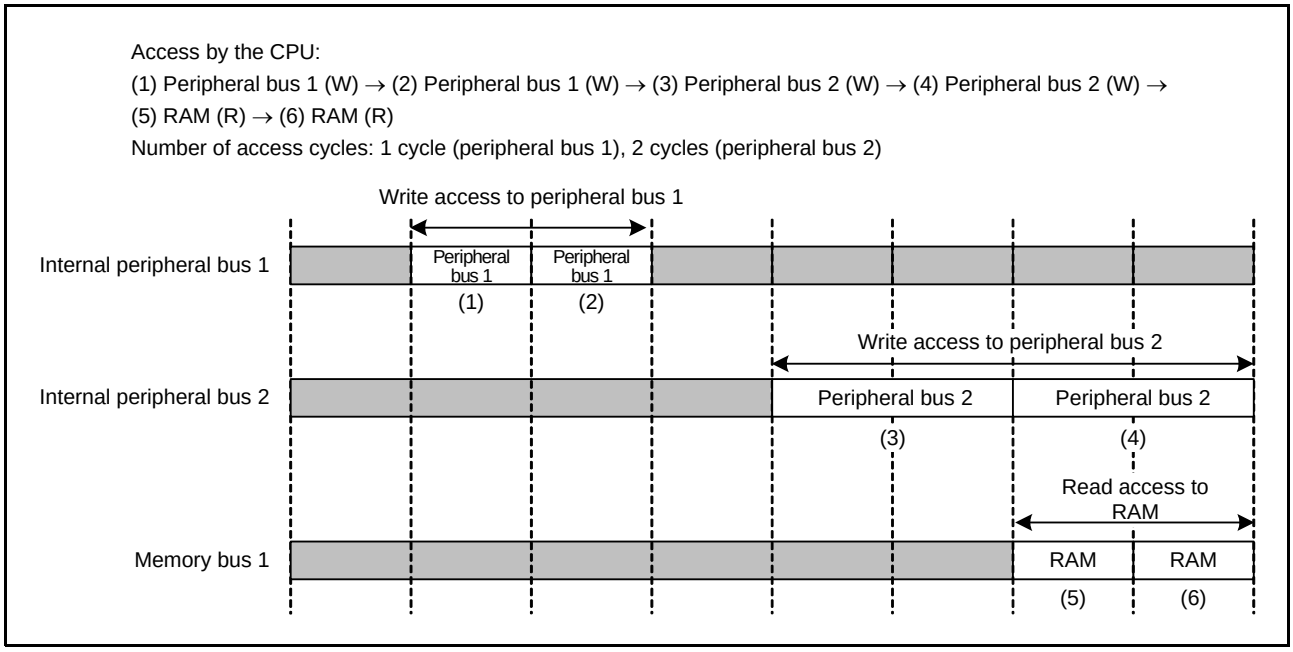


Figure 15.3 Write Buffer Function

15.2.6 External Bus

Table 15.5 lists the specifications of the external bus.

The external bus controller arbitrates requests for bus mastership on the external address space and on the external bus controller register (CSC) from the internal main bus 1 and the internal main bus 2.

The priority order of these two 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 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 internal main bus 2.

The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 15.4).

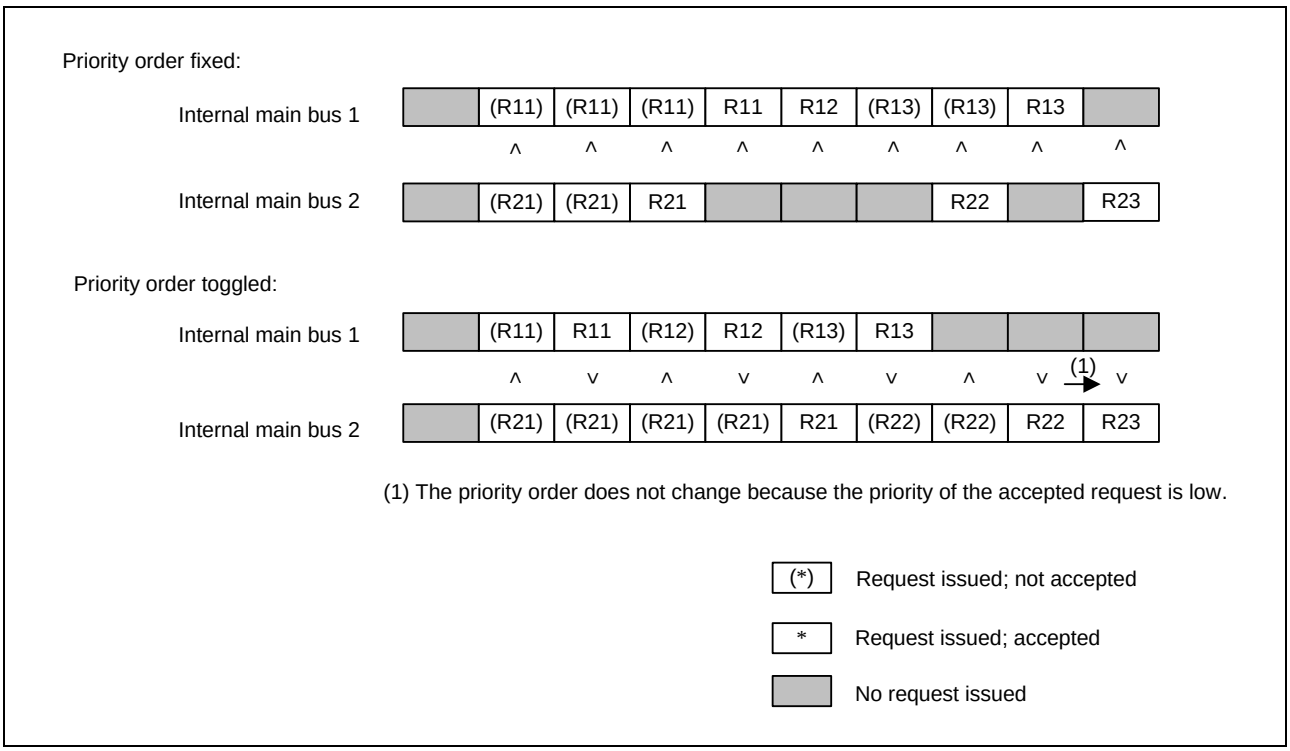


Figure 15.4 Priority Order of Internal Peripheral Bus Accesses

Table 15.5 Specifications of the External Bus

Item	Description
External address space	• An external address space is divided into four CS areas (CS0 to CS3) 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 CS3#) 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.
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).

Table 15.6 lists the input/output pins of the external bus.

Table 15.6 Pin Configuration of the External Bus

Pin Name	I/O	Description
A20 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)
RD#	Output	A strobe signal indicating that reading from an external address space (CS0 to CS3) is in progress
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 CS3) (Low: Wait request)

Note 1. The A0 and BC0# pin functions share the same pin, and either becomes effective according to the area, with the function being A0 in byte strobe mode and BC0# in single write strobe mode. Note that setting the 8-bit external bus width is prohibited in single write strobe mode. For information on other multiplexed pin functions, see section 20, I/O Ports.

15.2.7 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from code flash memory and an operand from RAM, the DMAC is able to handle transfer between a peripheral bus and the external bus at the same time.
An example of parallel operations is shown in Figure 15.5. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to code flash memory and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to RAM and code flash memory by the CPU.

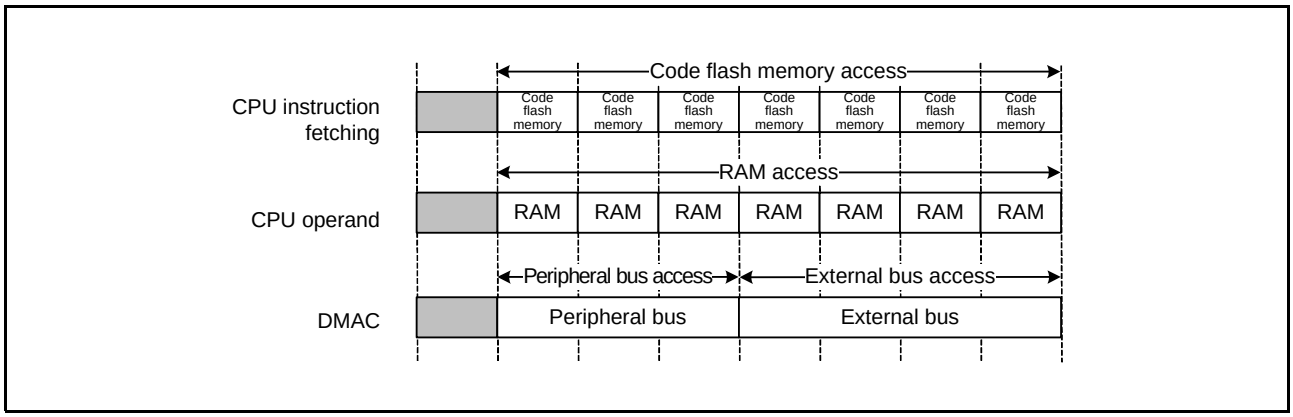


Figure 15.5 Example of Parallel Operations

15.2.8 Bus Settings

(1) Set the mode of the external bus in the CSn mode register (CSnMOD), CSn wait-control register 1 (CSnWCR1), CSn wait-control register 2 (CSnWCR2), CSn control register (CSnCR), CSn recovery-cycle setting register (CSnREC), CS recovery cycle insertion enable register (CSRECEN), bus error monitoring enable register (BEREN), and bus priority control register (BUSPRI).
(2) Make settings for pins in the CS output enable register (PFCSE), CS output pin select register 0 (PFCSS0), 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).

15.2.9 Restrictions

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

(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.

15.3 Register Descriptions

15.3.1 CSn Control Register (CSnCR) (n = 0 to 3)

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: 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 3)	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 3)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write the CSnCR register while the external bus is being accessed.

EXENB Bit (Operation Enable)

This bit enables or disables operation of the respective CS areas.

After this 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.

15.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 3)

Address(es): CS0REC 0008 380Ah, CS1REC 0008 381Ah, CS2REC 0008 382Ah, CS3REC 0008 383Ah

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RRCV[3:0]	Read Recovery	b3 b0 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	WRCV[3:0]	Write Recovery	b11 b8 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write the CSnREC register while the external bus is being accessed.
When the preceding bus access is a separate bus access, CSnREC is valid when the recovery cycle insertion is enabled with the separate bus recovery cycle insertion enable bit (RCVENj (j = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the multiplexed bus recovery cycle insertion enable bit (RCVENMj) in CSRECEN.

RRCV[3:0] Bits (Read Recovery)

These bits specify the number of recovery cycles to be inserted after a read access to the external bus.

When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] Bits (Write Recovery)

These bits specify the number of recovery cycles to be inserted after a write access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

15.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): 0008 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W

Do not attempt to write the CSRECEN register while the external bus is being accessed.

RCVEN0 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 Bit (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 15.7 Insertion of Recovery Cycles

Access Type	External Address Space	Insertion of Recovery Cycles	Corresponding Bits (Separate/Multiplexed)
Read access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN0/RCVENM0
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN1/RCVENM1
Read access after write access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN2/RCVENM2
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN3/RCVENM3
Write access after read access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN4/RCVENM4
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN6/RCVENM6
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN7/RCVENM7

15.3.4 CSn Mode Register (CSnMOD) (n = 0 to 3)

Address(es): CS0MOD 0008 3002h, CS1MOD 0008 3012h, CS2MOD 0008 3022h, CS3MOD 0008 3032h

	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 15.8 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.

15.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 3)

Address(es): CS0WCR1 0008 3004h, CS1WCR1 0008 3014h, CS2WCR1 0008 3024h, CS3WCR1 0008 3034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is valid only when the PWENB bit in CSnMOD is set to 1.
Note 2. The CSPRWAIT[2:0] value is valid only when the PRENB bit in CSnMOD is set to 1.

Do not attempt to write the CSnWCR1 register while the external bus is being accessed.
Set each of these bits within a range of the restrictions described in section 15.5.7 (1) Limitations on Using Separate Bus Interface or section 15.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

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.

15.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 3)

Address(es): CS0WCR2 0008 3008h, CS1WCR2 0008 3018h, CS2WCR2 0008 3028h, CS3WCR2 0008 3038h

	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 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 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 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 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not attempt to write the CSnWCR2 register while the external bus is being accessed.
Set each of these bits within a range of the restrictions described in section 15.5.7 (1) Limitations on Using Separate Bus Interface or section 15.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

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 3) 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 3) 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.

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: 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: 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 3) 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.

15.3.7 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.

15.3.8 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.

15.3.9 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: Reserved	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.

15.3.10 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

15.3.11 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BPEB[1:0]		BPFB[1:0]		BPHB[1:0]		BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 (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	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC and DMAC are written to more than one time, the 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 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 internal main bus 2.

15.4 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 area).

15.4.1 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 A20 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-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 15.6 and Figure 15.7.
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

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.6 Data Alignment (Little Endian) in 16-Bit Bus Space

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	15 8
			Second	8 bits	4n+2	7 0
	4n+2	One	First	16 bits	4n+2	15 8 7 0
	4n+3	Two	First	8 bits	4n+2	15 8
			Second	8 bits	4n+4	7 0
32 bits	4n	Two	First	16 bits	4n	31 24 23 16
			Second	16 bits	4n+2 (p)	15 8 7 0
	4n+1	Three	First	8 bits	4n	31 24
			Second	16 bits	4n+2	23 16 15 8
			Third	8 bits	4n+4	7 0
	4n+2	Two	First	16 bits	4n+2	31 24 23 16
			Second	16 bits	4n+4	15 8 7 0
	4n+3	Three	First	8 bits	4n+2	31 24
			Second	16 bits	4n+4	23 16 15 8
			Third	8 bits	4n+6	7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.7 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 A20 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- or 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 15.8 and Figure 15.9.

In 8-bit bus space, the valid positions of data external to the chip are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# / WR0#/BC0# / RD# / Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n+1		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+3		7 0
16 bits	4n	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
	4n+1	Two	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2	(p)	15 8
	4n+2	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3	(p)	15 8
	4n+3	Two	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
			Third	8 bits	4n+2	(p)	23 16
			Fourth	8 bits	4n+3	(p)	31 24
	4n+1	Four	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2	(p)	15 8
			Third	8 bits	4n+3	(p)	23 16
			Fourth	8 bits	4n+4		31 24
	4n+2	Four	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3	(p)	15 8
			Third	8 bits	4n+4		23 16
			Fourth	8 bits	4n+5	(p)	31 24
	4n+3	Four	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
			Third	8 bits	4n+5	(p)	23 16
			Fourth	8 bits	4n+6	(p)	31 24

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.8 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# D15 D8	WR0#/BC0# D7 D0
							RD# Data Bus	
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+1	(p)		7 0
	4n+1	Two	First	8 bits	4n+1			15 8
			Second	8 bits	4n+2	(p)		7 0
	4n+2	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+3	(p)		7 0
	4n+3	Two	First	8 bits	4n+3			15 8
			Second	8 bits	4n+4			7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n+1	(p)		23 16
			Third	8 bits	4n+2	(p)		15 8
			Fourth	8 bits	4n+3	(p)		7 0
	4n+1	Four	First	8 bits	4n+1			31 24
			Second	8 bits	4n+2	(p)		23 16
			Third	8 bits	4n+3	(p)		15 8
			Fourth	8 bits	4n+4			7 0
	4n+2	Four	First	8 bits	4n+2			31 24
			Second	8 bits	4n+3	(p)		23 16
			Third	8 bits	4n+4			15 8
			Fourth	8 bits	4n+5	(p)		7 0
	4n+3	Four	First	8 bits	4n+3			31 24
			Second	8 bits	4n+4			23 16
			Third	8 bits	4n+5	(p)		15 8
			Fourth	8 bits	4n+6	(p)		7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.9 Data Alignment (Big Endian) in 8-Bit Bus Space

15.5 Operation of CS Area Controller

15.5.1 Separate Bus

The various periods in the timing charts are described below.
The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). The operation cycles, such as wait cycles specified with the CSC register, are counted on BCLK. In the following description, frequencies of BCLK and BCLK pin output are the same, unless otherwise noted.
Access via the external bus starts at the same point as the output of a rising edge on the BCLK pin. However, if the external bus clock (BCLK) and the output on the BCLK pin are at different frequencies so that a single request from a bus master for transfer leads to two or more rounds of access via the external bus, the wait settings may cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the BCLK pin (see Figure 15.15 to Figure 15.19). If recovery cycles are inserted for bus access, the setting for the number of recovery cycles may also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the BCLK pin (see Figure 15.35).

(a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles are selectable within the range from zero to 31. Within this period, the timing of CSn#, RD#, and WRn# assertion (placing the signals at the low level) is determined by the respective wait settings. Specifically, the periods of waiting are controlled by the CS assert wait select bits (CSON[2:0]), the RD assert wait select bits (RDON[2:0]), the WR assert wait select bits (WRON[2:0]), and the write data output wait select bits (WDON[2:0]) in CSn wait control register 2 (CSnWCR2). The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

(b) Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If each wait select bit for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.
If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. Tend indicates the cycle where sampling of the wait signal starts.
After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (point (e) below) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (point (d) below) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

(c) Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF[2:0]) and the write-access CS extension cycle select bits (CSWOFF[2:0]) in the CSn wait control register 2 (CSnWCR2), respectively.
The number of cycles are counted from the cycle following the cycle where the strobe signal is valid.
In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.

For write access, setting the write data output extension cycle select bits (WDOFF) controls extension of the period where the address and output data are valid.

(d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).
In the case of normal access, this is inserted within the period of clock cycles of CS extension (point (c) above).
In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (point (c) above). Valid address and data output are extended over this period, and the WRn# signal is negated.

(e) Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.
CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

(f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, see section 15.5.4, 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 15.10 to Figure 15.12 show the normal access operations.

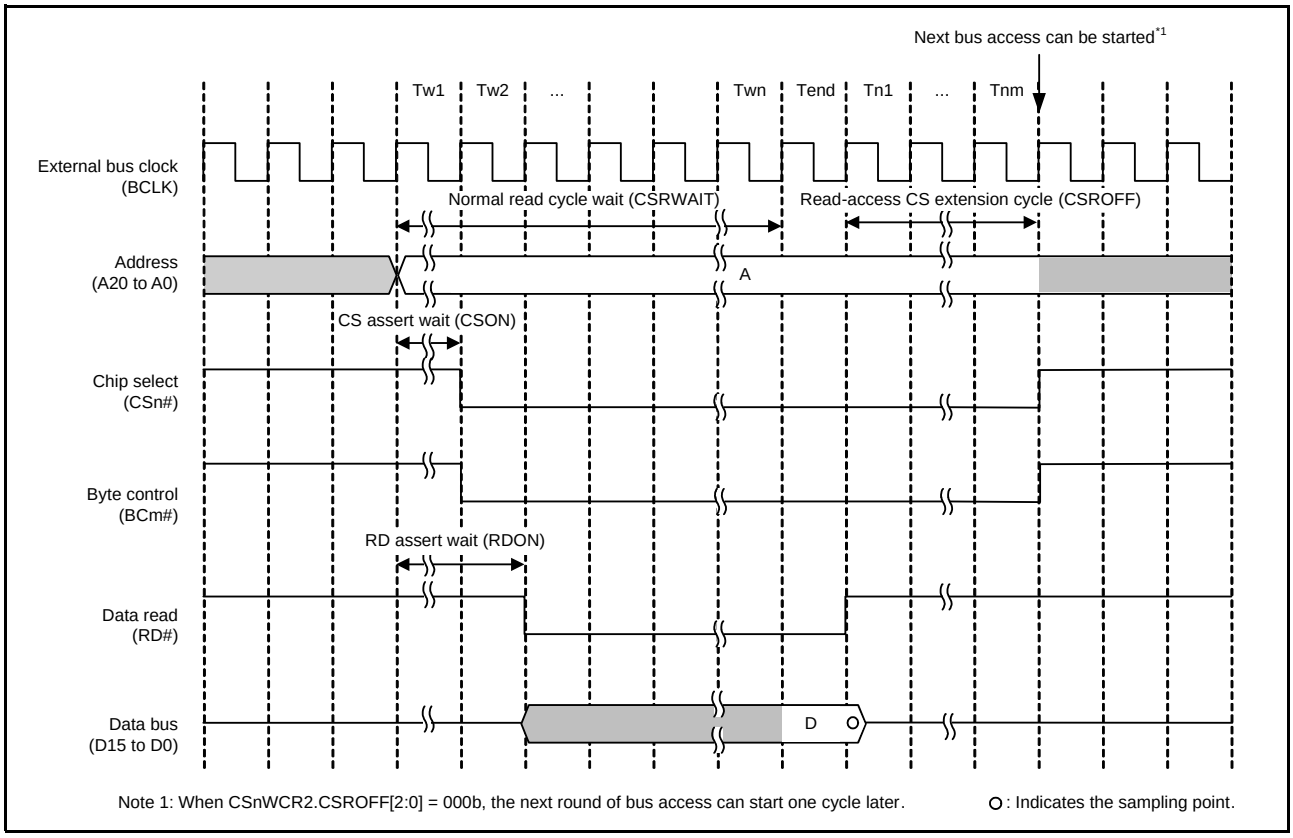


Figure 15.10 Bus Timing (Normal-Read Operation) (n = 0 to 3, m = 0, 1)

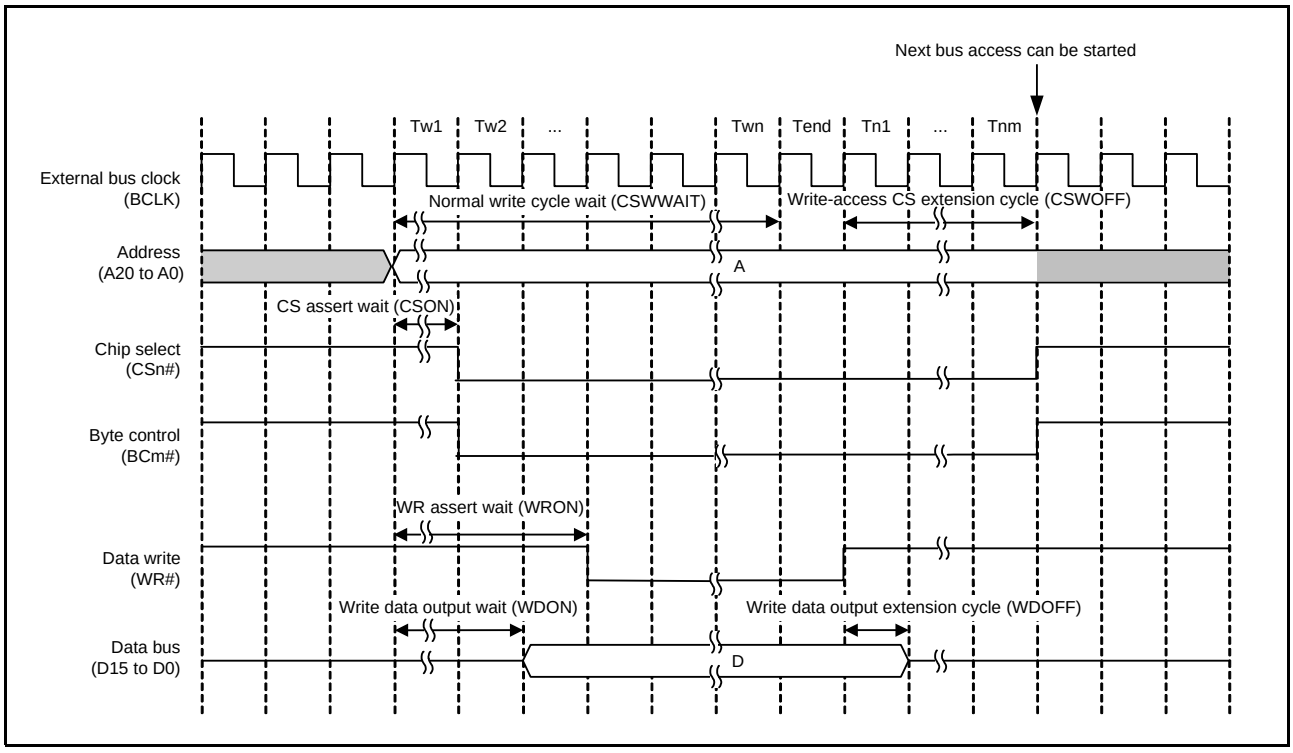


Figure 15.11 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 3, m = 0, 1)

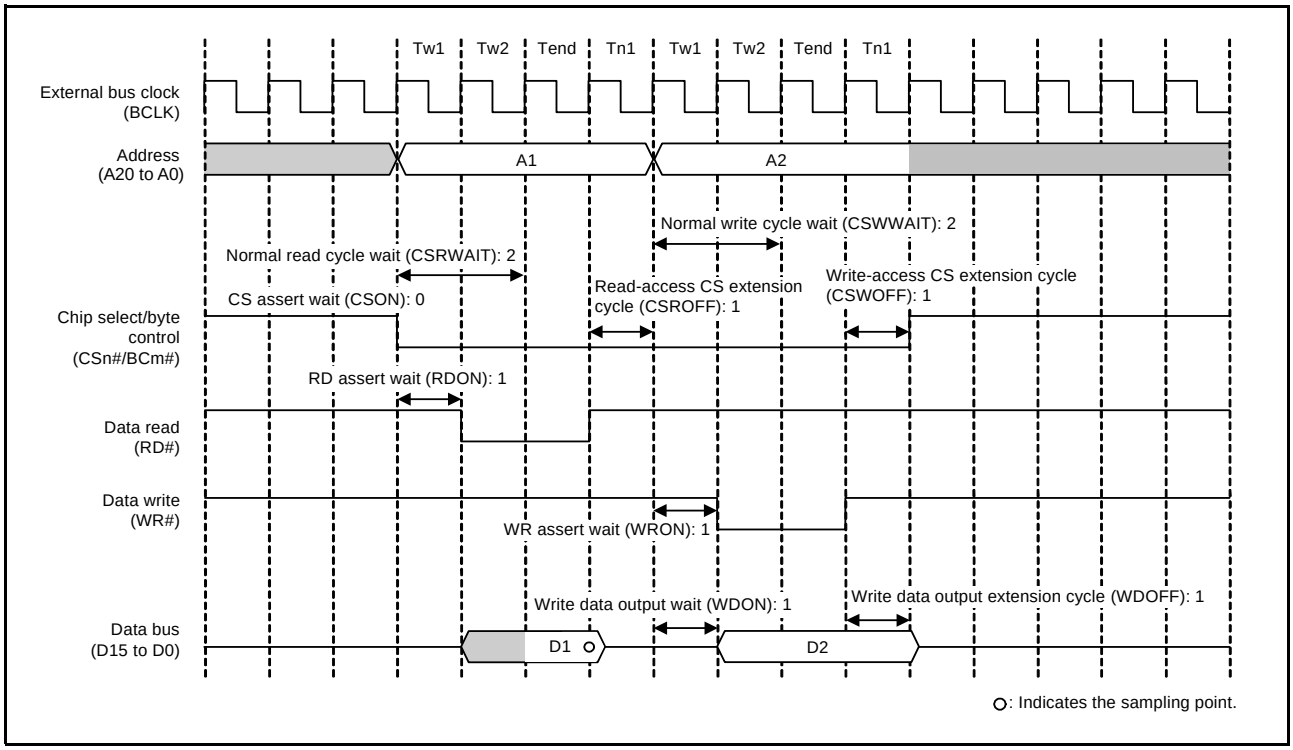


Figure 15.12 Example of Normal Access Operation (Read/Write) (n = 0 to 3, m = 0, 1)

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 15.13 and Figure 15.14 show examples of operations when two rounds of bus access are generated in response to a single request for transfer. If the recovery cycle insertion condition is satisfied, recovery cycles (step (f) above) are also inserted in the second and subsequent external bus accesses (see Figure 15.33).

The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

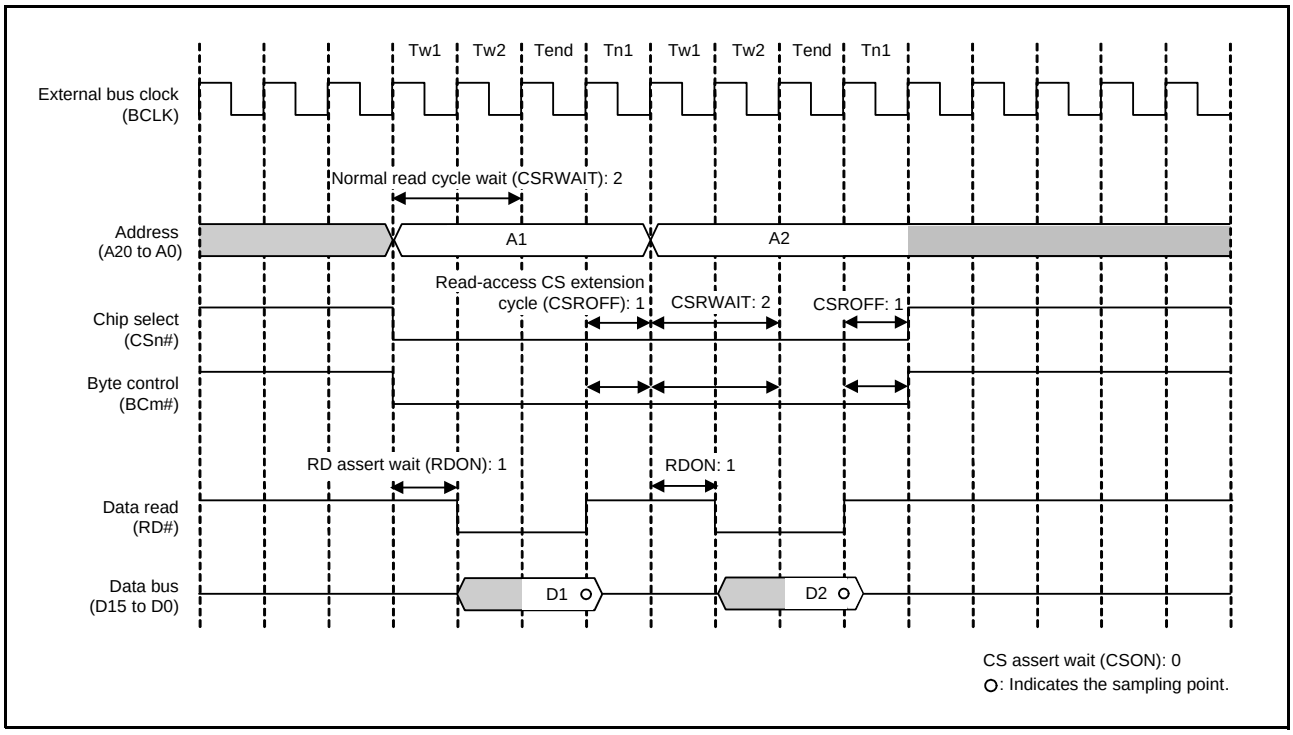


Figure 15.13 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3, m = 0, 1)

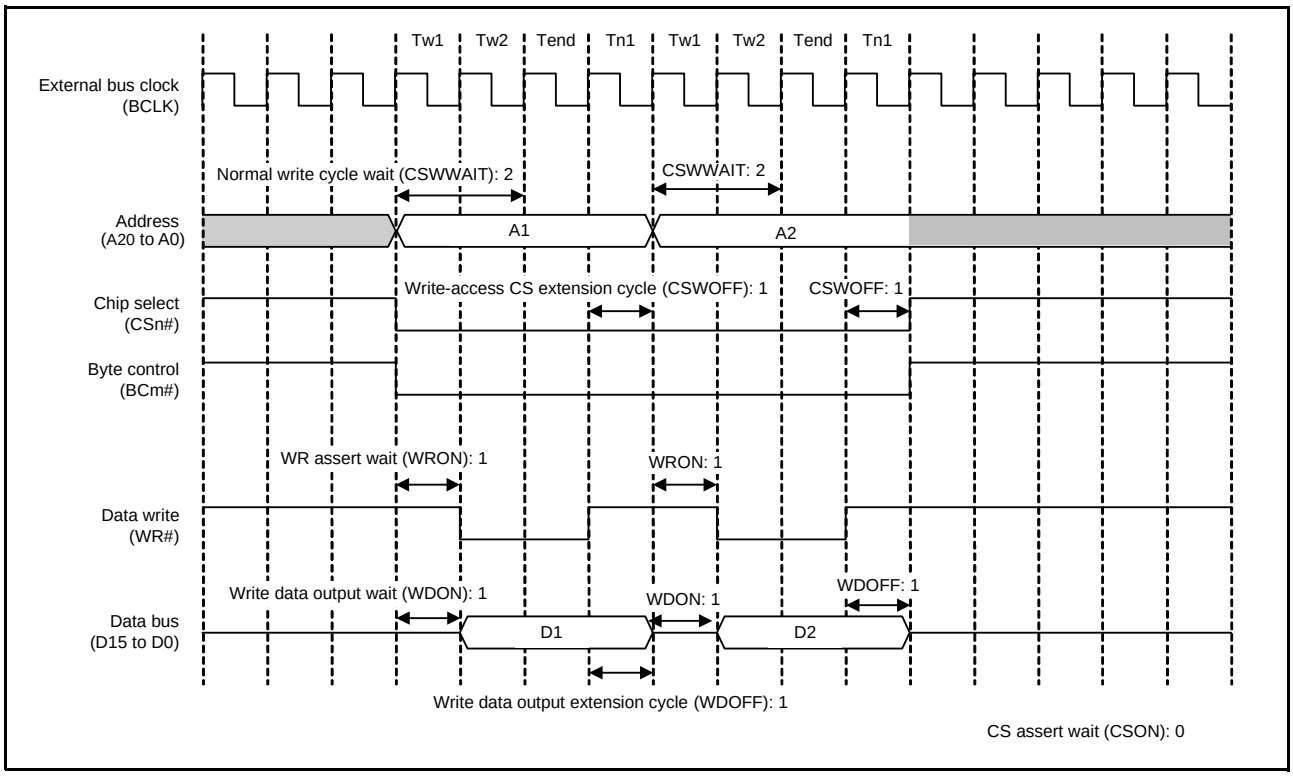


Figure 15.14 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 3, m = 0, 1)

Figure 15.15 to Figure 15.19 show examples of normal accesses made with the 1/2 BCLK selected with the BCLK pin output select bit.

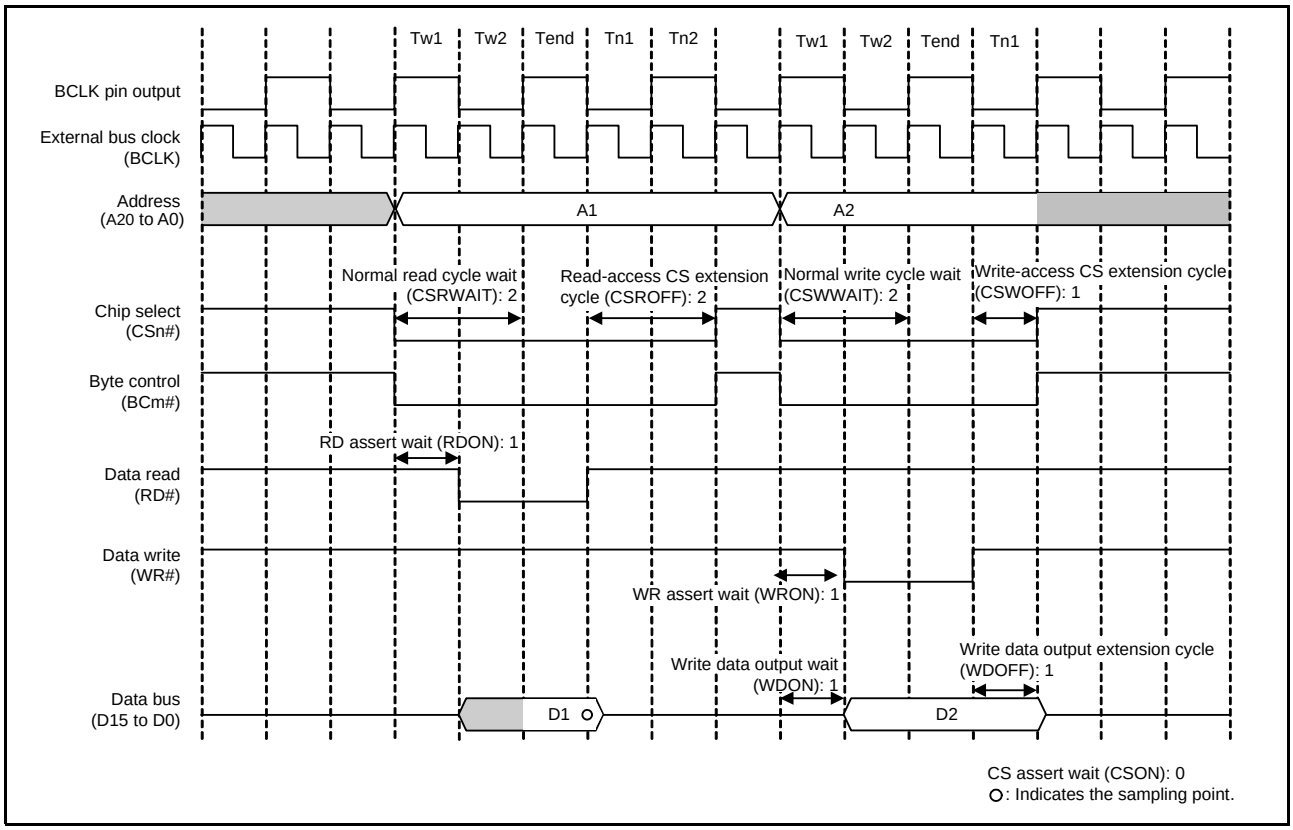


Figure 15.15 Example of Normal Access (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 3, m = 0, 1)

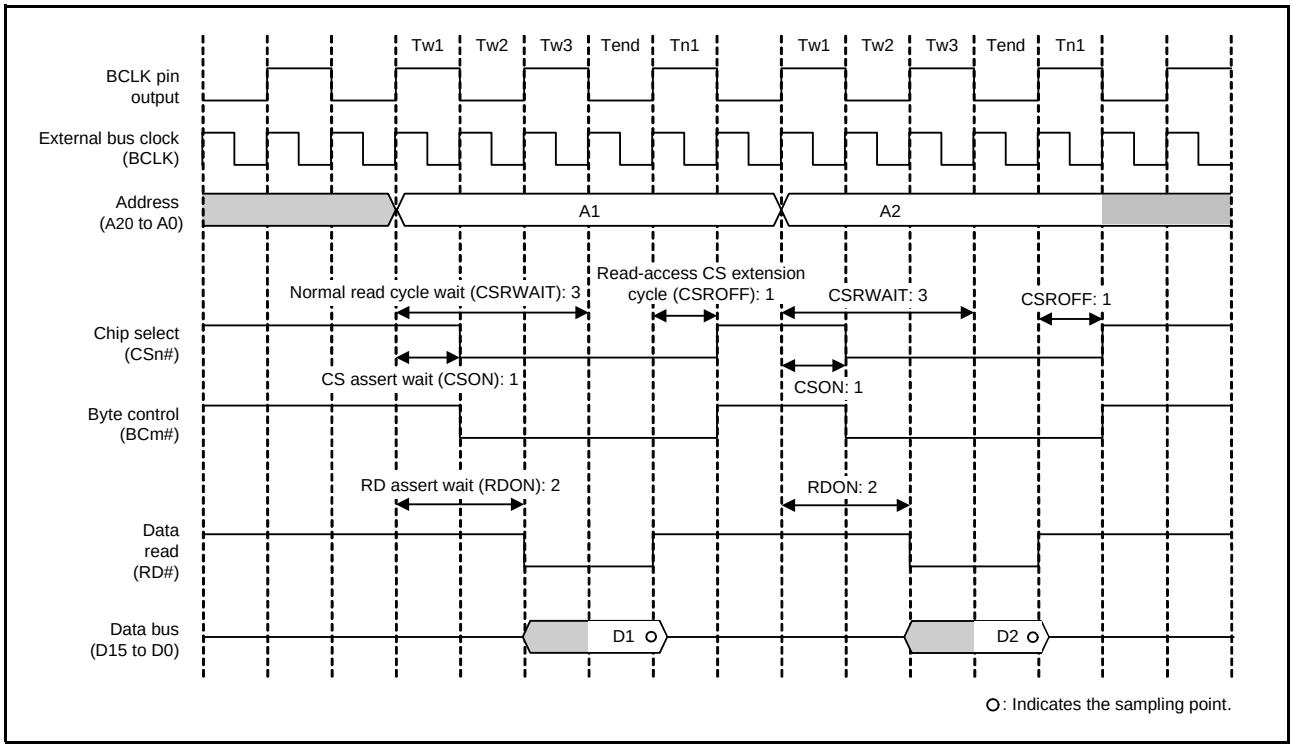


Figure 15.16 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 3, m = 0, 1)

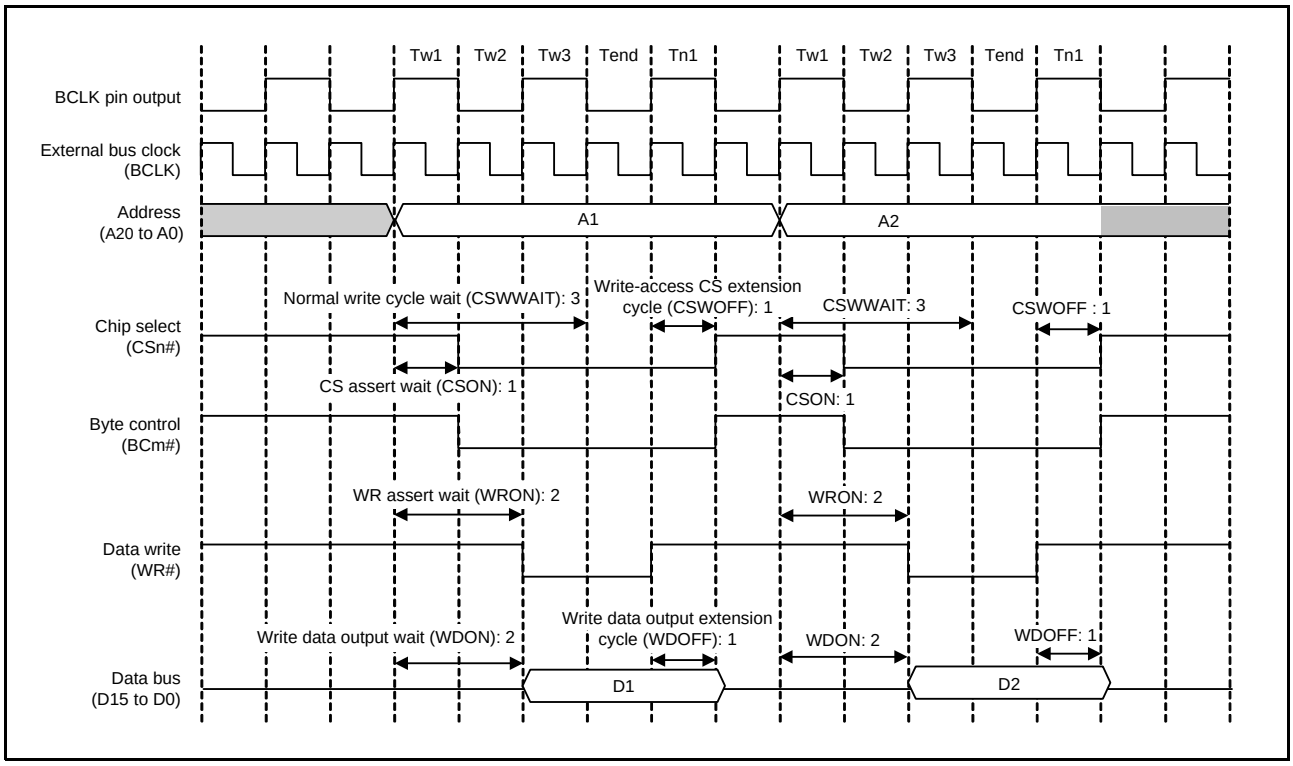


Figure 15.17 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 3, m = 0, 1)

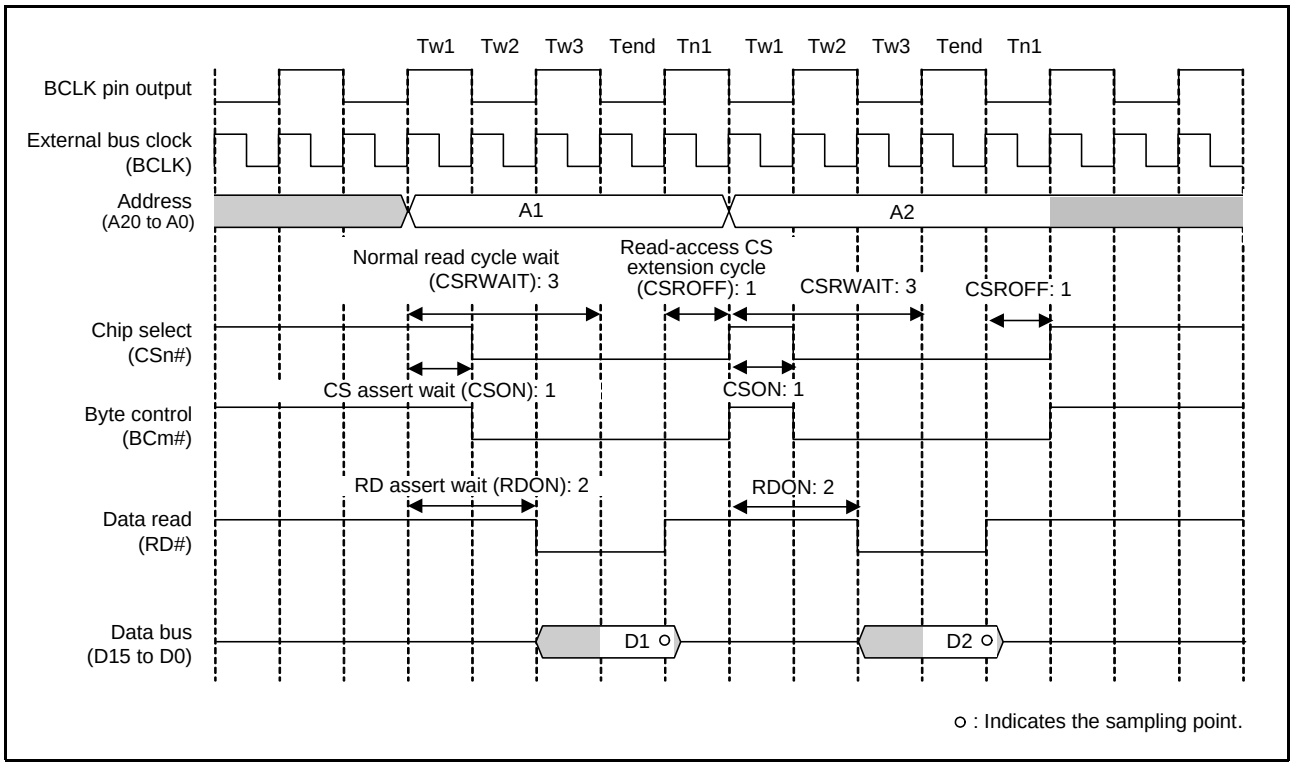


Figure 15.18 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3, m = 0, 1)

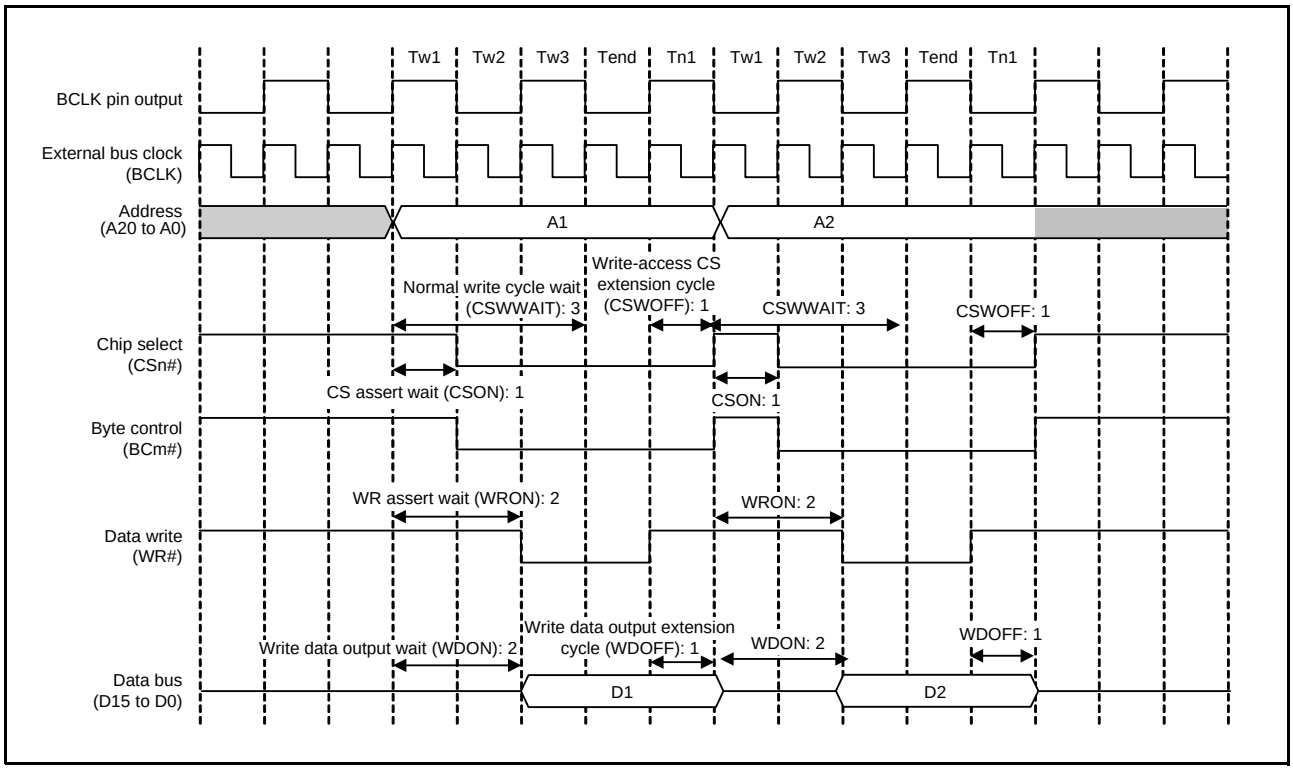


Figure 15.19 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3, m = 0, 1)

(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. See Figure 15.6 to Figure 15.9 for the conditions under which page access occurs.

Figure 15.20 and Figure 15.21 show examples of page access operations.

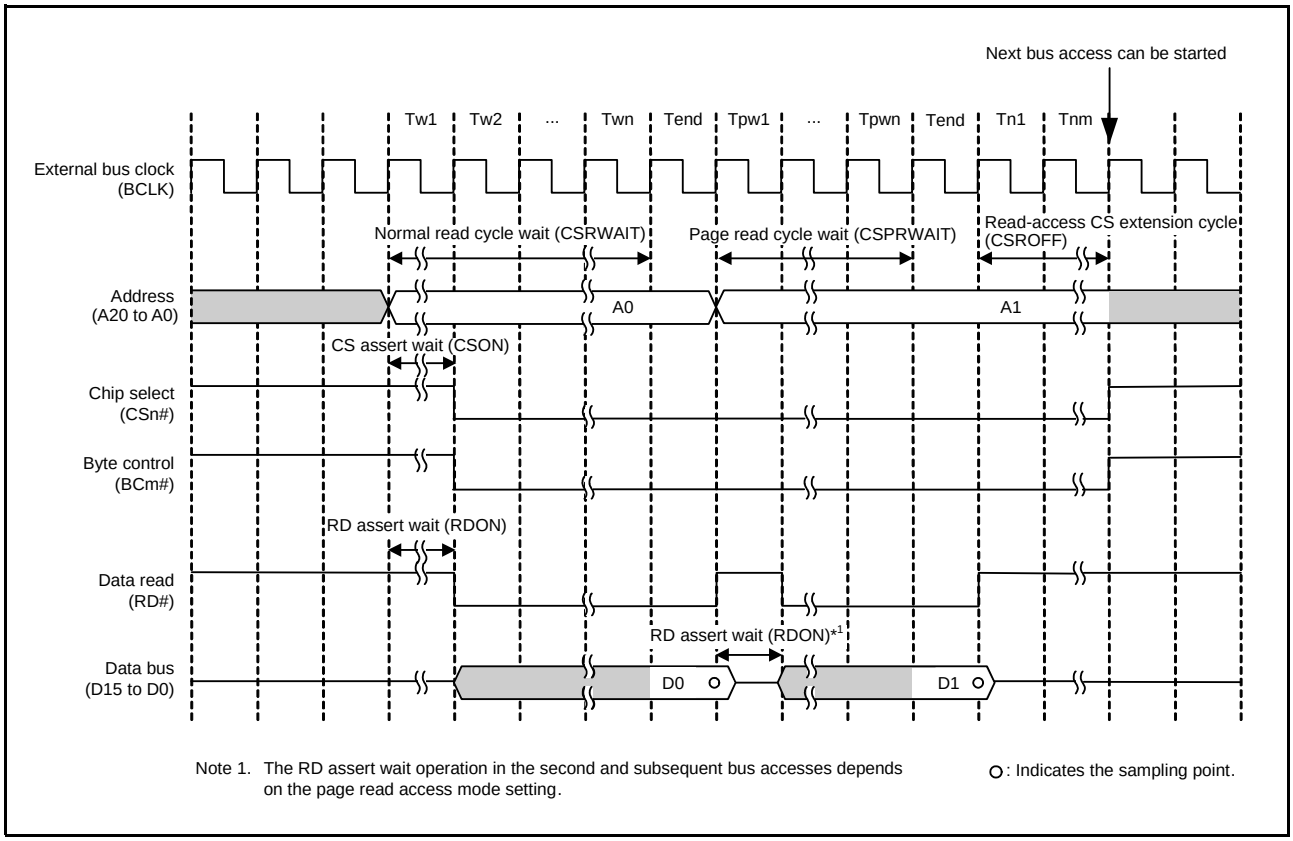


Figure 15.20 Page-Read Access Timing (n = 0 to 3, m = 0, 1)

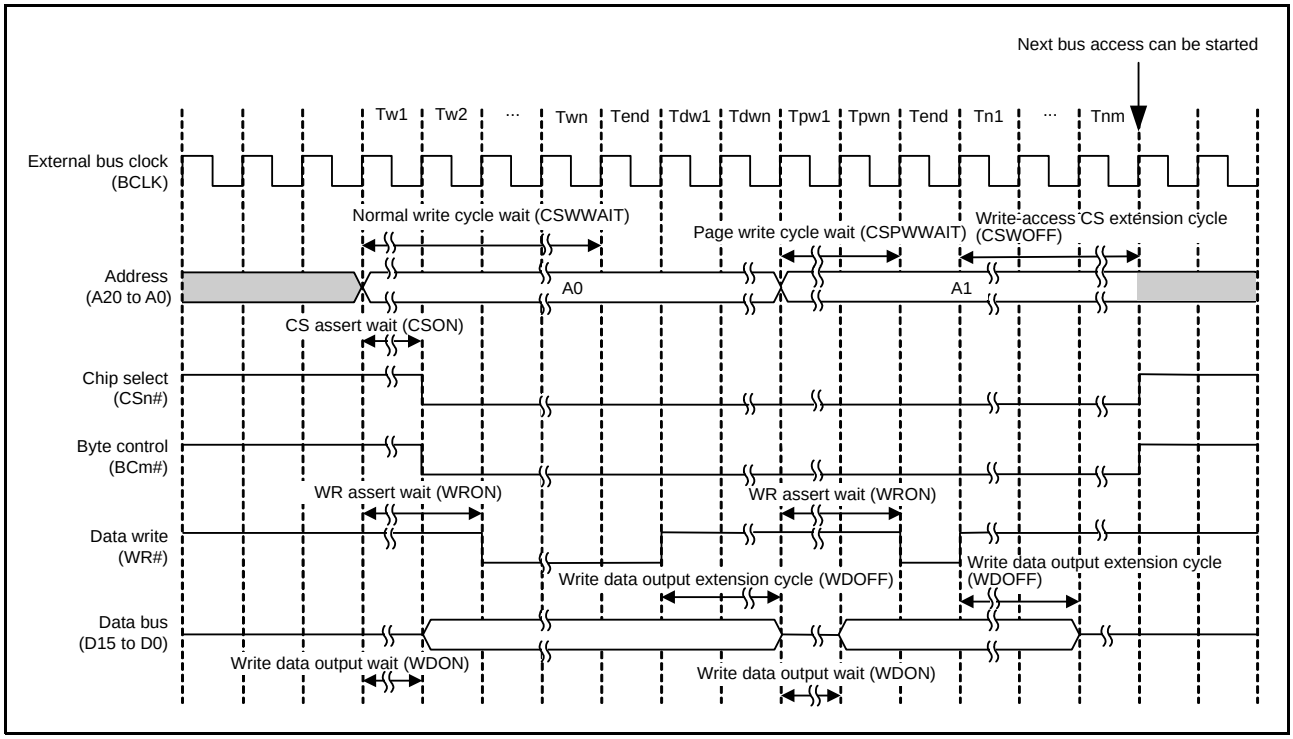


Figure 15.21 Page-Write Access Timing (n = 0 to 3, m = 0, 1)

Figure 15.22 and Figure 15.23 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.

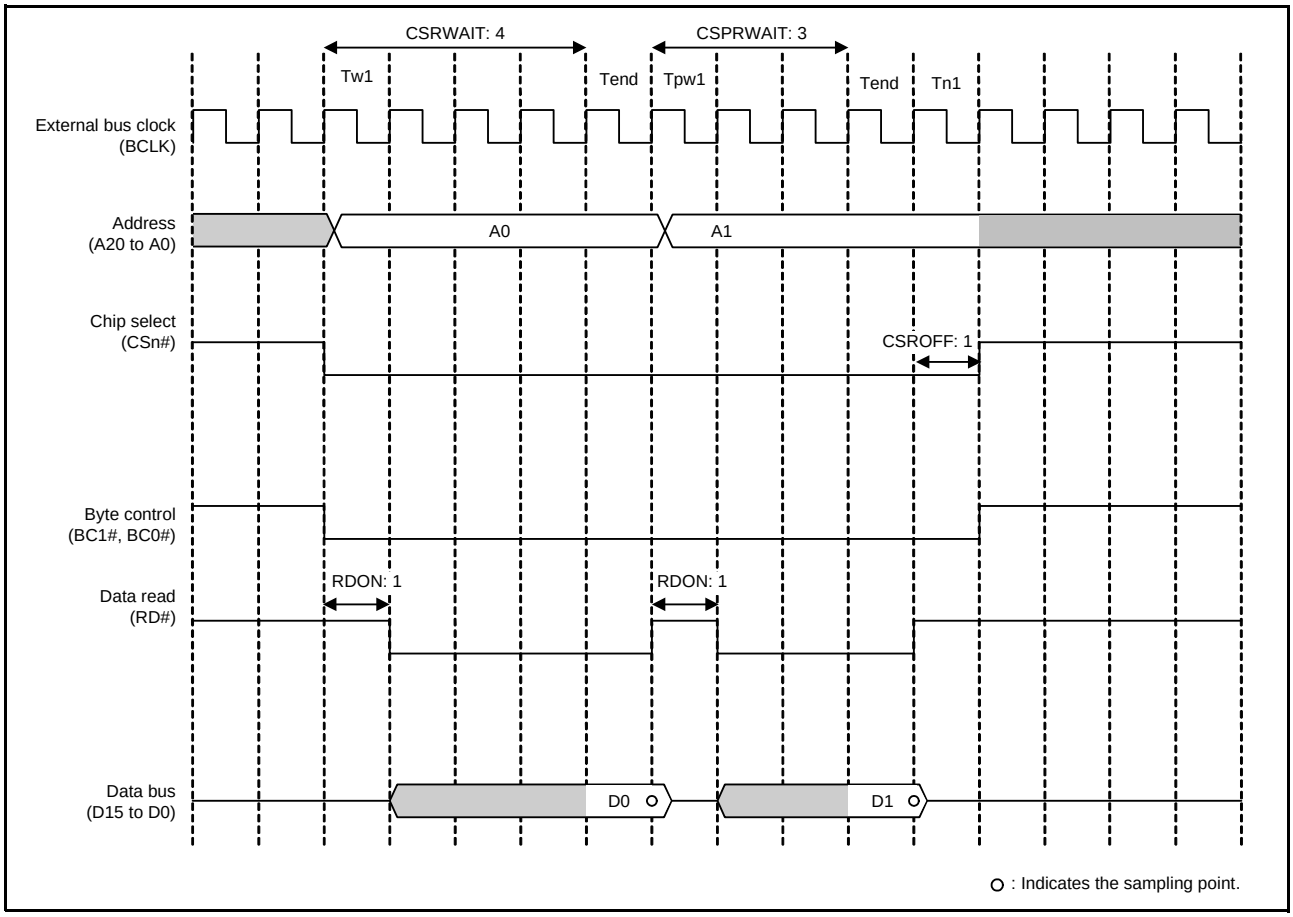


Figure 15.22 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits) (n = 0 to 3)

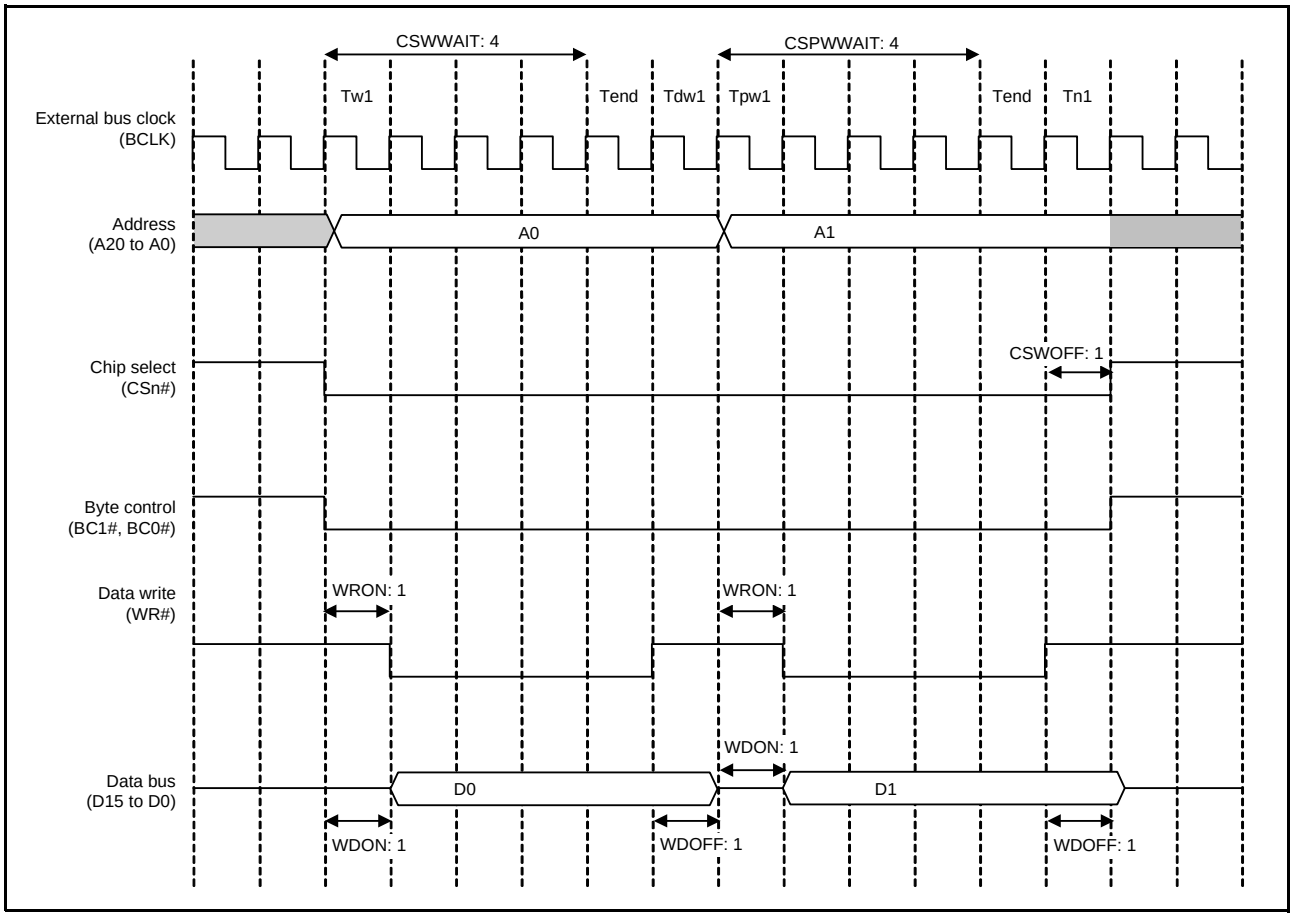


Figure 15.23 Example of Page-Write Access Operation
(when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 3)

Figure 15.24 and Figure 15.25 show examples of page access operations performed with the 1/2 BCLK selected with the BCLK pin output select bit.

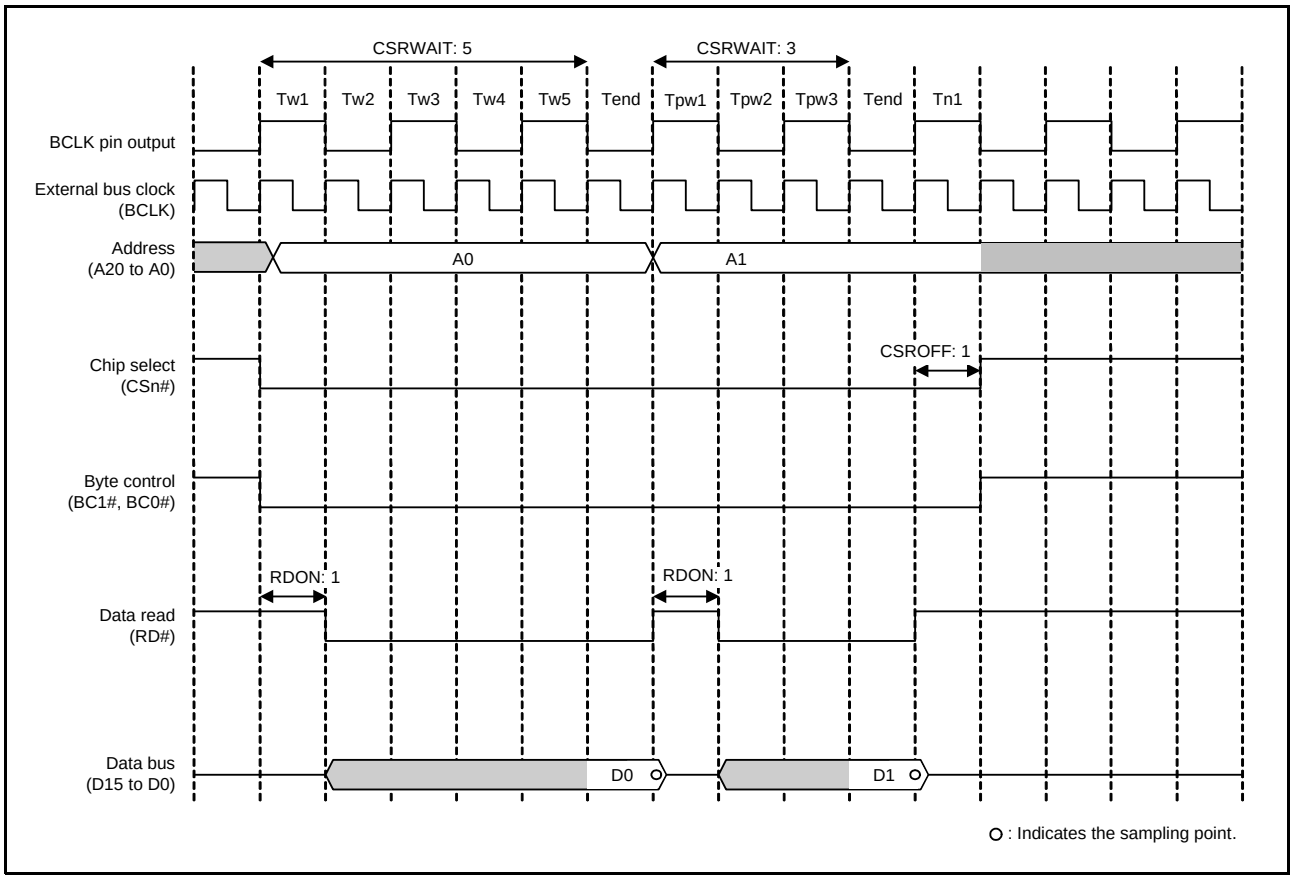


Figure 15.24 Example of Page Read Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3)

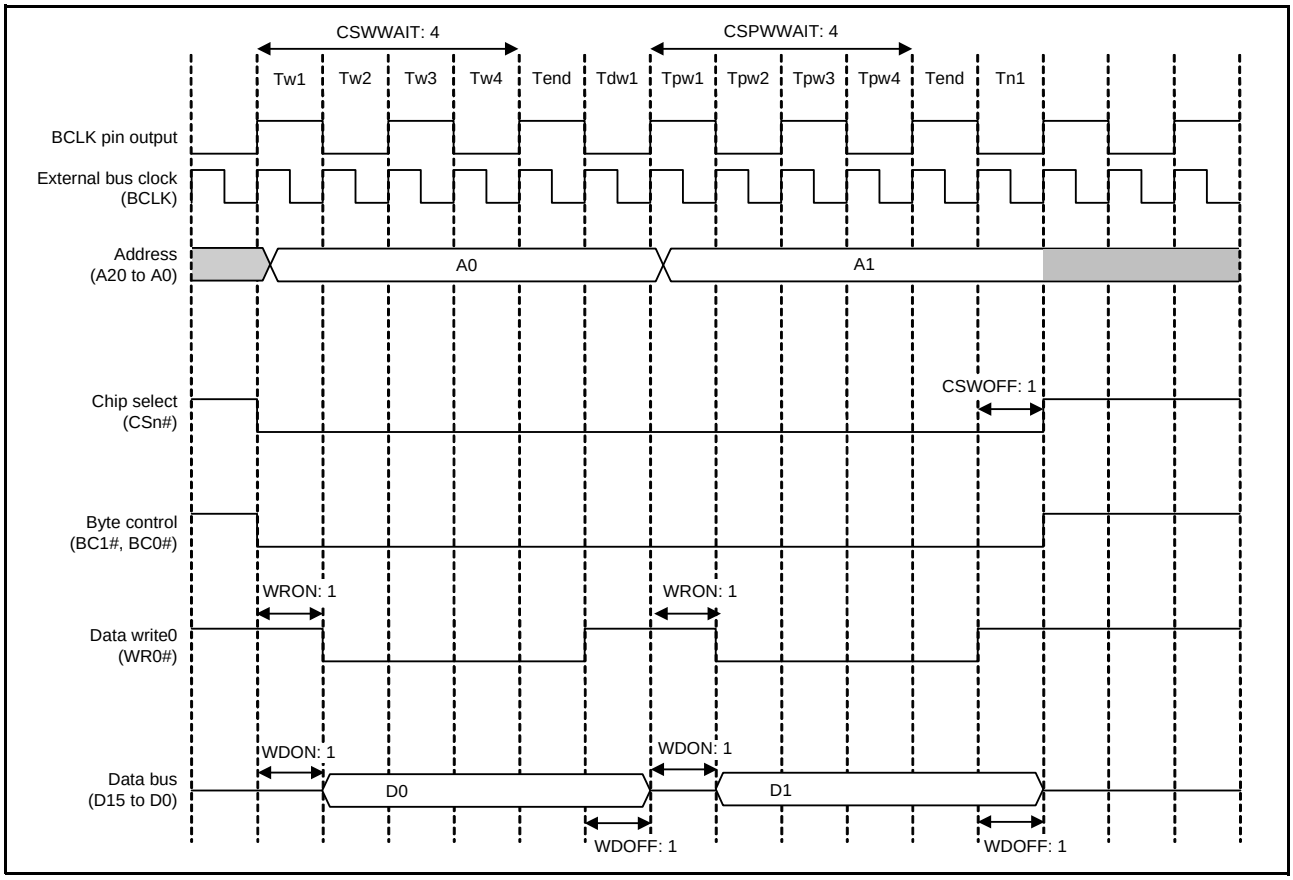


Figure 15.25 Example of Page Write Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 3)

15.5.2 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 15.26 to Figure 15.28 show examples of operations with the address/data multiplexed I/O interface.

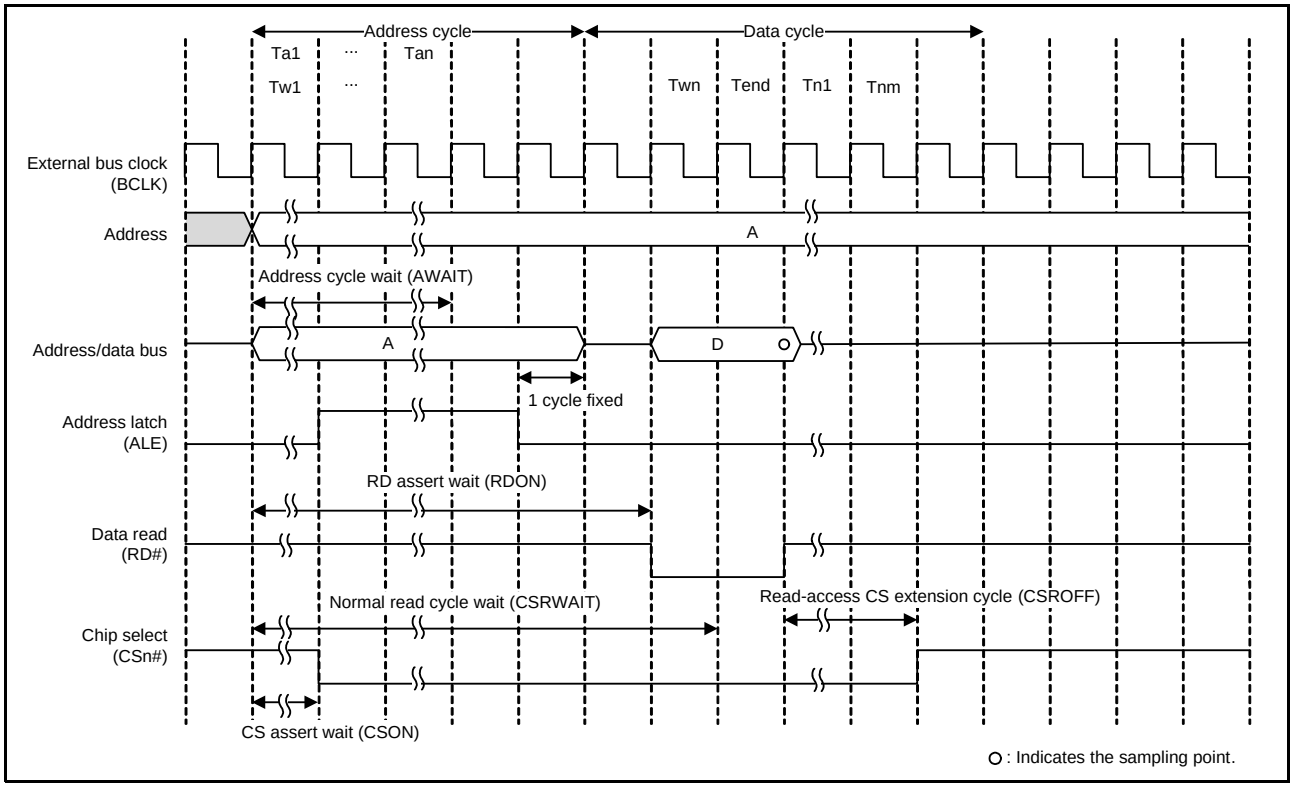


Figure 15.26 Example of Read Access Operation with Address/Data Multiplexed I/O Interface

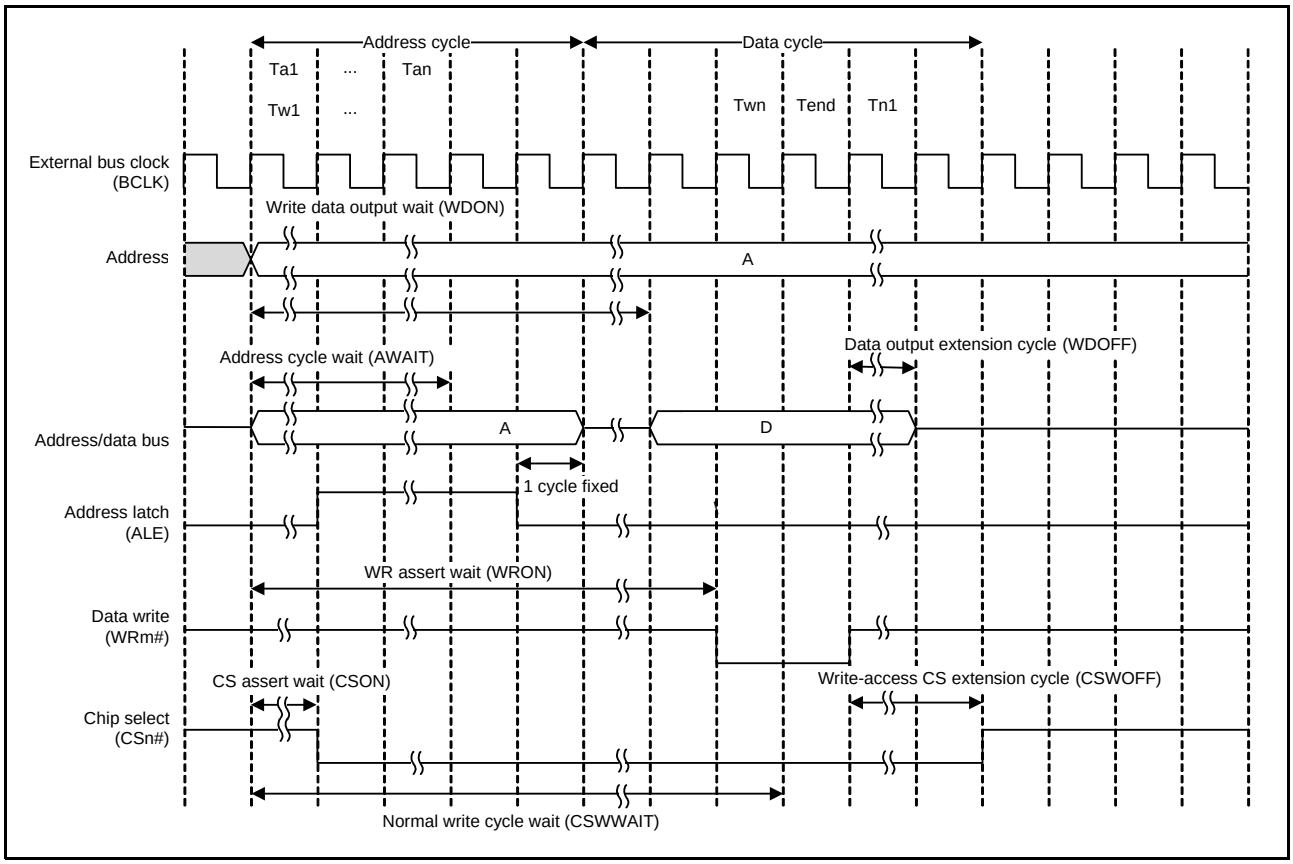


Figure 15.27 Example of Write Access Operation with Address/Data Multiplexed I/O Interface (m = 0, 1)

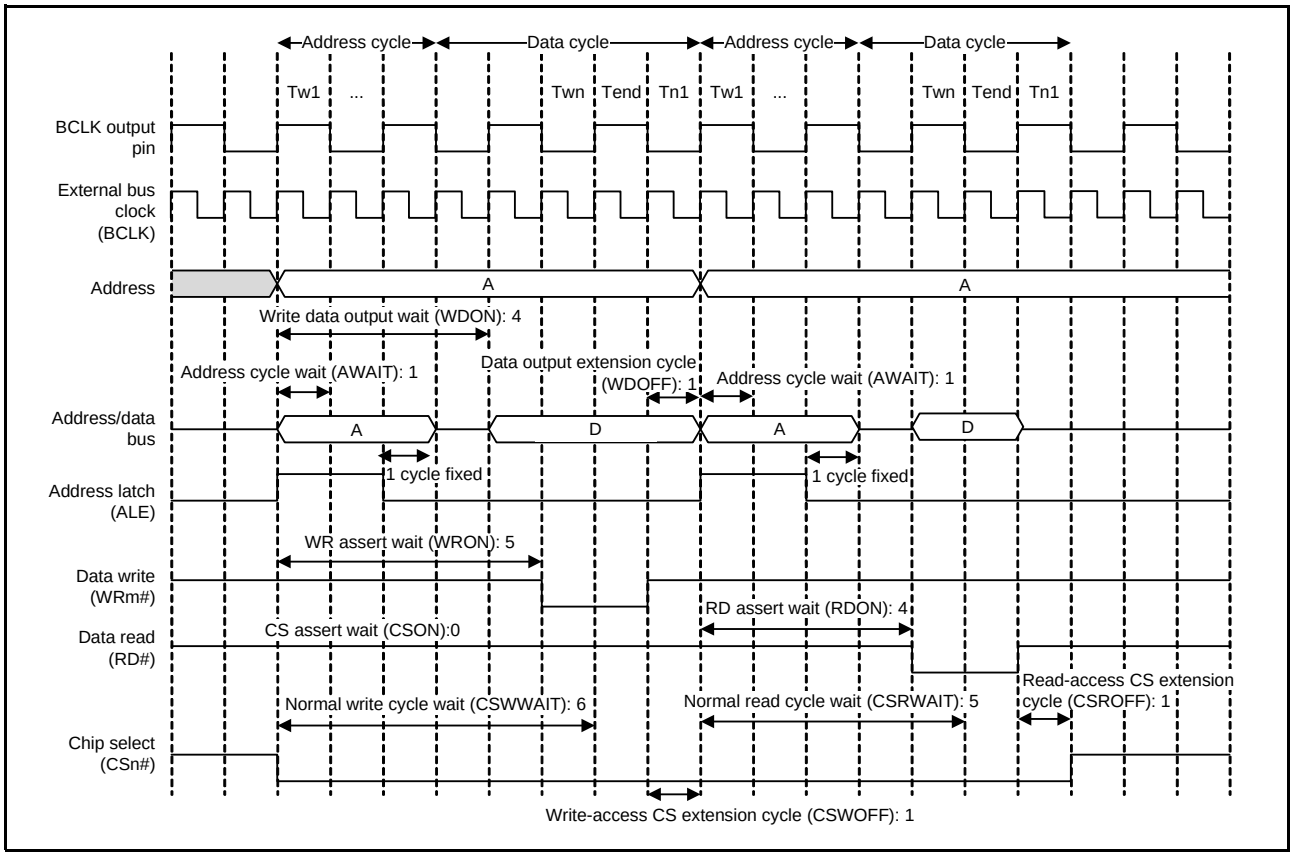


Figure 15.28 Example of Bus Timing with Address/Data Multiplexed I/O Interface (m = 0, 1)

15.5.3 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 15.29 and Figure 15.30 show examples of external wait insertion timing with the separate bus interface.

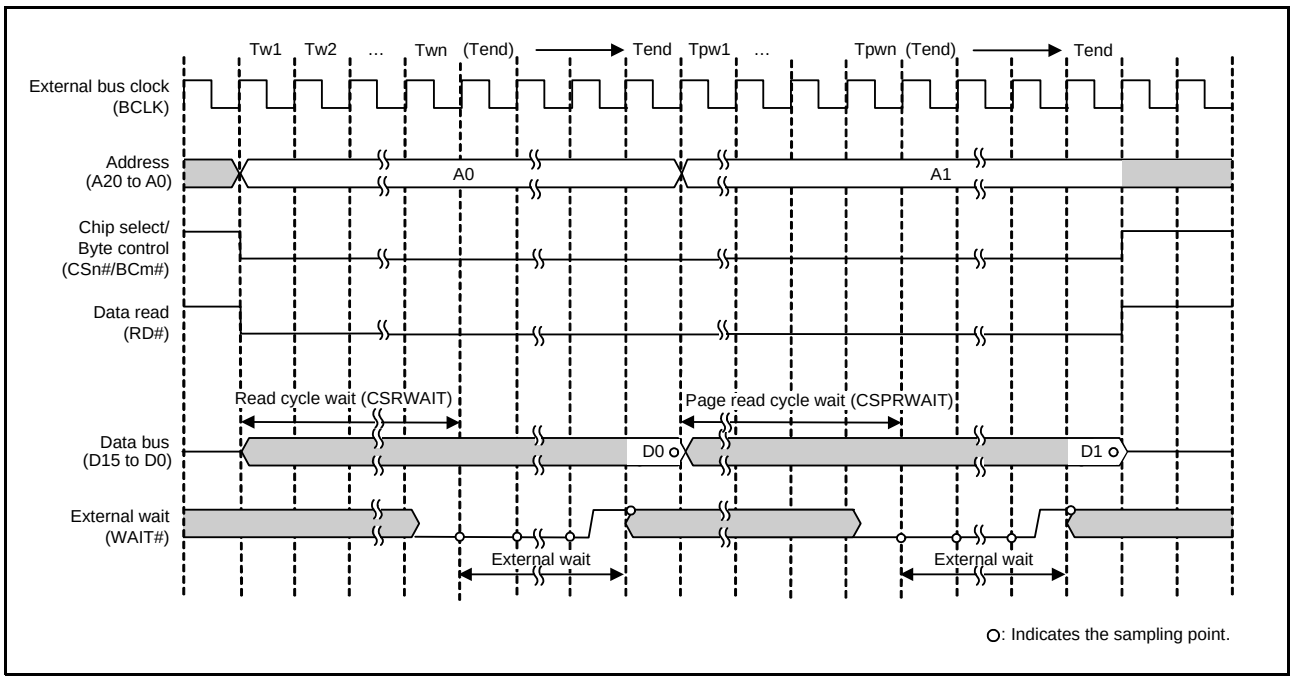


Figure 15.29 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space) (n = 0 to 3, m = 0, 1)

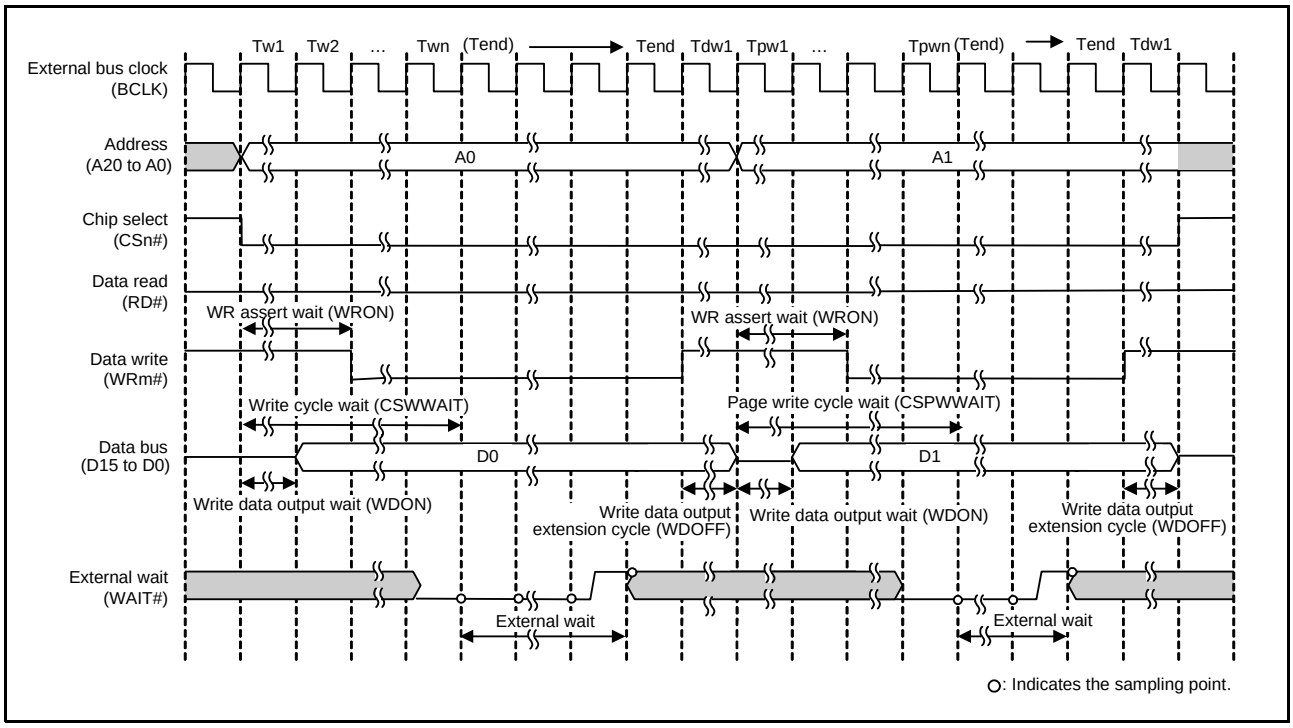


Figure 15.30 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode) (n = 0 to 3, m = 0, 1)

(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 15.31 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

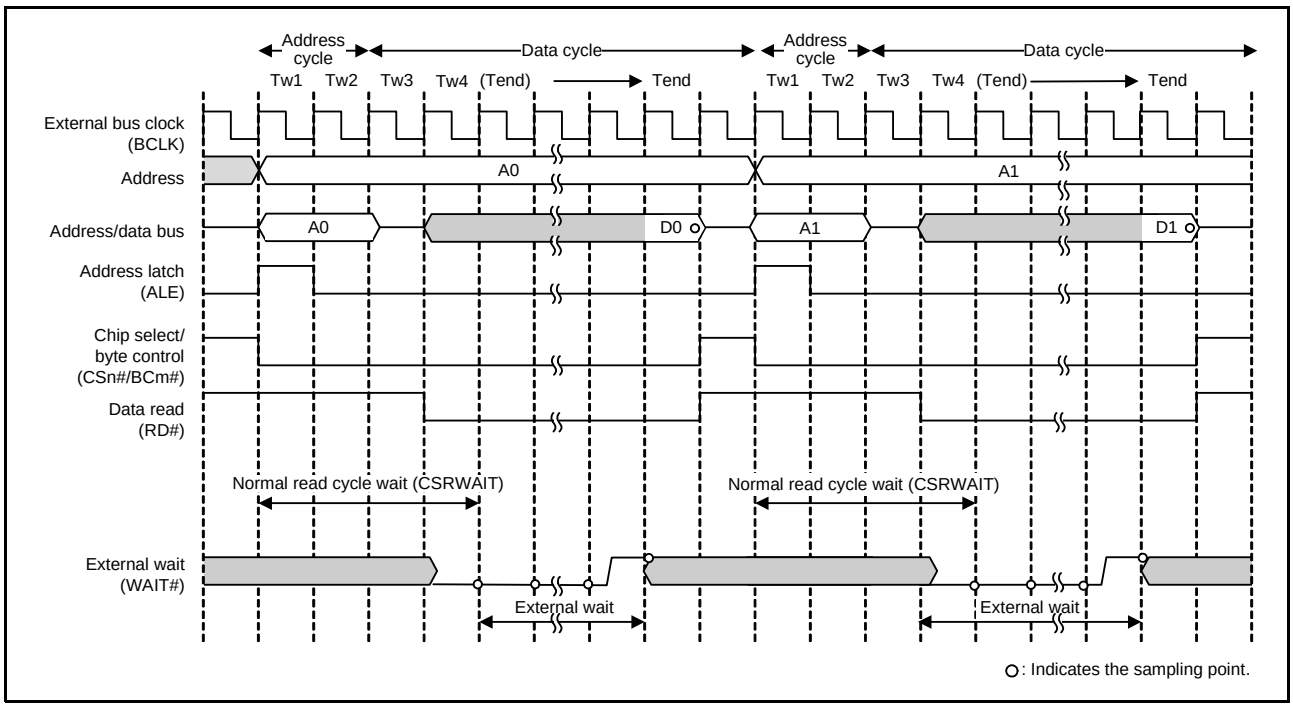


Figure 15.31 Example of External Wait Insertion Timing with Address/Data Multiplexed I/O Interface (m = 0, 1)

15.5.4 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 3) 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 15.34).
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 15.32 to Figure 15.34 show examples of recovery cycle insertion with the separate bus interface.

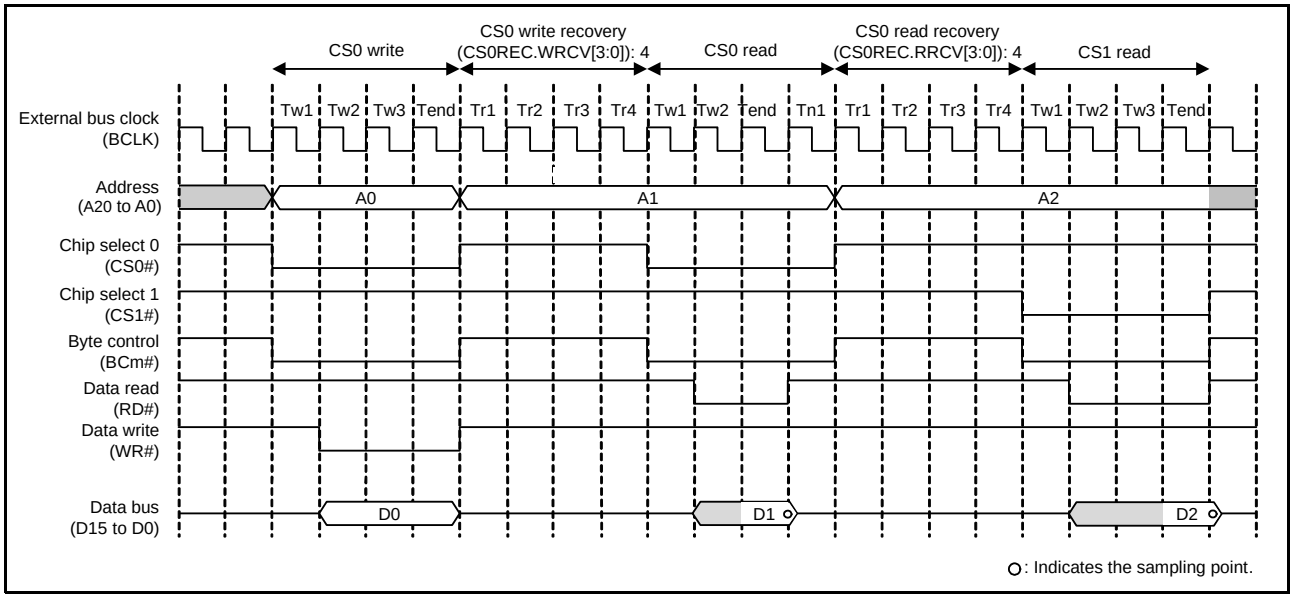


Figure 15.32 Example of Recovery Cycle Insertion with Separate Bus Interface (m = 0, 1)

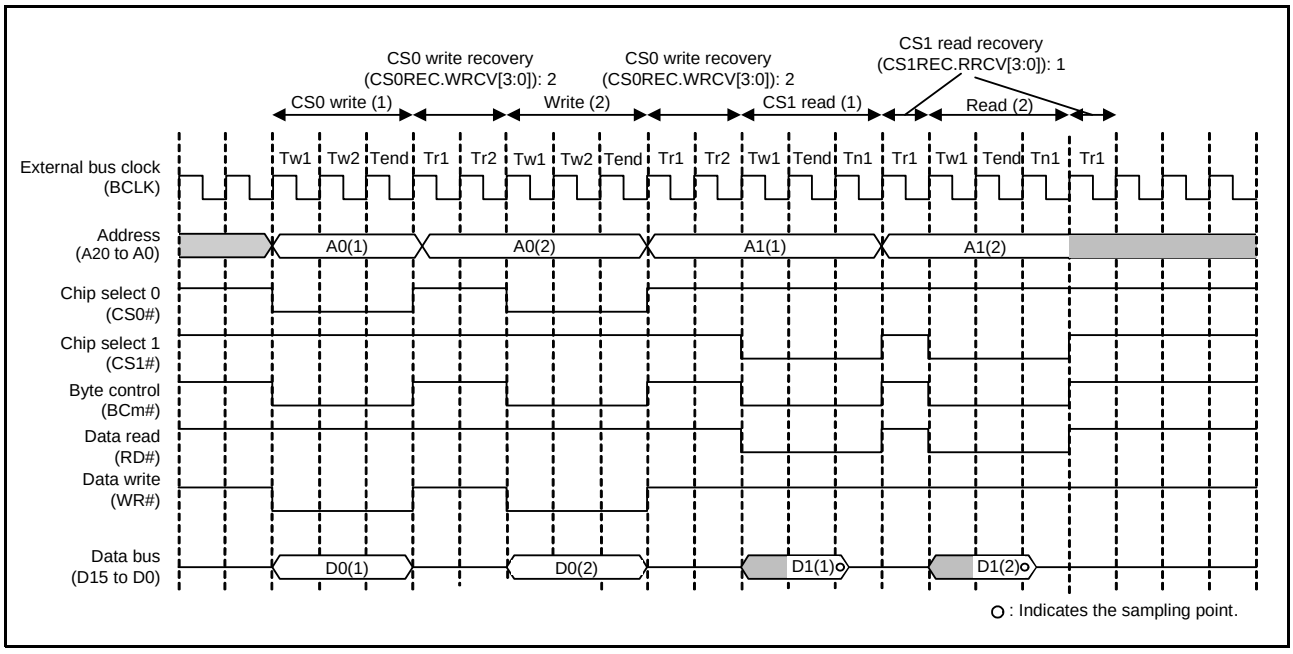


Figure 15.33 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0, 1)

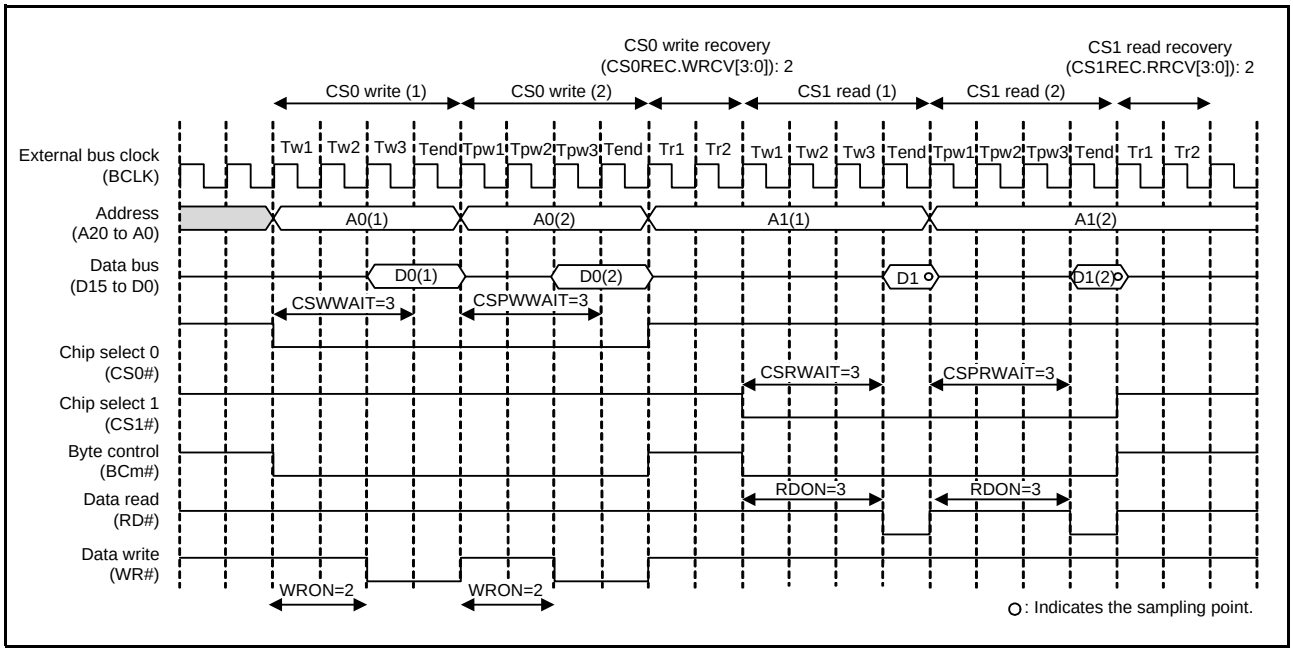


Figure 15.34 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0, 1)

Figure 15.35 shows examples of operations when the BCLK pin output selection bits are set for frequency-division of BCLK by 2.

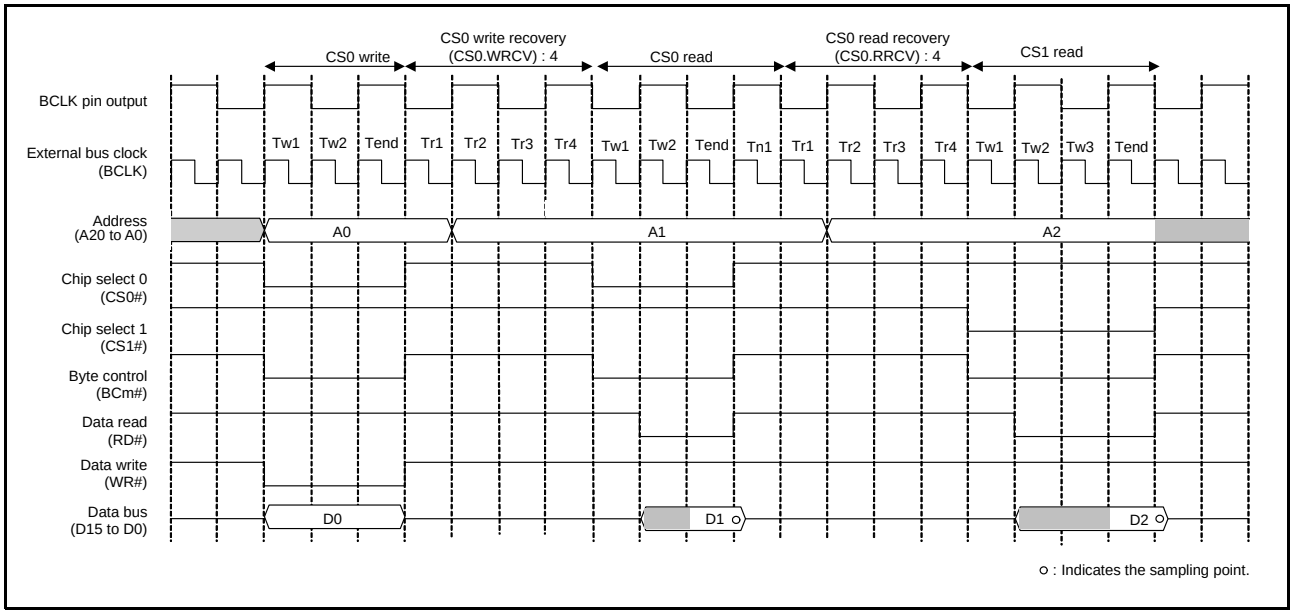


Figure 15.35 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access through a Separate Bus Interface) (m = 0, 1)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 15.36 and Figure 15.37 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

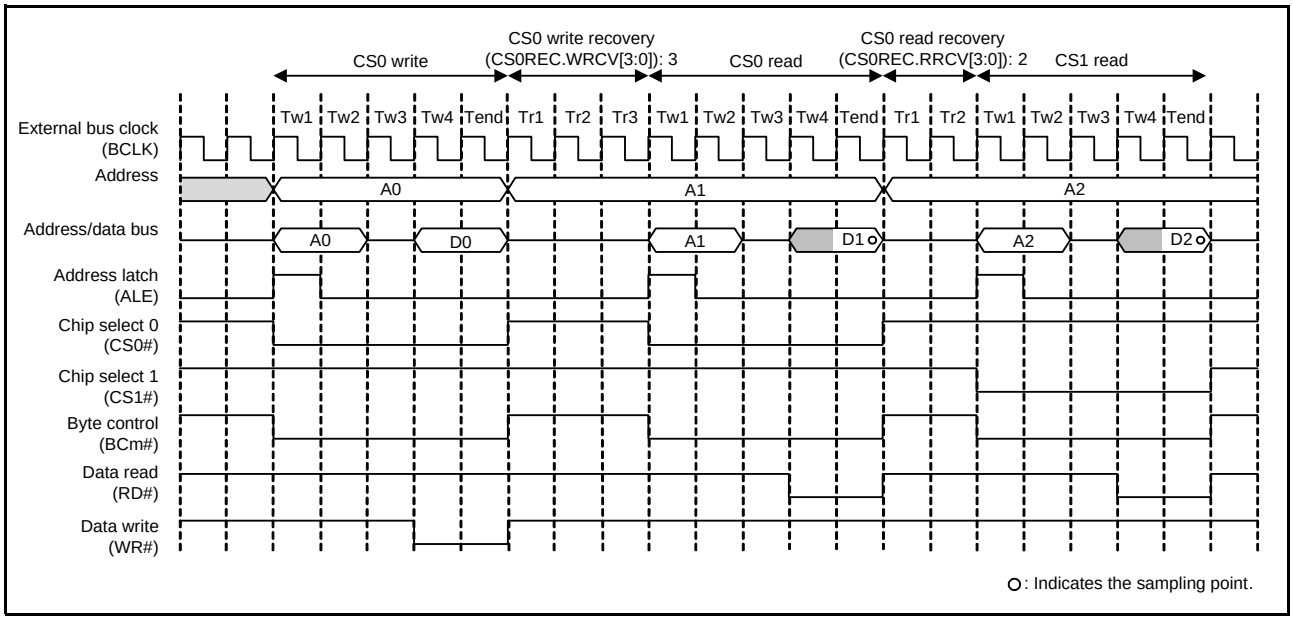


Figure 15.36 Example of Recovery Cycle Insertion with Address/Data Multiplexed I/O Interface (m = 0, 1)

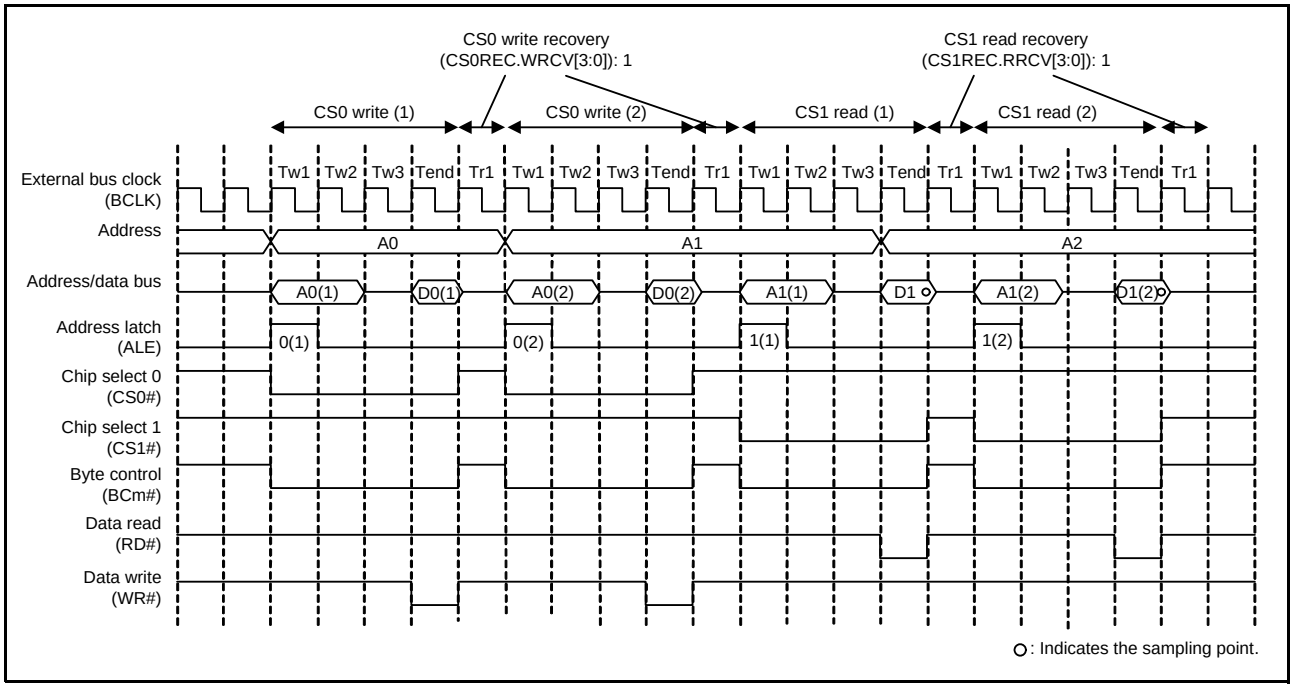


Figure 15.37 Example of Recovery Cycle Insertion When a Bus Access is Split with Address/Data Multiplexed I/O Interface (m = 0, 1)

15.5.5 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.

15.5.6 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 15.38 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.

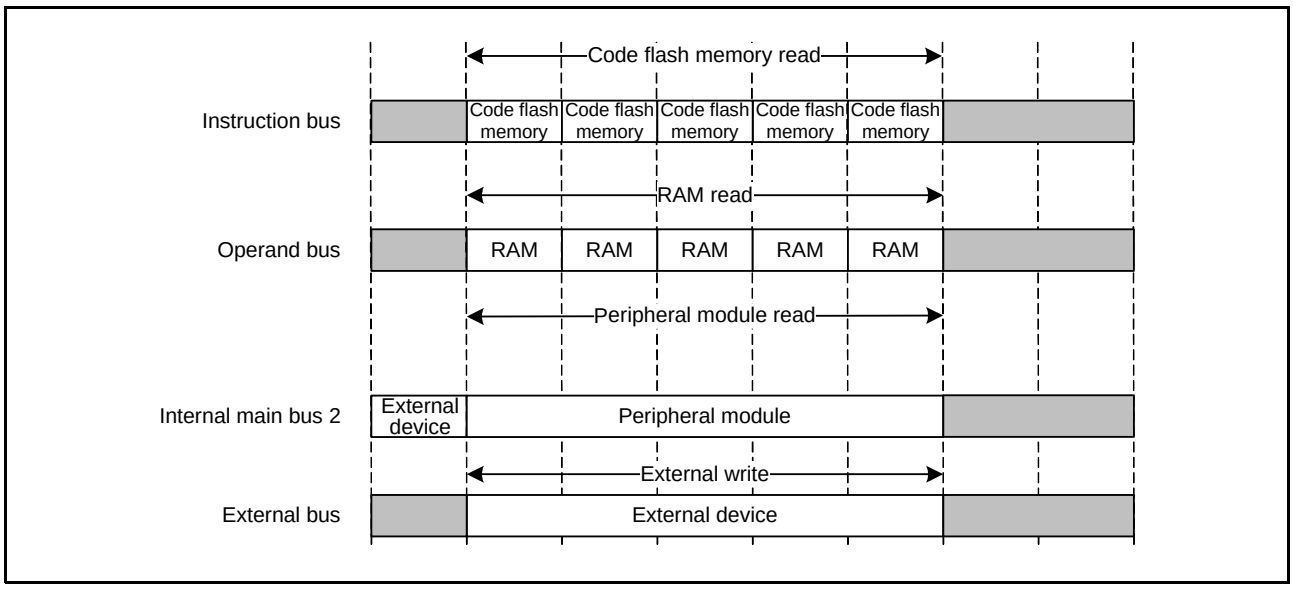


Figure 15.38 Example of Operation when the Write-Buffer Function is in Use

15.5.7 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 15.9.
  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 15.9 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 15.10 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access	
Reading	**Writing**
CSON[2:0] ≤ CSRWAIT	CSON[2:0] ≤ CSWWAIT
RDON[2:0] ≤ CSRWAIT	WRON[2:0] ≤ CSWWAIT
CSON[2:0] ≤ RDON	WDON[2:0] ≤ CSWWAIT
AWAIT[1:0] + 2 ≤ RDON	WDOFF[2:0] ≤ CSWOFF
CSON[2:0] ≤ AWAIT	WDON[2:0] ≤ WRON
	CSON[2:0] ≤ WRON
	AWAIT[1:0] + 2 ≤ WRON
	AWAIT[1:0] + 2 ≤ WDON
	CSON[2:0] ≤ AWAIT

(3) Limitation when a Pin is Multiplexed between A0 and BC0# Functions

When the A0 and BC0# pin functions share the same pin, setting the single write strobe mode is prohibited in the 8-bit bus space; otherwise the operation is not guaranteed.

(4) Limitations when 1/2 BCLK is Selected with BCLK Pin Output Select Bit

When 1/2 BCLK is selected through the BCLK pin output select bit, the external bus access cycle starts at the rising edge of the BCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle may start at the falling edge of the BCLK pin output depending on the wait cycle settings. Make appropriate register settings according to the specifications of the device to be connected.

(5) 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.

(6) 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.

(7) 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.

15.6 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.

15.6.1 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.

15.6.1.1 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)
- 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 15.11.

15.6.1.2 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 CS3): Bus access is not completed (the WAIT# signal is not negated) within 768 external bus clock (BCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.
- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access. In this MCU, a timeout error does not occur.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4): Bus access is not completed within 768 peripheral module clock (PCLKA) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKA cycles.
- 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.

15.6.2 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.

15.6.3 Conditions Leading to Bus Errors

Table 15.11 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 15.11 Types of Bus Errors

Address	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	Code Flash Memory		Code Flash Memory		Code Flash Memory	
	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled
0000 0000h to 0007 FFFFh	Memory bus 1		—		—	
0008 0000h to 0008 7FFFh	Internal peripheral bus 1		—		—	
0008 8000h to 0009 FFFFh	Internal peripheral bus 2		Δ		—	
000A 0000h to 000B FFFFh	Internal peripheral bus 3		Δ		—	
000C 0000h to 000D FFFFh	Internal peripheral bus 4		Δ		✓	
000E 0000h to 000F FFFFh	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 CS3, Reserved area)		[IA]		[TO]	
0800 0000h to 0FFF FFFFh	Reserved area		[IA]		—	—
1000 0000h to 7FFF FFFFh	Reserved area		✓		—	—
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	—	✓	—	—
FF00 0000h to FF7F FFFFh		External bus (CS0)	—	[IA]	—	[TO]
FF80 0000h to FFFF FFFFh			—		—	

—: A bus error does not result.
Δ: A bus error may or may not result.
✓: A bus error results.
[IA]: Access to this area leads to detection of a bus error if operation for this area is disabled (CSnCR.EXENB = 0; n = 0 to 3).
[TO]: Bus access not being completed within 768 cycles leads to detection of a bus error.
Note: The capacity of the RAM, data flash memory, and code flash memory differs depending on the product. For details, see section 43, RAM, section 44, Flash Memory (FLASH).

15.7 Interrupt

15.7.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 15.12 Interrupt Source

Name	Interrupt Source	DTC Trigger	DMAC Trigger
BUSERR	Illegal address access error or timeout	Not possible	Not possible

16. Memory-Protection Unit (MPU)

16.1 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 16.1 lists the specifications of the memory-protection unit, and Figure 16.1 shows a block diagram of the memory-protection unit.

Table 16.1 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.

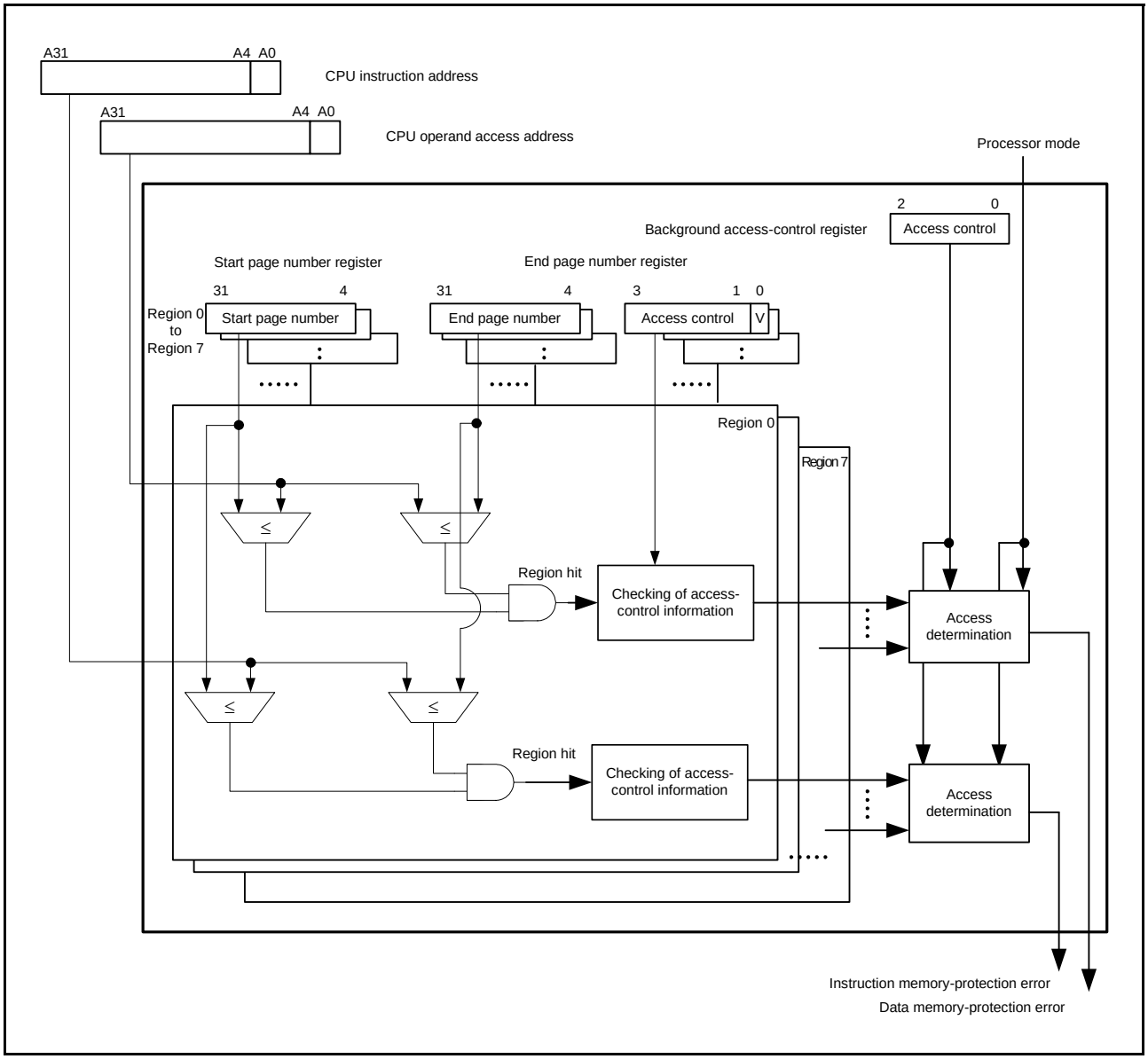


Figure 16.1 Block Diagram of the Memory-Protection Unit

16.1.1 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.

16.1.2 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.

16.1.3 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).

16.1.4 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.

16.1.5 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.

16.2 Register Descriptions

16.2.1 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.

16.2.2 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.

16.2.3 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.

16.2.4 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.

16.2.5 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.

16.2.6 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.

16.2.7 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.

16.2.8 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).

16.2.9 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.

16.2.10 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.

16.2.11 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.

16.2.12 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.

16.3 Functions

16.3.1 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.

16.3.2 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.

16.3.3 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.

16.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 16.2 shows the flow of determination in the case of data access and Figure 16.3 shows the flow of determination in the case of instruction access.

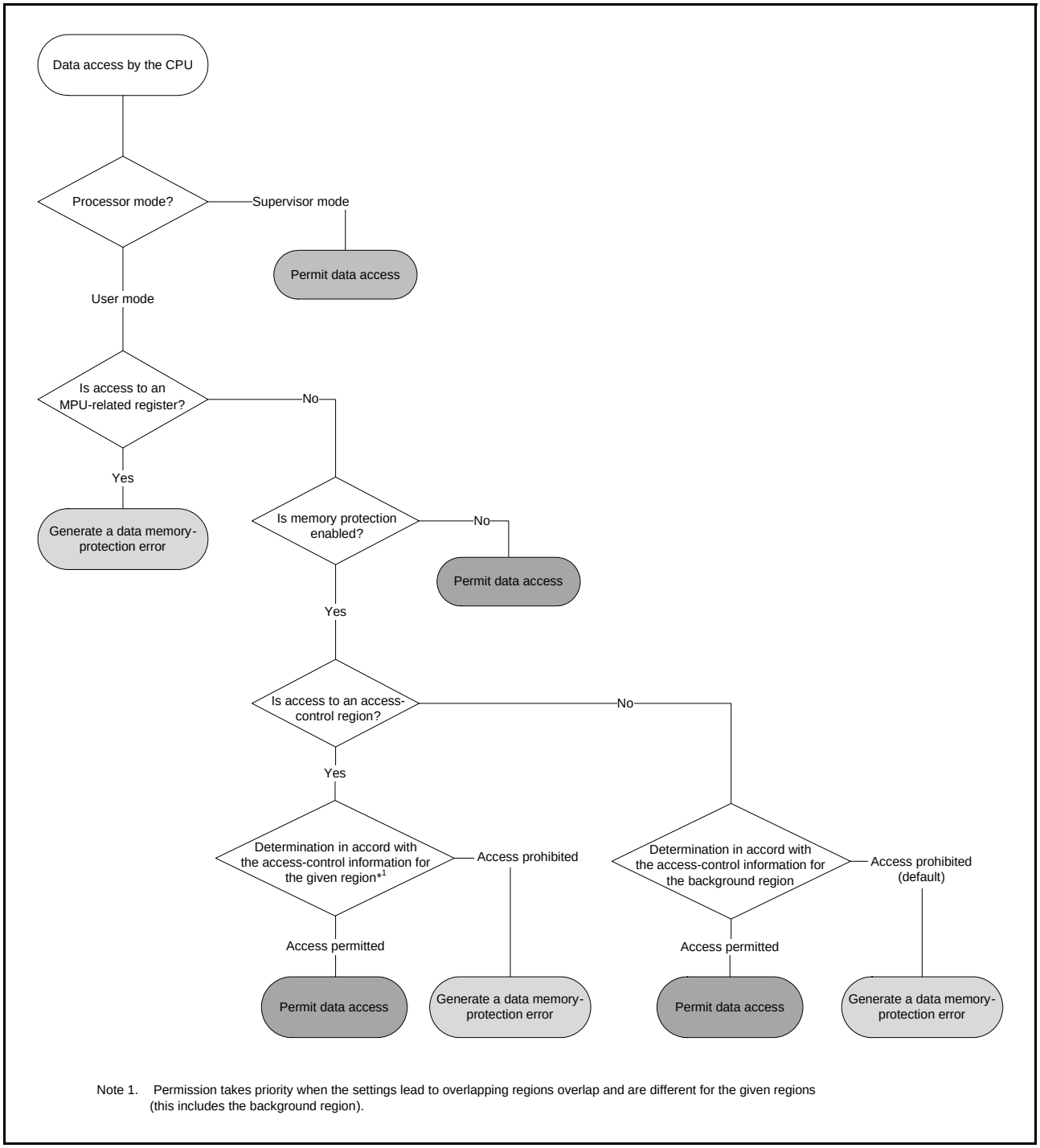
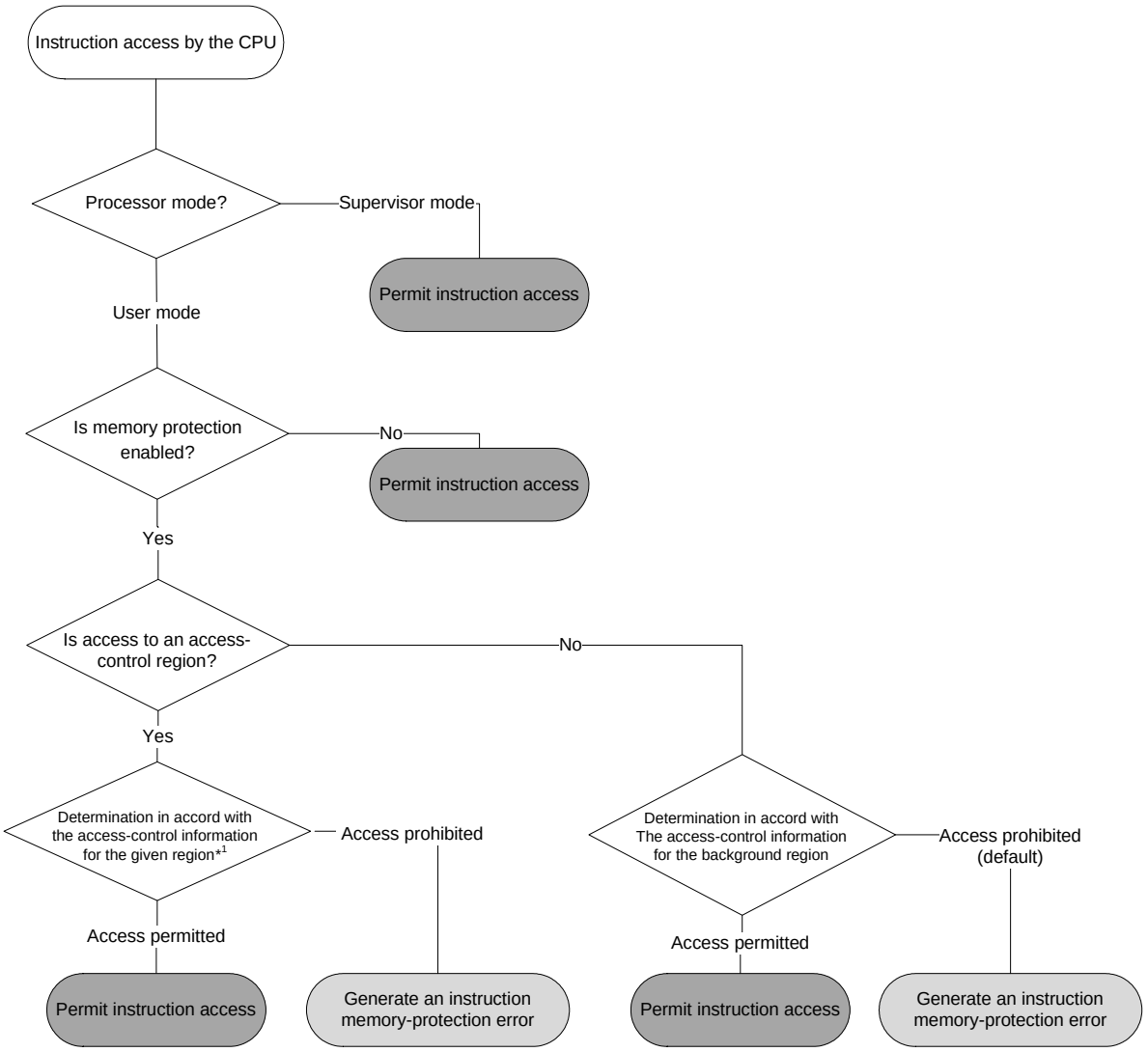


Figure 16.2 Flow of Determination for Data Access



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 16.3 Flow of Determination for Instruction Access

16.4 Procedures for Using Memory Protection

16.4.1 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).

16.4.2 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.

16.4.3 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.

16.4.4 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 13, 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.

17. DMA Controller (DMACAa)

This MCU incorporates an 8-channel direct memory access controller (DMAC).
The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

17.1 Overview

Table 17.1 lists the specifications of the DMAC, and Figure 17.1 shows a block diagram of the DMAC.

Table 17.1 Specifications of DMAC

Item		Description
Number of channels		8 (DMACm (m = 0 to 7))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		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 14.5, Interrupt Vector Table in section 14, Interrupt Controller (ICUF).

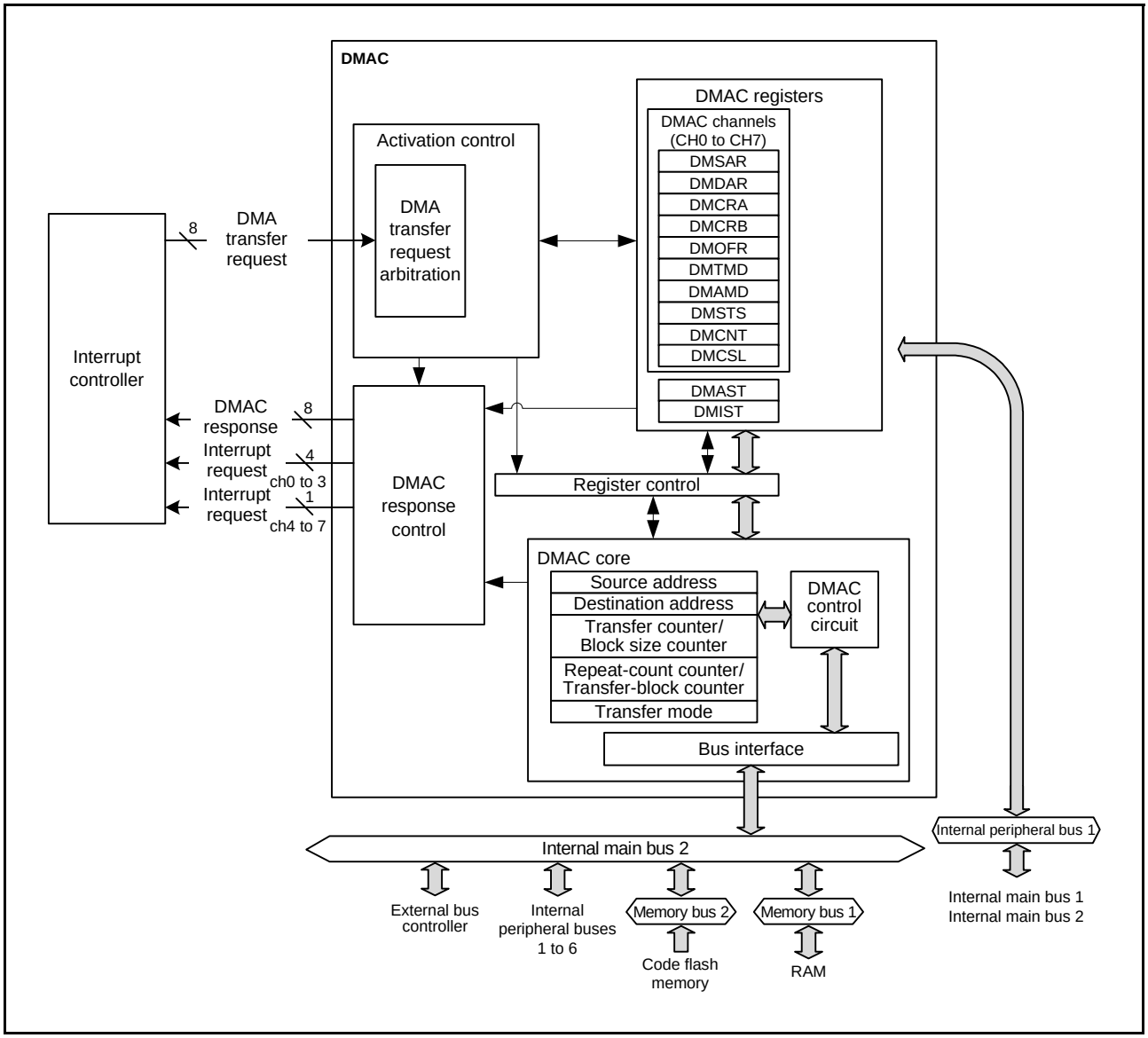
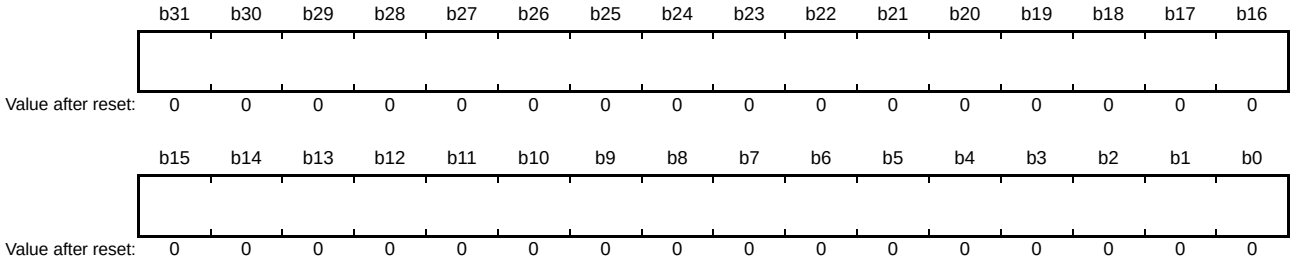


Figure 17.1 Block Diagram of DMAC

17.2 Register Descriptions

17.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h, DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h, DMAC4.DMSAR 0008 2100h, DMAC5.DMSAR 0008 2140h, DMAC6.DMSAR 0008 2180h, DMAC7.DMSAR 0008 21C0h



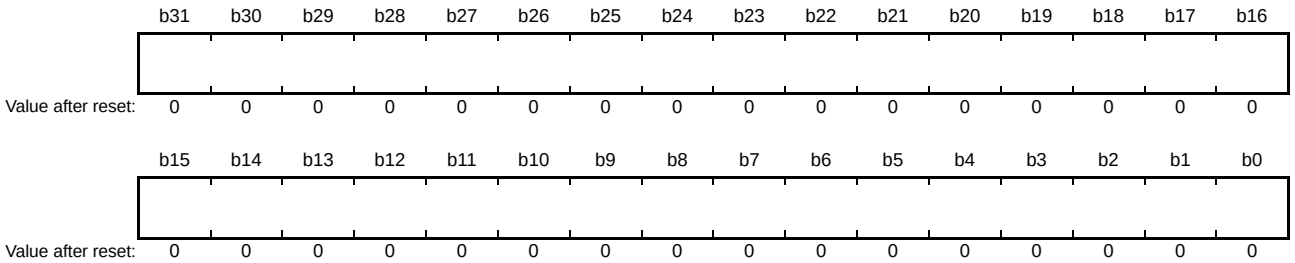
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set the DMSAR register while DMAC stops (DMAST.DMST bit = 0) or DMA transfer is disabled (DMCNT.DTE bit = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading the DMSAR register returns the extended value.

17.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h, DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h, DMAC4.DMDAR 0008 2104h, DMAC5.DMDAR 0008 2144h, DMAC6.DMDAR 0008 2184h, DMAC7.DMDAR 0008 21C4h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set the DMDAR register while DMAC stops (DMAST.DMST bit = 0) or DMA transfer is disabled (DMCNT.DTE bit = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading the DMDAR register returns the extended value.

17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h, DMAC4.DMCRA 0008 2108h, DMAC5.DMCRA 0008 2148h, DMAC6.DMCRA 0008 2188h, DMAC7.DMCRA 0008 21C8h

- Normal transfer mode

DMCRAH

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

DMCRAL

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

- Repeat transfer mode, block transfer mode

DMCRAH

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

DMCRAL

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

Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for 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.

17.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh, 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.

17.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h, DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h, 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 14.5, Interrupt Vector Table in section 14, Interrupt Controller (ICUF).

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.

17.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h, DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h, 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.

17.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h, DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h, 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 17.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, refer to Table 17.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: 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 17.2 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the 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 17.2 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 17.2 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.

17.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

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

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer). Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

17.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch, DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh, 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.

17.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh, 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.

17.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh, DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh, 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

17.2.12 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.

17.2.13 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

17.2.14 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.

17.3 Operation

17.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the 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 17.3 summarizes the register update operation in normal transfer mode, and Figure 17.2 shows the operation in normal transfer mode.

Table 17.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
DMACm.DMCRAL	Transfer 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.

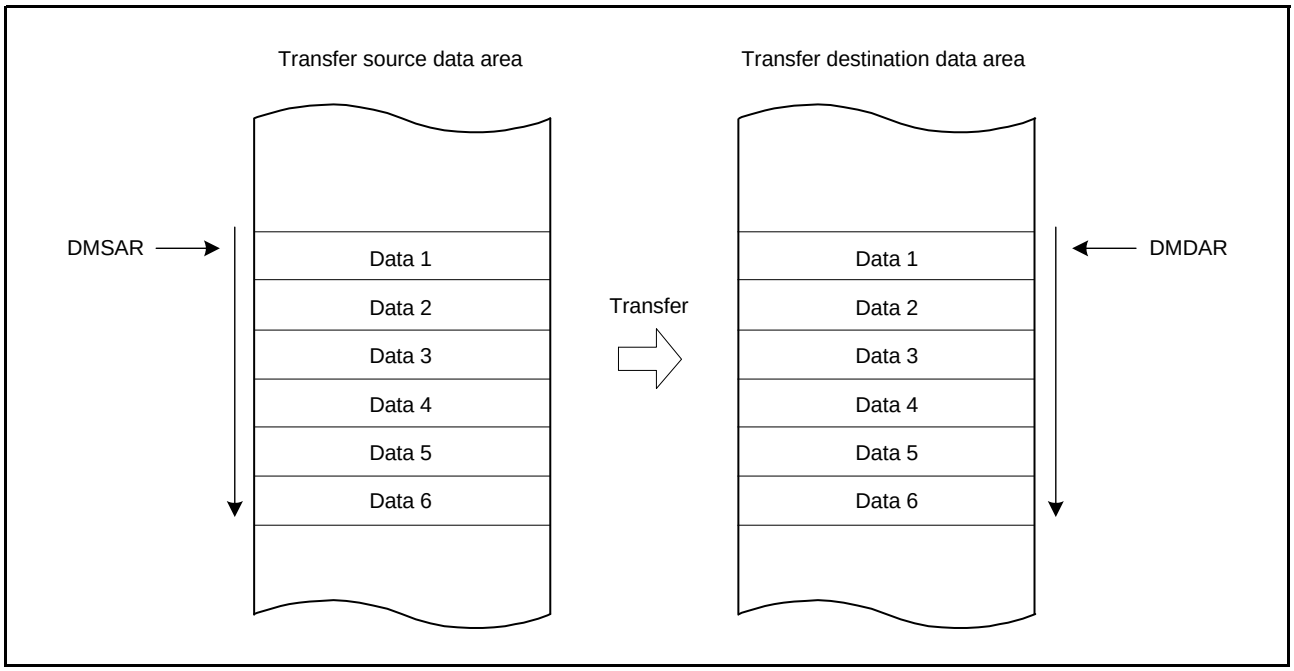


Figure 17.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using 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 17.4 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.4 Register Update Operation in Repeat Transfer Mode

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.

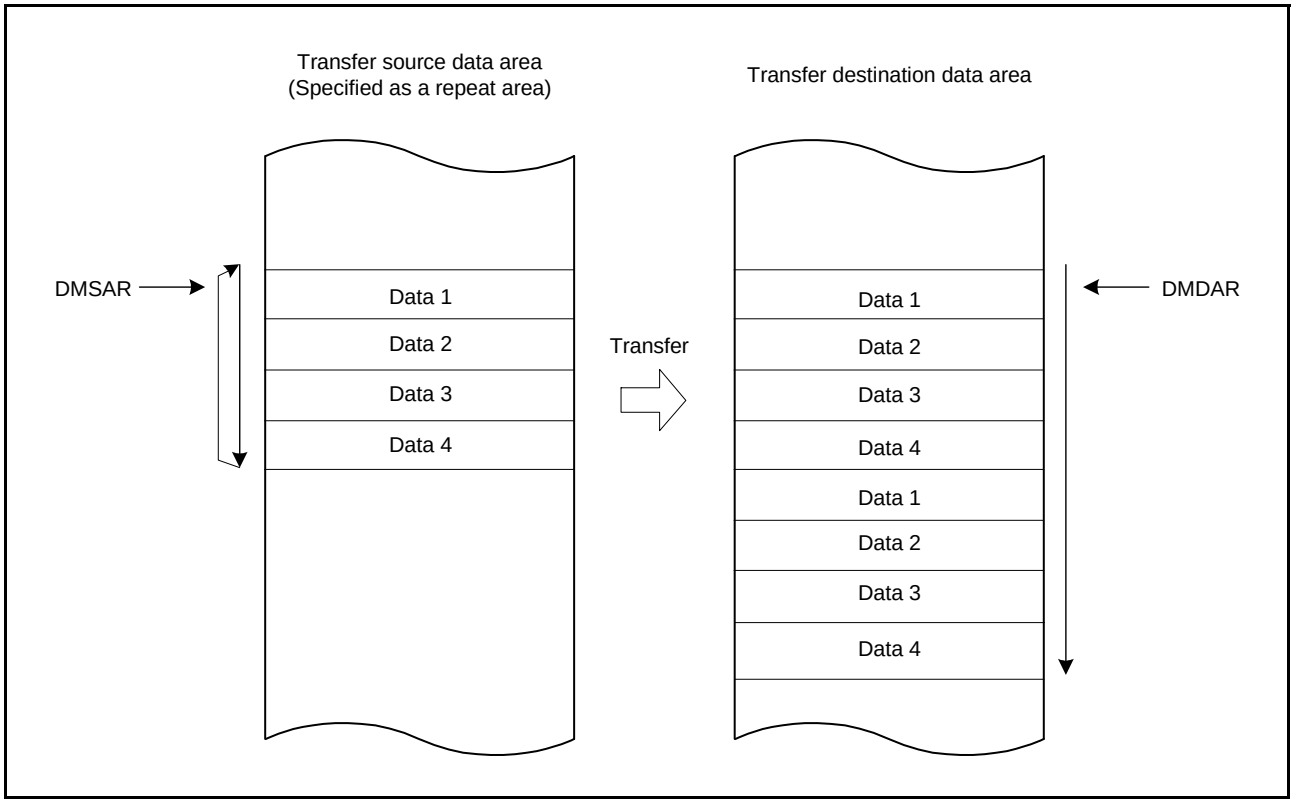


Figure 17.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.
A maximum of 1K data can be set as a total block transfer size using 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 17.5 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Block size counter	DMACm.DMCRAH
DMACm.DMCRB	Transfer-block counter	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

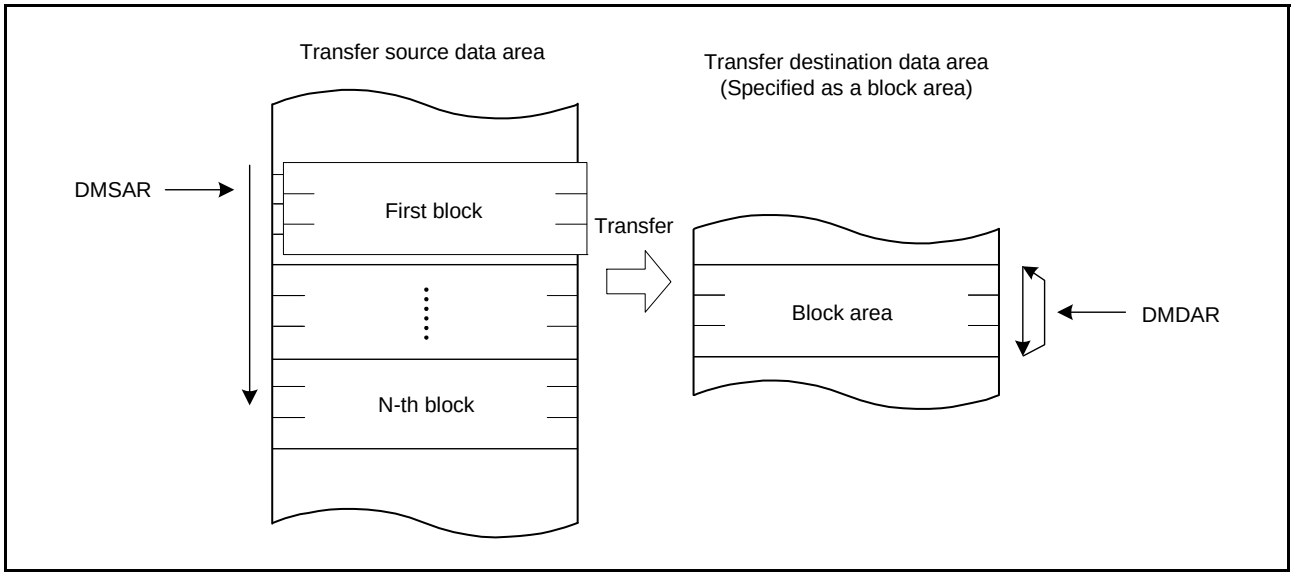


Figure 17.4 Operation in Block Transfer Mode

17.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the 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 17.5 shows an example of the extended repeat area operation.

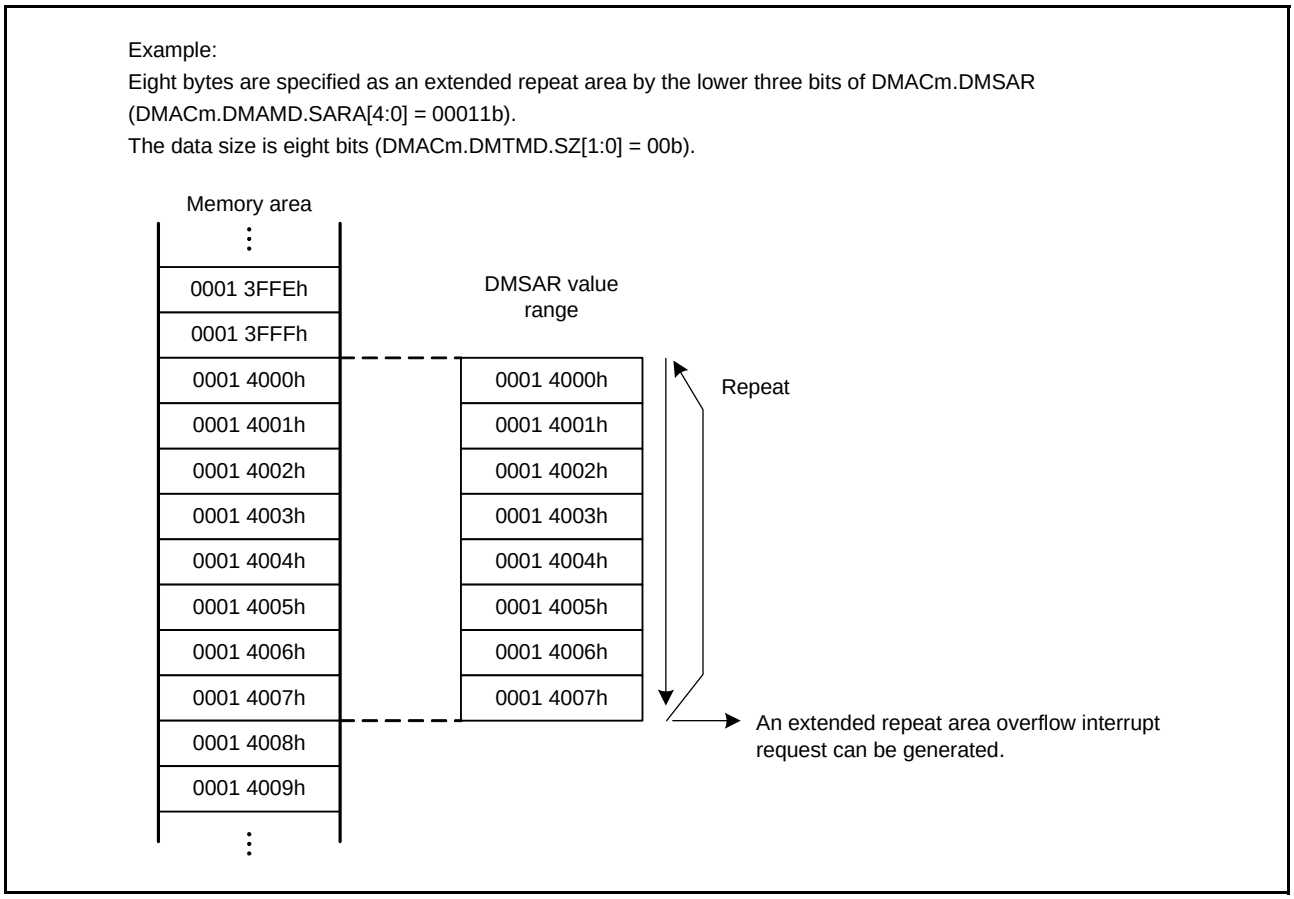


Figure 17.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.
When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 17.6 shows an example when the extended repeat area function is used in block transfer mode.

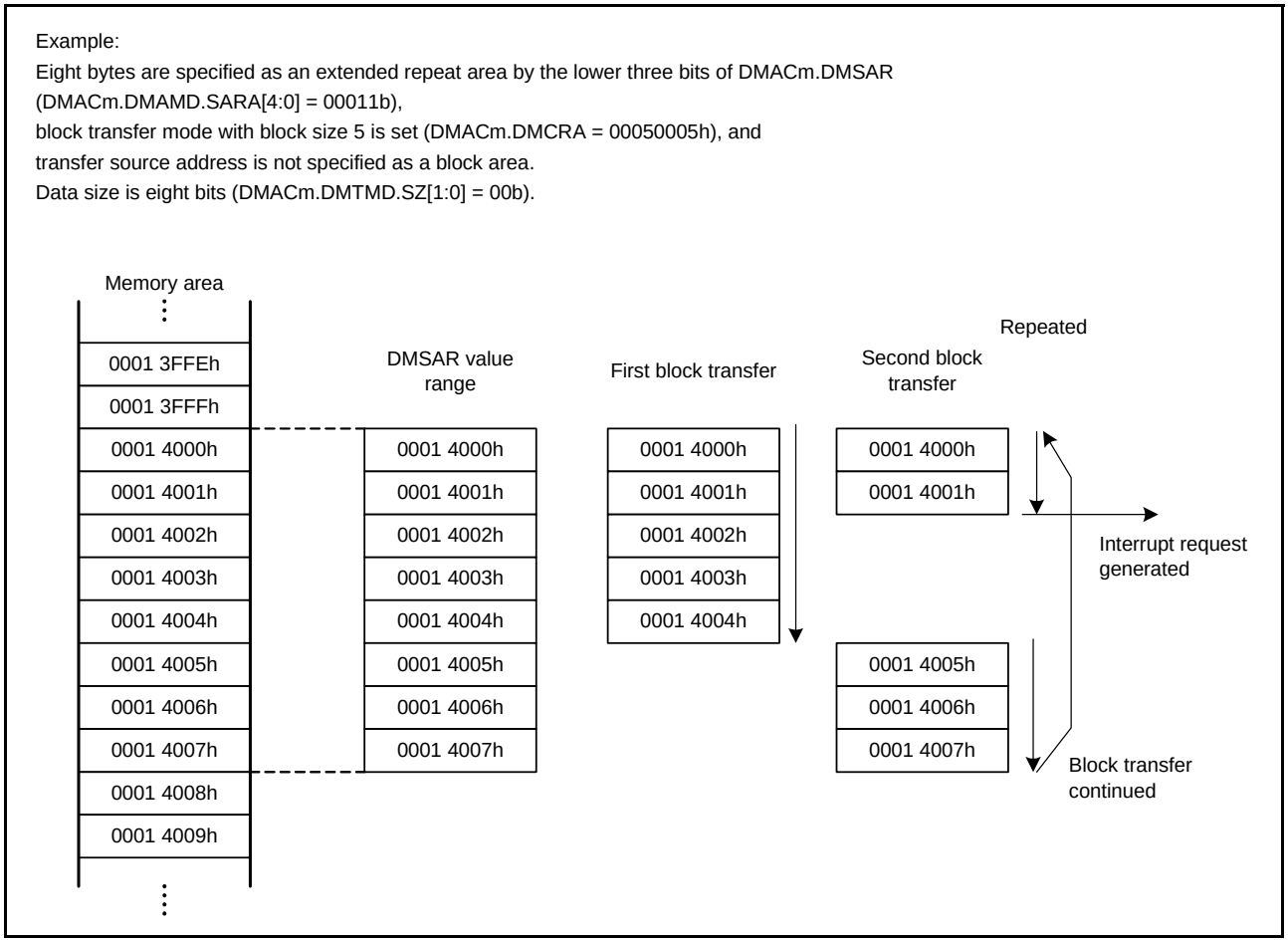


Figure 17.6 Example of Extended Repeat Area Function in Block Transfer Mode

17.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (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 17.6 lists the address update method in each address update mode.

Table 17.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different 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 17.7 shows an example of address updating using offset addition.

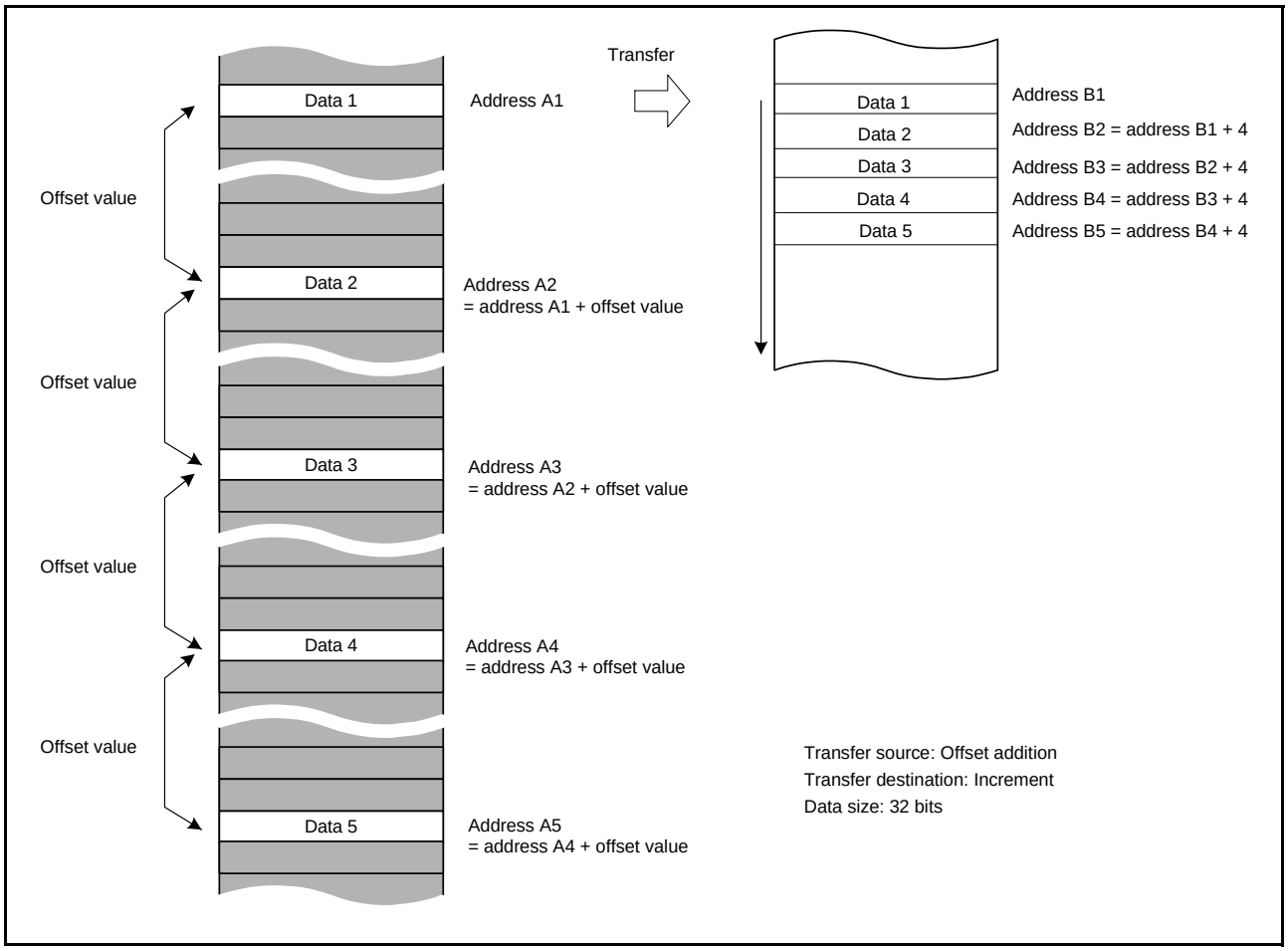


Figure 17.7 Example of Address Updating by Offset Addition

In Figure 17.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 17.8 shows the XY conversion using offset addition in repeat transfer mode.
Settings are as follows:

- DMAC0.DMAMD 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.

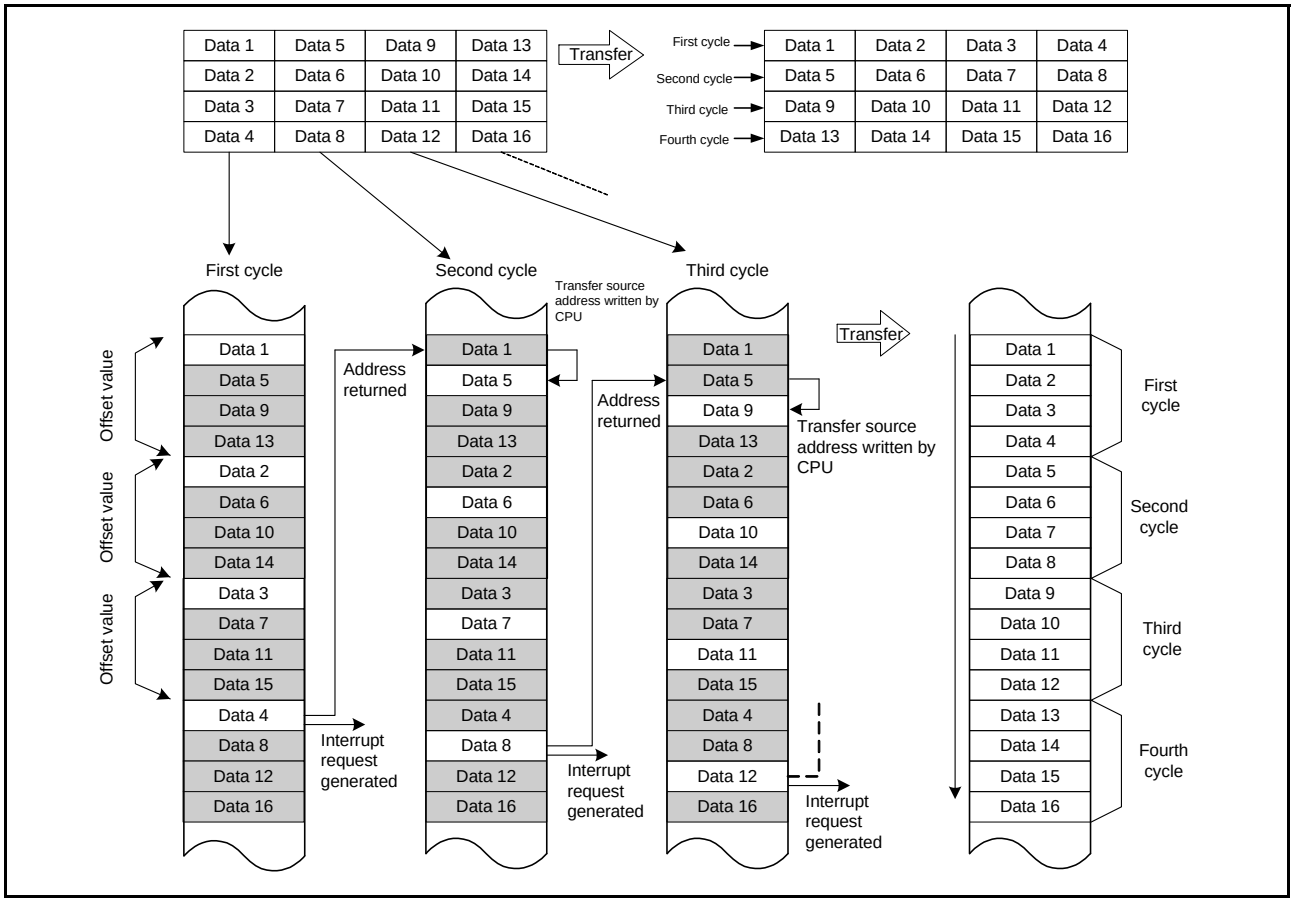


Figure 17.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the 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 17.9 shows a flowchart of the XY conversion.

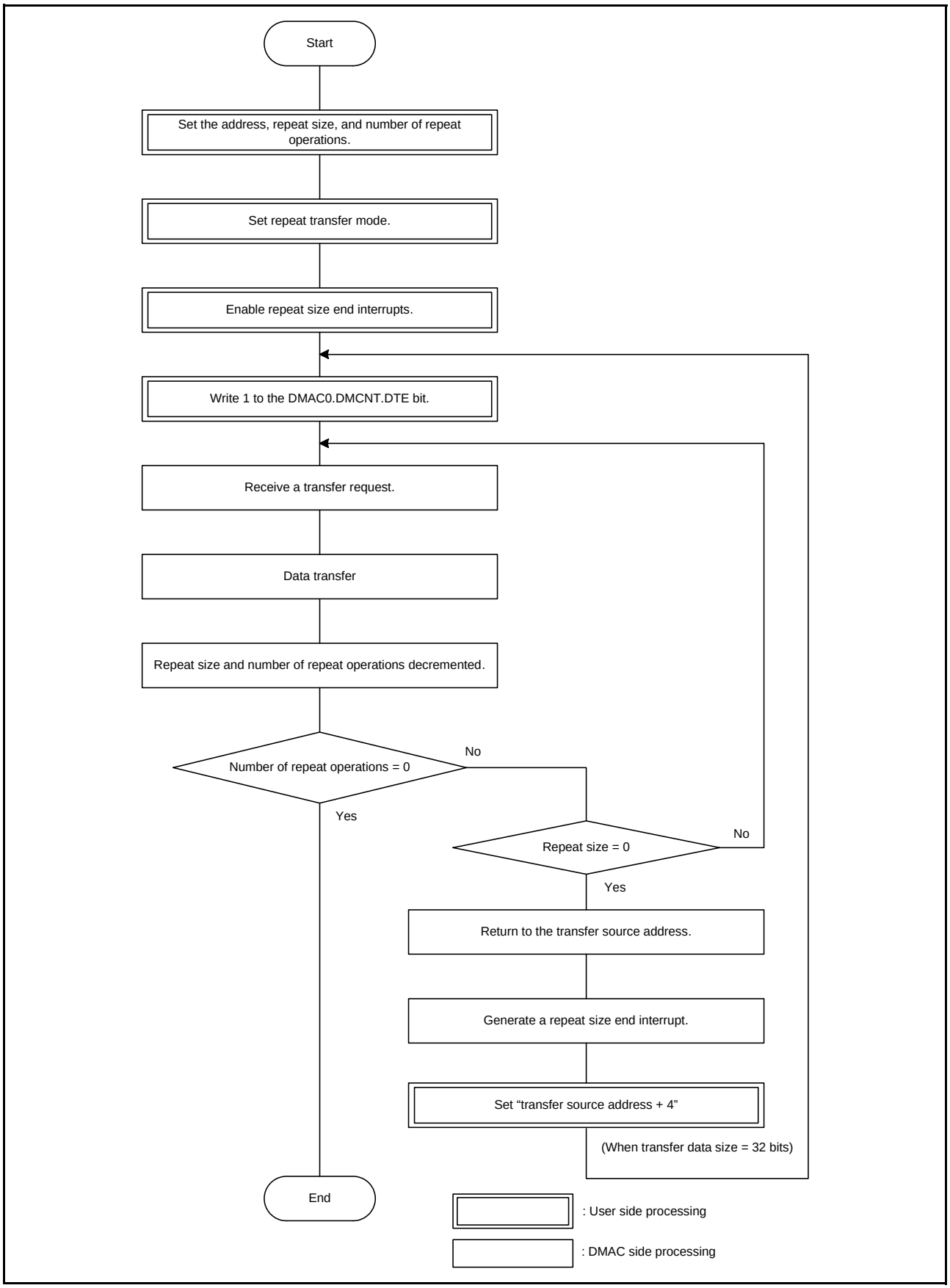


Figure 17.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

17.3.4 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 14.5, Interrupt Vector Table, in section 14, Interrupt Controller (ICUF).

17.3.5 Operation Timing

Figure 17.10 and Figure 17.11 show DMAC operation timing examples.

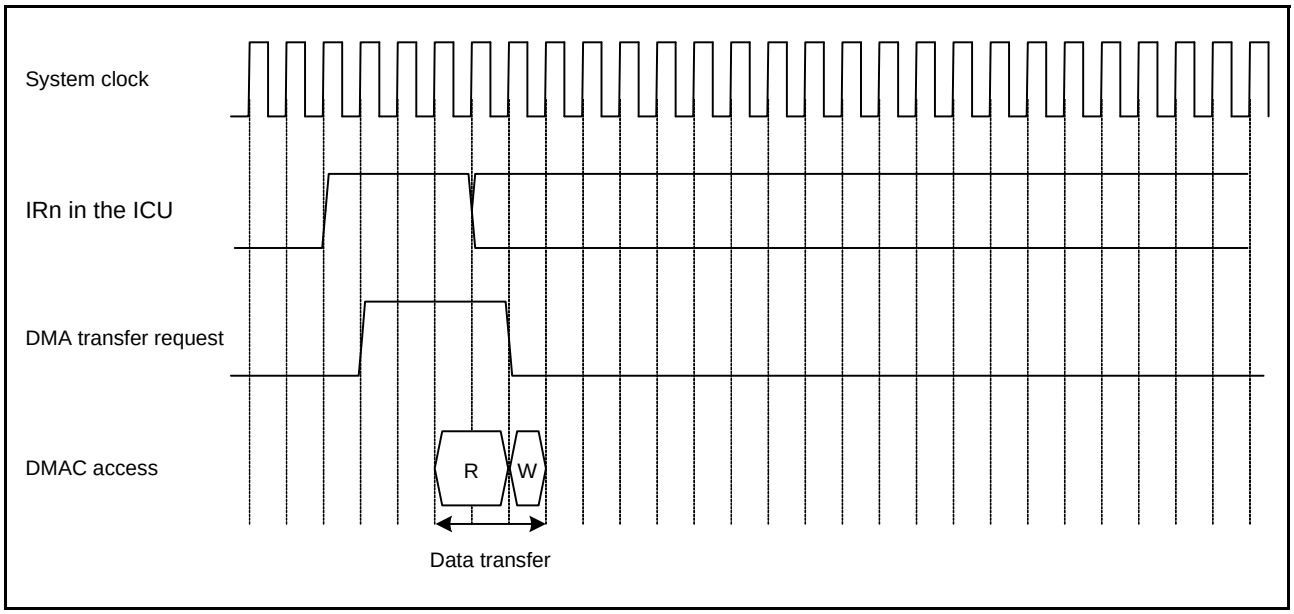


Figure 17.10 DMAC Operation Timing Example (1) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

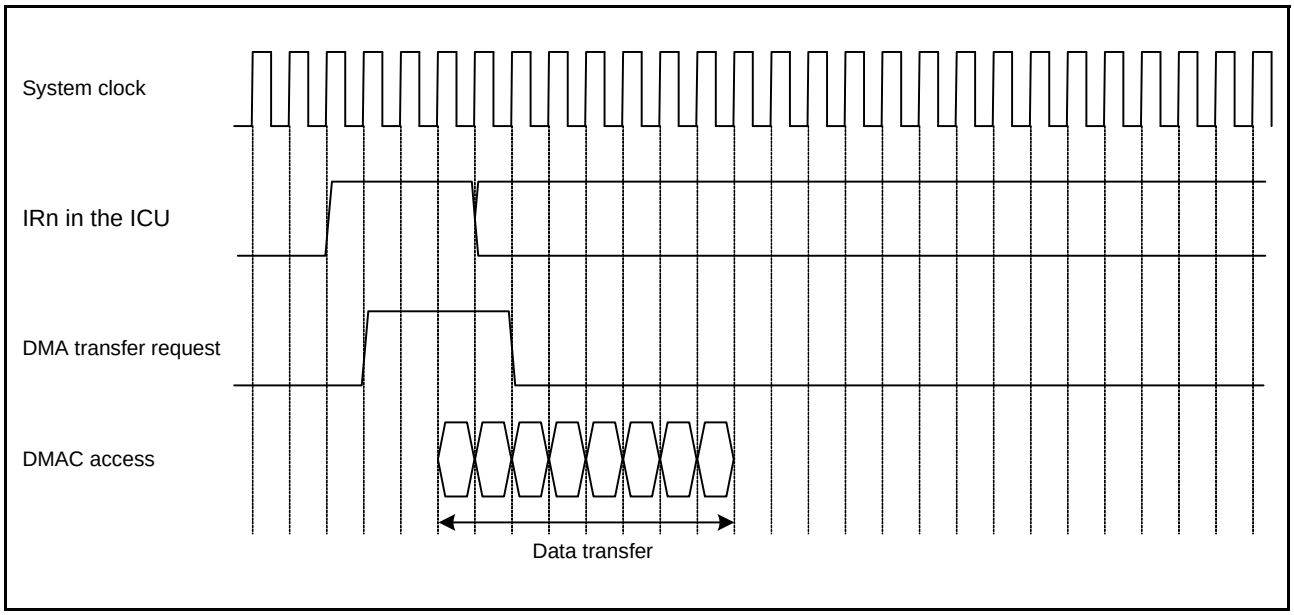


Figure 17.11 DMAC Operation Timing Example (2) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

17.3.6 DMAC Execution Cycles

Table 17.7 lists execution cycles in one DMAC data transfer operation.

Table 17.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

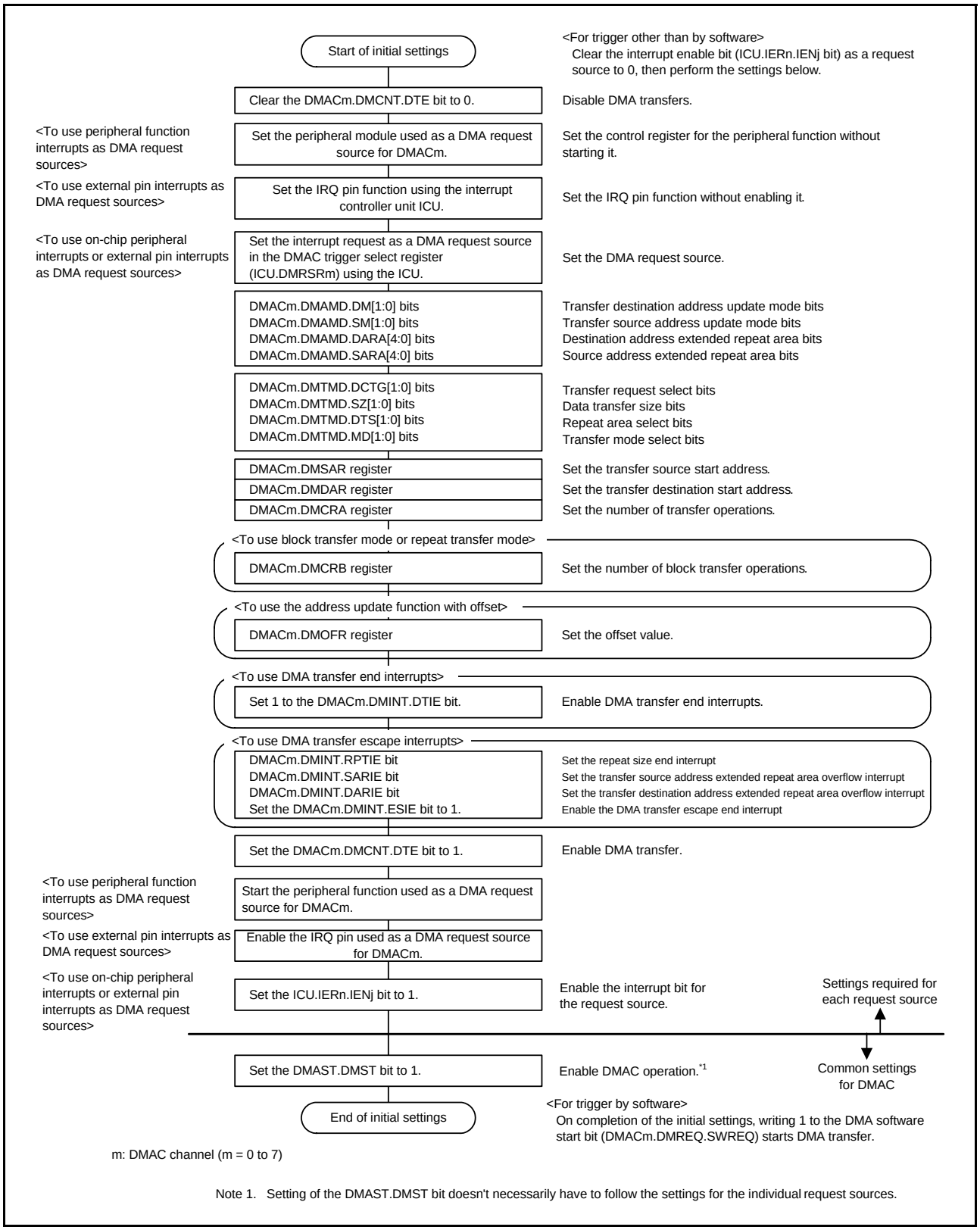
Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, refer to section 43, RAM, section 44, Flash Memory (FLASH), section 5, I/O Registers, and section 15.2.6, External Bus.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, refer to section 17.3.5, Operation Timing.

17.3.7 Activating the DMAC

Figure 17.12 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 17.12 Register Setting Procedure

17.3.8 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).

17.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

(1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of 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 17.3 to Table 17.5.

(2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of 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 17.3 to Table 17.5.

(3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically 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 14, Interrupt Controller (ICUF).

17.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.
The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 > channel 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.

17.4 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.

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of 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 14, Interrupt Controller (ICUF).

17.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the 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 14, Interrupt Controller (ICUF).

17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the 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 14, Interrupt Controller (ICUF).

17.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the external bus or the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 17.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 17.13 shows the schematic logic diagram of interrupt outputs (DMAC0 to DMAC3). Figure 17.14 shows the schematic logic diagram of interrupt outputs (DMAC4 to DMAC7). Figure 17.15 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 17.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

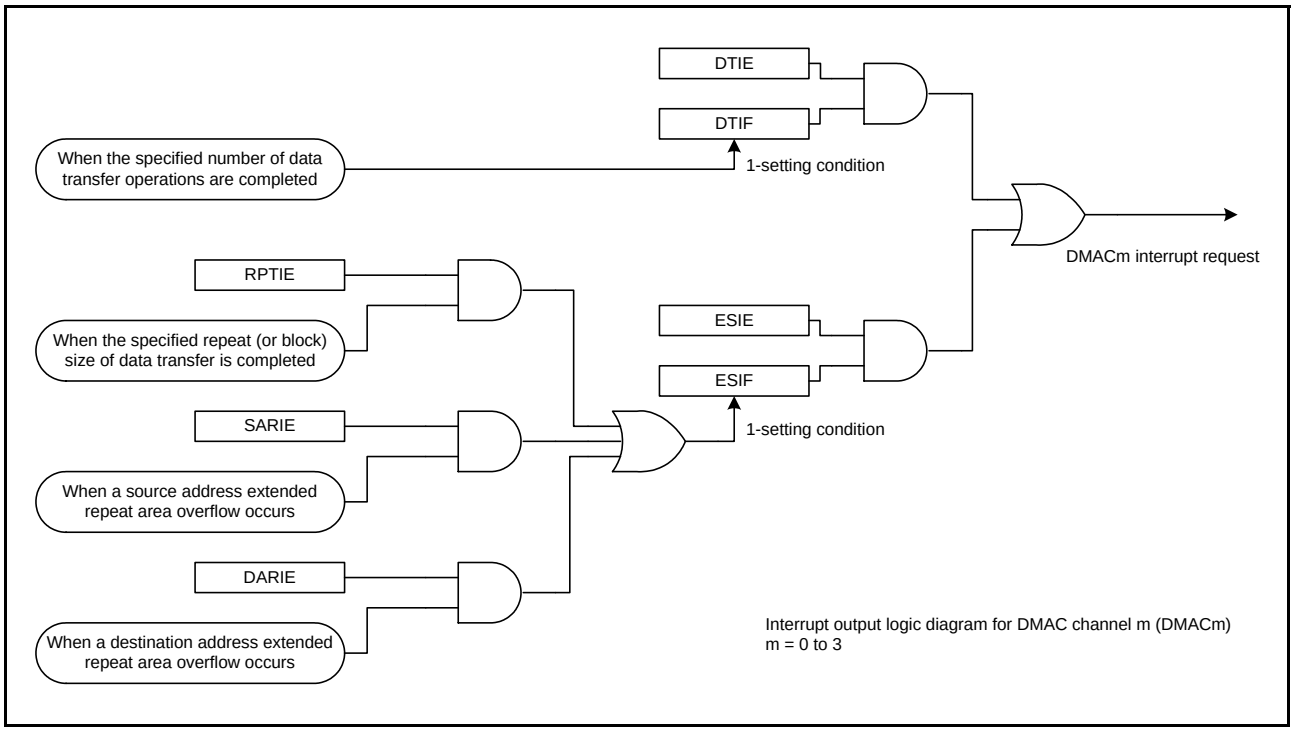


Figure 17.13 Schematic Logic Diagram of Interrupt Outputs (DMAC0 to DMAC3)

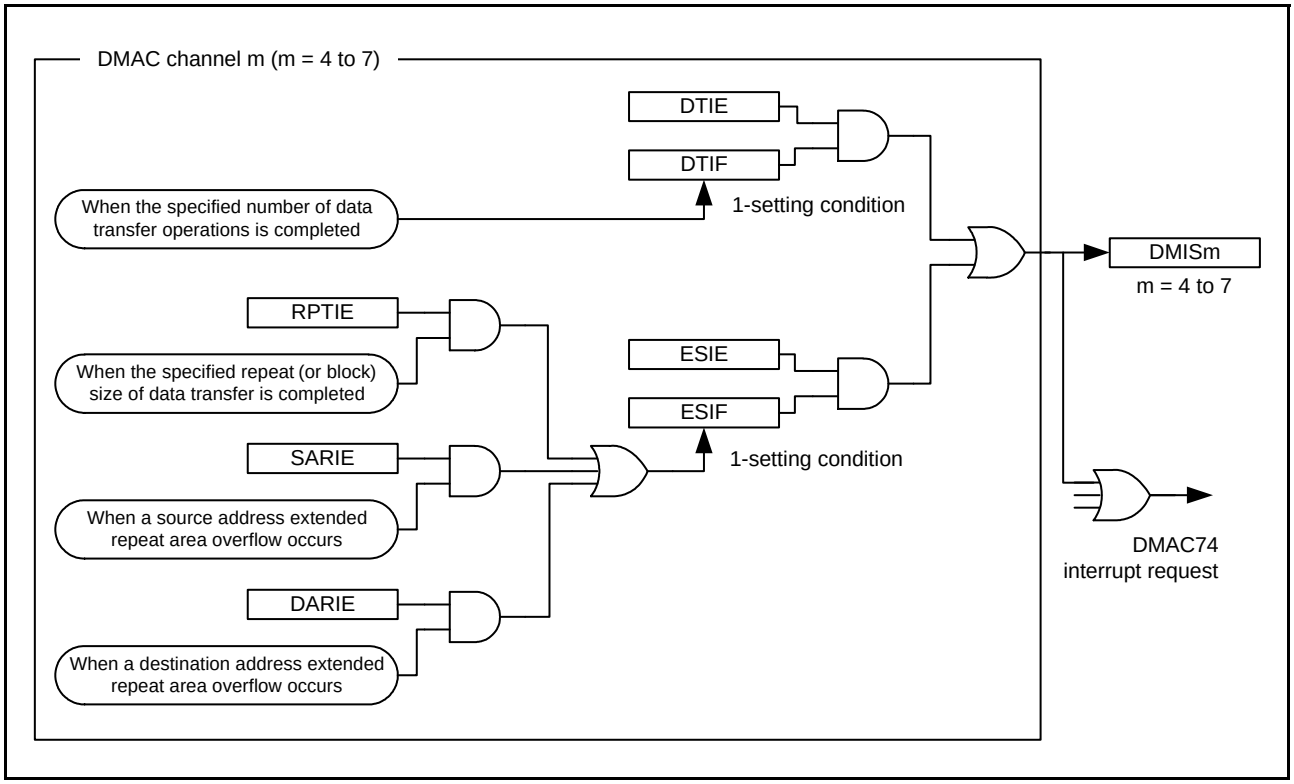


Figure 17.14 Schematic Logic Diagram of Interrupt Outputs (DMAC4 to DMAC7)

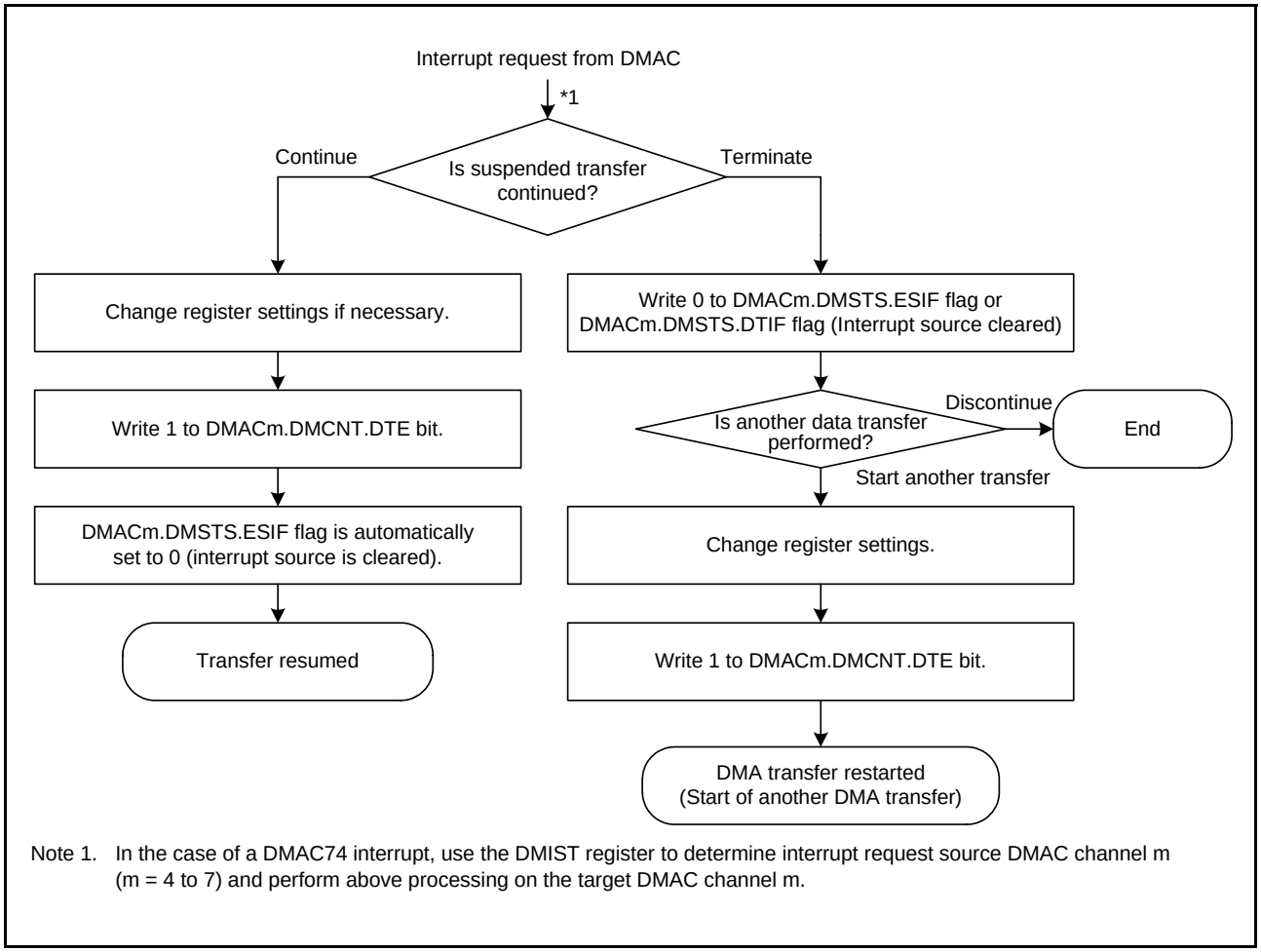
Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(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 17.15 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

17.6 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.

17.7 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 11.5.2.1, 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 11.5.3.1, Transition to Software Standby Mode or section 11.5.4.1, 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 11.6.6, Timing of WAIT Instruction 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 14, Interrupt Controller (ICUF), and then execute the WAIT instruction.

17.8 Usage Notes

17.8.1 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.

17.8.2 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.

17.8.3 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).

17.8.4 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.

17.8.5 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 14, Interrupt Controller (ICUF). For the DMACm.DMCSL.DISEL bit setting, refer to section 17.2.12, DMA Request Source Flag Control Register (DMCSL).

17.8.6 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 14, Interrupt Controller (ICUF).

17.8.7 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 17.3.7, Activating the DMAC.

18. 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.

18.1 Overview

Table 18.1 lists the specifications of the DTC, and Figure 18.1 shows a block diagram of the DTC.

Table 18.1 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.

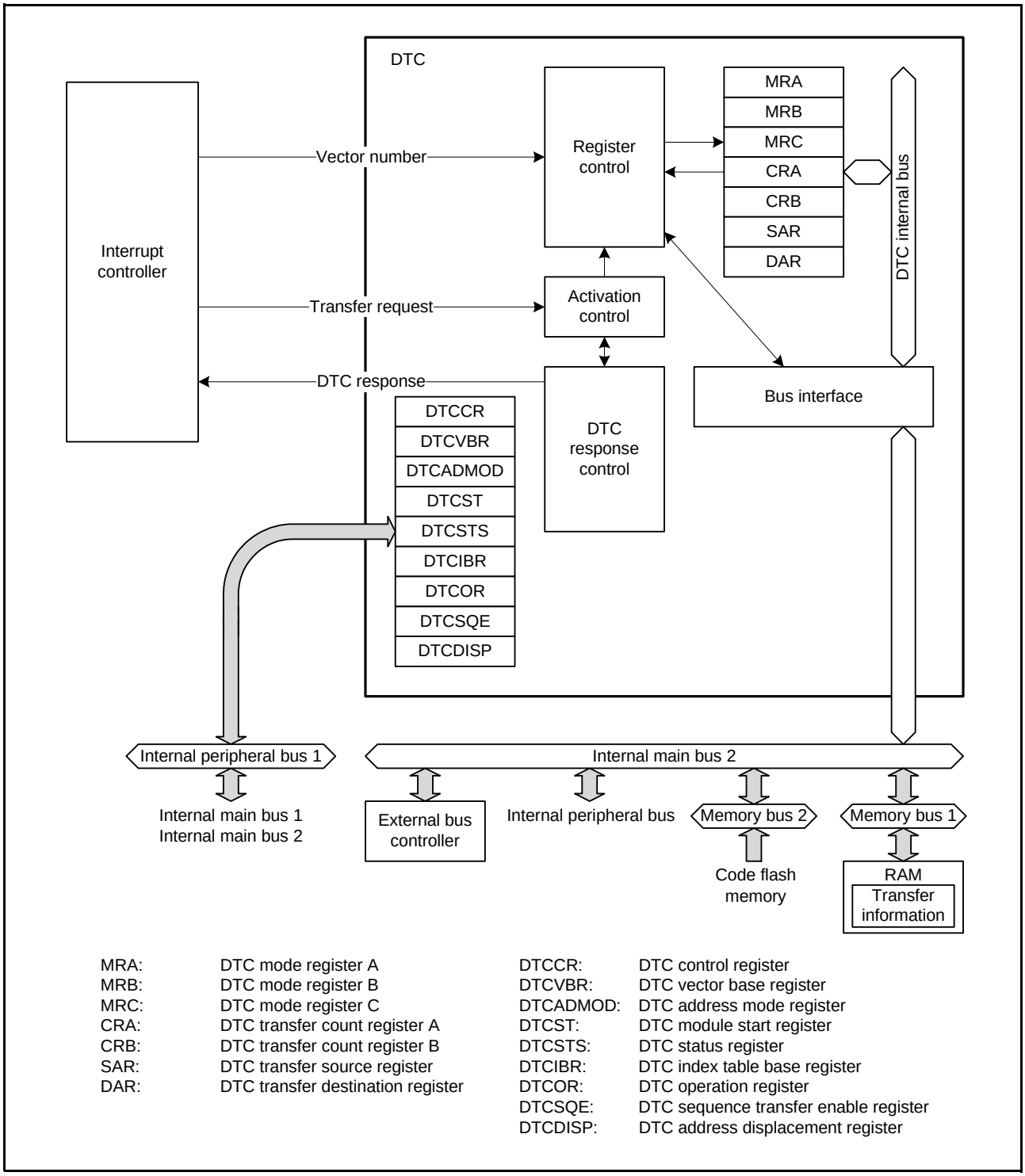


Figure 18.1 DTC Block Diagram

18.2 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.

18.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	WBDIS
Value after reset:	x	x	x	x	x	x	x	x

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).

18.2.2 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 18.2 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 18.2 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 18.2 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 18.4, 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 18.4.6, Chain Transfer.

Refer to Table 18.2 for the setting value to be used in the sequence transfer.

18.2.3 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).

18.2.4 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.

18.2.5 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

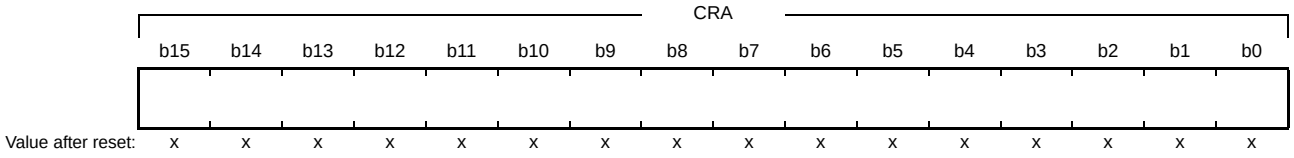
x: Undefined

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.

18.2.6 DTC Transfer Count Register A (CRA)

- Normal transfer mode

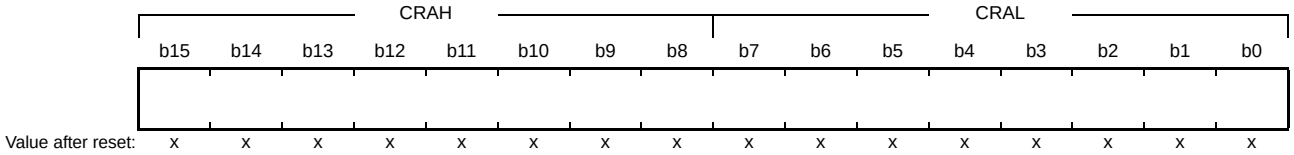
Address(es): (inaccessible directly from the CPU)



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.

18.2.7 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.

18.2.8 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.

18.2.9 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.

18.2.10 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).

18.2.11 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 18.9, Low Power Consumption Function, and section 11, Low Power Consumption.

18.2.12 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 14.5, Interrupt Vector Table in section 14, Interrupt Controller (ICUF).

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.

18.2.13 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.

18.2.14 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 18.2 to terminate the sequence transfer.
Writing 1 to the bit, while no sequence transfer is performed, have no effect.

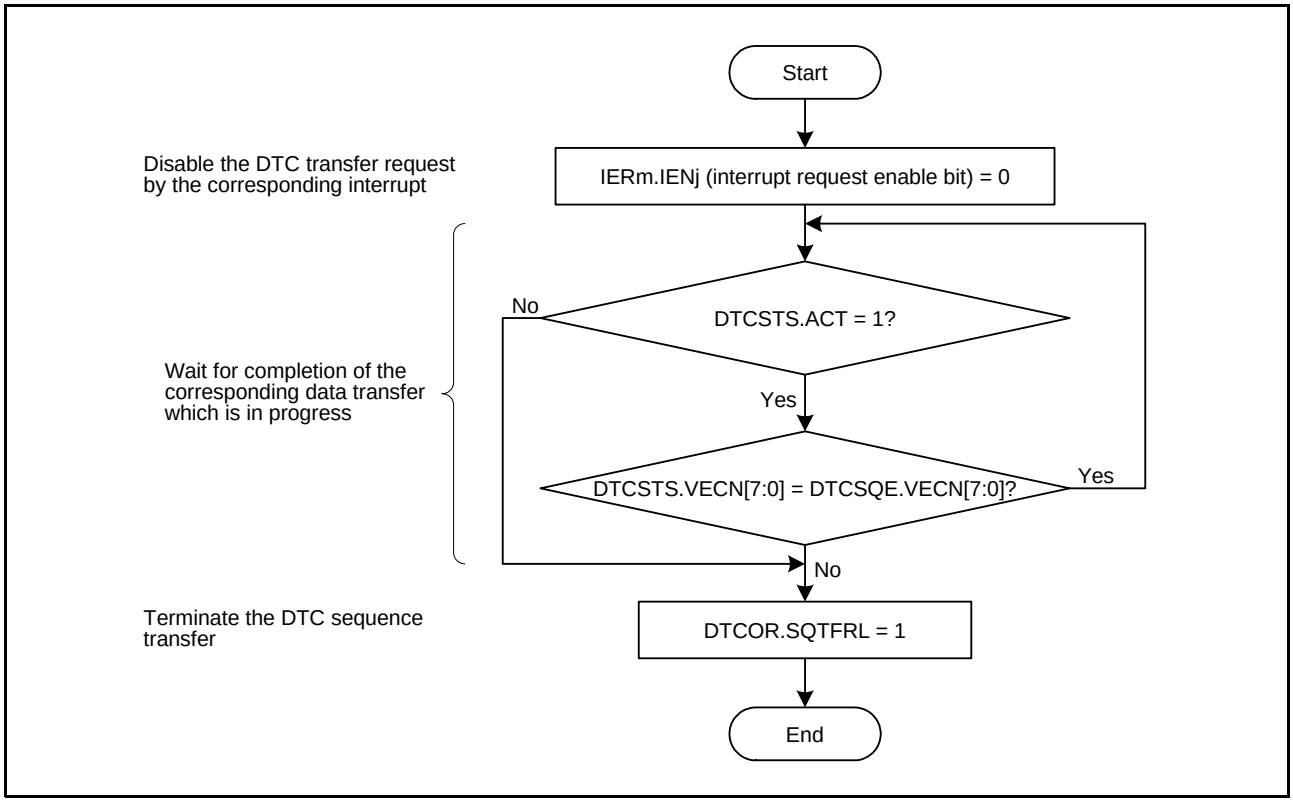


Figure 18.2 Procedure to Terminate Sequence Transfer

18.2.15 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 18.24 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 14.5, Interrupt Vector Table in section 14, Interrupt Controller (ICUF) 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.

18.2.16 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.

18.3 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 14.5, Interrupt Vector Table in section 14, Interrupt Controller (ICUF). For request by software, refer to section 14.2.5, Software Interrupt Generation Register (SWINTR) and section 14.2.6, Software Interrupt 2 Generation Register (SWINT2R) in section 14, Interrupt Controller (ICUF).
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.

18.3.1 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 18.3 shows the relationship between the DTC vector table and transfer information.
Figure 18.4 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 18.10.2, Allocating Transfer Information.

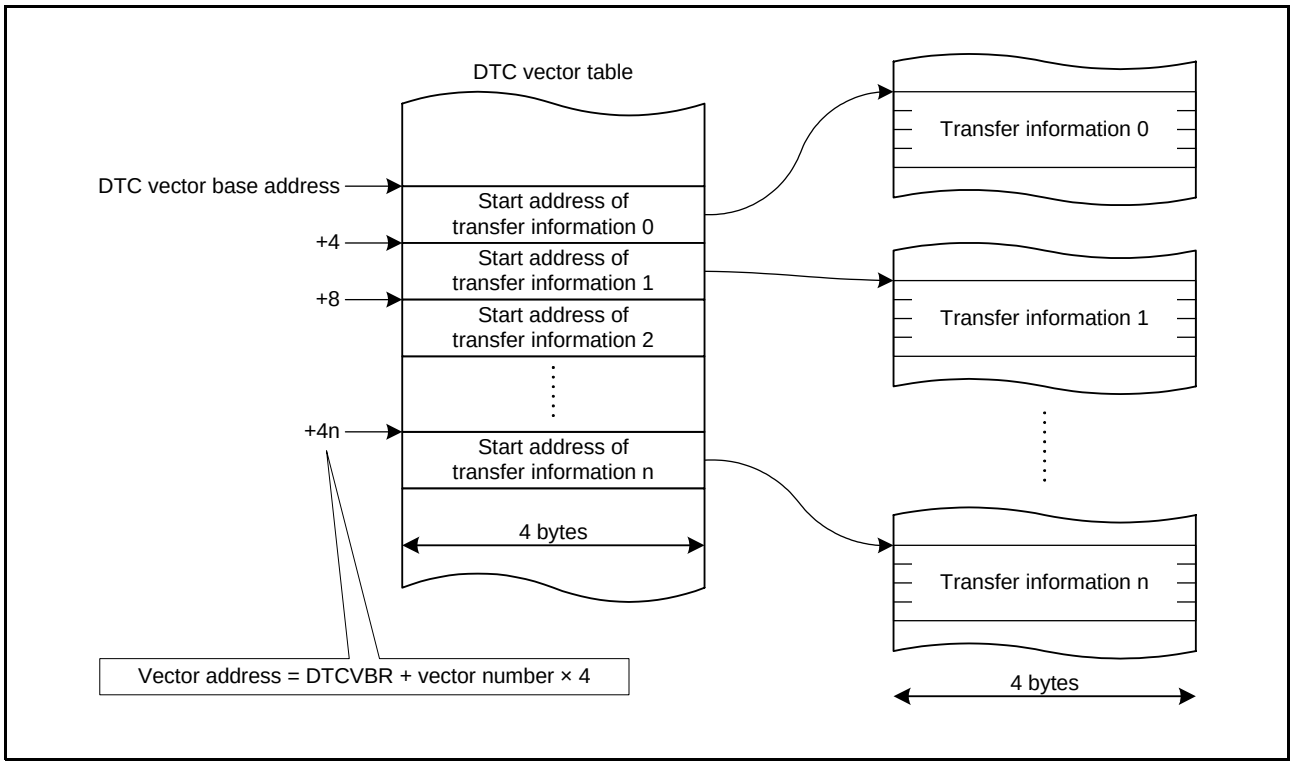


Figure 18.3 DTC Vector Table and Transfer Information

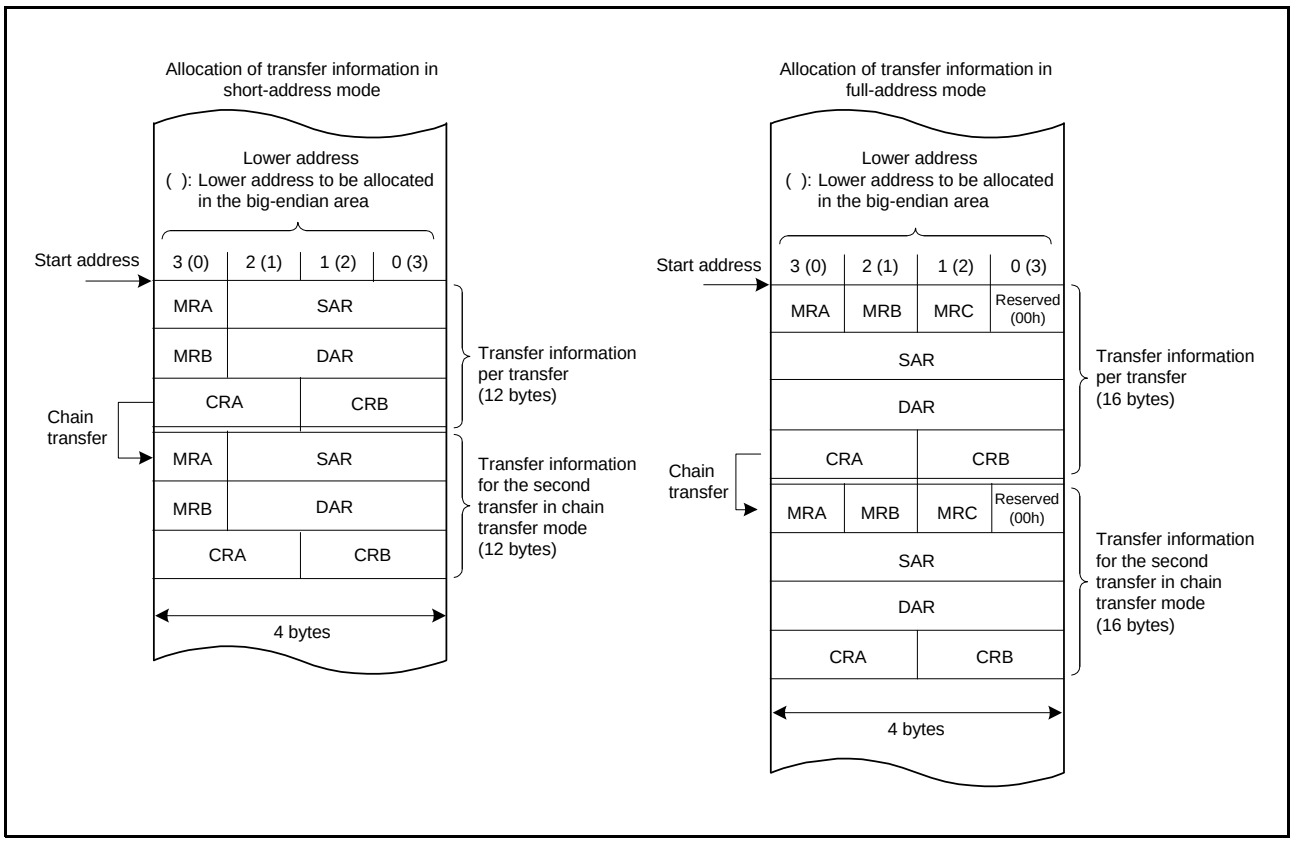


Figure 18.4 Allocation of Transfer Information in the RAM Area

18.4 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 18.3 lists transfer modes of the DTC.

Table 18.3 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 18.5 shows the operation flowchart of the DTC. Table 18.4 lists chain transfer conditions.

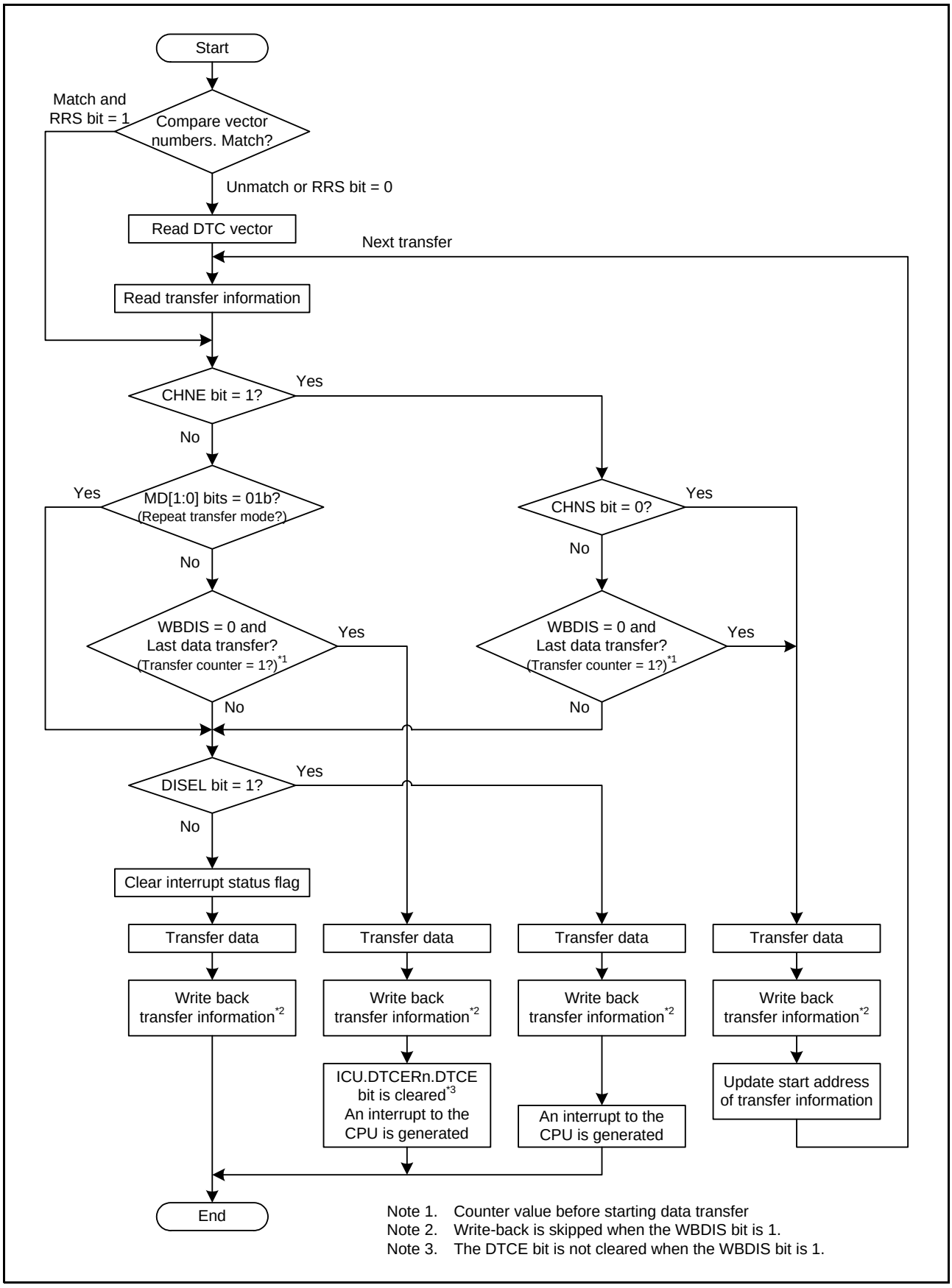


Figure 18.5 Operation Flowchart of the DTC

Table 18.4 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.

18.4.1 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 18.14 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.

18.4.2 Transfer Information Write-Back Skip Function

18.4.2.1 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 18.5 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 18.5 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		

18.4.2.2 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.

18.4.3 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 18.6 lists register functions in normal transfer mode, and Figure 18.6 shows the memory map of normal transfer mode.

Table 18.6 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.

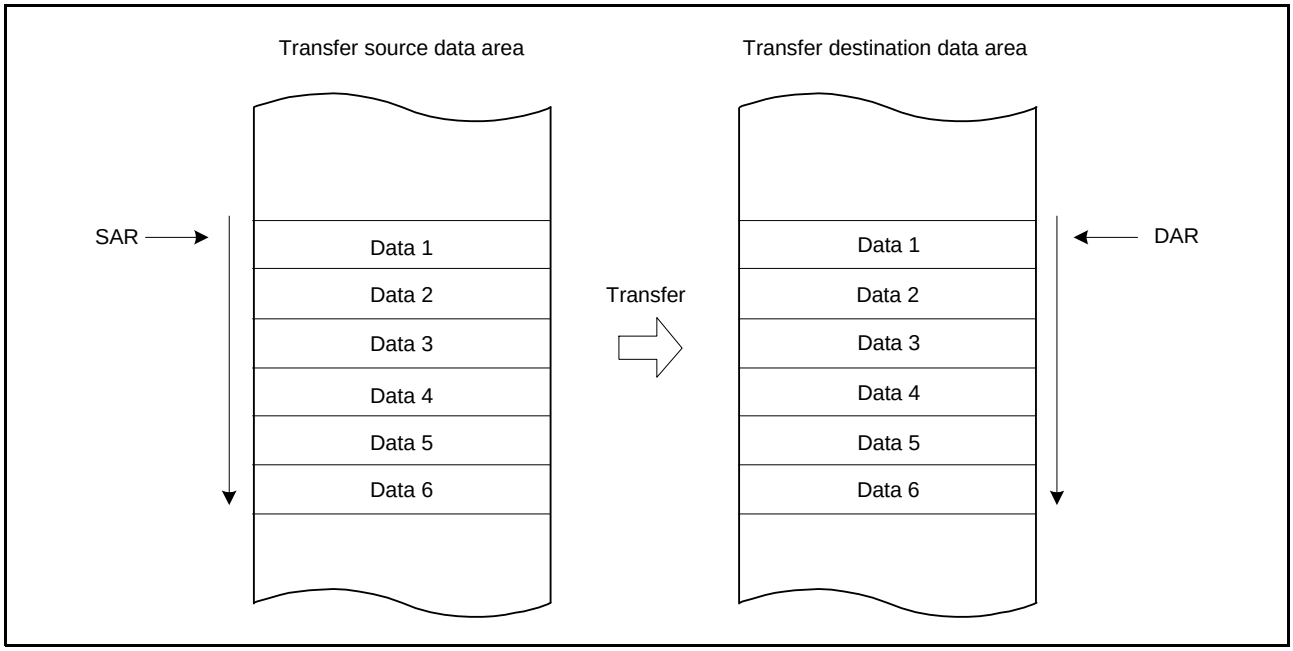


Figure 18.6 Memory Map of Normal Transfer Mode

18.4.4 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 18.7 lists the register functions in repeat transfer mode, and Figure 18.7 shows the memory map of repeat transfer mode.

Table 18.7 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.

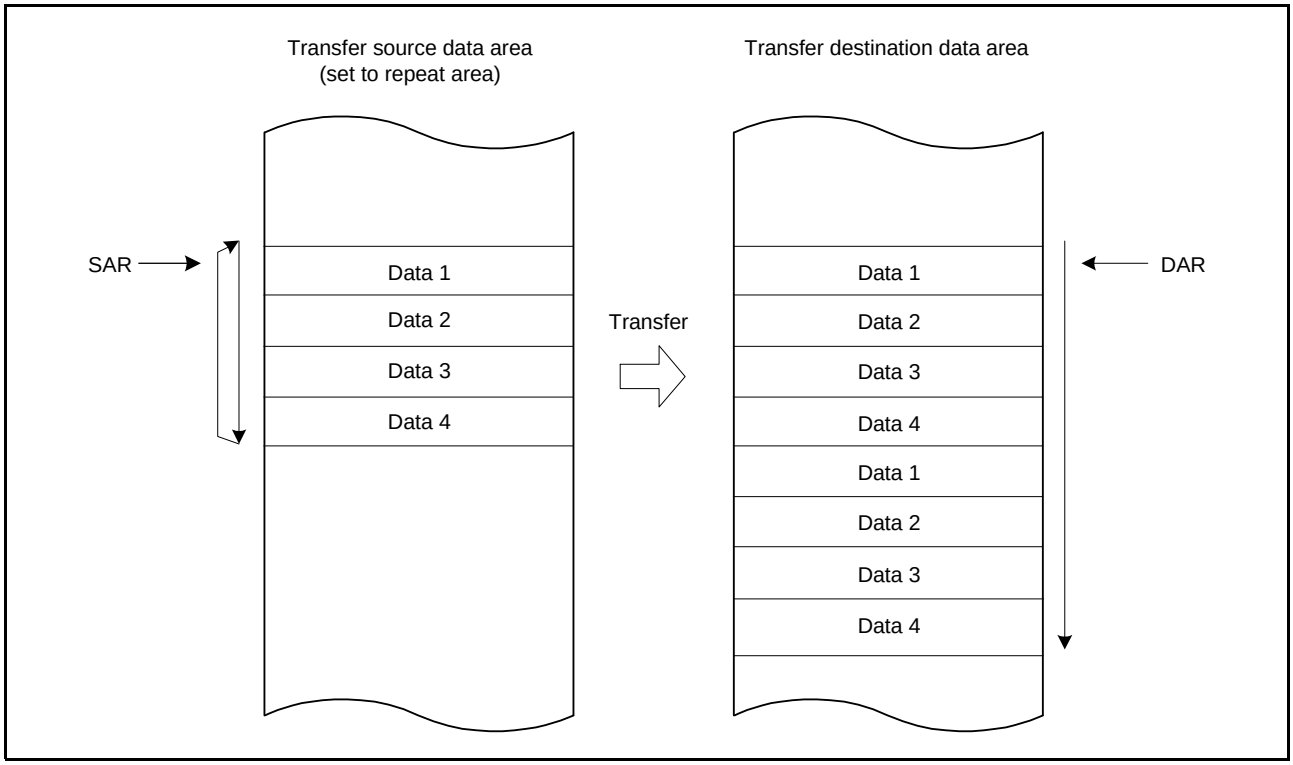


Figure 18.7 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

18.4.5 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 18.8 lists register functions in block transfer mode, and Figure 18.8 shows the memory map of block transfer mode.

Table 18.8 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.

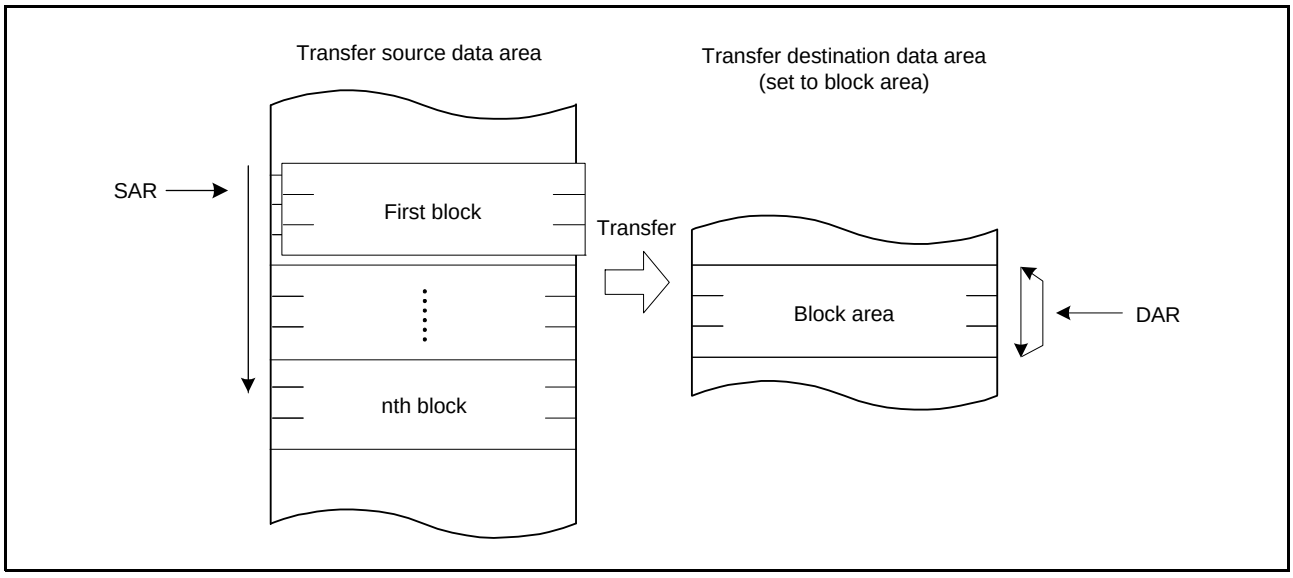


Figure 18.8 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

18.4.6 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 18.9 shows chain transfer operation.

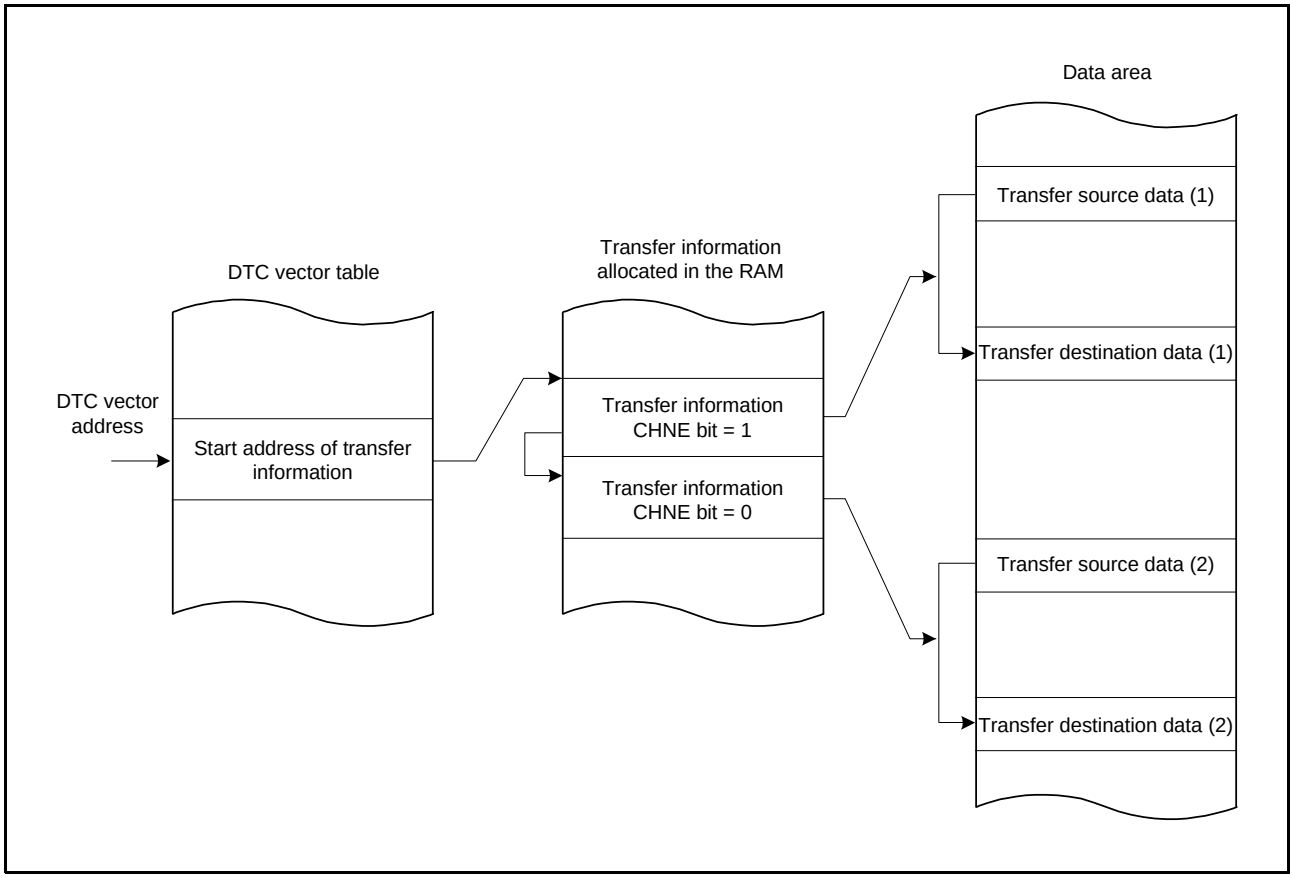


Figure 18.9 Chain Transfer Operation

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 18.4, Chain Transfer Conditions.

18.4.7 Operation Timing

Figure 18.10 to Figure 18.14 show examples of DTC operation timing.

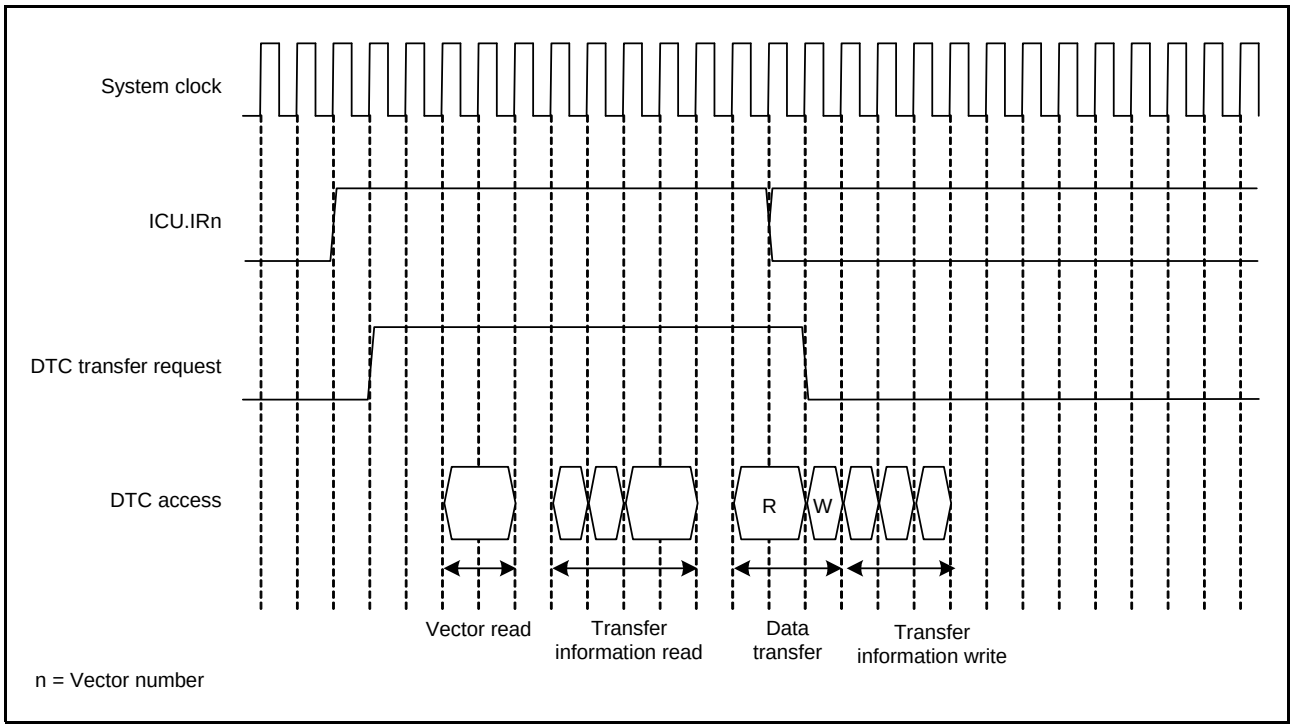


Figure 18.10 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

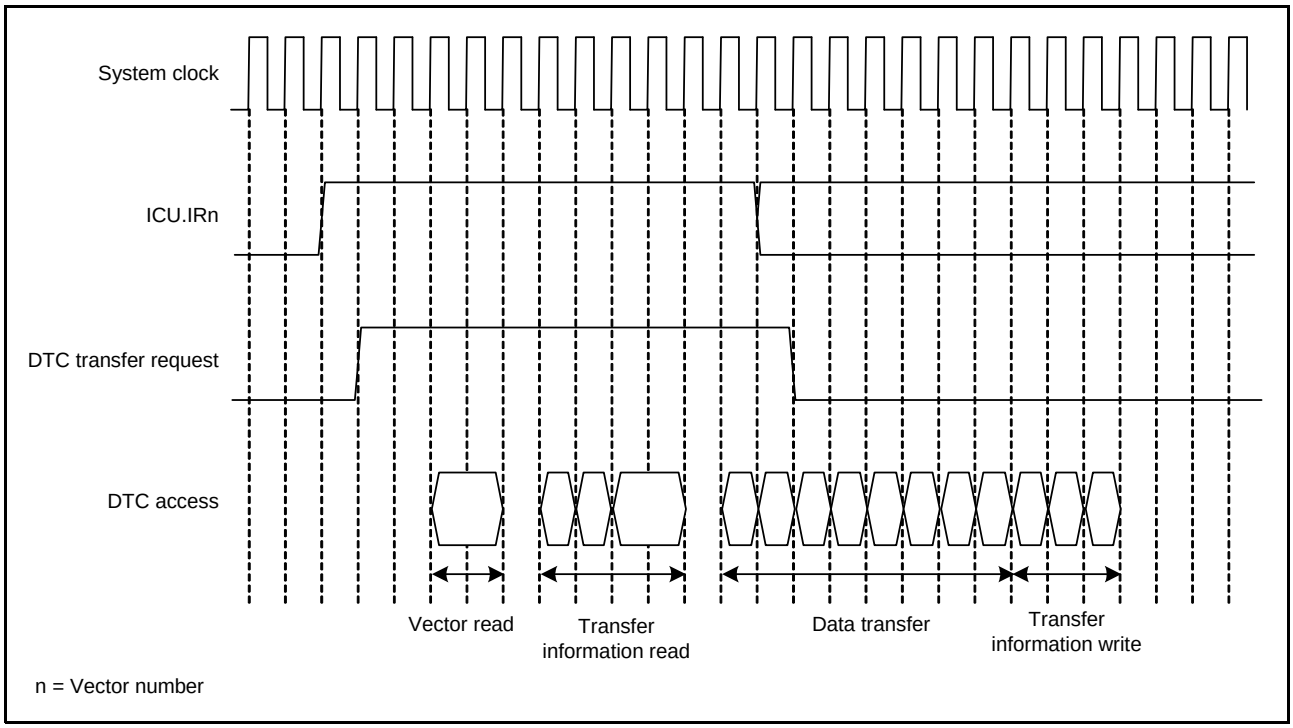


Figure 18.11 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

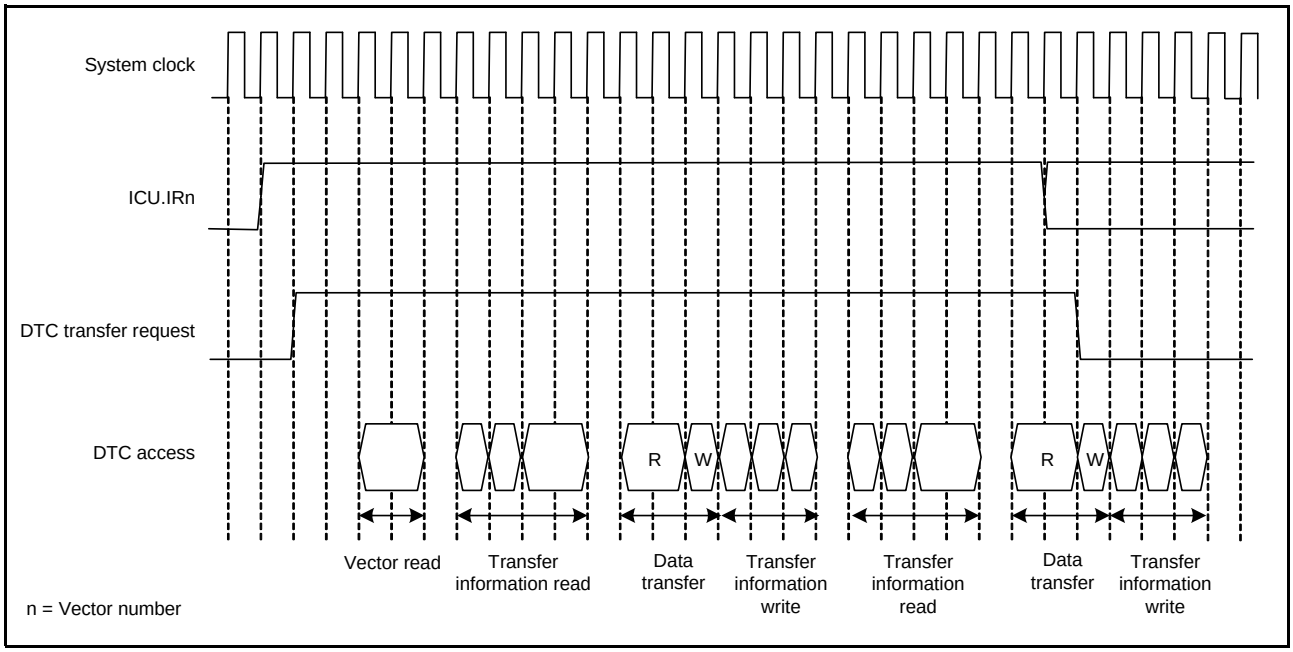


Figure 18.12 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

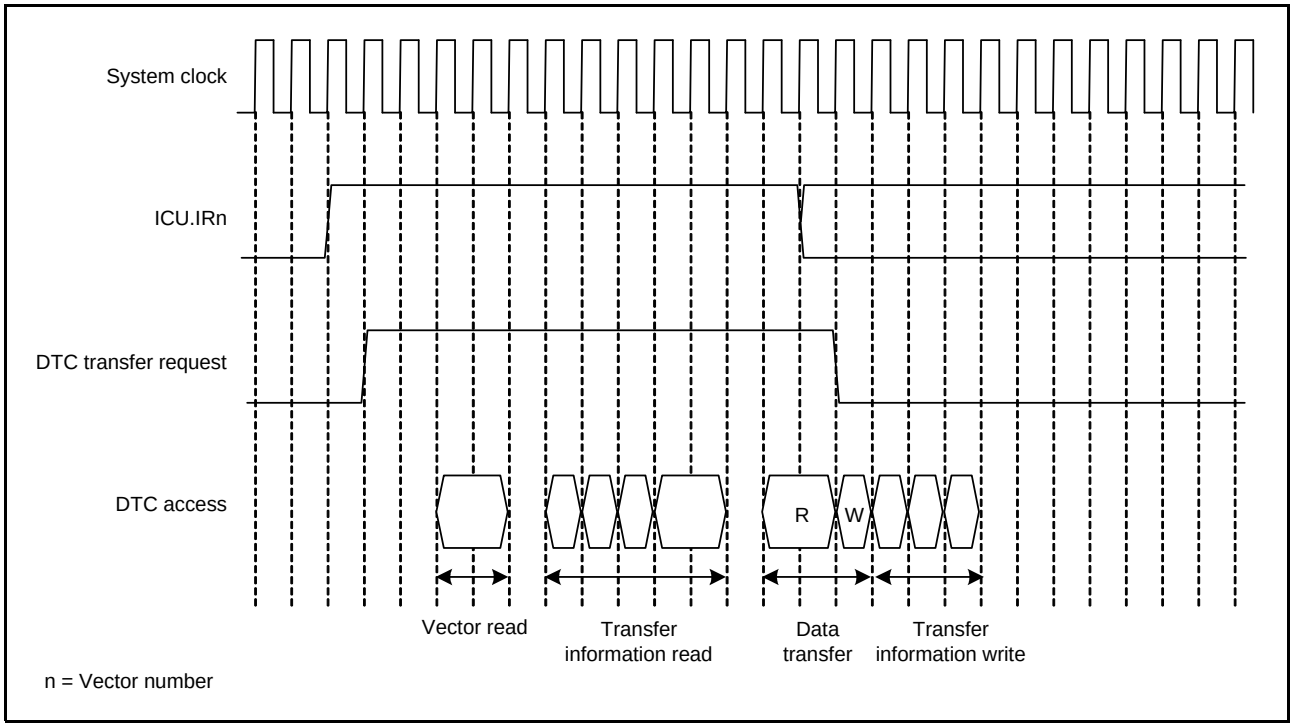
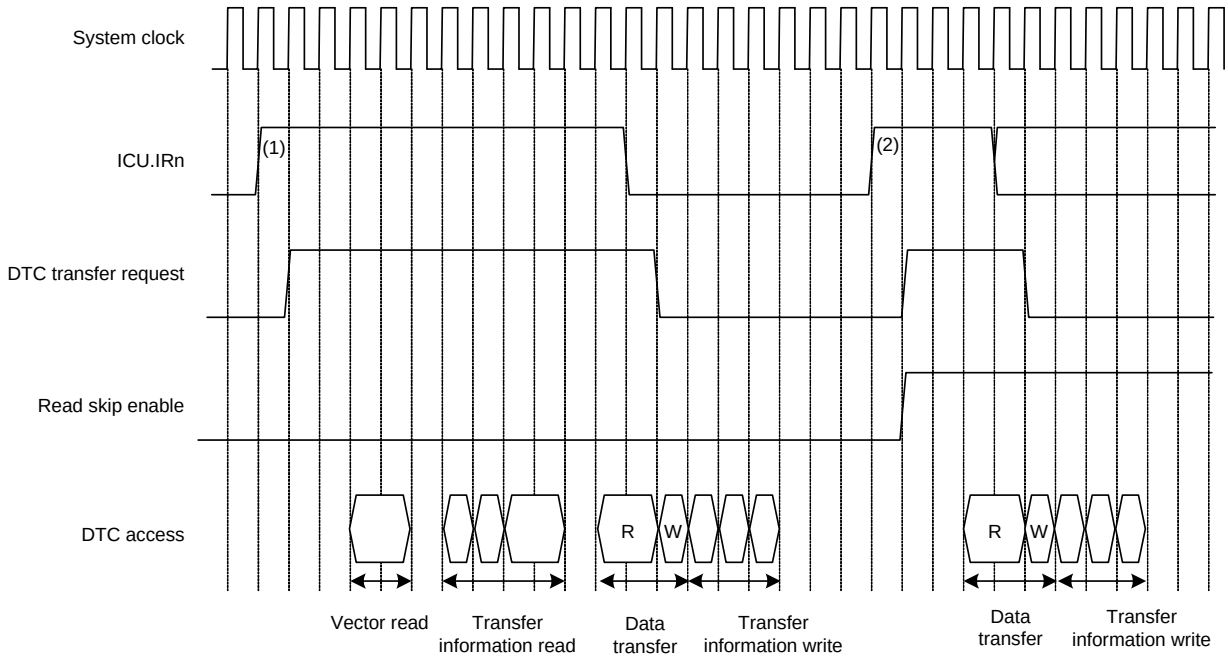


Figure 18.13 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



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 18.14 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)

18.4.8 Execution Cycles of the DTC

Table 18.9 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 18.4.7, Operation Timing.

Table 18.9 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 43, RAM, section 44, Flash Memory (FLASH), section 5, I/O Registers, and section 15.2.6, External Bus.)

18.4.9 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 15, Buses.

18.4.10 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 18.5 or section 14, Interrupt Controller (ICUF) for the conditions where the DTCE bit becomes 1.

Figure 18.15 and Figure 18.16 shows a basic sequence transfer operation.

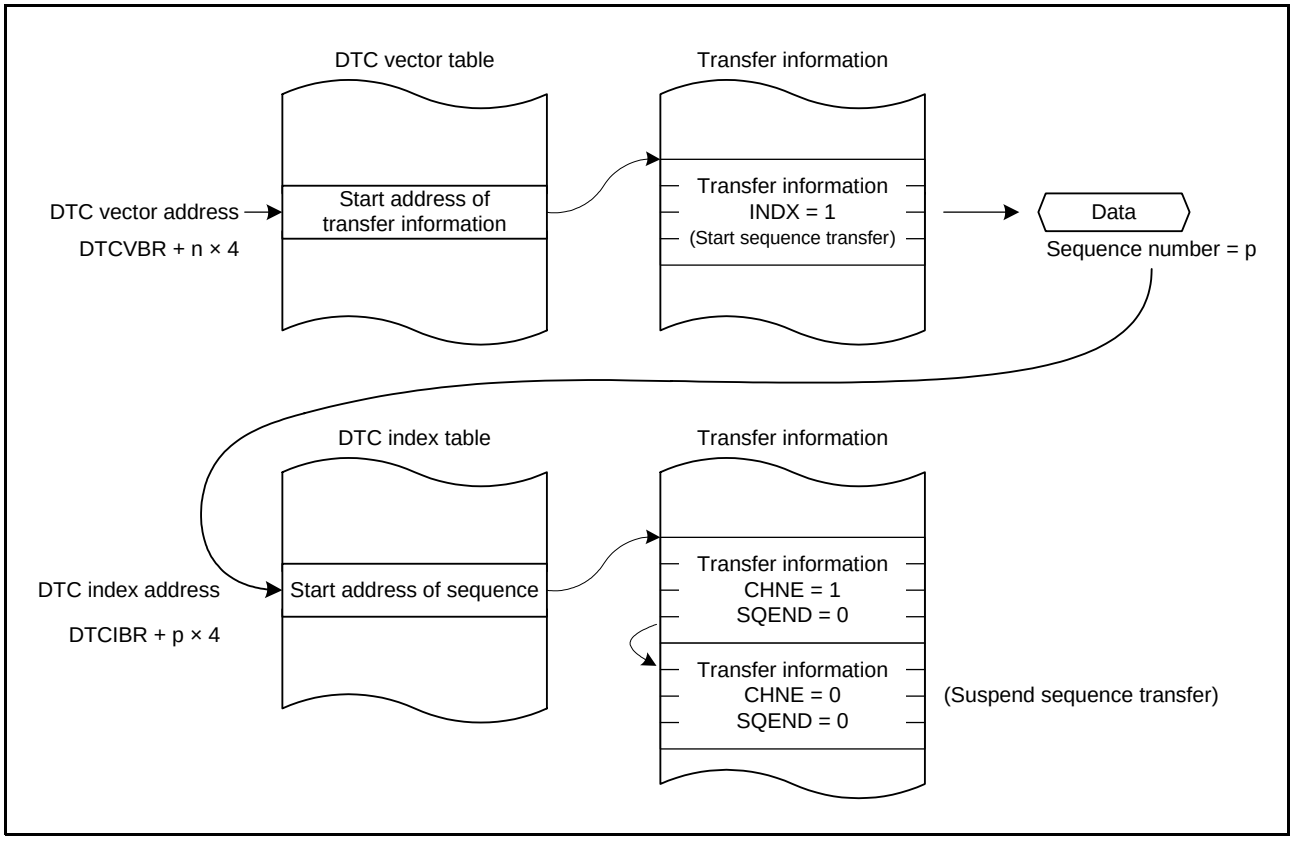


Figure 18.15 Start and Suspension of Sequence Transfer

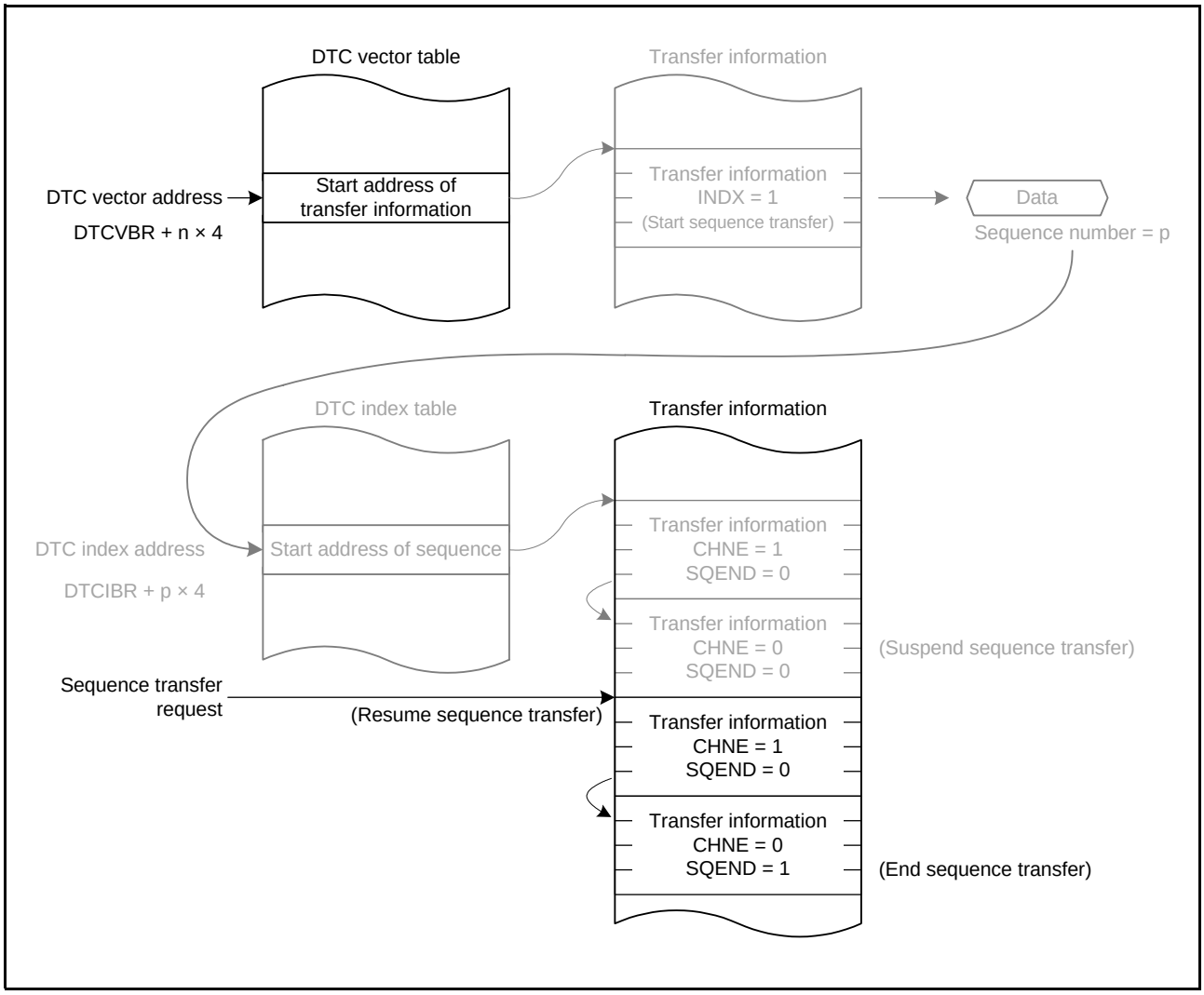


Figure 18.16 Resumption and End of Sequence Transfer

Table 18.10 lists the settings of bits CHNE, SQEND, and INDX during the sequence transfer.

Table 18.10 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.

18.4.11 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.

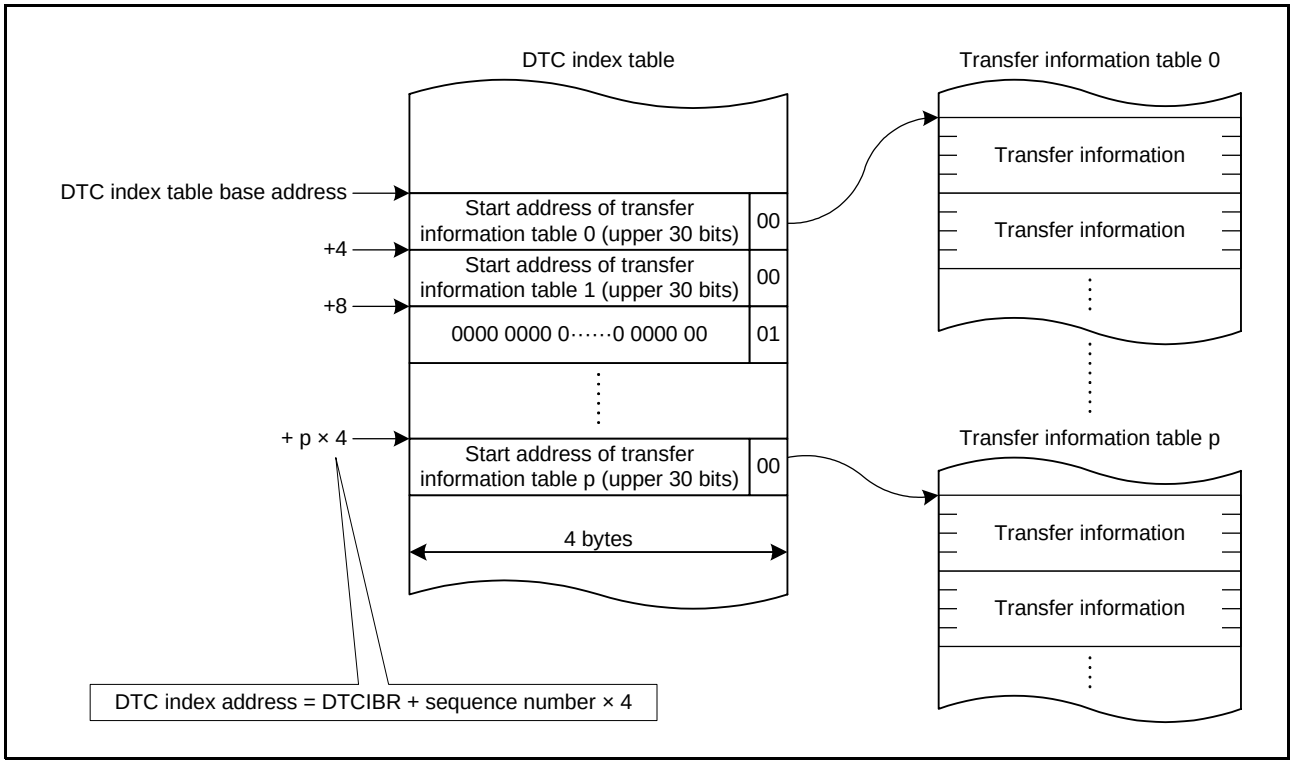
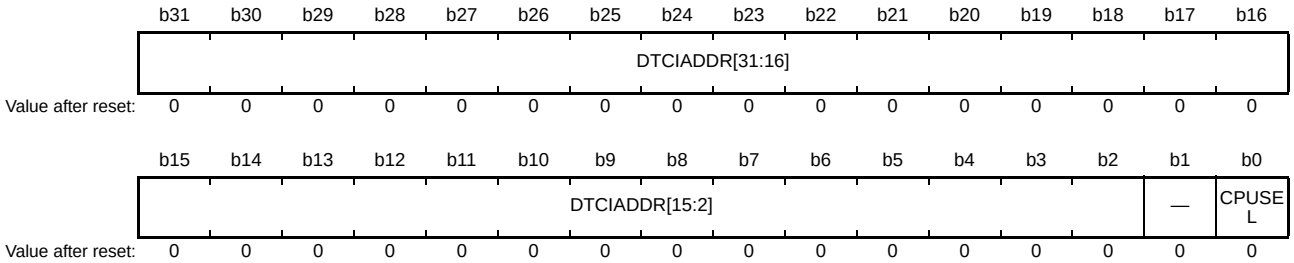


Figure 18.17 DTC Index Table

- 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.

18.4.12 Example of Sequence Transfer

Figure 18.18 shows a typical examples of a sequence transfer and Figure 18.19 to Figure 18.23 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.

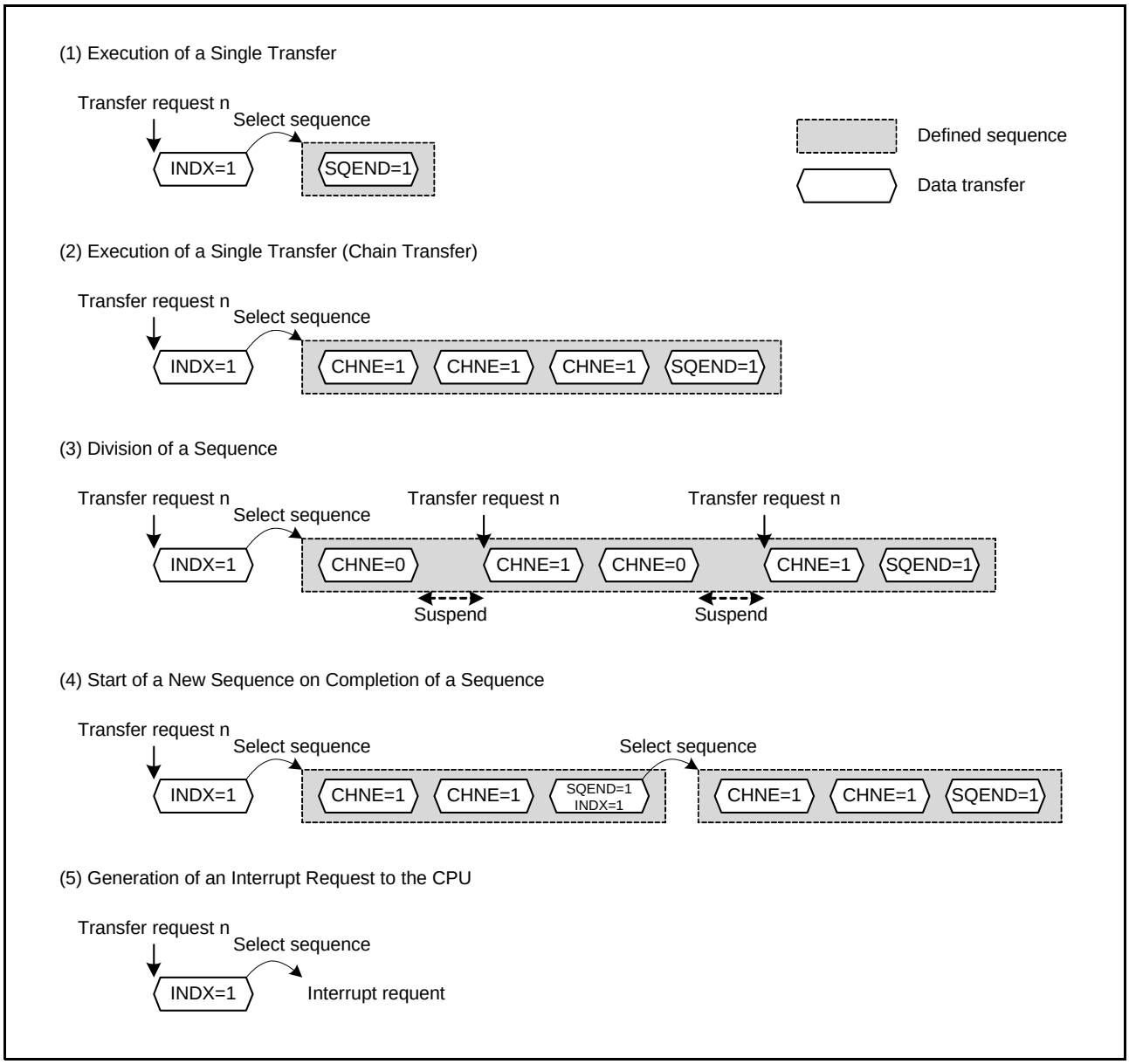


Figure 18.18 Examples of Sequence Transfers

(1) When Executing a Single Transfer

Figure 18.19 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.

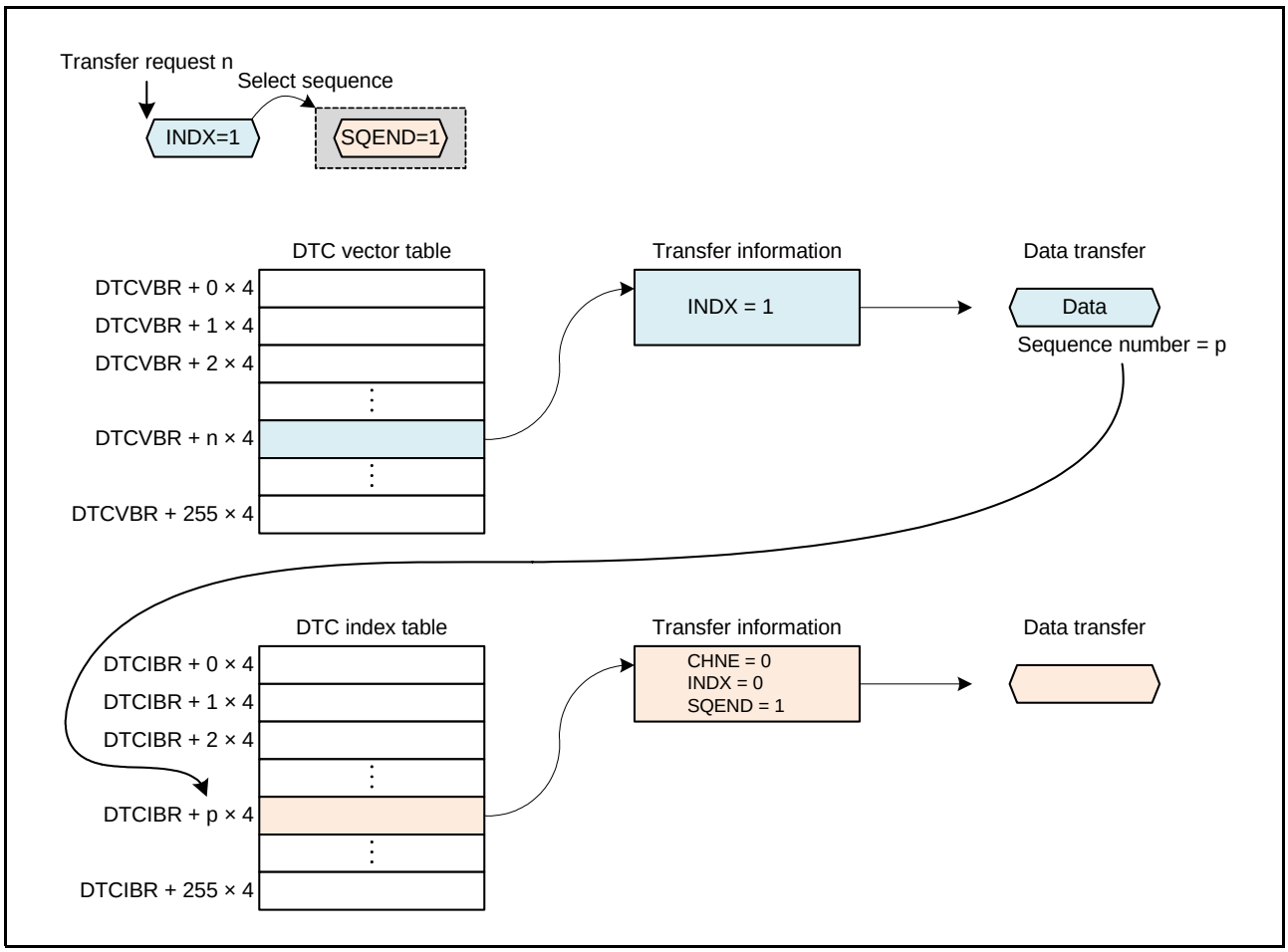


Figure 18.19 Example of a Sequence of Single Transfer

(2) When Executing a Single Chain Transfer

Figure 18.20 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.

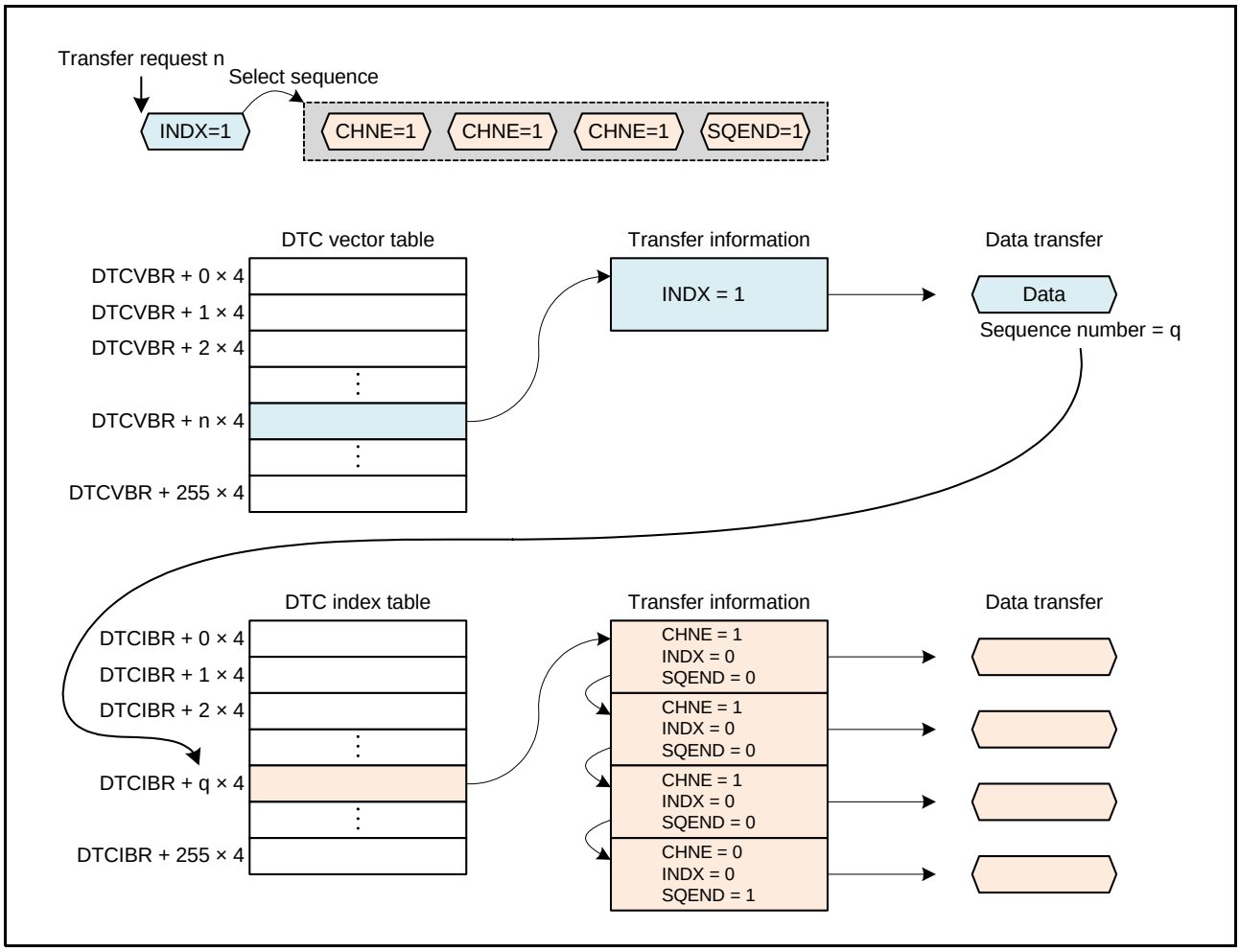


Figure 18.20 Example of Sequence of a Single Chain Transfer

(3) When Dividing a Sequence

Figure 18.21 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.

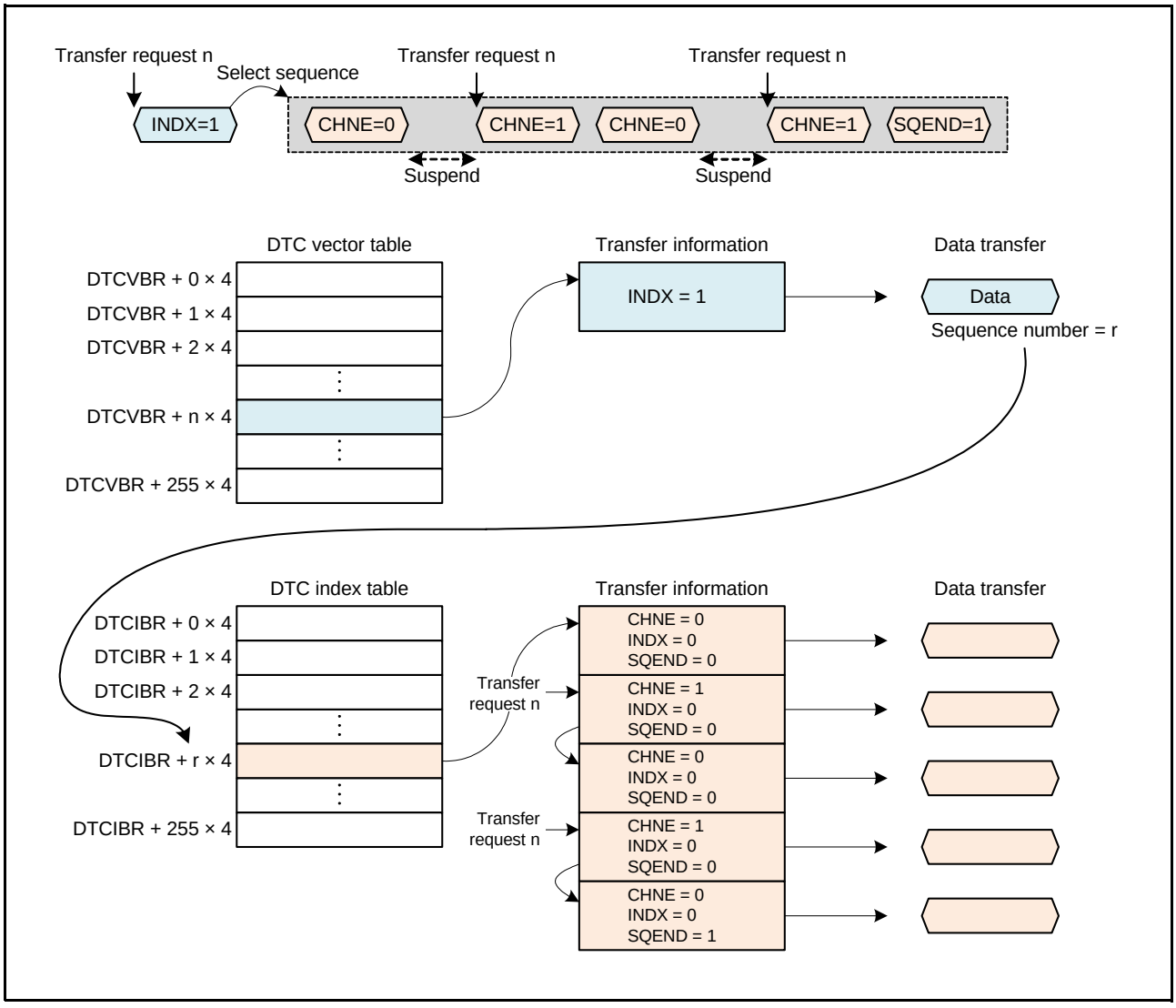


Figure 18.21 Example of Divided Sequence

(4) When Starting a New Sequence on completion of a Sequence

Figure 18.22 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.

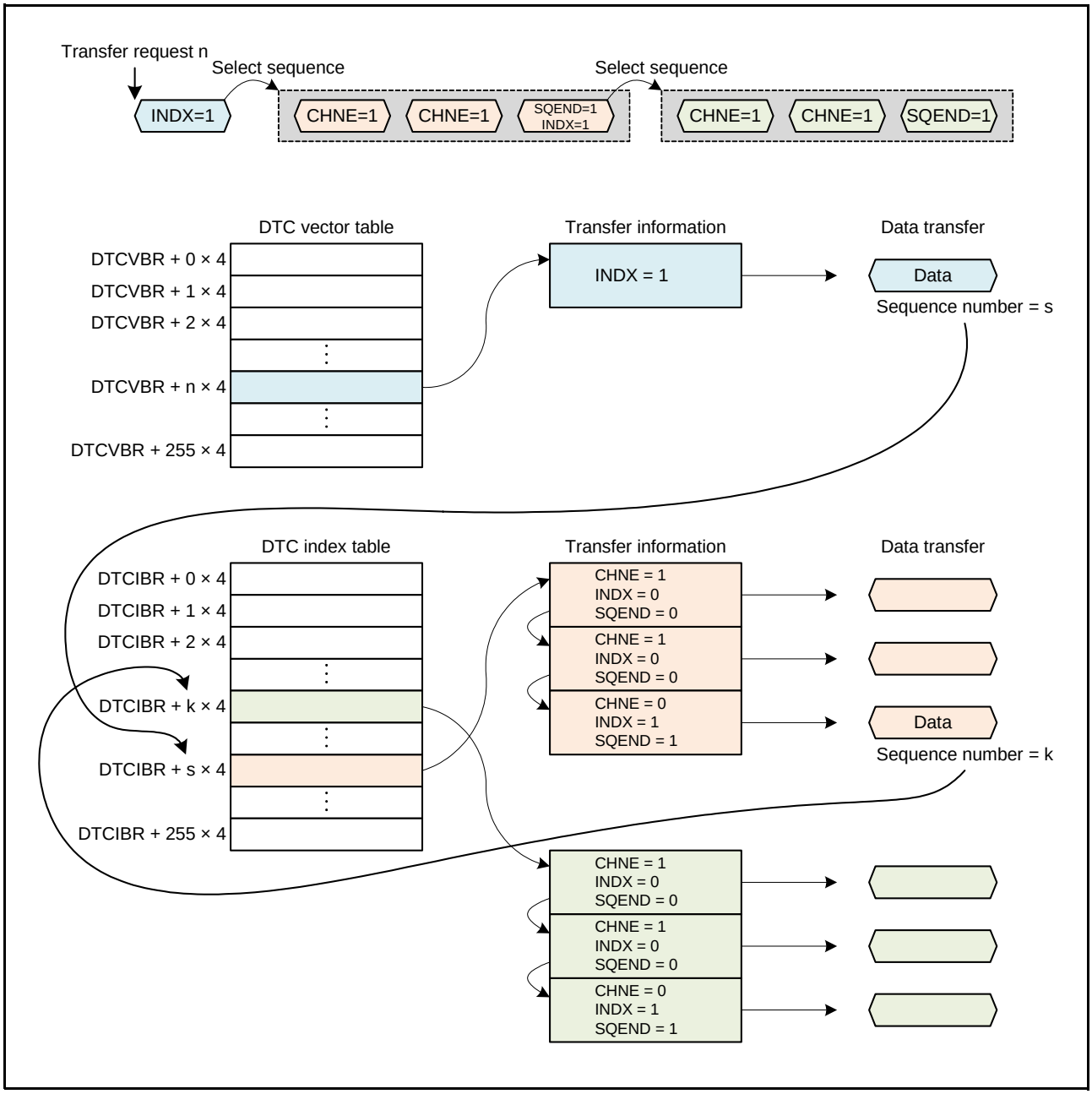


Figure 18.22 Example When Starting a New Sequence on Completion of a Sequence

(5) When Generating an Interrupt Request to the CPU

Figure 18.23 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.

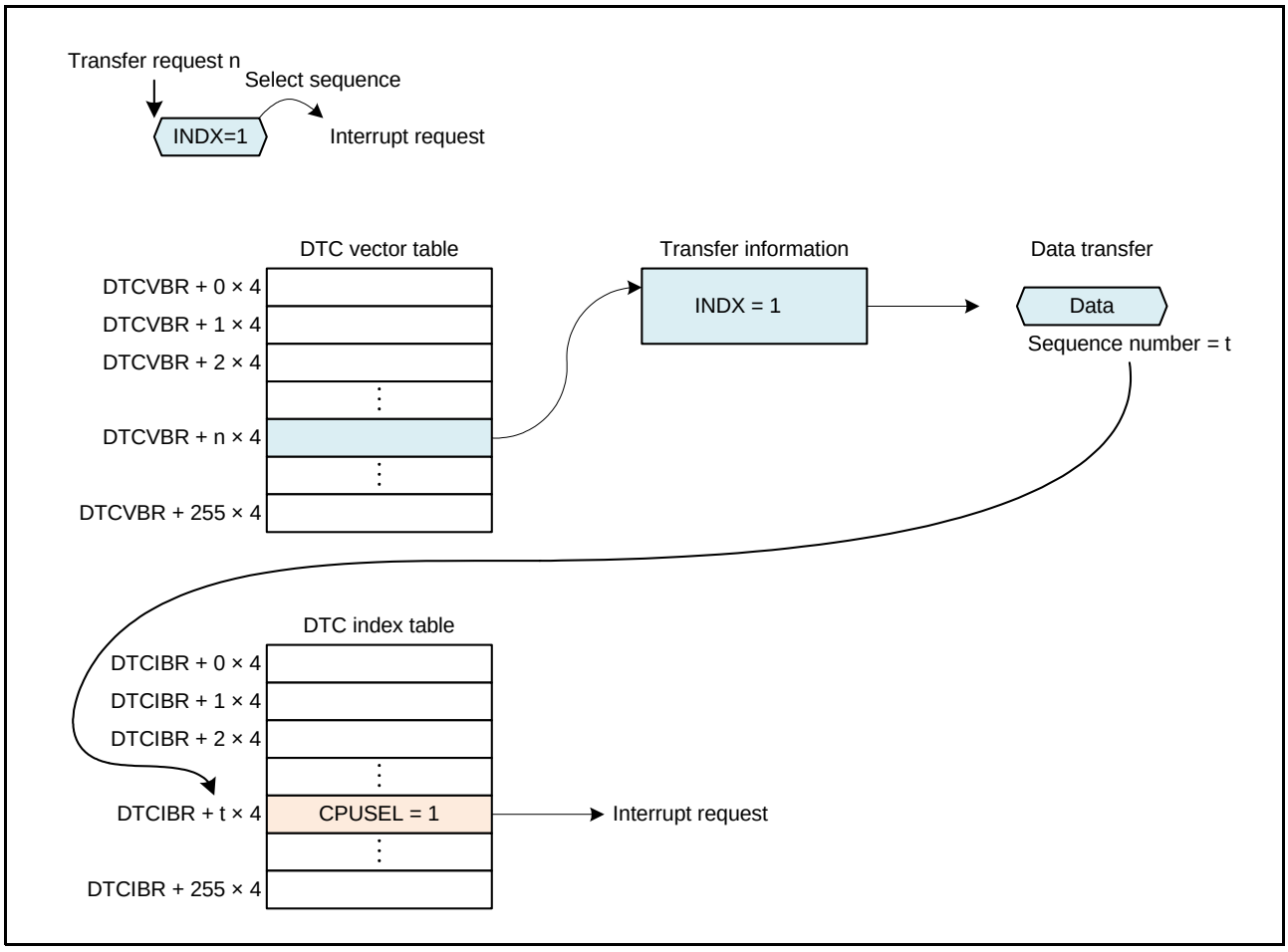


Figure 18.23 Example of Output of an Interrupt Request to the CPU

18.5 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 18.24 shows the procedure to set the DTC.

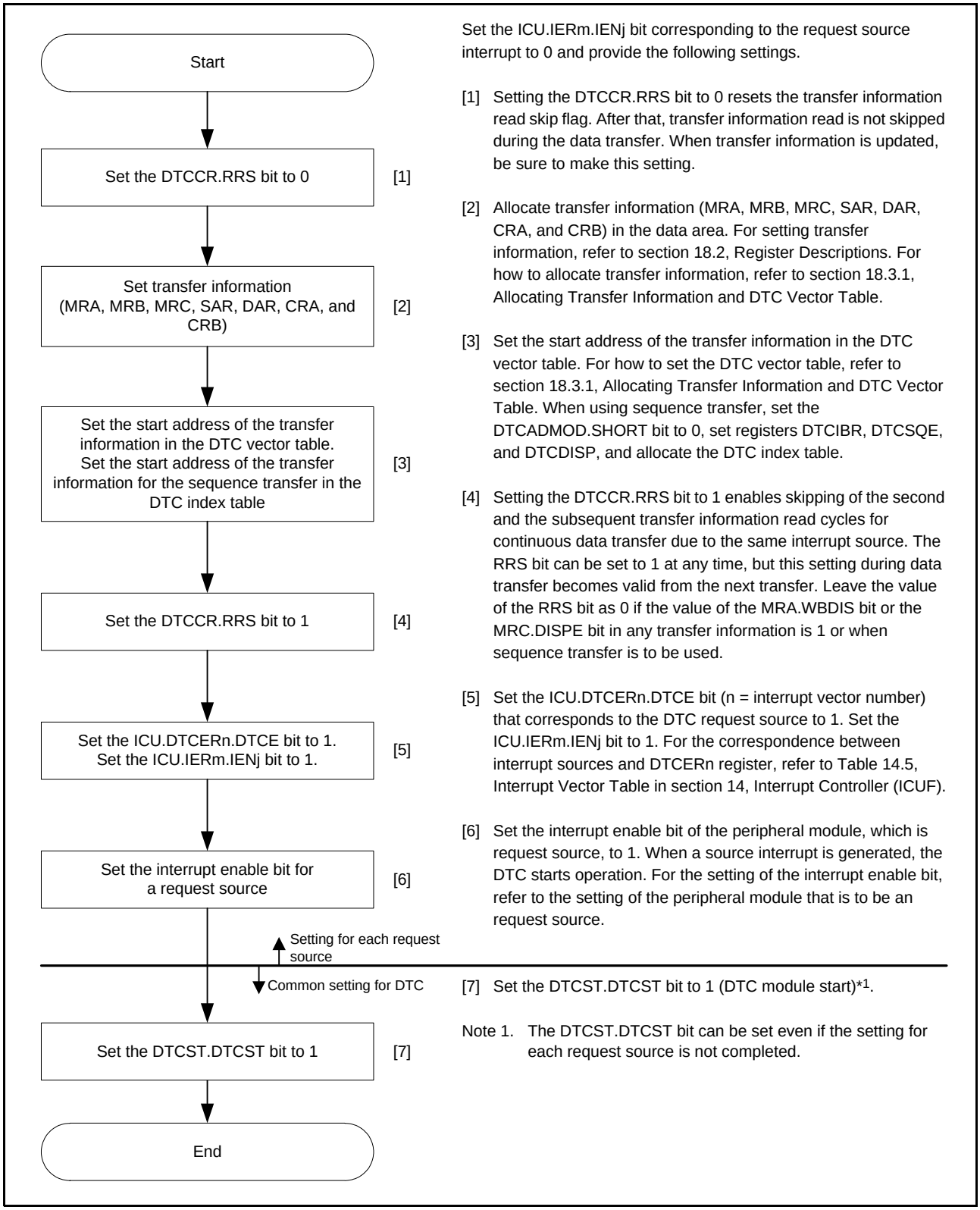


Figure 18.24 Procedure to Set the DTC

18.6 Examples of DTC Usage

18.6.1 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.

18.6.2 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 18.25 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.

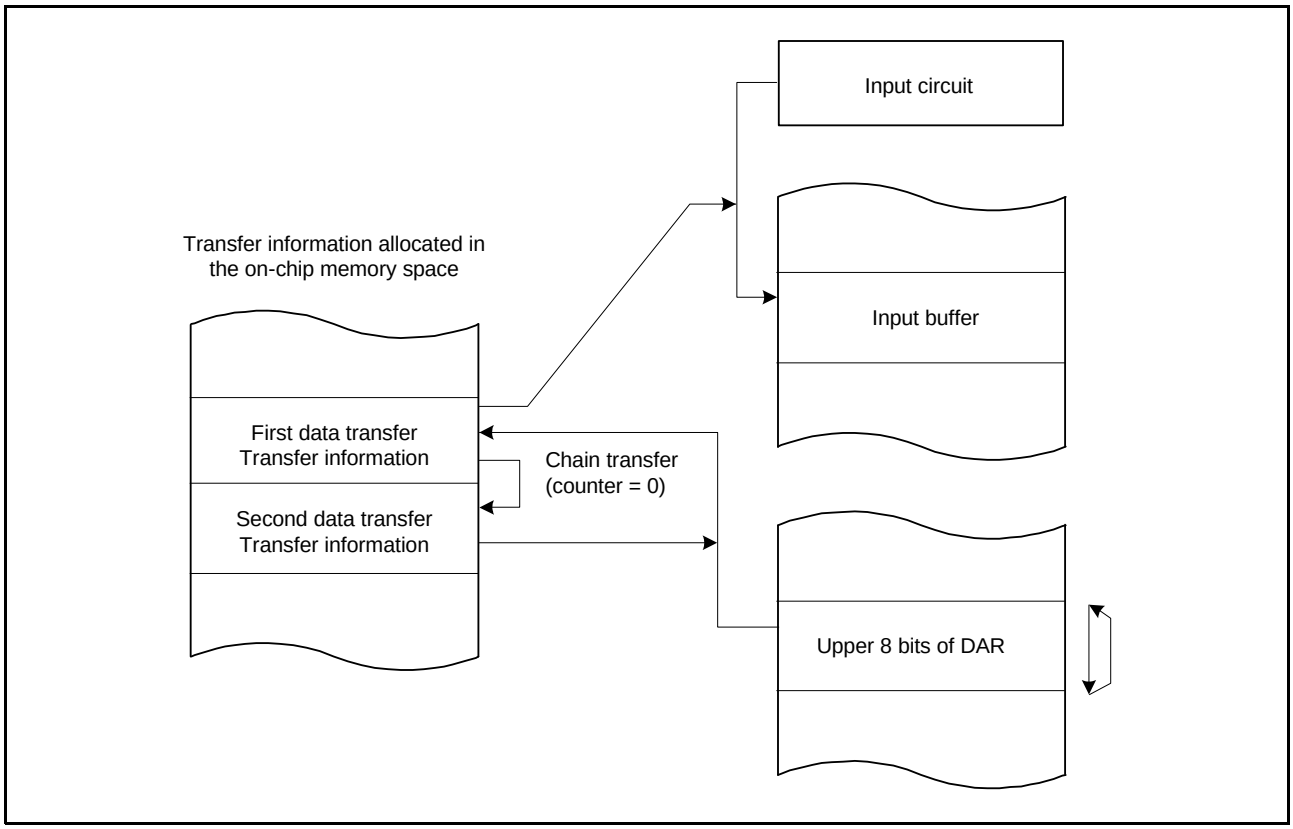


Figure 18.25 Chain Transfer When the Counter is 0

18.6.3 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.

18.7 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.

18.8 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.

18.9 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 11.5.2.1, 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 11.5.3.1, Transition to Software Standby Mode, or section 11.5.4.1, 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 11.6.6, Timing of WAIT Instruction 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 14.7.3.1, Interrupt Request Destination Setting Procedure in section 14, Interrupt Controller (ICUF), and then execute the WAIT instruction.

18.10 Usage Notes

18.10.1 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.

18.10.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 18.26. 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.

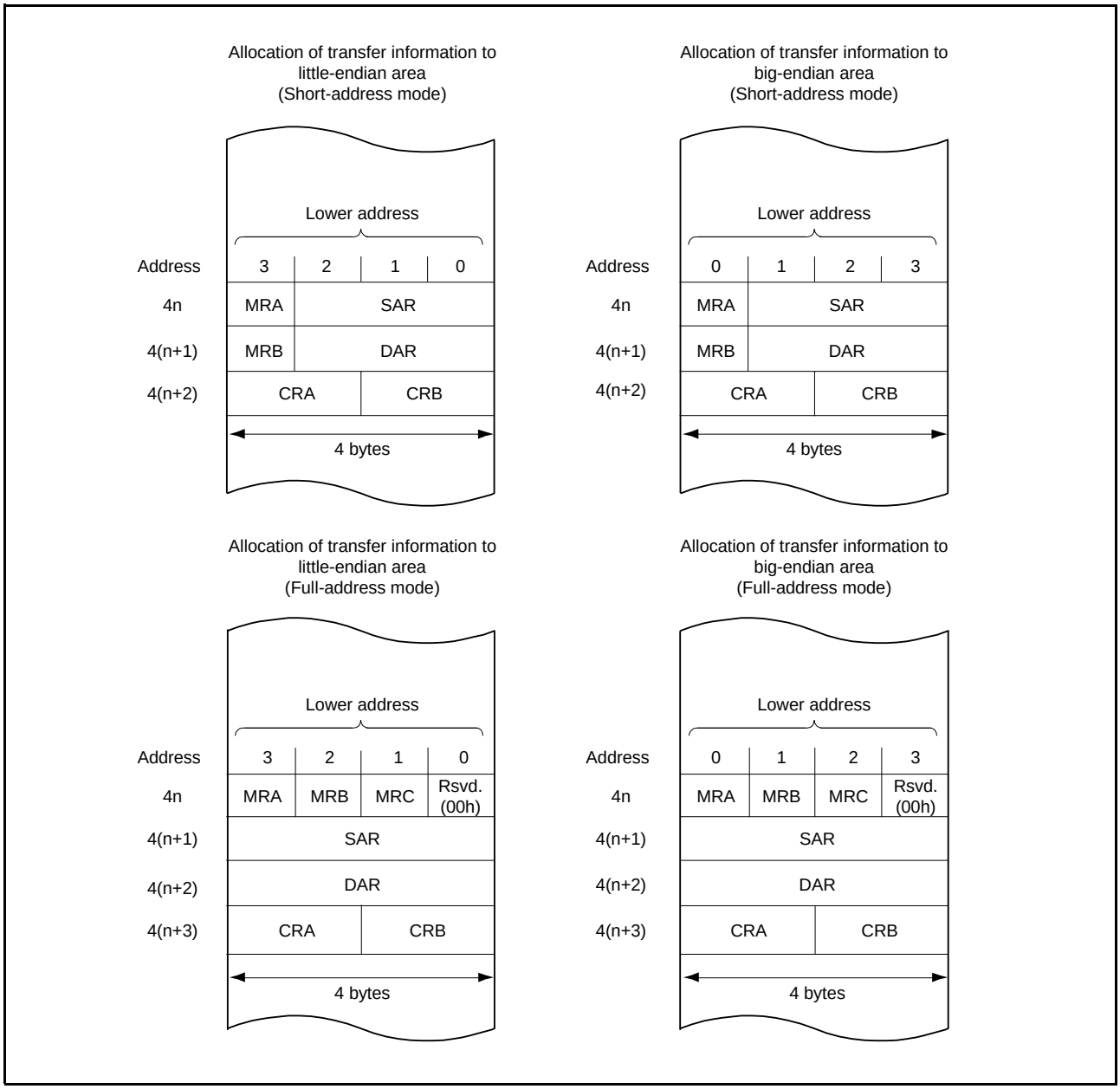


Figure 18.26 Allocation of Transfer Information

18.10.3 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 14, Interrupt Controller (ICUF).

18.10.4 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).

19. Event Link Controller (ELC)

19.1 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 19.1 lists the specifications of the ELC, and Figure 19.1 shows a block diagram of the ELC.

Table 19.1 ELC Specifications

Item	Description
Event link function	• 83 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.

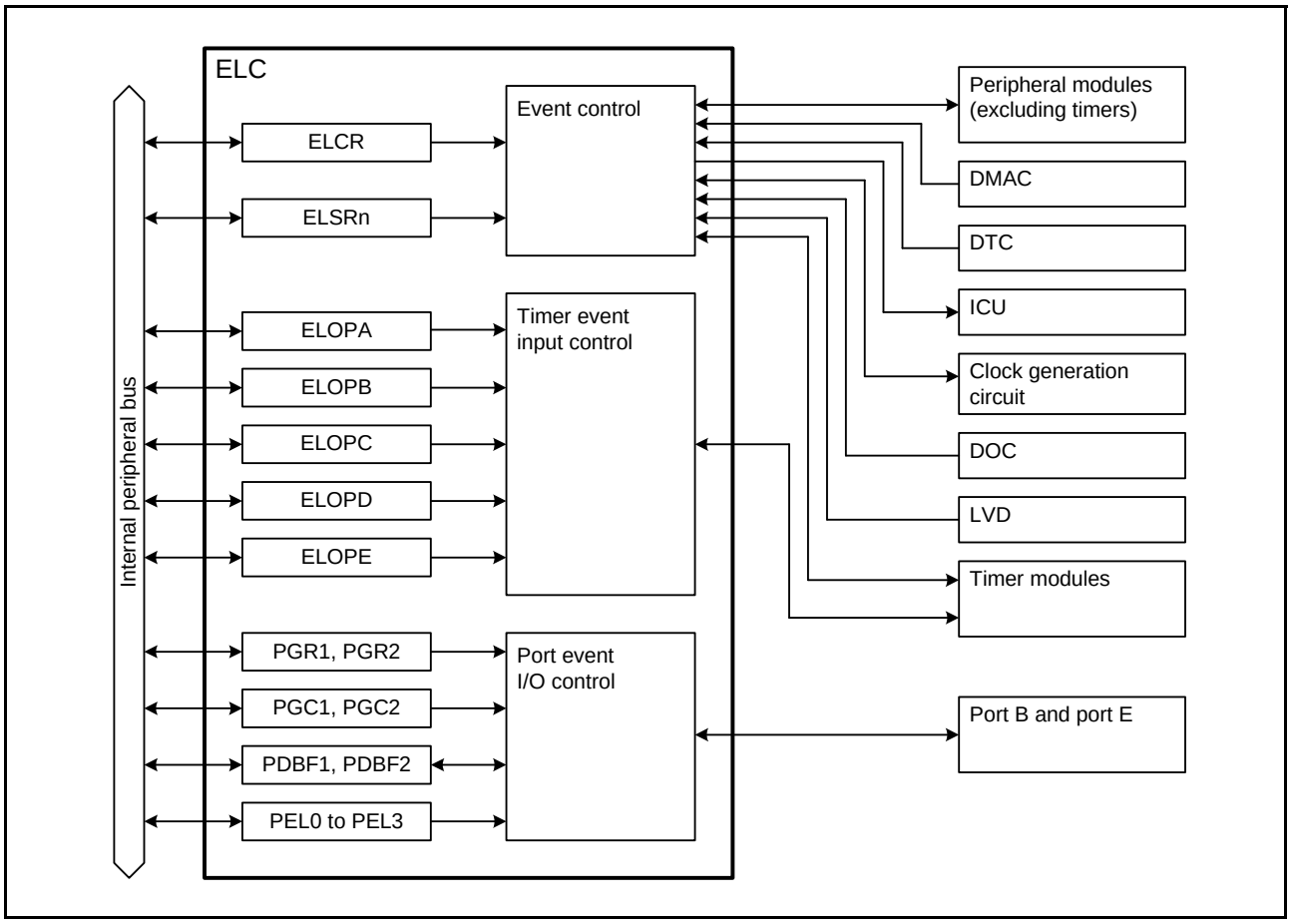


Figure 19.1 ELC Block Diagram (n = 0, 3, 4, 7, 10 to 13, 15, 16, 18 to 28, 30, 31, 32, 56)

19.2 Register Descriptions

19.2.1 Event Link Control Register (ELCR)

Address(es): ELC.ELCR 0008 B100h

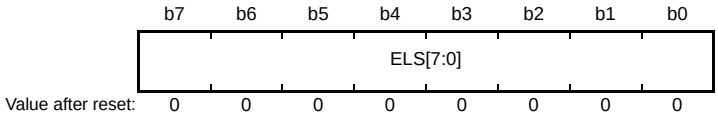
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

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.

19.2.2 Event Link Setting Register n (ELSRn) (n = 0, 3, 4, 7, 10 to 13, 15, 16, 18 to 28, 30, 31, 32, 56)

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.ELSR15 0008 B110h, ELC.ELSR16 0008 B111h, 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.ELSR30 0008 B12Eh, ELC.ELSR31 0008 B12Fh, ELC.ELSR32 0008 B130h, ELC.ELSR56 0008 B14Eh



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 F1h: 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 19.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 19.3 shows the correspondence between values set in the ELSRn register and event signals.

Table 19.2 Correspondence between the ELSRn Register and the Peripheral Modules

Register Name	Peripheral Module
ELSR0	MTU0
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR10	TMR0
ELSR11	TMR1
ELSR12	TMR2
ELSR13	TMR3
ELSR15	S12AD (ELCTRG00N)
ELSR16	DA0
ELSR18	ICU (Interrupt 1)*1
ELSR19	ICU (Interrupt 2)*1
ELSR20	Output port group 1
ELSR21	Output port group 2
ELSR22	Input port group 1
ELSR23	Input port group 2
ELSR24	Single port 0*2
ELSR25	Single port 1*2
ELSR26	Single port 2*2
ELSR27	Single port 3*2
ELSR28	Clock source switching to LOCO
ELSR30	MTU6
ELSR31	MTU7
ELSR32	MTU8
ELSR56	S12AD (ELCTRG01N)

Note 1. Specify an event number from among EAh to F1h. Do not set other values.
Note 2. Do not set the DOC data operation condition met signal (F1h) in the ELSR24, ELSR25, ELSR26, and ELSR27 registers.

Table 19.3 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
1Eh		MTU6 compare match 6A
1Fh		MTU6 compare match 6B
20h		MTU6 compare match 6C
21h		MTU6 compare match 6D
22h		MTU6 overflow
23h		MTU7 compare match 7A
24h		MTU7 compare match 7B
25h		MTU7 compare match 7C
26h		MTU7 compare match 7D
27h		MTU7 overflow
28h		MTU7 underflow
29h		MTU8 compare match 8A
2Ah		MTU8 compare match 8B
2Bh		MTU8 compare match 8C
2Ch		MTU8 compare match 8D
2Dh		MTU8 overflow
37h	Compare match timer	CMT1 compare match 1

Table 19.3 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
3Ch	8-bit timers	TMR0 compare match A0
3Dh		TMR0 compare match B0
3Eh		TMR0 overflow
3Fh		TMR1 compare match A1
40h		TMR1 compare match B1
41h		TMR1 overflow
42h		TMR2 compare match A2
43h		TMR2 compare match B2
44h		TMR2 overflow
45h		TMR3 compare match A3
46h		TMR3 compare match B3
47h		TMR3 overflow
ACh	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)
AFh	Independent watchdog timer	IWDT underflow or refresh error
B8h	Serial communications interfaces	SCI5 error (receive error or error signal detection)
B9h		SCI5 receive data full
BAh		SCI5 transmit data empty
BBh		SCI5 transmit end
CCh	I^2C-bus interface	RIIC0 communication error or event generation
CDh		RIIC0 receive data full
CEh		RIIC0 transmit data empty
CFh		RIIC0 transmit end
D0h	Serial peripheral interface	RSPI0 error (mode fault, overrun, underrun, or parity error)
D1h		RSPI0 idle
D2h		RSPI0 receive buffer full
D3h		RSPI0 transmit buffer empty
D4h		RSPI0 transmit end
D6h	12-bit A/D converter	S12AD A/D conversion end
DCh	Comparator C	Comparison result change of comparator C0
DDh		Comparison result change of comparator C1
DEh		Comparison result change of comparator C2
DFh		Comparison result change of comparator C3
E2h	Voltage detection circuit	LVD1 voltage detection
E3h		LVD2 voltage detection
E4h	DMA controller	DMAC0 transfer end
E5h		DMAC1 transfer end
E6h		DMAC2 transfer end
E7h		DMAC3 transfer end
E8h	Data transfer controller	DTC transfer end
E9h	Clock generation circuit	Oscillation stop detection of clock generation circuit

Table 19.3 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
EAh	I/O ports	Input edge detection of input port group 1
EBh		Input edge detection of input port group 2
ECh		Input edge detection of single input port 0
EDh		Input edge detection of single input port 1
EEh		Input edge detection of single input port 2
EFh		Input edge detection of single input port 3
F0h	Event link controller	Software event
F1h	Data operation circuit	DOC data operation condition met
Settings other than above are prohibited.		

19.2.3 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.

19.2.4 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.

19.2.5 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.

19.2.6 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.

19.2.7 Event Link Option Setting Register E (ELOPE)

Address(es): ELC.ELOPE 0008 B13Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MTU8MD[1:0]		MTU7MD[1:0]		MTU6MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU6MD[1:0]	MTU6 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	MTU7MD[1:0]	MTU7 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	MTU8MD[1:0]	MTU8 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	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MTU6.TCNT value is captured into the MTU6.TGRA register.
Note 2. The MTU7.TCNT value is captured into the MTU7.TGRA register.
Note 3. The MTU8.TCNT value is captured into the MTU8.TGRA register.

The ELOPE register specifies the operations of MTU6, MTU7, and MTU8 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.8 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 19.4 shows the PGRn register and corresponding ports.

Table 19.4 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

19.2.9 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 19.4 for the PGCn register and corresponding ports.

19.2.10 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 19.3, 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 19.3.5, I/O Port Operation When Event Signal is Input and Event Generation for the PDBFn register operations. Refer to Table 19.4 for the PDBFn register and corresponding ports.

19.2.11 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.

19.2.12 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.

19.3 Operation

19.3.1 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 19.2 shows the relation between the interrupt handling and ELC.

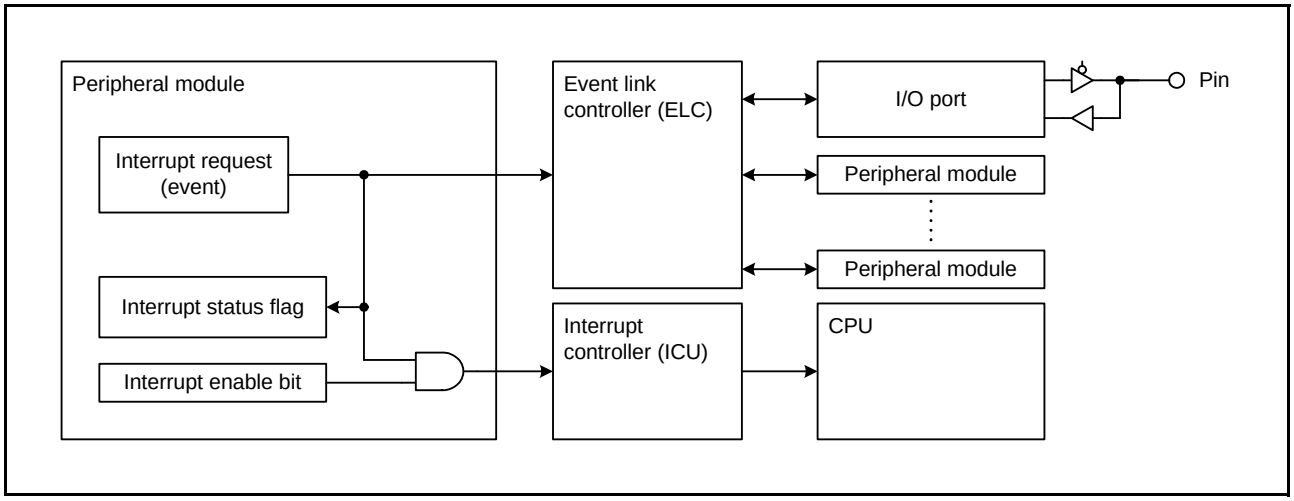


Figure 19.2 Relation between Interrupt Handling and ELC

19.3.2 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 19.5 lists the operations of peripheral modules when an event signal is input.

Table 19.5 Operations of Peripheral Modules When Event Signal is Input

Peripheral Module	Operations When Event Signal is Input		
MTU CMT TMR	The following operations can be selected by setting the ELOPA to ELOPE registers: • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events (CMT, TMR). • Performs input-capture operation when an event signal is input (MTU).		
A/D converter	Starts A/D conversion when an event signal is input.		
D/A converter	Starts D/A conversion when an event signal is input.		
I/O ports (output)	The value of PODR 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).

19.3.3 Operation of Peripheral Timer Modules When Event Signal is Input

For the timer modules, set the ELOPA to ELOPE 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.

19.3.4 Operation of A/D and D/A Converters When Event Signal is Input

When an event signal is input, the ADCSR.ADST bit and the DACR.DAOE0 bit*1 are set to 1 and the A/D and D/A converter start A/D and D/A conversion, respectively.

Note 1. Refer to the bit descriptions in the A/D converter and D/A converter sections.

19.3.5 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 19.3 (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 19.3 (2).

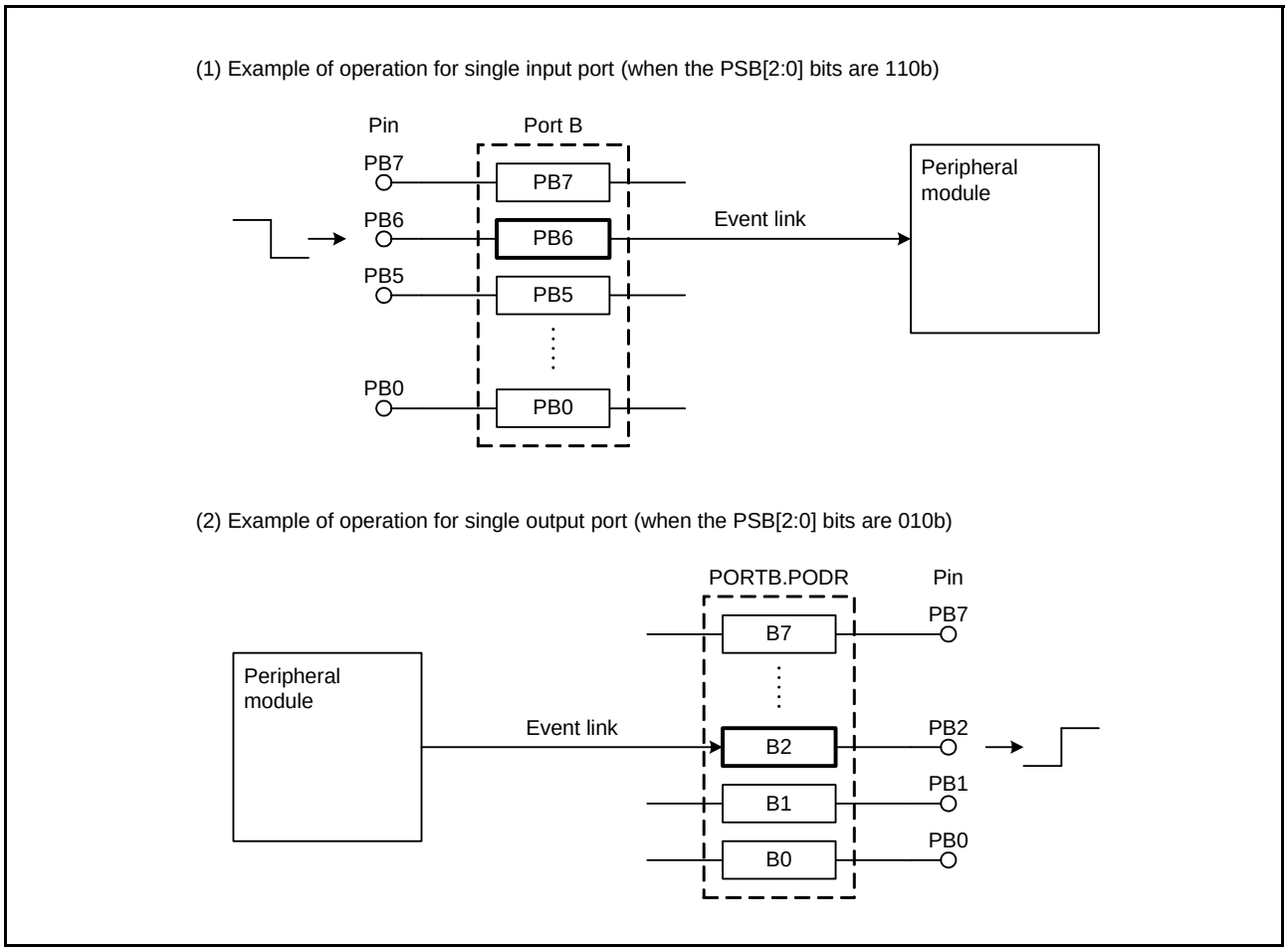


Figure 19.3 Event Linkage Related to Single Ports (Port B)

(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 19.4.

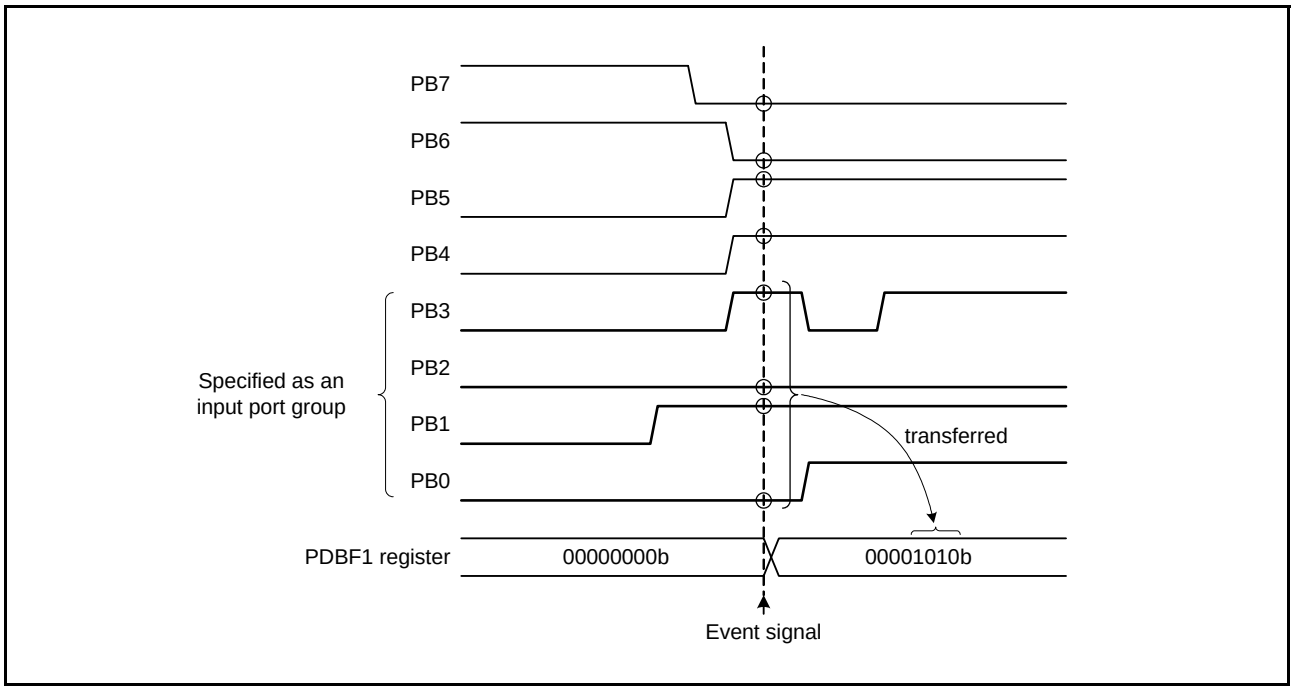


Figure 19.4 Event Linkage Related to Input Port Groups (Port B)

(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 19.5.

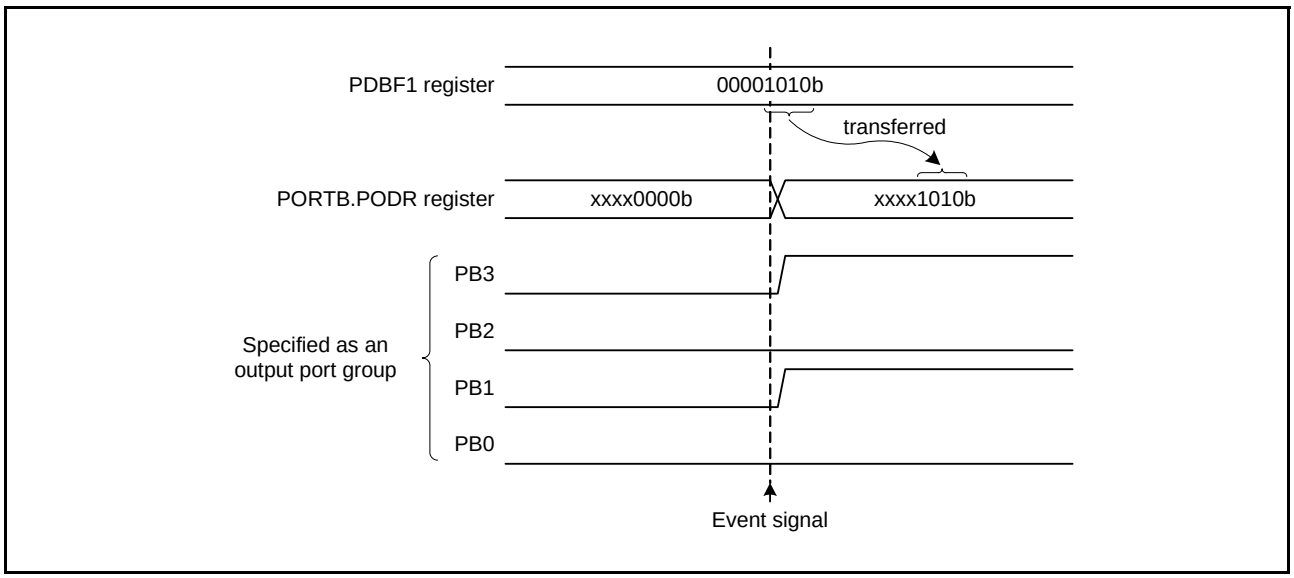


Figure 19.5 Event Linkage Related to Output Port Groups (Port B)

(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 19.6.

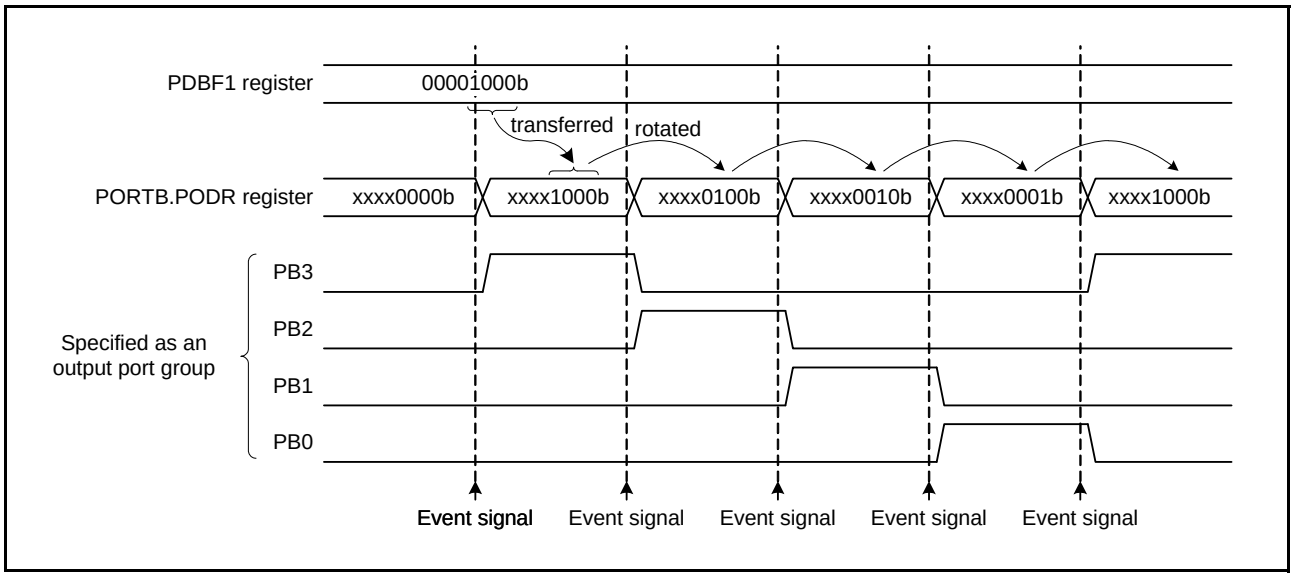


Figure 19.6 Bit-Rotating Operation of Output Port Groups (Port B)

(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.

19.3.6 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 ELOPE 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: 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.

19.4 Usage Notes

19.4.1 Setting ELSRn Register

(1) Setting ELSR18 and ELSR19 Registers

Specify an event number from among EAh to F1h. 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 (F1h).

19.4.2 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.

19.4.3 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.

19.4.4 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).

19.4.5 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.

20. I/O Ports

20.1 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) that selects the drive capacity, and the port mode register (PMR) that specifies the pin function of each port.

For details on PMR, refer to section 21, Multi-Function Pin Controller (MPC).

The configuration of the I/O ports differs depending on the package. Table 20.1 shows the specifications of I/O ports, Table 20.2 lists the port functions.

Table 20.1 Specifications of I/O Ports

Port	Package 144 Pins	Number of Pin	Package 100 Pins	Number of Pin	Package 80 Pins	Number of Pin	Package 64 Pins	Number of Pin	Package 48 Pins	Number of Pin
PORT0	P00 to P07	8	P03*1 to P07	5	P03 to P07	5	P03, P07	2	Not provided	0
PORT1	P12 to P17	6	P12 to P17	6	P12 to P17	6	P14 to P17	4	P14 to P17	4
PORT2	P20 to P27	8	P20 to P27	8	P20, P21, P26, P27	4	P26, P27	2	P26, P27	2
PORT3	P30 to P37	8	P30 to P37	8	P30 to P32, P34 to P37	7	P30 to P32, P35 to P37	6	P30, P31, P35 to P37	5
PORT4	P40 to P47	8	P40 to P47	8	P40 to P47	8	P40 to P47	8	P40 to P42, P45 to P47	6
PORT5	P50 to P56	7	P50 to P55	6	P54, P55	2	P54, P55	2	Not provided	0
PORT6	P60 to P67	8	Not provided	0	Not provided	0	Not provided	0	Not provided	0
PORT7	P70 to P77	8	Not provided	0	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	Not provided	0
PORT9	P90 to P93	4	Not provided	0	Not provided	0	Not provided	0	Not provided	0
PORTA	PA0 to PA7	8	PA0 to PA7	8	PA0 to PA6	7	PA0, PA1, PA3, PA4, PA6	5	PA1, PA3, PA4, PA6	4
PORTB	PB0 to PB7	8	PB0 to PB7	8	PB0 to PB7	8	PB0, PB1, PB3, PB5 to PB7	6	PB0, PB1, PB3, PB5	4
PORTC	PC0 to PC7	8	PC0 to PC7	8	PC2 to PC7	6	PC2 to PC7	6	PC4 to PC7	4
PORTD	PD0 to PD7	8	PD0 to PD7	8	PD0 to PD2	3	Not provided	0	Not provided	0
PORTE	PE0 to PE7	8	PE0 to PE7	8	PE0 to PE5	6	PE0 to PE5	6	PE1 to PE4	4
PORTF	PF5 to PF7	3	Not provided	0	Not provided	0	Not provided	0	Not provided	0
PORTH	PH0 to PH3, PH6*2, PH7*2	6	PH0 to PH3, PH6*2, PH7*2	6	PH0 to PH3, PH6*2, PH7*2	6	PH0 to PH3, PH6*2, PH7*2	6	PH0 to PH3	4
PORTJ	PJ1, PJ3 to PJ7	6	PJ1, PJ3, PJ6, PJ7	4	PJ1, PJ6, PJ7	3	PJ6, PJ7	2	PJ6, PJ7	2
PORTK	PK2 to PK5	4	Not provided	0	Not provided	0	Not provided	0	Not provided	0
PORTL	PL0, PL1	2	Not provided	0	Not provided	0	Not provided	0	Not provided	0
PORTN	PN6, PN7*3	2	PN6	1	PN6	1	PN6	1	PN6	1
	Total of pins	134	Total of pins	92	Total of pins	72	Total of pins	56	Total of pins	40

Note 1. Products with a JTAG interface do not have the P03 pin.
Note 2. Products with a sub-clock oscillator do not have the PH6 and PH7 pins.
Note 3. Products with a JTAG interface do not have the PN7 pin.

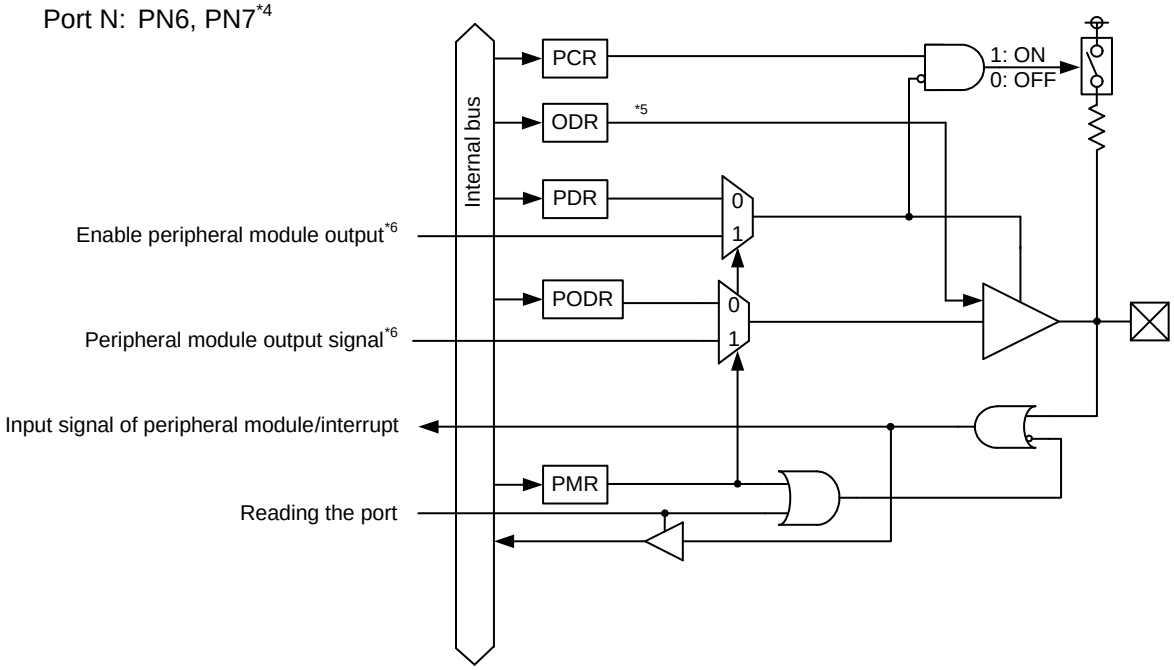
Table 20.2 Port Functions

Port	Pin	Input Pull-up	Open-Drain Output	Drive Capacity Switching	5-V Tolerant
PORT0	P00 to P02	✓	✓	Normal drive/high drive	—
	P03	✓	✓	Fixed to normal output	—
	P04	✓	✓	Normal drive/high drive	—
	P05 to P07	✓	✓	Fixed to normal output	—
PORT1	P12, P13	✓	✓	Normal drive/high drive	✓
	P14, P15	✓	✓	Normal drive/high drive	—
	P16, P17	✓	✓	Normal drive/high drive	✓
PORT2	P20 to P27	✓	✓	Normal drive/high drive	—
PORT3	P30 to P34	✓	✓	Normal drive/high drive	—
	P35	—	—	—	—
	P36, P37	✓	✓	Fixed to normal output	—
PORT4	P40 to P47	✓	✓	Fixed to normal output	—
PORT5	P50 to P56	✓	✓	Normal drive/high drive	—
PORT6	P60 to P67	✓	✓	Normal drive/high drive	—
PORT7	P70 to P77	✓	✓	Normal drive/high drive	—
PORT8	P80 to P83, P86, P87	✓	✓	Normal drive/high drive	—
PORT9	P90 to P93	✓	✓	Normal drive/high drive	—
PORTA	PA0 to PA7	✓	✓	Normal drive/high drive	—
PORTB	PB0 to PB7	✓	✓	Normal drive/high drive	—
PORTC	PC0 to PC7	✓	✓	Normal drive/high drive	—
PORTD	PD0 to PD7	✓	✓	Normal drive/high drive	—
PORTE	PE0 to PE7	✓	✓	Normal drive/high drive	—
PORTF	PF5 to PF7	✓	✓	Normal drive/high drive	—
PORTH	PH0 to PH3, PH6, PH7	✓	✓	Normal drive/high drive	—
PORTJ	PJ1, PJ3 to PJ5	✓	✓	Normal drive/high drive	—
	PJ6, PJ7	✓	✓	Fixed to normal output	—
PORTK	PK2 to PK5	✓	✓	Normal drive/high drive	—
PORTL	PL0, PL1	✓	✓	Normal drive/high drive	—
PORTN	PN6, PN7	✓	✓	Normal drive/high drive	—

Specifying input pull-up, open-drain output, switching of drive capacity, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

20.2 I/O Port Configuration

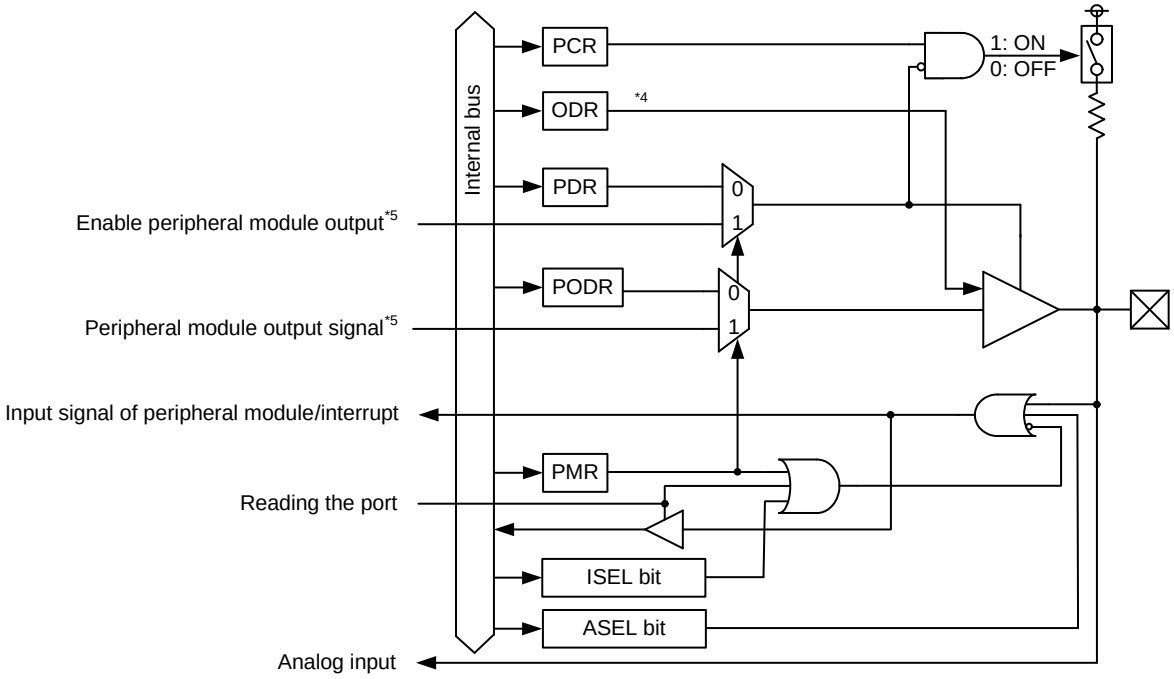
Port 0: P04*1, P06*1
Port F: PF6*2, PF7*2
Port H: PH0, PH3, PH6*3, PH7*3
Port J: PJ1*1, PJ4*2, PJ6, PJ7
Port K: PK2*2, PK3*2, PK4*2, PK5*2
Port L: PL0*2, PL1*2
Port N: PN6, PN7*4



Note 1. Not provided on the 64-pin LFQFP and 48-pin LFQFP
Note 2. Only 144-pin LFQFP products have this pin.
Note 3. Products with a sub-clock oscillator do not have this pin.
Note 4. Only products without a JTAG interface and in a 144-pin LFQFP have this pin.
Note 5. Control signal for N-channel open-drain output
Note 6. Output from peripheral modules is multiplexed on this pin.

Figure 20.1 I/O Port Configuration (1)

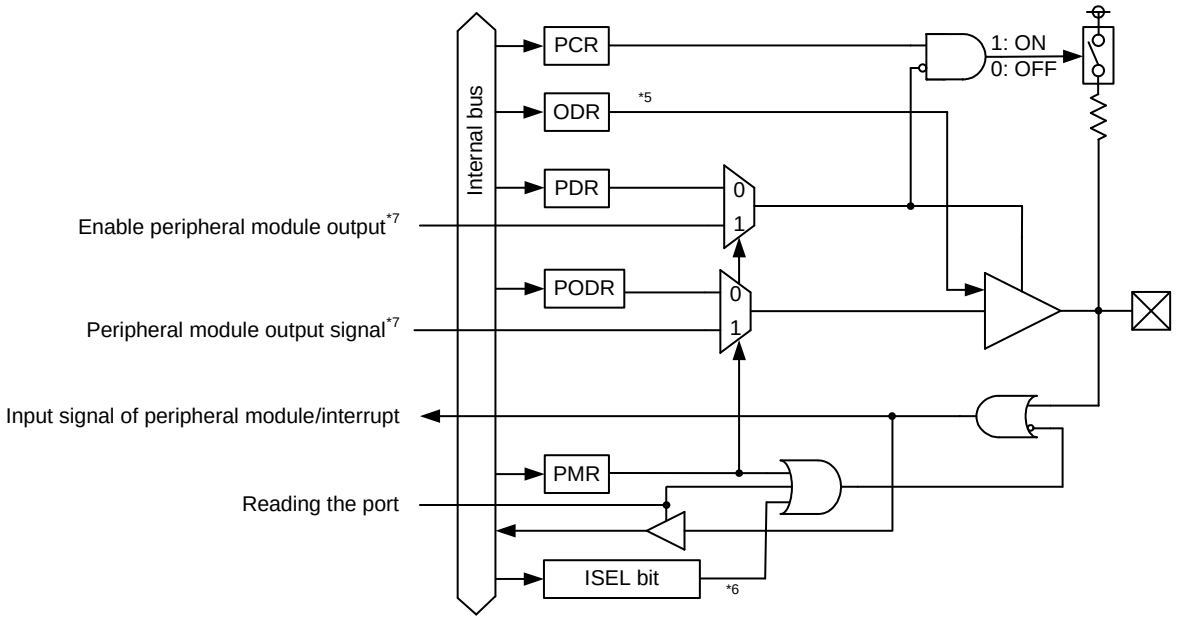
Port 0: P03[*1, *2], P05[*2, *3]
Port 1: P14
Port 4: P40 to P42, P43[*2], P44[*2], P45 to P47



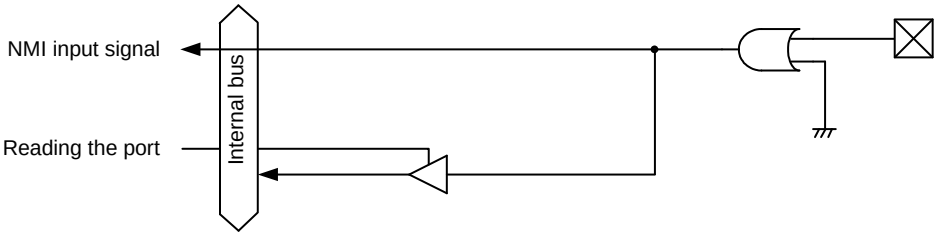
Note 1. Not provided on the 100-pin LFQFP products with a JTAG interface.
Note 2. Not provided on the 48-pin LFQFP
Note 3. Not provided on the 64-pin LFQFP
Note 4. Control signal for N-channel open-drain output
Note 5. Output from peripheral modules is multiplexed on this pin.

Figure 20.2 I/O Port Configuration (2)

Port 0: P00[*1], P01[*1], P02[*1], P07[*2]
Port 1: P12[*3], P13[*3], P16, P17
Port 2: P20[*3], P21[*3], P22[*4], P23[*4]
Port 3: P30, P31, P32[*2], P33[*4], P34[*3], P36, P37
Port 5: P56[*1]
Port 6: P65[*1], P66[*1], P67[*1]
Port 7: P70[*1], P75[*1], P76[*1], P77[*1]
Port 8: P80[*1], P81[*1], P82[*1], P83[*1], P86[*1], P87[*1]
Port F: PF5[*1]
Port H: PH1, PH2
Port J: PJ3[*4], PJ5[*1]



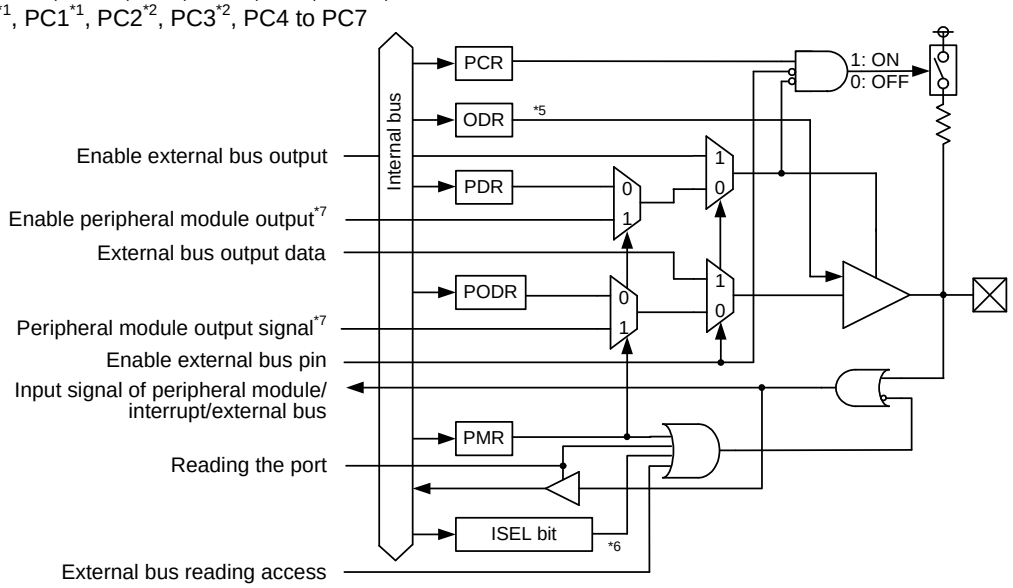
Port 3: P35



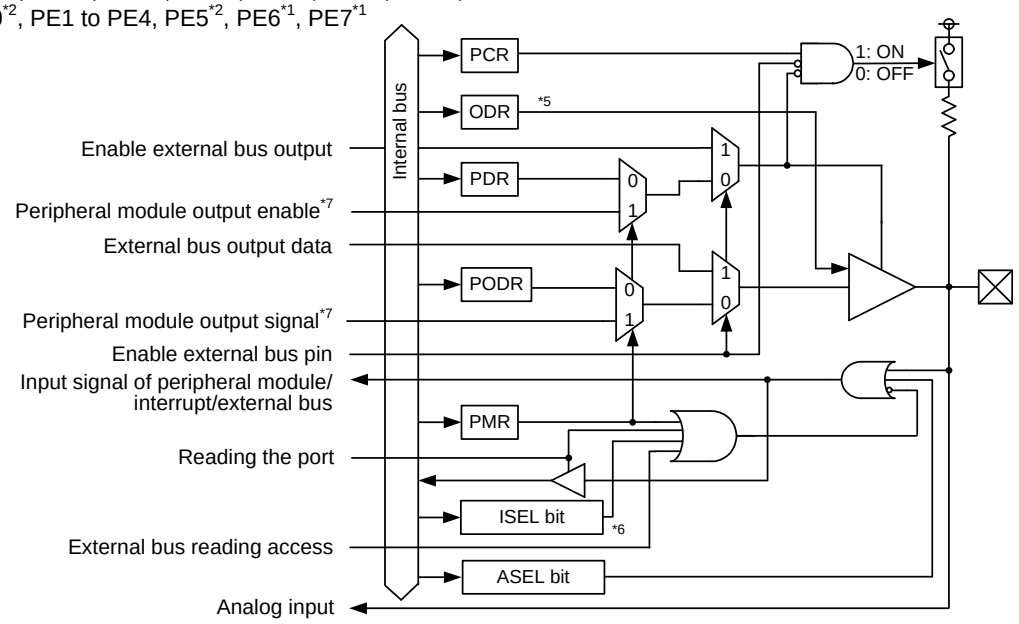
Note 1. Only 144-pin LFQFP products have this pin.
Note 2. Not provided on the 48-pin LFQFP
Note 3. Not provided on the 64-pin LFQFP and 48-pin LFQFP
Note 4. Not provided on the 80-pin LFQFP, 64-pin LFQFP and 48-pin LFQFP
Note 5. Control signal for N-channel open-drain output
Note 6. An external interrupt function is multiplexed on this pin.
Note 7. Output from peripheral modules is multiplexed on this pin.

Figure 20.3 I/O Port Configuration (3)

Port 2: P24[*1], P25[*1]
Port 5: P50[*1], P51[*1], P52[*1], P53[*1], P54[*2], P55[*2]
Port 6: P60[*3], P61[*3], P62[*3], P63[*3], P64[*3]
Port 7: P71[*3], P72[*3], P73[*3], P74[*3]
Port 9: P90[*3], P91[*3], P92[*3], P93[*3]
Port A: PA0[*2], PA1, PA2[*4], PA5[*4], PA6, PA7[*1]
Port B: PB0, PB1, PB2[*4], PB3, PB4[*4], PB5, PB6[*2], PB7[*2]
Port C: PC0[*1], PC1[*1], PC2[*2], PC3[*2], PC4 to PC7



Port 1: P15
Port 2: P26, P27
Port A: PA3, PA4
Port D: PD0[*4], PD1[*4], PD2[*4], PD3[*1], PD4[*1], PD5[*1], PD6[*1], PD7[*1]
Port E: PE0[*2], PE1 to PE4, PE5[*2], PE6[*1], PE7[*1]



Note 1. Not provided on the 80-pin LFQFP, 64-pin LFQFP and 48-pin LFQFP
Note 2. Not provided on the 48-pin LFQFP
Note 3. Only 144-pin LFQFP products have this pin.
Note 4. Not provided on the 64-pin LFQFP and 48-pin LFQFP
Note 5. Control signal for N-channel open-drain output
Note 6. An external interrupt function is multiplexed on this pin.
Note 7. Output from peripheral modules is multiplexed on this pin.

Figure 20.4 I/O Port Configuration (4)

20.3 Register Descriptions

20.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT5.PDR 0008 C005h, PORT6.PDR 0008 C006h, PORT7.PDR 0008 C007h, PORT8.PDR 0008 C008h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTF.PDR 0008 C00Fh, PORTH.PDR 0008 C011h, PORTJ.PDR 0008 C012h, PORTK.PDR 0008 C013h, PORTL.PDR 0008 C014h, PORTN.PDR 0008 C016h

	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, J to L, and N

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 20.4, Initialization of the Port Direction Register (PDR).

The B5 bit in PORT3.PDR is reserved, because the P35 pin is input only.

20.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORT6.PODR 0008 C026h, PORT7.PODR 0008 C027h, PORT8.PODR 0008 C028h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTF.PODR 0008 C02Fh, PORTH.PODR 0008 C031h, PORTJ.PODR 0008 C032h, PORTK.PODR 0008 C033h, PORTL.PODR 0008 C034h, PORTN.PODR 0008 C036h

	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, J to L, and N

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.

20.3.3 Port Input Register (PIDR)

Address(es): PORT0.PIDR 0008 C040h, PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORT5.PIDR 0008 C045h, PORT6.PIDR 0008 C046h, PORT7.PIDR 0008 C047h, PORT8.PIDR 0008 C048h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTF.PIDR 0008 C04Fh, PORTH.PIDR 0008 C051h, PORTJ.PIDR 0008 C052h, PORTK.PIDR 0008 C053h, PORTL.PIDR 0008 C054h, PORTN.PIDR 0008 C056h

	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, J to L, and N

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.

20.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORT5.PMR 0008 C065h, PORT6.PMR 0008 C066h, PORT7.PMR 0008 C067h, PORT8.PMR 0008 C068h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTF.PMR 0008 C06Fh, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h, PORTK.PMR 0008 C073h, PORTL.PMR 0008 C074h, PORTN.PMR 0008 C076h

	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, J to L, and N

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.

20.3.5 Open-Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT4.ODR0 0008 C088h, PORT5.ODR0 0008 C08Ah, PORT6.ODR0 0008 C08Ch, PORT7.ODR0 0008 C08Eh, PORT8.ODR0 0008 C090h, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTD.ODR0 0008 C09Ah, PORTE.ODR0 0008 C09Ch, PORTH.ODR0 0008 C0A2h, PORTJ.ODR0 0008 C0A4h, PORTK.ODR0 0008 C0A6h, PORTL.ODR0 0008 C0A8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	—	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	0: CMOS output 1: N-channel open-drain output	R/W
b1	—	Reserved		R/W
b2	B2	Pm1 Output Type Select		R/W
b3	—	Reserved		R/W
b4	B4	Pm2 Output Type Select		R/W
b5	—	Reserved		R/W
b6	B6	Pm3 Output Type Select		R/W
b7	—	Reserved		R/W

m = 0 to 9, A to E, H, and J to L

ODR0 is a register which is used to select an output type for the pins of the port.
The odd bits (b1, b3, b5, and b7) in the ODR0 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.

20.3.6 Open-Drain Control Register 1 (ODR1)

Address(es): PORT0.ODR1 0008 C081h, PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT4.ODR1 0008 C089h, PORT5.ODR1 0008 C08Bh, PORT6.ODR1 0008 C08Dh, PORT7.ODR1 0008 C08Fh, PORT8.ODR1 0008 C091h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTD.ODR1 0008 C09Bh, PORTE.ODR1 0008 C09Dh, PORTF.ODR1 0008 C09Fh, PORTH.ODR1 0008 C0A3h, PORTJ.ODR1 0008 C0A5h, PORTK.ODR1 0008 C0A7h, PORTN.ODR1 0008 C0ADh

	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, H, J, K, and N

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.

20.3.7 Pull-Up Resistor Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT5.PCR 0008 C0C5h, PORT6.PCR 0008 C0C6h, PORT7.PCR 0008 C0C7h, PORT8.PCR 0008 C0C8h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTF.PCR 0008 C0CFh, PORTH.PCR 0008 C0D1h, PORTJ.PCR 0008 C0D2h, PORTK.PCR 0008 C0D3h, PORTL.PCR 0008 C0D4h, PORTN.PCR 0008 C0D6h

	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, J to L, and N

PCR is a register which enables or disables an input pull-up resistor for individual pins of the port.
While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.
When a pin is set as an external bus pin (except as the WAIT pin), a general-purpose output port pin, or a peripheral module output pin, the pull-up resistor for the pin is disabled regardless of the setting of the PCR register.
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.

20.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT0.DSCR 0008 C0E0h, PORT1.DSCR 0008 C0E1h, PORT2.DSCR 0008 C0E2h, PORT3.DSCR 0008 C0E3h, PORT5.DSCR 0008 C0E5h, PORT6.DSCR 0008 C0E6h, 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, PORTF.DSCR 0008 C0EFh, PORTH.DSCR 0008 C0F1h, PORTJ.DSCR 0008 C0F2h, PORTK.DSCR 0008 C0F3h, PORTL.DSCR 0008 C0F4h, PORTN.DSCR 0008 C0F6h

	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 3, 5 to 9, A to F, H, J to L, and N

DSCR is a register which is used to switch the drive capacity of the port.
The bit corresponding to a pin that does not exist or whose drive capacity is fixed is reserved. A reserved bit is always read as 0. The write value should always be 0.

20.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 20.3, Table 20.4, Table 20.5, Table 20.6, and Table 20.7.

- The blank columns in Table 20.3, Table 20.4, Table 20.5,Table 20.6, and Table 20.7 indicate the bits corresponding to the pins listed in Table 20.1, 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 20.3, Table 20.4, Table 20.5,Table 20.6, and Table 20.7 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 20.3, Table 20.4, Table 20.5,Table 20.6, and Table 20.7.
  When setting a value to a reserved bit, access in byte units.

Table 20.3 PDR Register Settings in 144-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0								
PORT1							1	1
PORT2								
PORT3			0					
PORT4								
PORT5	1							
PORT6								
PORT7								
PORT8			1	1				
PORT9	1	1	1	1				
PORTA								
PORTB								
PORTC								
PORTD								
PORTE								
PORTF				1	1	1	1	1
PORTH*1			1	1				
PORTJ						1		1
PORTK	1	1					1	1
PORTL	1	1	1	1	1	1		
PORTN*2			1	1	1	1	1	1

Note 1. Since products with a sub-clock oscillator do not have the PH6 and PH7 pins, set b6 and b7 of such a product to 1.
Note 2. Since products with a JTAG interface do not have the PN7 pin, set b7 of such a product to 1.

Table 20.4 PDR Register Settings in 100-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0*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*2			1	1				
PORTJ			1	1		1		1
PORTK	1	1	1	1	1	1	1	1
PORTL	1	1	1	1	1	1	1	1
PORTN	1		1	1	1	1	1	1

Note 1. Since products with a JTAG interface do not have the P03 pin, set b3 of such a product to 1.
Note 2. Since products with a sub-clock oscillator do not have the PH6 and PH7 pins, set b6 and b7 of such a product to 1.

Table 20.5 PDR Register Settings in 80-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0						1	1	1
PORT1							1	1
PORT2			1	1	1	1		
PORT3			0		1			
PORT4								
PORT5	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							
PORTB								
PORTC							1	1
PORTD	1	1	1	1	1			
PORTE	1	1						
PORTF	1	1	1	1	1	1	1	1
PORTH*1			1	1				
PORTJ			1	1	1	1		1
PORTK	1	1	1	1	1	1	1	1
PORTL	1	1	1	1	1	1	1	1
PORTN	1		1	1	1	1	1	1

Note 1. Since products with a sub-clock oscillator do not have the PH6 and PH7 pins, set b6 and b7 of such a product to 1.

Table 20.6 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
PORT1					1	1	1	1
PORT2			1	1	1	1	1	1
PORT3			0	1	1			
PORT4								
PORT5	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		
PORTC							1	1
PORTD	1	1	1	1	1	1	1	1
PORTE	1	1						
PORTF	1	1	1	1	1	1	1	1
PORTH*1			1	1				
PORTJ			1	1	1	1	1	1
PORTK	1	1	1	1	1	1	1	1
PORTL	1	1	1	1	1	1	1	1
PORTN	1		1	1	1	1	1	1

Note 1. Since products with a sub-clock oscillator do not have the PH6 and PH7 pins, set b6 and b7 of such a product to 1.

Table 20.7 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	1		
PORT4				1	1			
PORT5	1	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		
PORTC					1	1	1	1
PORTD	1	1	1	1	1	1	1	1
PORTE	1	1	1					1
PORTF	1	1	1	1	1	1	1	1
PORTH	1	1	1	1				
PORTJ			1	1	1	1	1	1
PORTK	1	1	1	1	1	1	1	1
PORTL	1	1	1	1	1	1	1	1
PORTN	1		1	1	1	1	1	1

20.5 Handling of Unused Pins

Details on the handling of unused pins are given in Table 20.8.

Table 20.8 Handling of Unused Pins

Pin Name	Handling
EMLE	Connect this pin to VSS via a resistor (pulling down).
MD	Use this as a mode pin.
RES#	Connect this pin to VCC via a resistor (pulling up).
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, H, J to L, and N.
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, H, J to L, and N. When an external clock is input to the EXTAL pin, keep the pin open.
XCIN	Connect this pin to VSS via a resistor (pulling down).
XCOUT	Keep this pin open.
Port 0 to 9, A to F, H, J to L, and N	• 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	Reference power supply pin for the 12-bit A/D converter. Connect this pin to VCC when the 12-bit A/D converter is not to be used.
VREFL0	Reference ground pin for the 12-bit A/D converter. Connect this pin to VSS when the 12-bit A/D converter is not to be used.
AVCC0	Analog power supply pin for the 12-bit A/D converter, 12-bit D/A converter, comparator C, and temperature sensor. Connect this pin to VCC when the 12-bit A/D converter, 12-bit D/A converter, comparator C, and temperature sensor are not to be used.
AVSS0	Analog ground pin for the 12-bit A/D converter, 12-bit D/A converter, comparator C, and temperature sensor. Connect this pin to VSS when the 12-bit A/D converter, 12-bit D/A converter, comparator C, and temperature sensor are not to be used.

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.

21. Multi-Function Pin Controller (MPC)

21.1 Overview

The multi-function pin controller (MPC) selects and assigns input/output of peripheral functions and interrupt input signals from multiple ports. The MPC also assigns the port of external bus related signals.
Table 21.1 lists the functions assigned to each multiplexed pin. The symbols ✓ and × in the table indicate whether the pin is available or unavailable for the package. Selecting a single function for multiple pins is prohibited.

Table 21.1 Functions Assigned to Each Multiplexed Pin (1/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Interrupt		NMI (input)	P35	✓	✓	✓	✓	✓
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	✓	×	×	×	×
			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 21.1 Functions Assigned to Each Multiplexed Pin (2/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Interrupt	IRQ4	IRQ4-DS (input)	PB1	✓	✓	✓	✓	✓
		IRQ4 (input)	P14	✓	✓	✓	✓	✓
			P34	✓	✓	✓	×	×
			P37	✓	✓	✓	✓	✓
			P54	✓	✓	✓	✓	×
			P64	✓	×	×	×	×
			PB4	✓	✓	✓	×	×
			PD4	✓	✓	×	×	×
			PF5	✓	×	×	×	×
	IRQ5	IRQ5-DS (input)	PA4	✓	✓	✓	✓	✓
		IRQ5 (input)	P15	✓	✓	✓	✓	✓
			P25	✓	✓	×	×	×
			P36	✓	✓	✓	✓	✓
			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 21.1 Functions Assigned to Each Multiplexed Pin (3/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Interrupt	IRQ10	IRQ10-DS (input)	P42	✓	✓	✓	✓	✓
		IRQ10 (input)	P02	✓	×	×	×	×
			P55	✓	✓	✓	✓	×
			P72	✓	×	×	×	×
			P92	✓	×	×	×	×
			PA2	✓	✓	✓	×	×
			PC2	✓	✓	✓	✓	×
	IRQ11	IRQ11-DS (input)	P43	✓	✓	✓	✓	×
		IRQ11 (input)	P03	✓	✓*1	✓	✓	×
			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	✓	✓	✓	×	×
			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 21.1 Functions Assigned to Each Multiplexed Pin (4/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Multi-function timer unit 3	MTU0	MTIOC0A (input/output)	P34	✓	✓	✓	×	×
			PB3	✓	✓	✓	✓	✓
			PC4	✓	✓	✓	✓	✓
		MTIOC0B (input/output)	P13	✓	✓	✓	×	×
			P15	✓	✓	✓	✓	✓
			PA1	✓	✓	✓	✓	✓
		MTIOC0C (input/output)	P32	✓	✓	✓	✓	×
			PB1	✓	✓	✓	✓	✓
			PC5	✓	✓	✓	✓	✓
		MTIOC0D (input/output)	P33	✓	✓	×	×	×
			PA3	✓	✓	✓	✓	✓
	MTU1	MTIOC1A (input/output)	P20	✓	✓	✓	×	×
			PE4	✓	✓	✓	✓	✓
		MTIOC1B (input/output)	P21	✓	✓	✓	×	×
			PB5	✓	✓	✓	✓	✓
			PE3	✓	✓	✓	✓	✓
	MTU2	MTIOC2A (input/output)	P26	✓	✓	✓	✓	✓
			PB5	✓	✓	✓	✓	✓
		MTIOC2B (input/output)	P27	✓	✓	✓	✓	✓
			PE5	✓	✓	✓	✓	×
	MTU3	MTIOC3A (input/output)	P14	✓	✓	✓	✓	✓
			P17	✓	✓	✓	✓	✓
			PC1	✓	✓	×	×	×
			PC7	✓	✓	✓	✓	✓
			PJ1	✓	✓	✓	×	×
		MTIOC3B (input/output)	P17	✓	✓	✓	✓	✓
			P22	✓	✓	×	×	×
			P80	✓	×	×	×	×
			PA1	✓	✓	✓	✓	✓
			PB7	✓	✓	✓	✓	×
			PC5	✓	✓	✓	✓	✓
			PE1	✓	✓	✓	✓	✓
			PH0	✓	✓	✓	✓	✓
		MTIOC3C (input/output)	P16	✓	✓	✓	✓	✓
			P56	✓	×	×	×	×
			PC0	✓	✓	×	×	×
			PC6	✓	✓	✓	✓	✓
			PJ3	✓	✓	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (5/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Multi-function timer unit 3	MTU3	MTIOC3D (input/output)	P16	✓	✓	✓	✓	✓
			P23	✓	✓	×	×	×
			P81	✓	×	×	×	×
			PA6	✓	✓	✓	✓	✓
			PB0	✓	✓	✓	✓	✓
			PB6	✓	✓	✓	✓	×
			PC4	✓	✓	✓	✓	✓
			PE0	✓	✓	✓	✓	×
			PH1	✓	✓	✓	✓	✓
	MTU4	MTIOC4A (input/output)	P21	✓	✓	✓	×	×
			P24	✓	✓	×	×	×
			P55	✓	✓	✓	✓	×
			P82	✓	×	×	×	×
			PA0	✓	✓	✓	✓	×
			PB3	✓	✓	✓	✓	✓
			PE2	✓	✓	✓	✓	✓
			PE4	✓	✓	✓	✓	✓
		MTIOC4B (input/output)	P17	✓	✓	✓	✓	✓
			P30	✓	✓	✓	✓	✓
			P54	✓	✓	✓	✓	×
			PC2	✓	✓	✓	✓	×
			PD1	✓	✓	✓	×	×
			PE3	✓	✓	✓	✓	✓
		MTIOC4C (input/output)	P25	✓	✓	×	×	×
			P83	✓	×	×	×	×
			P87	✓	×	×	×	×
			PA4	✓	✓	✓	✓	✓
			PB1	✓	✓	✓	✓	✓
			PE1	✓	✓	✓	✓	✓
			PE5	✓	✓	✓	✓	×
			PH2	✓	✓	✓	✓	✓
		MTIOC4D (input/output)	P31	✓	✓	✓	✓	✓
			P55	✓	✓	✓	✓	×
			P86	✓	×	×	×	×
			PA3	✓	✓	✓	✓	✓
			PC3	✓	✓	✓	✓	×
			PD2	✓	✓	✓	×	×
			PE4	✓	✓	✓	✓	✓
			PH3	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (6/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Multi-function timer unit 3	MTU5	MTIC5U (input)	P12	✓	✓	✓	×	×
			PA4	✓	✓	✓	✓	✓
			PD7	✓	✓	×	×	×
		MTIC5V (input)	PA3	✓	✓	✓	✓	✓
			PA6	✓	✓	✓	✓	✓
			PD6	✓	✓	×	×	×
		MTIC5W (input)	PB0	✓	✓	✓	✓	✓
			PD5	✓	✓	×	×	×
	MTU6	MTIOC6A (input/output)	PE7	✓	✓	×	×	×
		MTIOC6B (input/output)	PA5	✓	✓	✓	×	×
			PA6	✓	✓	✓	✓	×
		MTIOC6C (input/output)	PE6	✓	✓	×	×	×
		MTIOC6D (input/output)	PA0	✓	✓	✓	✓	×
	MTU7	MTIOC7A (input/output)	PA2	✓	✓	✓	×	×
			PE2	✓	✓	✓	✓	✓
		MTIOC7B (input/output)	PA1	✓	✓	✓	✓	✓
		MTIOC7C (input/output)	P67	✓	×	×	×	×
			PA4	✓	✓	✓	✓	✓
		MTIOC7D (input/output)	P66	✓	×	×	×	×
			PE4	✓	✓	✓	✓	✓
	MTU8	MTIOC8A (input/output)	PD6	✓	✓	×	×	×
		MTIOC8B (input/output)	PD4	✓	✓	×	×	×
		MTIOC8C (input/output)	PD5	✓	✓	×	×	×
		MTIOC8D (input/output)	PD3	✓	✓	×	×	×
	MTU	MTCLKA (input)	P14	✓	✓	✓	✓	✓
			P24	✓	✓	×	×	×
			PA4	✓	✓	✓	✓	✓
			PC6	✓	✓	✓	✓	✓
		MTCLKB (input)	P15	✓	✓	✓	✓	✓
			P25	✓	✓	×	×	×
			PA6	✓	✓	✓	✓	✓
			PC7	✓	✓	✓	✓	✓
		MTCLKC (input)	P22	✓	✓	×	×	×
			PA1	✓	✓	✓	✓	✓
			PC4	✓	✓	✓	✓	✓
		MTCLKD (input)	P23	✓	✓	×	×	×
			PA3	✓	✓	✓	✓	✓
			PC5	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (7/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Port output enable 3	POE0	POE0# (input)	P32	✓	✓	✓	✓	×
			P93	✓	×	×	×	×
			PC4	✓	✓	✓	✓	✓
			PD1	✓	✓	✓	×	×
			PD7	✓	✓	×	×	×
	POE4	POE4# (input)	P33	✓	✓	×	×	×
			P92	✓	×	×	×	×
			PB5	✓	✓	✓	✓	✓
			PD0	✓	✓	✓	×	×
			PD6	✓	✓	×	×	×
	POE8	POE8# (input)	P17	✓	✓	✓	✓	✓
			P30	✓	✓	✓	✓	✓
			PD3	✓	✓	×	×	×
			PE3	✓	✓	✓	✓	✓
			PJ5	✓	×	×	×	×
	POE10	POE10# (input)	P32	✓	✓	✓	✓	×
			P34	✓	✓	✓	×	×
			PA6	✓	✓	✓	✓	✓
			PD5	✓	✓	×	×	×
	POE11	POE11# (input)	P33	✓	✓	×	×	×
			PB3	✓	✓	✓	✓	✓
			PD4	✓	✓	×	×	×
8-bit timer	TMR0	TMO0 (output)	P22	✓	✓	×	×	×
			PB3	✓	✓	✓	✓	✓
			PH1	✓	✓	✓	✓	✓
		TMCI0 (input)	P01	✓	×	×	×	×
			P21	✓	✓	✓	×	×
			PB1	✓	✓	✓	✓	✓
			PH3	✓	✓	✓	✓	✓
		TMRI0 (input)	P00	✓	×	×	×	×
			P20	✓	✓	✓	×	×
			PA4	✓	✓	✓	✓	✓
			PH2	✓	✓	✓	✓	✓
	TMR1	TMO1 (output)	P17	✓	✓	✓	✓	✓
			P26	✓	✓	✓	✓	✓
		TMCI1 (input)	P02	✓	×	×	×	×
			P12	✓	✓	✓	×	×
			P54	✓	✓	✓	✓	×
			PC4	✓	✓	✓	✓	✓
		TMRI1 (input)	P24	✓	✓	×	×	×
			PB5	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (8/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
8-bit timer	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	✓	✓	×	×	×
Compare match timer W	CMTW0	TOC0 (output)	PC7	✓	✓	✓	✓	✓
		TIC0 (input)	PC6	✓	✓	✓	✓	✓
		TOC1 (output)	PE7	✓	✓	×	×	×
			PH2	✓	✓	✓	✓	✓
		TIC1 (input)	PE6	✓	✓	×	×	×
			PH1	✓	✓	✓	✓	✓
	CMTW1	TOC2 (output)	PB5	✓	✓	✓	✓	✓
			PD3	✓	✓	×	×	×
		TIC2 (input)	PB3	✓	✓	✓	✓	✓
			PD2	✓	✓	✓	×	×
		TOC3 (output)	PE3	✓	✓	✓	✓	✓
		TIC3 (input)	PE2	✓	✓	✓	✓	✓
Realtime clock		RTCOUT (output)*2	P16	✓	✓	✓	✓	×
			P32	✓	✓	✓	✓	×
		RTCIC0 (input) *2, *3	P30	✓	✓	✓	✓	×
		RTCIC1 (input) *2, *3	P31	✓	✓	✓	✓	×
		RTCIC2 (input) *2, *3	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	✓	✓	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (9/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Serial communications interface	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	✓	×	×	×	×
	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	✓	✓	✓	✓	✓
			PK4	✓	×	×	×	×
		TXD4 (output) /SMOSI4 (input/output) / SSDA4 (input/output)	PB1	✓	✓	✓	✓	✓
			PK5	✓	×	×	×	×
		SCK4 (input/output)	P70	✓	×	×	×	×
			PB3	✓	✓	✓	✓	✓
		CTS4# (input) /RTS4# (output) / SS4# (input)	PB2	✓	✓	✓	×	×
			PE6	✓	✓	×	×	×
	SCI5	RXD5 (input) /SMISO5 (input/output) / SSCL5 (input/output)	PA2	✓	✓	✓	×	×
			PA3	✓	✓	✓	✓	✓
			PC2	✓	✓	✓	✓	×
		TXD5 (output) /SMOSI5 (input/output) / SSDA5 (input/output)	PA4	✓	✓	✓	✓	✓
			PC3	✓	✓	✓	✓	×
		SCK5 (input/output)	PA1	✓	✓	✓	✓	✓
			PC1	✓	✓	×	×	×
			PC4	✓	✓	✓	✓	✓
		CTS5# (input) /RTS5# (output) / SS5# (input)	PA6	✓	✓	✓	✓	✓
			PC0	✓	✓	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (10/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Serial communications interface	SCI6	RXD6 (input) /SMISO6 (input/output) / SSCL6 (input/output)	P01	✓	×	×	×	×
			P33	✓	✓	×	×	×
			PB0	✓	✓	✓	✓	✓
		TXD6 (output) /SMOSI6 (input/output) / SSDA6 (input/output)	P00	✓	×	×	×	×
			P32	✓	✓	✓	✓	×
			PB1	✓	✓	✓	✓	✓
		SCK6 (input/output)	P02	✓	×	×	×	×
			P34	✓	✓	✓	×	×
			PB3	✓	✓	✓	✓	✓
		CTS6# (input) /RTS6# (output) / SS6# (input)	PB2	✓	✓	✓	×	×
			PJ3	✓	✓	×	×	×
	SCI7	RXD7 (input) /SMISO7 (input/output) / SSCL7 (input/output)	P92	✓	×	×	×	×
		TXD7 (output) /SMOSI7 (input/output) / SSDA7 (input/output)	P55	✓	×	×	×	×
			P90	✓	×	×	×	×
		SCK7 (input/output)	P56	✓	×	×	×	×
			P91	✓	×	×	×	×
		CTS7# (input) /RTS7# (output) / SS7# (input)	P93	✓	×	×	×	×
	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	✓	✓	✓	✓	×
			PK3	✓	×	×	×	×
		TXD9 (output) /SMOSI9 (input/output) / SSDA9 (input/output)	PB7	✓	✓	✓	✓	×
			PK2	✓	×	×	×	×
		SCK9 (input/output)	P60	✓	×	×	×	×
			PB5	✓	✓	✓	✓	×
		CTS9# (input) /RTS9# (output) / SS9# (input)	P61	✓	×	×	×	×
			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	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (11/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Serial communications interface	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)	PA2	✓	✓	✓	×	×
			PE2	✓	✓	✓	✓	✓
		TXD12 (output) / SMOSI12 (input/output) / SSDA12 (input/output) / TXDX12 (output) / SIOX12 (input/output)	PA4	✓	✓	✓	✓	✓
			PE1	✓	✓	✓	✓	✓
		SCK12 (input/output)	PA1	✓	✓	✓	✓	✓
			PE0	✓	✓	✓	✓	×
		CTS12# (input) /RTS12# (output) / SS12# (input)	PA6	✓	✓	✓	✓	✓
			PE3	✓	✓	✓	✓	✓
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) / RTS010# (output) /DE010 (output)	P80	✓	×	×	×	×
		SCK010 (input/output) / CTS010# (input) /SS010# (input)	P83	✓	×	×	×	×
		SCK010 (input/output)	PC5	✓	✓	✓	✓	✓
		CTS010# (input) /RTS010# (output) / SS010# (input) /DE010 (output)	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) / RTS011# (output) /DE011 (output)	P75	✓	×	×	×	×
		SCK011(input/output)	PB5	✓	✓	✓	✓	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (12/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
Serial communications interface	RSCI11	TXDA011(output)	PC1	✓	✓	×	×	×
		TXDB011(output)	PC2	✓	✓	✓	✓	×
		CTS011# (input) /SS011# (input)	P74	✓	×	×	×	×
		CTS011# (input) /RTS011# (output) / SS011# (input) /DE011 (output)	PB4	✓	✓	✓	×	×
I^2C bus interface	RIIC0	SCL0 (input/output)	P12	✓	✓	✓	×	×
		SDA0 (input/output)	P13	✓	✓	✓	×	×
	RIIC2	SCL2 (input/output)	P16	✓	✓	✓	✓	✓
		SDA2 (input/output)	P17	✓	✓	✓	✓	✓
CAN FD module	CANFD0	CRX0 (input)	P15	✓	✓	✓	✓	✓
			P33	✓	✓	×	×	×
			P55	✓	✓	✓	✓	×
			PD2	✓	✓	✓	×	×
		CTX0 (output)	P14	✓	✓	✓	✓	✓
			P32	✓	✓	✓	✓	×
			P54	✓	✓	✓	✓	×
			PD1	✓	✓	✓	×	×
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PA5	✓	✓	✓	×	×
			PB0	✓	✓	✓	✓	✓
			PC5	✓	✓	✓	✓	✓
		MOSIA (input/output)	P16	✓	✓	✓	✓	✓
			PA6	✓	✓	✓	✓	✓
			PC6	✓	✓	✓	✓	✓
		MISOA (input/output)	P17	✓	✓	✓	✓	✓
			PA7	✓	✓	×	×	×
			PC7	✓	✓	✓	✓	✓
		SSLA0 (input/output)	PA4	✓	✓	✓	✓	✓
			PC4	✓	✓	✓	✓	✓
		SSLA1 (output)	PA0	✓	✓	✓	✓	×
			PC0	✓	✓	×	×	×
		SSLA2 (output)	PA1	✓	✓	✓	✓	✓
			PC1	✓	✓	×	×	×
		SSLA3 (output)	PA2	✓	✓	✓	×	×
			PC2	✓	✓	✓	✓	×
12-bit A/D converter		AN000 (input) *3	P40	✓	✓	✓	✓	✓
		AN001 (input) *3	P41	✓	✓	✓	✓	✓
		AN002 (input) *3	P42	✓	✓	✓	✓	✓
		AN003 (input) *3	P43	✓	✓	✓	✓	×
		AN004 (input) *3	P44	✓	✓	✓	✓	×
		AN005 (input) *3	P45	✓	✓	✓	✓	✓
		AN006 (input) *3	P46	✓	✓	✓	✓	✓
		AN007 (input) *3	P47	✓	✓	✓	✓	✓
		AN008 (input) *3	PE0	✓	✓	✓	✓	×
		AN009 (input) *3	PE1	✓	✓	✓	✓	✓
		AN010 (input) *3	PE2	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (13/13)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				144-pin	100-pin	80-pin	64-pin	48-pin
12-bit A/D converter		AN011 (input) *3	PE3	✓	✓	✓	✓	✓
		AN012 (input) *3	PE4	✓	✓	✓	✓	✓
		AN013 (input) *3	PE5	✓	✓	✓	✓	×
		AN014 (input) *3	PE6	✓	✓	×	×	×
		AN015 (input) *3	PE7	✓	✓	×	×	×
		AN016 (input) *3	PD0	✓	✓	✓	×	×
		AN017 (input) *3	PD1	✓	✓	✓	×	×
		AN018 (input) *3	PD2	✓	✓	✓	×	×
		AN019 (input) *3	PD3	✓	✓	×	×	×
		AN020 (input) *3	PD4	✓	✓	×	×	×
		AN021 (input) *3	PD5	✓	✓	×	×	×
		AN022 (input) *3	PD6	✓	✓	×	×	×
		AN023 (input) *3	PD7	✓	✓	×	×	×
		ADST0 (output)	PA4	✓	✓	✓	✓	✓
			PH1	✓	✓	✓	✓	✓
		ADTRG0# (input)	P07	✓	✓	✓	✓	×
			P16	✓	✓	✓	✓	✓
			P25	✓	✓	×	×	×
			PA1	✓	✓	✓	✓	✓
			PH0	✓	✓	✓	✓	✓
12-bit D/A converter		DA0 (output)	P03	✓	✓*1	✓	✓	×
		DA1 (output)	P05	✓	✓	✓	×	×
Clock frequency accuracy measurement circuit		CACREF (input)	PA0	✓	✓	✓	✓	×
			PC7	✓	✓	✓	✓	✓
			PH0	✓	✓	✓	✓	✓
Remote control signal receiver	REMC0	PMC0 (input)	P51	✓	✓	×	×	×
			P53	✓	✓	×	×	×
			PB3	✓	✓	✓	✓	✓
			PC3	✓	✓	✓	✓	×
			PC4	✓	✓	✓	✓	✓
			PC5	✓	✓	✓	✓	✓
Comparator C		CMPC00 (input)	PE1	✓	✓	✓	✓	✓
		CMPC10 (input)	PA3	✓	✓	✓	✓	✓
		CMPC20 (input)	P15	✓	✓	✓	✓	✓
		CMPC30 (input)	P26	✓	✓	✓	✓	✓
		COMP0 (output)	PE5	✓	✓	✓	✓	×
		COMP1 (output)	PB1	✓	✓	✓	✓	✓
		COMP2 (output)	P17	✓	✓	✓	✓	✓
		COMP3 (output)	P30	✓	✓	✓	✓	✓
		CVREFC0 (input)	PE2	✓	✓	✓	✓	✓
		CVREFC1 (input)	PA4	✓	✓	✓	✓	✓
		CVREFC2 (input)	P14	✓	✓	✓	✓	✓
		CVREFC3 (input)	P27	✓	✓	✓	✓	✓

Note 1. This pin function is not provided for products with a JTAG interface.
Note 2. This pin function is not available in products with no sub-clock oscillator.
Note 3. To use this pin function, set the corresponding pin as general input (set the PORTm.PDR.Bn and PORTm.PMR.Bn bits to 0).

21.2 Register Descriptions

The registers and bits of unsupported pins, depending on the package, are reserved. The write value to the reserved bits is the value after a reset.

21.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are 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.

21.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0 to 3, 5, 7)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h, P02PFS 0008 C142h, P03PFS 0008 C143h, P05PFS 0008 C145h, P07PFS 0008 C147h

	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 21.2 and Table 21.3.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P00: IRQ8 (144 pin) P01: IRQ9 (144 pin) P02: IRQ10 (144 pin) P03: IRQ11 (144/100*1/80/64 pin) P05: IRQ13 (144/100/80 pin) P07: IRQ15 (144/100/80/64 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P03: DA0 (144/100*1/80/64 pin) P05: DA1 (144/100/80 pin)	R/W

Note 1. Products with a JTAG interface do not have this pin function.

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 21.2 Register Settings for Input/Output Pin Function in 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 21.3 Register Settings for Input/Output Pin Function in 100-/80-/64-Pin LFQFP

	Pin
PSEL[5:0] Settings	**P07**
000000b (initial value)	Hi-Z
001001b	ADTRG0#

21.2.3 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

	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 21.4 and Table 21.5.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P12: IRQ2 (144/100/80 pin) P13: IRQ3 (144/100/80 pin) P14: IRQ4 (144/100/80/64/48 pin) P15: IRQ5 (144/100/80/64/48 pin) P16: IRQ6 (144/100/80/64/48 pin) P17: IRQ7 (144/100/80/64/48 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P14: CVREFC2 (144/100/80/64/48 pin) P15: CMPC20 (144/100/80/64/48 pin)	R/W

Table 21.4 Register Settings for Input/Output Pin Function in 144-/100-/80-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
000101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
000111b	—	—	—	—	RTCOUT*1	POE8#
001000b	—	—	—	—	—	MTIOC4B
001001b	—	—	—	—	ADTRG0#	—
001010b	RXD2*2 SMISO2*2 SSCL2*2	TXD2*2 SMOSI2*2 SSDA2*2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
001101b	—	—	—	—	MOSIA	MISOA
001111b	SCL0	SDA0	—	—	SCL2	SDA2
010000b	—	—	CTX0	CRX0	—	—
011110b	—	—	—	—	—	COMP2

—: Do not specify this value.

Note 1. This pin function is not available in products with no sub-clock oscillator.

Note 2. This pin function is only available in products other than those in an 80-pin package.

Table 21.5 Register Settings for Input/Output Pin Function in 64-/48-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P14**	**P15**	**P16**	**P17**
000000b (initial value)	Hi-Z			
000001b	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
000010b	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
000101b	TMRI2	TMCI2	TMO2	TMO1
000111b	—	—	RTCOUT*1, *2	POE8#
001000b	—	—	—	MTIOC4B
001001b	—	—	ADTRG0#	—
001010b	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
001011b	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
001101b	—	—	MOSIA	MISOA
001111b	—	—	SCL2	SDA2
010000b	CTX0	CRX0	—	—
011110b	—	—	—	COMP2

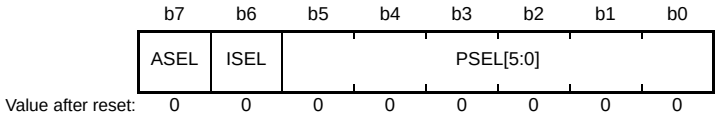
—: Do not specify this value.

Note 1. This pin function is not available in products with no sub-clock oscillator.

Note 2. This pin function is only available in products other than those in a 48-pin package.

21.2.4 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



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 21.6, Table 21.7, and Table 21.8.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P20: IRQ8 (144/100/80 pin) P21: IRQ9 (144/100/80 pin) P22: IRQ15 (144/100 pin) P23: IRQ3 (144/100 pin) P24: IRQ12 (144/100 pin) P25: IRQ5 (144/100 pin) P26: IRQ6 (144/100/80/64/48 pin) P27: IRQ7 (144/100/80/64/48 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P26: CMPC30 (144/100/80/64/48 pin) P27: CVREFC3 (144/100/80/64/48 pin)	R/W

Table 21.6 Register Settings for Input/Output Pin Function in 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	—	—
000101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
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#	—

—: Do not specify this value.

Table 21.7 Register Settings for Input/Output Pin Function in 80-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P20**	**P21**	**P26**	**P27**
000000b (initial value)	Hi-Z			
000001b	MTIOC1A	MTIOC1B	MTIOC2A	MTIOC2B
000101b	TMRI0	TMCI0	TMO1	TMCI3
001000b	—	MTIOC4A	—	—
001010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	CTS3# RTS3# SS3#	—

—: Do not specify this value.

Table 21.8 Register Settings for Input/Output Pin Function in 64-/48-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**P26**	**P27**
000000b (initial value)	Hi-Z	
000001b	MTIOC2A	MTIOC2B
000101b	TMO1	TMCI3*1
001010b	TXD1 SMOSI1 SSDA1	SCK1
001011b	CTS3# RTS3# SS3#	—

—: Do not specify this value.

Note 1. This pin function is only available in products other than those in a 48-pin package.

21.2.5 P3n Pin Function Control Register (P3nPFS) (n = 0 to 4, 6, 7)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh, P34PFS 0008 C15Ch, P36PFS 0008 C15Eh, P37PFS 0008 C15Fh

	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 21.9, Table 21.10, Table 21.11, and Table 21.12.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P30: IRQ0-DS (144/100/80/64/48 pin) P31: IRQ1-DS (144/100/80/64/48 pin) P32: IRQ2-DS (144/100/80/64 pin) P33: IRQ3-DS (144/100 pin) P34: IRQ4 (144/100/80 pin) P36: IRQ5 (144/100/80/64/48 pin) P37: IRQ4 (144/100/80/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.9 Register Settings for Input/Output Pin Function in 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
000101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
000111b	POE8#	—	RTCOUT*1	—	POE10#
001000b	—	—	POE0#	POE4#	—
001010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
001011b	—	CTS1# RTS1# SS1#	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0
010000b	—	—	CTX0	CRX0	—
011110b	COMP3	—	—	—	—
100001b	—	—	POE10#	POE11#	—

—: Do not specify this value.

Note 1. This pin function is not available in products with no sub-clock oscillator.

Table 21.10 Register Settings for Input/Output Pin Function in 80-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P30**	**P31**	**P32**	**P34**
000000b (initial value)	Hi-Z			
000001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0A
000101b	TMRI3	TMCI2	TMO3	TMCI3
000111b	POE8#	—	RTCOUT*1	POE10#
001000b	—	—	POE0#	—
001010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6	SCK6
001011b	—	CTS1# RTS1# SS1#	TXD0 SMOSI0 SSDA0	SCK0
010000b	—	—	CTX0	—
011110b	COMP3	—	—	—
100001b	—	—	POE10#	—

—: Do not specify this value.
Note 1. This pin function is not available in products with no sub-clock oscillator.

Table 21.11 Register Settings for Input/Output Pin Function in 64-Pin LFQFP

	Pin		
PSEL[5:0] Settings	**P30**	**P31**	**P32**
000000b (initial value)	Hi-Z		
000001b	MTIOC4B	MTIOC4D	MTIOC0C
000101b	TMRI3	TMCI2	TMO3
000111b	POE8#	—	RTCOUT*1
001000b	—	—	POE0#
001010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6
001011b	—	CTS1# RTS1# SS1#	—
010000b	—	—	CTX0
011110b	COMP3	—	—
100001b	—	—	POE10#

—: Do not specify this value.
Note 1. This pin function is not available in products with no sub-clock oscillator.

Table 21.12 Register Settings for Input/Output Pin Function in 48-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**P30**	**P31**
000000b (initial value)	Hi-Z	
000001b	MTIOC4B	MTIOC4D
000101b	—	TMCI2
000111b	POE8#	—
001010b	RXD1 SMISO1 SSCL1	—
001011b	—	CTS1# RTS1# SS1#
011110b	COMP3	—

—: Do not specify this value.

21.2.6 P4n Pin Function Control Register (P4nPFS) (n = 0 to 7)

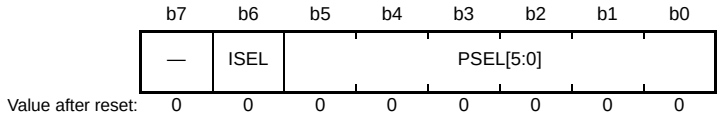
Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

	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 (144/100/80/64/48 pin) P41: IRQ9-DS (144/100/80/64/48 pin) P42: IRQ10-DS (144/100/80/64/48 pin) P43: IRQ11-DS (144/100/80/64 pin) P44: IRQ12-DS (144/100/80/64 pin) P45: IRQ13-DS (144/100/80/64/48 pin) P46: IRQ14-DS (144/100/80/64/48 pin) P47: IRQ15-DS (144/100/80/64/48 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P40: AN000 (144/100/80/64/48 pin) P41: AN001 (144/100/80/64/48 pin) P42: AN002 (144/100/80/64/48 pin) P43: AN003 (144/100/80/64 pin) P44: AN004 (144/100/80/64 pin) P45: AN005 (144/100/80/64/48 pin) P46: AN006 (144/100/80/64/48 pin) P47: AN007 (144/100/80/64/48 pin)	R/W

21.2.7 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 21.13, Table 21.14, and Table 21.15.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P50: IRQ0 (144/100 pin) P51: IRQ1 (144/100 pin) P52: IRQ2 (144/100 pin) P53: IRQ3 (144/100 pin) P54: IRQ4 (144/100/80/64 pin) P55: IRQ10 (144/100/80/64 pin) P56: IRQ6 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.13 Register Settings for Input/Output Pin Function in 144-Pin LFQFP

	Pin						
PSEL[5:0] Settings	**P50**	**P51**	**P52**	**P53**	**P54**	**P55**	**P56**
000000b (initial value)	Hi-Z						
000001b	—	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C
000010b	—	—	—	—	—	MTIOC4A	—
000101b	—	—	—	—	TMCI1	TMO3	—
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	TXD7 SMOSI7 SSDA7	SCK7
001011b	—	—	—	—	CTS2# RTS2# SS2#	—	—
010000b	—	—	—	—	CTX0	CRX0	—
100110b	—	PMC0	—	PMC0	—	—	—

—: Do not specify this value.

Table 21.14 Register Settings for Input/Output Pin Function in 100-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**P50**	**P51**	**P52**	**P53**	**P54**	**P55**
000000b (initial value)	Hi-Z					
000001b	—	—	—	—	MTIOC4B	MTIOC4D
000010b	—	—	—	—	—	MTIOC4A
000101b	—	—	—	—	TMCI1	TMO3
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—
001011b	—	—	—	—	CTS2# RTS2# SS2#	—
010000b	—	—	—	—	CTX0	CRX0
100110b	—	PMC0	—	PMC0	—	—

—: Do not specify this value.

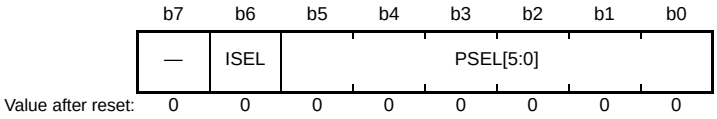
Table 21.15 Register Settings for Input/Output Pin Function in 80-/64-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**P54**	**P55**
000000b (initial value)	Hi-Z	
000001b	MTIOC4B	MTIOC4D
000010b	—	MTIOC4A
000101b	TMCI1	TMO3
010000b	CTX0	CRX0

—: Do not specify this value.

21.2.8 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



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 21.16.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P60: IRQ0 (144 pin) P61: IRQ1 (144 pin) P62: IRQ2 (144 pin) P63: IRQ3 (144 pin) P64: IRQ4 (144 pin) P65: IRQ13 (144 pin) P66: IRQ14 (144 pin) P67: IRQ15 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.16 Register Settings for Input/Output Pin Function in 144-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P60**	**P61**	**P66**	**P67**
000000b (initial value)	Hi-Z			
001000b	—	—	MTIOC7D	MTIOC7C
001010b	SCK9	CTS9# RTS9# SS9#	—	—

—: Do not specify this value.

21.2.9 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

	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 21.17.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P70: IRQ0 (144 pin) P71: IRQ1 (144 pin) P72: IRQ10 (144 pin) P73: IRQ8 (144 pin) P74: IRQ12 (144 pin) P75: IRQ13 (144 pin) P76: IRQ14 (144 pin) P77: IRQ7 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

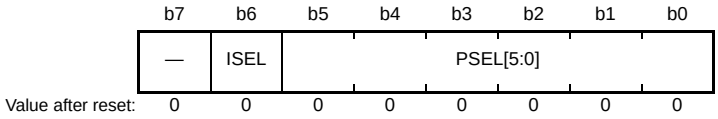
Table 21.17 Register Settings for Input/Output Pin Function in 144-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**P70**	**P74**	**P75**	**P76**	**P77**
000000b (initial value)	Hi-Z				
001010b	SCK4	—	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
001011b	—	CTS11# SS11#	RTS11#	—	—
101100b	—	—	SCK011	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011
101101b	—	CTS011# SS011#	RTS011#	—	—
101110b	—	—	DE011	—	—

—: Do not specify this value.

21.2.10 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 21.18.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P80: IRQ8 (144 pin) P81: IRQ9 (144 pin) P82: IRQ2 (144 pin) P83: IRQ3 (144 pin) P86: IRQ14 (144 pin) P87: IRQ15 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

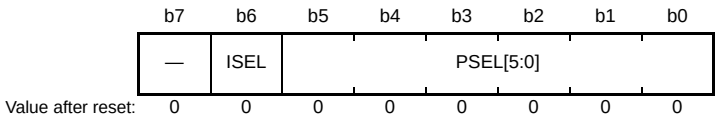
Table 21.18 Register Settings for Input/Output Pin Function in 144-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**P80**	**P81**	**P82**	**P83**	**P86**	**P87**
000000b (initial value)	Hi-Z					
000001b	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	—	—
001000b	—	—	—	—	MTIOC4D	MTIOC4C
001010b	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
001011b	RTS10#	—	—	CTS10# SS10#	—	—
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.

21.2.11 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 21.19.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P90: IRQ0 (144 pin) P91: IRQ9 (144 pin) P92: IRQ10 (144 pin) P93: IRQ11 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.19 Register Settings for Input/Output Pin Function in 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.

21.2.12 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

	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 21.20, Table 21.21, Table 21.22, and Table 21.23.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PA0: IRQ0 (144/100/80/64 pin) PA1: IRQ11 (144/100/80/64/48 pin) PA2: IRQ10 (144/100/80 pin) PA3: IRQ6-DS (144/100/80/64/48 pin) PA4: IRQ5-DS (144/100/80/64/48 pin) PA5: IRQ5 (144/100/80 pin) PA6: IRQ14 (144/100/80/64/48 pin) PA7: IRQ7 (144/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. PA3: CMPC10 (144/100/80/64/48 pin) PA4: CVREFC1 (144/100/80/64/48 pin)	R/W

Table 21.20 Register Settings for Input/Output Pin Function in 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	—
000101b	—	—	—	—	TMRI0	—	TMCI3	—
000111b	CACREF	—	—	—	—	—	POE10#	—
001000b	MTIOC6D	MTIOC7B	MTIOC7A	MTIC5V	MTIOC4C	MTIOC6B	MTIOC3D	—
001001b	—	ADTRG0#	—	—	ADST0	—	—	—
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	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
100111b	—	MTIOC3B	—	MTIOC4D	MTIOC7C	—	MTIOC6B	—

—: Do not specify this value.

Table 21.21 Register Settings for Input/Output Pin Function in 80-Pin LFQFP

	Pin						
PSEL[5:0] Settings	**PA0**	**PA1**	**PA2**	**PA3**	**PA4**	**PA5**	**PA6**
000000b (initial value)	Hi-Z						
000001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V
000010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB
000101b	—	—	—	—	TMRI0	—	TMCI3
000111b	CACREF	—	—	—	—	—	POE10#
001000b	MTIOC6D	MTIOC7B	MTIOC7A	MTIC5V	MTIOC4C	MTIOC6B	MTIOC3D
001001b	—	ADTRG0#	—	—	ADST0	—	—
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	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA
100111b	—	MTIOC3B	—	MTIOC4D	MTIOC7C	—	MTIOC6B

—: Do not specify this value.

Table 21.22 Register Settings for Input/Output Pin Function in 64-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**PA0**	**PA1**	**PA3**	**PA4**	**PA6**
000000b (initial value)	Hi-Z				
000001b	MTIOC4A	MTIOC0B	MTIOC0D	MTIC5U	MTIC5V
000010b	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000101b	—	—	—	TMRI0	TMCI3
000111b	CACREF	—	—	—	POE10#
001000b	MTIOC6D	MTIOC7B	MTIC5V	MTIOC4C	MTIOC3D
001001b	—	ADTRG0#	—	ADST0	—
001010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—
001011b	—	—	—	—	CTS5# RTS5# SS5#
001100b	—	SCK12	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	CTS12# RTS12# SS12#
001101b	SSLA1	SSLA2	—	SSLA0	MOSIA
100111b	—	MTIOC3B	MTIOC4D	MTIOC7C	MTIOC6B

—: Do not specify this value.

Table 21.23 Register Settings for Input/Output Pin Function in 48-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**PA1**	**PA3**	**PA4**	**PA6**
000000b (initial value)	Hi-Z			
000001b	MTIOC0B	MTIOC0D	MTIC5U	MTIC5V
000010b	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000101b	—	—	TMRI0	—
000111b	—	—	—	POE10#
001000b	MTIOC7B	MTIC5V	MTIOC4C	MTIOC3D
001001b	ADTRG0#	—	ADST0	—
001010b	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—
001011b	—	—	—	CTS5# RTS5# SS5#
001100b	SCK12	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	CTS12# RTS12# SS12#
001101b	SSLA2	—	SSLA0	MOSIA
100111b	MTIOC3B	MTIOC4D	MTIOC7C	—

—: Do not specify this value.

21.2.13 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 21.24, Table 21.25, and Table 21.26.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PB0: IRQ12 (144/100/80/64/48 pin) PB1: IRQ4-DS (144/100/80/64/48 pin) PB2: IRQ2 (144/100/80 pin) PB3: IRQ3 (144/100/80/64/48 pin) PB4: IRQ4 (144/100/80 pin) PB5: IRQ13 (144/100/80/64/48 pin) PB6: IRQ6 (144/100/80/64 pin) PB7: IRQ15 (144/100/80/64 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.24 Register Settings for Input/Output Pin Function in 144-/100-/80-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	MTIOC3D	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
000101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
000111b	—	—	—	POE11#	—	POE4#	—	—
001010b	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4	CTS4# RTS4# SS4#	SCK4	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
001011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
001101b	RSPCKA	—	—	—	—	—	—	—
011101b	—	—	—	TIC2	—	TOC2	—	—
011110b	—	COMP1	—	—	—	—	—	—
100100b	—	—	—	—	CTS11# RTS11# SS11#	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
100110b	—	—	—	PMC0	—	—	—	—
101100b	—	—	—	—	CTS011# RTS011# SS011#	SCK011	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011
101110b	—	—	—	—	DE011	—	—	—

—: Do not specify this value.

Table 21.25 Register Settings for Input/Output Pin Function in 64-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**PB0**	**PB1**	**PB3**	**PB5**	**PB6**	**PB7**
000000b (initial value)	Hi-Z					
000001b	MTIC5W	MTIOC0C	MTIOC0A	MTIOC2A	MTIOC3D	MTIOC3B
000010b	MTIOC3D	MTIOC4C	MTIOC4A	MTIOC1B	—	—
000101b	—	TMCI0	TMO0	TMRI1	—	—
000111b	—	—	POE11#	POE4#	—	—
001010b	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4	SCK4	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
001011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—	—	—
001101b	RSPCKA	—	—	—	—	—
011101b	—	—	TIC2	TOC2	—	—
011110b	—	COMP1	—	—	—	—
100100b	—	—	—	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
100110b	—	—	PMC0	—	—	—
101100b	—	—	—	SCK011	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011

—: Do not specify this value.

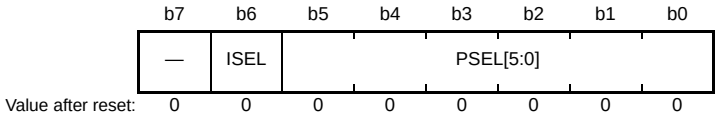
Table 21.26 Register Settings for Input/Output Pin Function in 48-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**PB0**	**PB1**	**PB3**	**PB5**
000000b (initial value)	Hi-Z			
000001b	MTIC5W	MTIOC0C	MTIOC0A	MTIOC2A
000010b	MTIOC3D	MTIOC4C	MTIOC4A	MTIOC1B
000101b	—	TMCI0	TMO0	TMRI1
000111b	—	—	POE11#	POE4#
001010b	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4	SCK4	—
001011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—
001101b	RSPCKA	—	—	—
011101b	—	—	TIC2	TOC2
011110b	—	COMP1	—	—
100110b	—	—	PMC0	—

—: Do not specify this value.

21.2.14 PCn Pin Function Control Register (PCnPFS) (n = 0 to 7)

Address(es): PC0PFS 0008 C1A0h, PC1PFS 0008 C1A1h, PC2PFS 0008 C1A2h, PC3PFS 0008 C1A3h, PC4PFS 0008 C1A4h, PC5PFS 0008 C1A5h, PC6PFS 0008 C1A6h, PC7PFS 0008 C1A7h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.27, Table 21.28, and Table 21.29.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PC0: IRQ14 (144/100 pin) PC1: IRQ12 (144/100 pin) PC2: IRQ10 (144/100/80/64 pin) PC3: IRQ11 (144/100/80/64 pin) PC4: IRQ12 (144/100/80/64/48 pin) PC5: IRQ5 (144/100/80/64/48 pin) PC6: IRQ13 (144/100/80/64/48 pin) PC7: IRQ14 (144/100/80/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.27 Register Settings for Input/Output Pin Function in 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
000101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
000111b	—	—	—	—	POE0#	—	—	CACREF
001000b	—	—	—	—	MTIOC0A	MTIOC0C	—	—
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	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
011101b	—	—	—	—	—	—	TIC0	TOC0
100100b	—	—	—	—	CTS10# RTS10# SS10#	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
100110b	—	—	—	PMC0	PMC0	PMC0	—	—
101100b	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011 TXDA011	TXDB011	—	CTS010# RTS010# SS010#	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101110b	—	—	—	—	DE010	—	—	—

—: Do not specify this value.

Table 21.28 Register Settings for Input/Output Pin Function in 80-/64-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
000000b (initial value)	Hi-Z					
000001b	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
000010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000101b	—	—	TMCI1	TMRI2	TMCI2	TMO2
000111b	—	—	POE0#	—	—	CACREF
001000b	—	—	MTIOC0A	MTIOC0C	—	—
001010b	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
001011b	—	—	CTS8# RTS8# SS8#	—	—	—
001101b	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
011101b	—	—	—	—	TIC0	TOC0
100100b	—	—	CTS10# RTS10# SS10#	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
100110b	—	PMC0	PMC0	PMC0	—	—
101100b	TXDB011	—	CTS010# RTS010# SS010#	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101110b	—	—	DE010	—	—	—

—: Do not specify this value.

Table 21.29 Register Settings for Input/Output Pin Function in 48-Pin LFQFP

	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
001000b	MTIOC0A	MTIOC0C	—	—
001010b	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
001011b	CTS8# RTS8# SS8#	—	—	—
001101b	SSLA0	RSPCKA	MOSIA	MISOA
011101b	—	—	TIC0	TOC0
100100b	CTS10# RTS10# SS10#	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
100110b	PMC0	PMC0	—	—
101100b	CTS010# RTS010# SS010#	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101110b	DE010	—	—	—

—: Do not specify this value.

21.2.15 PDn Pin Function Control Register (PDnPFS) (n = 0 to 7)

Address(es): PD0PFS 0008 C1A8h, PD1PFS 0008 C1A9h, PD2PFS 0008 C1AAh, PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh, PD7PFS 0008 C1AFh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[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 21.30 and Table 21.31.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PD0: IRQ0 (144/100/80 pin) PD1: IRQ1 (144/100/80 pin) PD2: IRQ2 (144/100/80 pin) PD3: IRQ3 (144/100 pin) PD4: IRQ4 (144/100 pin) PD5: IRQ5 (144/100 pin) PD6: IRQ6 (144/100 pin) PD7: IRQ7 (144/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. PD0: AN016 (144/100/80 pin) PD1: AN017 (144/100/80 pin) PD2: AN018 (144/100/80 pin) PD3: AN019 (144/100 pin) PD4: AN020 (144/100 pin) PD5: AN021 (144/100 pin) PD6: AN022 (144/100 pin) PD7: AN023 (144/100 pin)	R/W

Table 21.30 Register Settings for Input/Output Pin Function in 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7
000000b (initial value)	Hi-Z							
000001b	—	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
000111b	—	—	—	POE8#	POE11#	POE10#	POE4#	POE0#
001000b	POE4#	POE0#	—	MTIOC8D	MTIOC8B	MTIOC8C	MTIOC8A	—
010000b	—	CTX0	CRX0	—	—	—	—	—
011101b	—	—	TIC2	TOC2	—	—	—	—

—: Do not specify this value.

Table 21.31 Register Settings for Input/Output Pin Function in 80-Pin LFQFP

	Pin		
PSEL[5:0] Settings	PD0	PD1	PD2
000000b (initial value)	Hi-Z		
000001b	—	MTIOC4B	MTIOC4D
001000b	POE4#	POE0#	—
010000b	—	CTX0	CRX0
011101b	—	—	TIC2

—: Do not specify this value.

21.2.16 PEn Pin Function Control Register (PEnPFS) (n = 0 to 7)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h, PE4PFS 0008 C1B4h, PE5PFS 0008 C1B5h, PE6PFS 0008 C1B6h, PE7PFS 0008 C1B7h

	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 21.32, Table 21.33, and Table 21.34.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PE0: IRQ8 (144/100/80/64 pin) PE1: IRQ9 (144/100/80/64/48 pin) PE2: IRQ7-DS (144/100/80/64/48 pin) PE3: IRQ11 (144/100/80/64/48 pin) PE4: IRQ12 (144/100/80/64/48 pin) PE5: IRQ5 (144/100/80/64 pin) PE6: IRQ6 (144/100 pin) PE7: IRQ7 (144/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. PE0: AN008 (144/100/80/64 pin) PE1: AN009 (144/100/80/64/48 pin) PE2: AN010 (144/100/80/64/48 pin) PE3: AN011 (144/100/80/64/48 pin) PE4: AN012 (144/100/80/64/48 pin) PE5: AN013 (144/100/80/64 pin) PE6: AN014 (144/100 pin) PE7: AN015 (144/100 pin)	R/W

Table 21.32 Register Settings for Input/Output Pin Function in 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	—	—
000111b	—	—	—	POE8#	—	—	—	—
001000b	MTIOC3D	MTIOC3B	MTIOC7A	MTIOC1B	MTIOC4A	—	MTIOC6C	MTIOC6A
001010b	—	—	—	—	—	—	CTS4# RTS4# SS4#	—
001100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
011101b	—	—	TIC3	TOC3	—	—	TIC1	TOC1
011110b	—	—	—	—	—	COMP0	—	—
100111b	—	—	—	—	MTIOC7D	—	—	—

—: Do not specify this value.

Table 21.33 Register Settings for Input/Output Pin Function in 80-/64-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**PE0**	**PE1**	**PE2**	**PE3**	**PE4**	**PE5**
000000b (initial value)	Hi-Z					
000001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C
000010b	—	—	—	—	MTIOC1A	MTIOC2B
000111b	—	—	—	POE8#	—	—
001000b	MTIOC3D	MTIOC3B	MTIOC7A	MTIOC1B	MTIOC4A	—
001100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—
011101b	—	—	TIC3	TOC3	—	—
011110b	—	—	—	—	—	COMP0
100111b	—	—	—	—	MTIOC7D	—

—: Do not specify this value.

Table 21.34 Register Settings for Input/Output Pin Function in 48-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**PE1**	**PE2**	**PE3**	**PE4**
000000b (initial value)	Hi-Z			
000001b	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D
000010b	—	—	—	MTIOC1A
000111b	—	—	POE8#	—
001000b	MTIOC3B	MTIOC7A	MTIOC1B	MTIOC4A
001100b	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—
011101b	—	TIC3	TOC3	—
100111b	—	—	—	MTIOC7D

—: Do not specify this value.

21.2.17 PF5 Pin Function Control Register (PF5PFS)

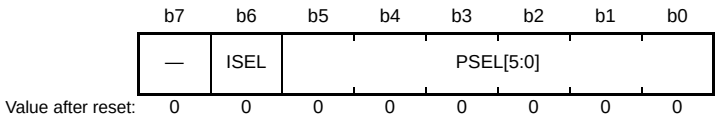
Address(es): PF5PFS 0008 C1BDh

	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 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

21.2.18 PHn Pin Function Control Register (PHnPFS) (n = 0 to 3)

Address(es): PH0PFS 0008 C1C8h, PH1PFS 0008 C1C9h, PH2PFS 0008 C1CAh, PH3PFS 0008 C1CBh



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 21.35.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PH1: IRQ0 (144/100/80/64/48 pin) PH2: IRQ1 (144/100/80/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.35 Register Settings for Input/Output Pin Function in 144-/100-/80-/64-/48-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**PH0**	**PH1**	**PH2**	**PH3**
000000b (initial value)	Hi-Z			
000001b	MTIOC3B	MTIOC3D	MTIOC4C	MTIOC4D
000101b	—	TMO0	TMRI0	TMCI0
000111b	CACREF	—	—	—
001001b	ADTRG0#	ADST0	—	—
011101b	—	TIC1	TOC1	—

—: Do not specify this value.

21.2.19 PJn Pin Function Control Register (PJnPFS) (n = 1, 3, 5)

Address(es): PJ1PFS 0008 C1D1h, 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 21.36, Table 21.37, and Table 21.38.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PJ3: IRQ11 (144/100 pin) PJ5: IRQ13 (144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.36 Register Settings for Input/Output Pin Function in 144-Pin LFQFP

	Pin		
PSEL[5:0] Settings	**PJ1**	**PJ3**	**PJ5**
000000b (initial value)	Hi-Z		
000001b	MTIOC3A	MTIOC3C	—
001010b	—	CTS6# RTS6# SS6#	—
001011b	—	CTS0# RTS0# SS0#	CTS2# RTS2# SS2#
100001b	—	—	POE8#

—: Do not specify this value.

Table 21.37 Register Settings for Input/Output Pin Function in 100-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**PJ1**	**PJ3**
000000b (initial value)	Hi-Z	
000001b	MTIOC3A	MTIOC3C
001010b	—	CTS6# RTS6# SS6#
001011b	—	CTS0# RTS0# SS0#

—: Do not specify this value.

Table 21.38 Register Settings for Input/Output Pin Function in 80-Pin LFQFP

	Pin
PSEL[5:0] Settings	**PJ1**
000000b (initial value)	Hi-Z
000001b	MTIOC3A

21.2.20 PKn Pin Function Control Register (PKnPFS) (n = 2 to 5)

Address(es): PK2PFS 0008 C1DAh, PK3PFS 0008 C1DBh, PK4PFS 0008 C1DCh, PK5PFS 0008 C1DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.39.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.39 Register Settings for Input/Output Pin Function in 144-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**PK2**	**PK3**	**PK4**	**PK5**
000000b (initial value)	Hi-Z			
001010b	TXD9 SMOSI9 SSDA9	RXD9 SMISO9 SSCL9	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4

21.2.21 CS Output Enable Register (PFCSE)

Address(es): 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	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 3)	R/W
b1	CS1E	CS1 Enable		R/W
b2	CS2E	CS2 Enable		R/W
b3	CS3E	CS3 Enable		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CSnE Bit (CSn Enable) (n = 0 to 3)

These bits enable or disable the corresponding pins to output the CSn# signal.

To enable output of the CSn# signal, set the corresponding CSnE bit in PFCSE to 1.

21.2.22 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[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS0S[1:0]	CS0# Output Pin Select*1	b1 b0 0 0: Set P24 as CS0# output pin 0 1: Set P60 as CS0# output pin 1 X: Set PC7 as CS0# output pin	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select*2	b3 b2 0 0: Set P25 as CS1# output pin 0 1: Set P61 as CS1# output pin 1 0: Set P71 as CS1# output pin 1 1: Set PC6 as CS1# output pin	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select*3	b5 b4 0 0: Set P26 as CS2# output pin 0 1: Set P62 as CS2# output pin 1 0: Set P72 as CS2# output pin 1 1: Set PC5 as CS2# output pin	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select*4	b7 b6 0 0: Set P27 as CS3# output pin 0 1: Set P63 as CS3# output pin 1 0: Set P73 as CS3# output pin 1 1: Set PC4 as CS3# output pin	R/W

X: Don't care

Note 1. Since P60 is not present in 100-pin products, do not set the CS0S[1:0] bits of those products to 01b.
Note 2. Since P61 and P71 are not present in 100-pin products, do not set the CS1S[1:0] bits of those products to 01b or 10b.
Note 3. Since P62 and P72 are not present in 100-pin products, do not set the CS2S[1:0] bits of those products to 01b or 10b.
Note 4. Since P63 and P73 are not present in 100-pin products, do not set the CS3S[1:0] bits of those products to 01b or 10b.

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 0 to 3)

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

21.2.23 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

21.2.24 Address Output Enable Register 1 (PFAOE1)

Address(es): 0008 C105h

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

21.2.25 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 A20 Output Enable	See Table 21.40.	R/W
b2	ADRHMS2	A16 to A20 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 21.40 Settings for External Address Buses A16 to A20

ADRHMS Bit	ADRHMS2 Bit	Settings for External Address Buses A16 to A20
0	0	Set PC0 to PC4.
0	1	Set PC0, PC1, P71, P72, and P74.
1	0	Set P90 to P93. (No allocation of A20)
1	1	Setting prohibited

21.2.26 External Bus Control Register 1 (PFBCR1)

Address(es): 0008 C107h

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

Note 1. Even if these bits are set to 00b in 144- and 100-pin products, P55 is set as WAIT# input pin.

WAITS[1:0] Bits (WAIT Select)

To use the external wait when the external bus is enabled, follow the procedure below.

1. Set the external wait enable bit in the relevant CSn mode register (CSnMOD.EWENB) to 1 to enable the external wait.
2. Set the WAITS[1:0] bits to a value other than 00 to specify the port pin to be used for the WAIT# function.
3. Set the bits corresponding to the selected port pin in the port mode register (PMR) and port direction register (PDR) to 0.

To use all of the pins for functions other than WAIT# when the external bus is enabled, disable the external wait function by setting the external wait enable bit in the CSn mode register (CSnMOD.EWENB) to 0, and set the WAITS[1:0] bits to 00.

ALEOE Bit (ALE Output Enable)

This bit enables or disables output of the ALE pin.

21.2.27 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.

21.2.28 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.

21.3 How to Set the External Bus Interface

If the external bus interface is to be used, set the MPC registers according to Table 21.41 and then set the external bus enable bit (EXBE) in system control register 0 (SYSCR0) to 1.
Table 21.41 lists how to set up port pins to act as the external bus interface. For details on the relevant registers of the MPC, refer to section 21.2, Register Descriptions.

Table 21.41 How to Set the External Bus Interface (1/2)

Port	Output Signal	Settings of MPC Registers	
		144-Pin	100-Pin
P24	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S[1:0] = 00	
P25	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 00	
P26	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 00	
P27	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 00	
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[1:0] = 01	(not provided)
P61	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 01	(not provided)
	D0[A0/D0]	PFBCR2.D0S[1:0] = 11	(not provided)
P62	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 01	(not provided)
	D1[A1/D1]	PFBCR2.D1S[1:0] = 11	(not provided)
P63	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 01	(not provided)
	D2[A2/D2]	PFBCR2.D2S[1:0] = 11	(not provided)
P64	D3[A3/D3]	PFBCR2.D3S[1:0] = 11	(not provided)
P71	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 10	(not provided)
	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1	(not provided)
P72	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 10	(not provided)
	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1	(not provided)
P73	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 10	(not provided)
P74	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1	(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	

Table 21.41 How to Set the External Bus Interface (2/2)

Port	Output Signal	Settings of MPC Registers	
		144-Pin	100-Pin
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] = 11	
PC5	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 11	
	WAIT#	PFBCR1.WAITS[1:0] = 10	
	D3[A3/D3]	PFBCR2.D3S[1:0] = 10	
PC6	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 11	
	D2[A2/D2]	PFBCR2.D2S[1:0] = 10	
PC7	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S[1:0] = 10/11	
PD0	D0[A0/D0]	PFBCR2.D0S[1:0] = 00	
PD1	D1[A1/D1]	PFBCR2.D1S[1:0] = 00	
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	

21.4 Usage Notes

21.4.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the 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, J and K, 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.

21.4.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is 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 to 2, 4, A, D, and E also function as analog Input/output pins for the A/D converter, comparator, and D/A converter. When using these ports as analog Input/output pins, set them to general input pins by clearing the corresponding bits in the port mode register (PMR) and port direction register (PDR) to 0 and set the PmnPFS.ASEL bit to 1, to avoid degradation of accuracy.
(5) The initial value of the time capture event input pin enable bit (TCEN) of the time capture control register y (RTCCRy, y = 0 to 2) is undefined after a reset. Therefore, set this bit to 0 to avoid unnecessary input.
(6) Points to note regarding the port mode register (PMR), port direction register (PDR), and the PmnPFS register settings for pins that have multiplexed pin functions are listed in Table 21.42. 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 21.42 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 21.2 to Table 21.39)	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.

21.4.3 Notes on the Use of Analog Functions

To use an analog function, set the corresponding bits in both the port mode register (PMR) and port direction register (PDR) to 0 so that the pin acts as a general input port. After that, set the pin function select bit in the Pmn pin function control register (PmnPFS.ASEL) to 1.

22. Multi-Function Timer Pulse Unit 3 (MTU3a)

22.1 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 22.1 shows the specifications of the MTU and Table 22.2 lists the functions of the MTU. Figure 22.1 and Figure 22.2 show block diagrams of the MTU.

Table 22.1 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 22.2 MTU Functions (1/2)

Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
Count clock	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTIOC1A	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB
External clock for phase counting mode	—	MTCLKA MTCLKB	MTCLKA MTCLKB MTCLKC MTCLKD	MTCLKA MTCLKB MTCLKC MTCLKD	—	—	—	—	—	—
General registers (TGR)	TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRALW TGRBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB	TGRA TGRB
General registers/ buffer registers	TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	TGRC TGRD
I/O pins	MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D	MTIOC8A MTIOC8B MTIOC8C MTIOC8D
Counter clear function	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output: 0 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Compare match output: 1 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Compare match output: Toggle output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Input capture function	✓	✓	✓	✓*1	✓	✓	✓	✓	✓	✓*2
Synchronous operation	✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 1	✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 2	✓	✓	✓	—	—	—	—	—	—	—
Complementary PWM mode	—	—	—	—	✓	✓	—	✓	✓	—
Reset-synchronized PWM mode	—	—	—	—	✓	✓	—	✓	✓	—
AC synchronous motor drive mode	✓	—	—	—	✓	✓	—	—	—	—
Phase counting mode	—	✓	✓	✓	—	—	—	—	—	—
Buffer operation	✓	—	—	—	✓	✓	—	✓	✓	✓
Dead time compensation counter function	—	—	—	—	—	—	✓	—	—	—
DMAC/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 22.2 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	—	Five sources • Compare match 6A • Compare match 6B • Compare match 6C • Compare match 6D • Overflow	Six sources • Compare match 7A • Compare match 7B • Compare match 7C • Compare match 7D • Overflow • Underflow*3	Five sources • Compare match 8A • Compare match 8B • Compare match 8C • Compare match 8D • Overflow
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)	—	• 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									

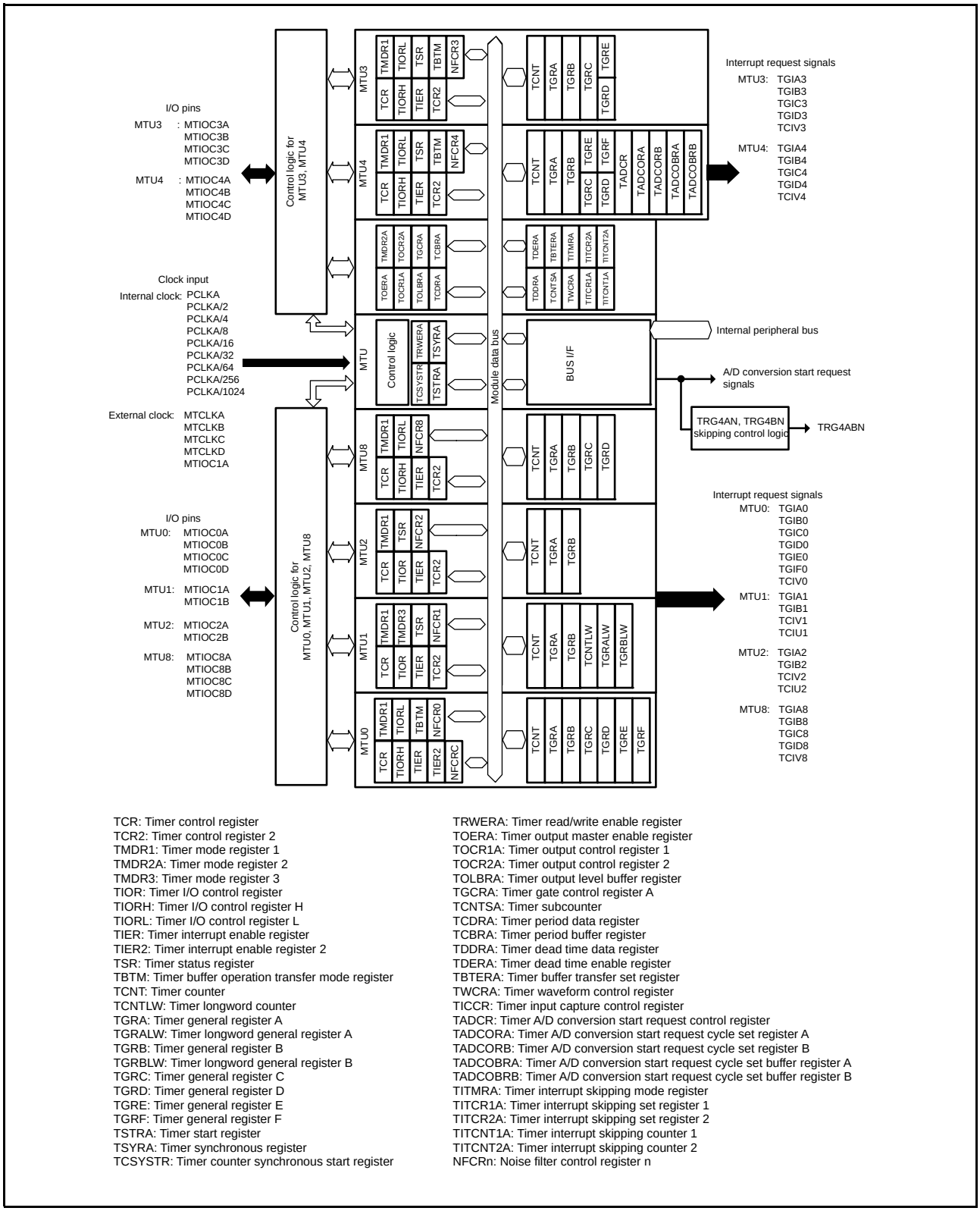
✓: Possible —: Not possible

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.

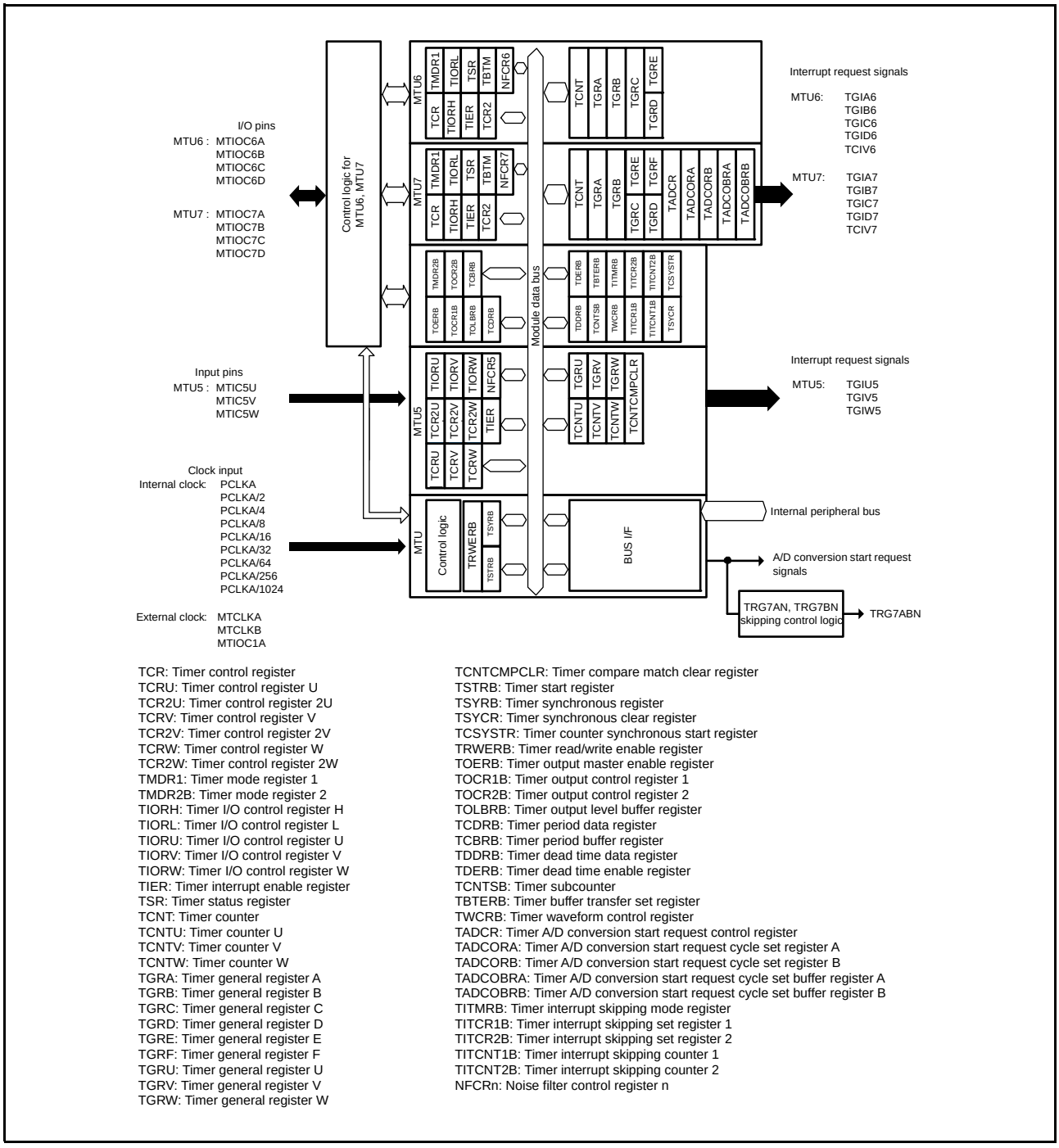
Note 3. Underflow is available only in complementary PWM mode.

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 22.1 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 22.2 Block Diagram of MTU (MTU5 to MTU7)

Table 22.3 shows the configuration of pins for the MTU.

Table 22.3 Pin Configuration of the MTU

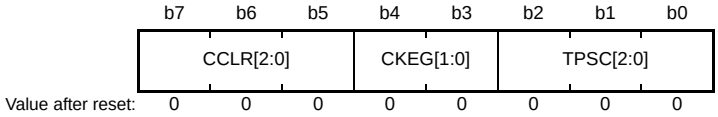
Channel	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 and MTU2 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 and MTU2 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	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

22.2 Register Descriptions

22.2.1 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 22.6 to Table 22.9.	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 22.4 and Table 22.5.	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 22.6 to Table 22.9 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 22.4 and Table 22.5 for details.

Table 22.4 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 22.5 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 22.10.	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 22.10 for details.

22.2.2 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 22.6 to Table 22.9.	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)

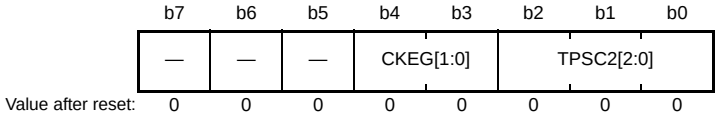
These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 22.6 to Table 22.9 for details.

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 22.3.6, 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 22.10.	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 22.10 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 22.6 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 22.7 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 22.8 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 22.9 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 22.10 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

22.2.3 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 22.11 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 22.11 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 22.50 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 22.50 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.

22.2.4 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.

22.2.5 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 22.12.

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 22.65, Clock Input Pins in Phase Counting Mode for details.

Table 22.12 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.

22.2.6 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 22.27 MTU1.TIOR: Table 22.29 MTU2.TIOR: Table 22.30 MTU3.TIORH: Table 22.31 MTU4.TIORH: Table 22.33 MTU6.TIORH: Table 22.35 MTU7.TIORH: Table 22.37 MTU8.TIORH: Table 22.39	R/W
b7 to b4	IOB[3:0]	I/O Control B*1	Refer to the following tables. MTU0.TIORH: Table 22.13 MTU1.TIOR: Table 22.15 MTU2.TIOR: Table 22.16 MTU3.TIORH: Table 22.17 MTU4.TIORH: Table 22.19 MTU6.TIORH: Table 22.21 MTU7.TIORH: Table 22.23 MTU8.TIORH: Table 22.25	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 22.28 MTU3.TIORL: Table 22.32 MTU4.TIORL: Table 22.34 MTU6.TIORL: Table 22.36 MTU7.TIORL: Table 22.38 MTU8.TIORL: Table 22.40	R/W
b7 to b4	IOD[3:0]	I/O Control D*1	Refer to the following tables. MTU0.TIORL: Table 22.14 MTU3.TIORL: Table 22.18 MTU4.TIORL: Table 22.20 MTU6.TIORL: Table 22.22 MTU7.TIORL: Table 22.24 MTU8.TIORL: Table 22.26	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 22.41	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 22.13 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 22.14 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 22.15 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 22.16 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 22.17 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 22.18 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 22.19 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 22.20 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 22.21 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 22.22 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 22.23 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 22.24 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 22.25 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 22.26 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 22.27 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 22.28 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 22.29 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 22.30 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 22.31 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 22.32 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 22.33 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 22.34 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 22.35 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 22.36 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 22.37 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 22.38 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 22.39 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 22.40 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 22.41 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 22.3.11, External Pulse Width Measurement and section 22.3.12, Dead Time Compensation.

22.2.7 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).

22.2.8 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).

22.2.9 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.

22.2.10 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.

22.2.11 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.

22.2.12 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.

22.2.13 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 22.2.5, Timer Mode Register 3 (TMDR3) for details.

22.2.14 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 22.2.5, Timer Mode Register 3 (TMDR3) for details. This register can only be used in 32-bit phase counting mode.

22.2.15 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 22.2.5, Timer Mode Register 3 (TMDR3) for details.

22.2.16 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 22.2.5, 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.

22.2.17 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

22.2.18 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].

22.2.19 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

22.2.20 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)

22.2.21 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 20, 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 22.44 and Figure 22.48).

- 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 20, 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 22.44 and Figure 22.48).

22.2.22 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 22.42.	R/W
b1	OLSN	Output Level Select N*1, *3	Refer to Table 22.43.	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 22.42 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 22.43 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 22.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

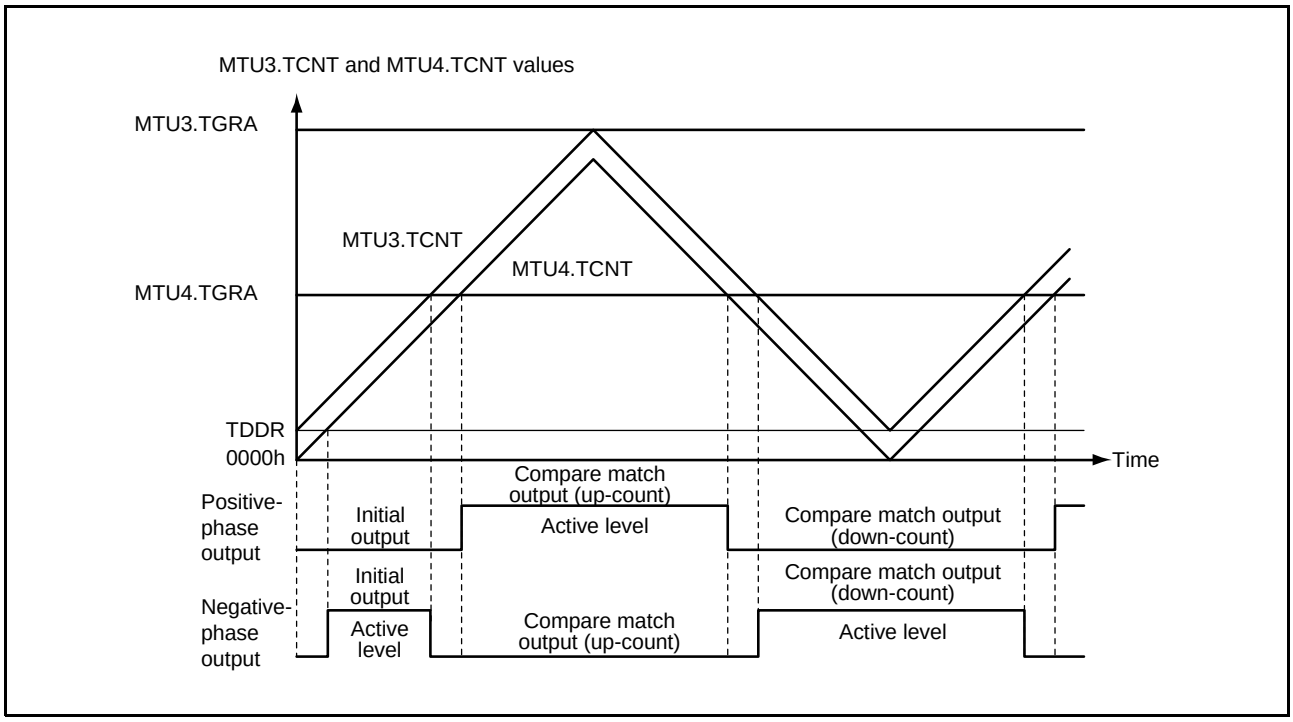


Figure 22.3 Example of Output in Complementary PWM Mode

22.2.23 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 22.44.	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 22.45.	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 22.46.	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 22.47.	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 22.48.	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 22.49.	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 22.50 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 22.44 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 22.45 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 22.46 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 22.47 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 22.48 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 22.49 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 22.50 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

22.2.24 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 22.4 shows an example of the PWM output level setting procedure in buffer operation.

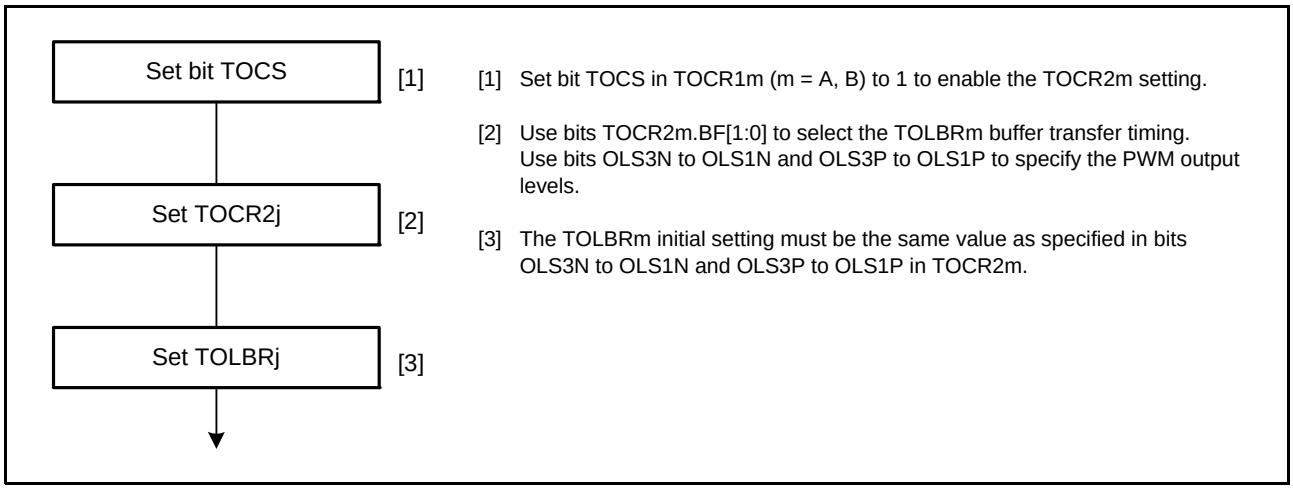


Figure 22.4 Example of PWM Output Level Setting Procedure in Buffer Operation

22.2.25 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 22.51.	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 22.51 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 22.51 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

22.2.26 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.

22.2.27 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.

22.2.28 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.

22.2.29 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.

22.2.30 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

22.2.31 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 22.52.	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 22.52 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 22.3.8, 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.

22.2.32 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 22.50.
[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 22.50.
[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

22.2.33 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.

22.2.34 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.

22.2.35 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 22.53 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 22.53 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 22.54 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 22.54 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

22.2.36 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 22.3.9 (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.

22.2.37 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.

22.2.38 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.

22.2.39 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 22.55.	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 22.56.	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 22.57.	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 22.58.	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 22.55 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 22.56 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 22.57 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 22.58 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.

22.2.40 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

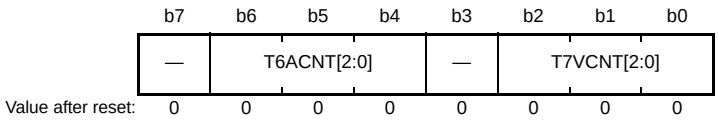
T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[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

22.2.41 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 22.59.	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 22.59 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 22.60.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 22.60 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.

22.2.42 Timer Interrupt Skipping Counter 2m (TITCNT2m) (m = A, B)

- MTU.TITCNT2A

Address(es): MTU.TITCNT2A 000C 123Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter	These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0.	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)

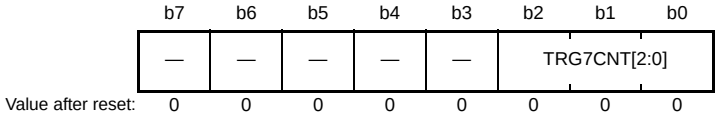
These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[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



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

22.3 Operation

22.3.1 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 22.5 shows an example of the count operation setting procedure.

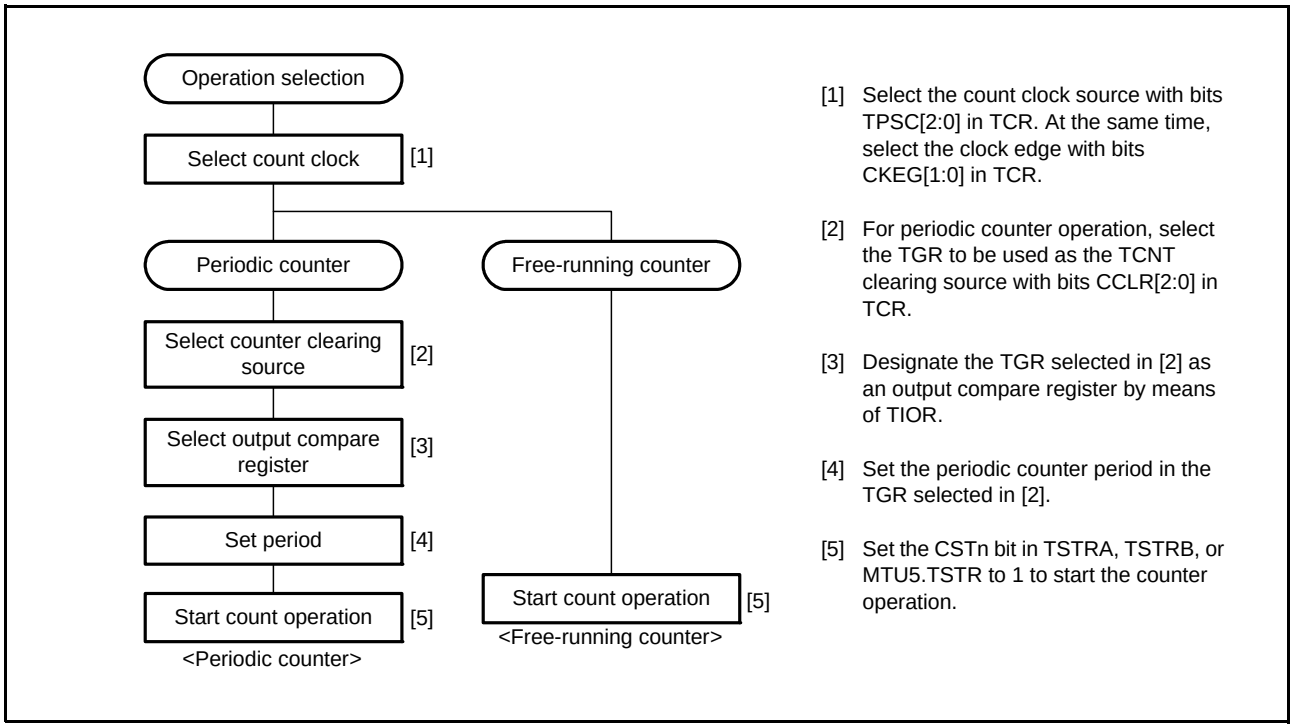


Figure 22.5 Example of Count Operation Setting Procedure

(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 22.6 illustrates free-running counter operation.

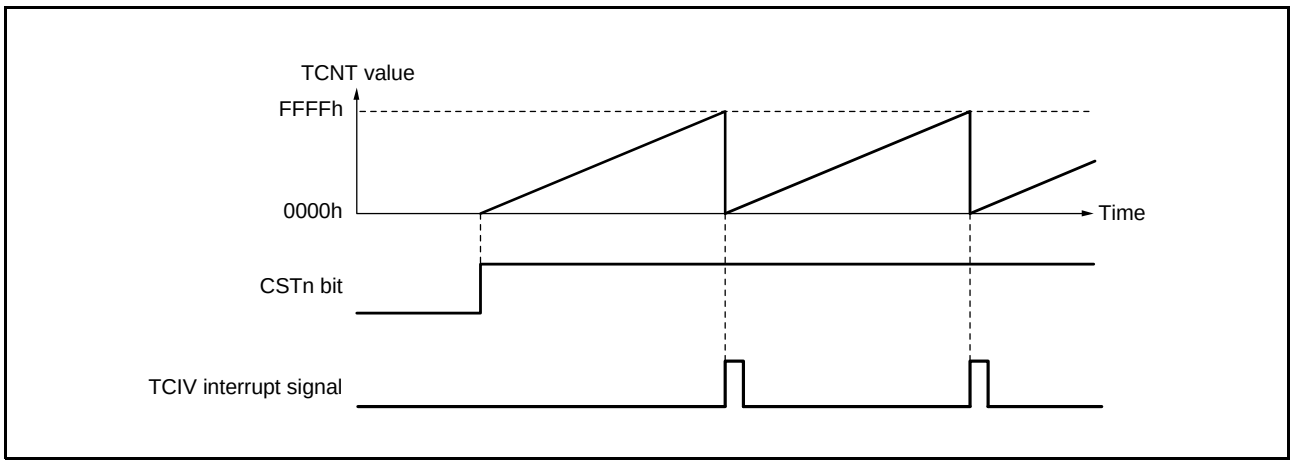


Figure 22.6 Free-Running Counter Operation

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 22.7 illustrates periodic counter operation.

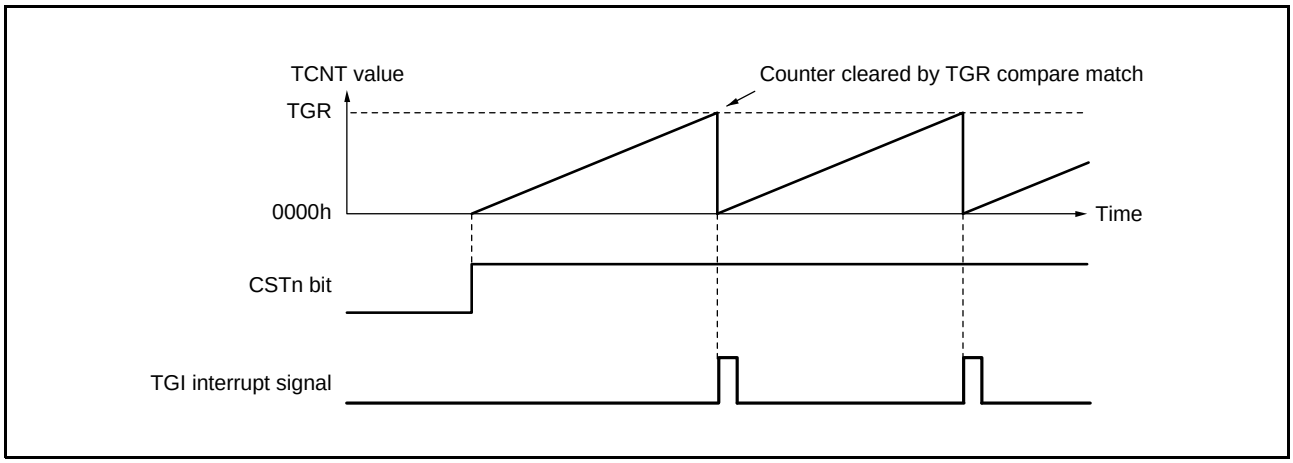


Figure 22.7 Periodic Counter Operation

(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 22.8 shows an example of the procedure for setting waveform output by compare match

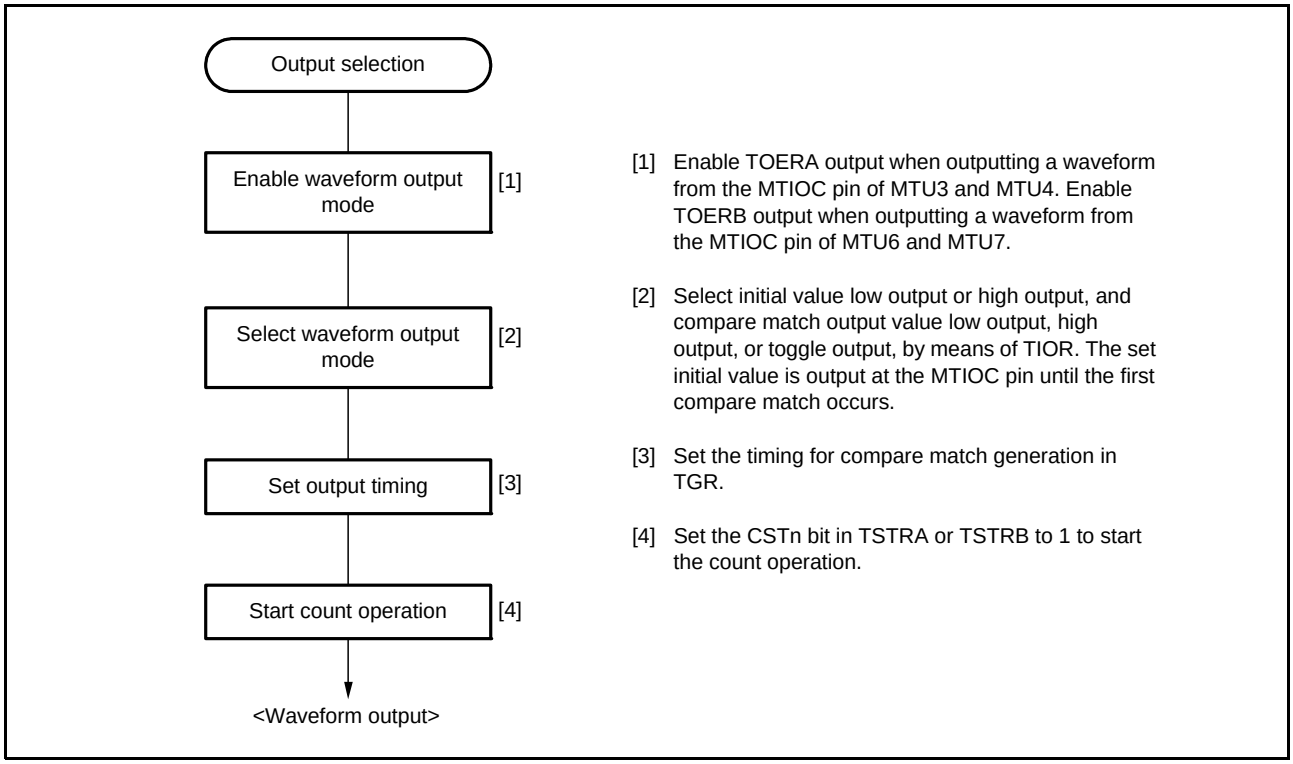


Figure 22.8 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 22.9 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.

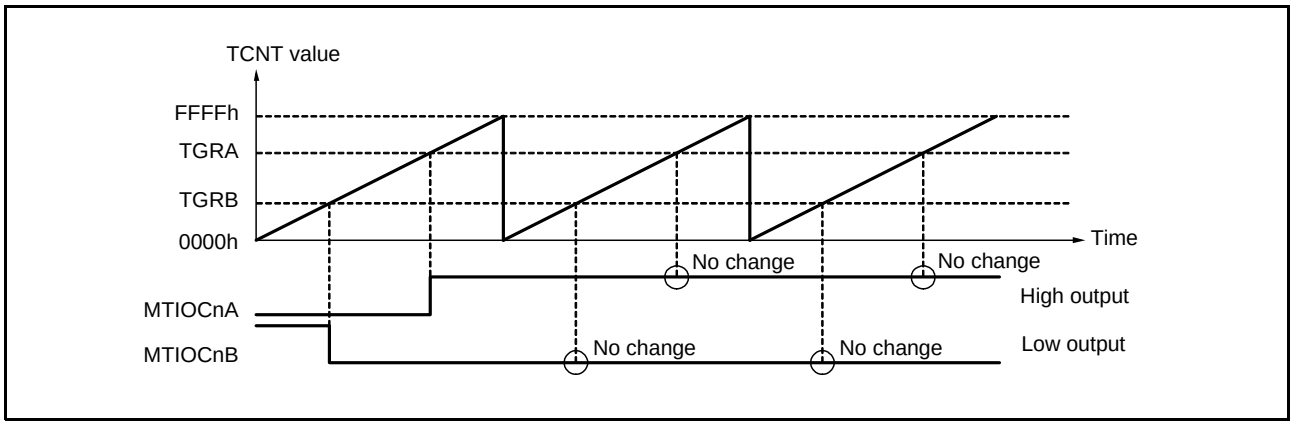


Figure 22.9 Example of low output and high output Operation (n = 0 to 4, 6, 7, 8)

Figure 22.10 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.

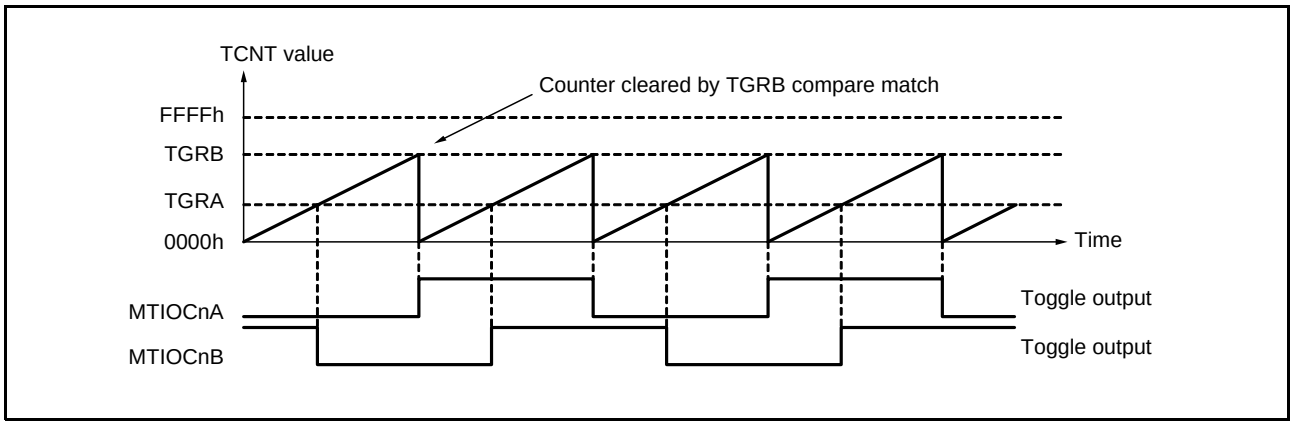


Figure 22.10 Example of Toggle Output Operation (n = 0 to 4, 6, 7, 8)

(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 22.11 shows an example of the input capture operation setting procedure.

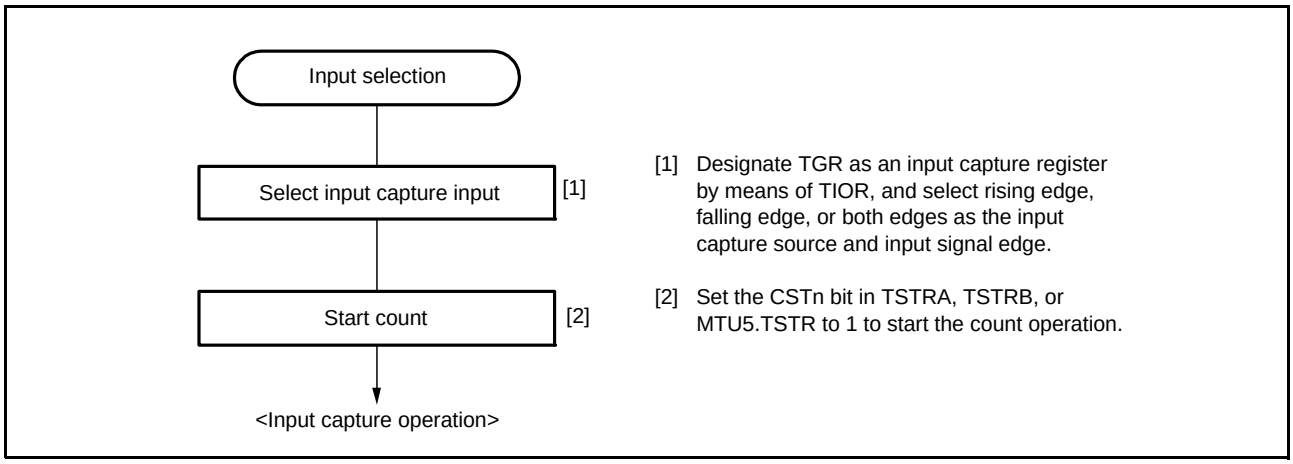


Figure 22.11 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 22.12 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)

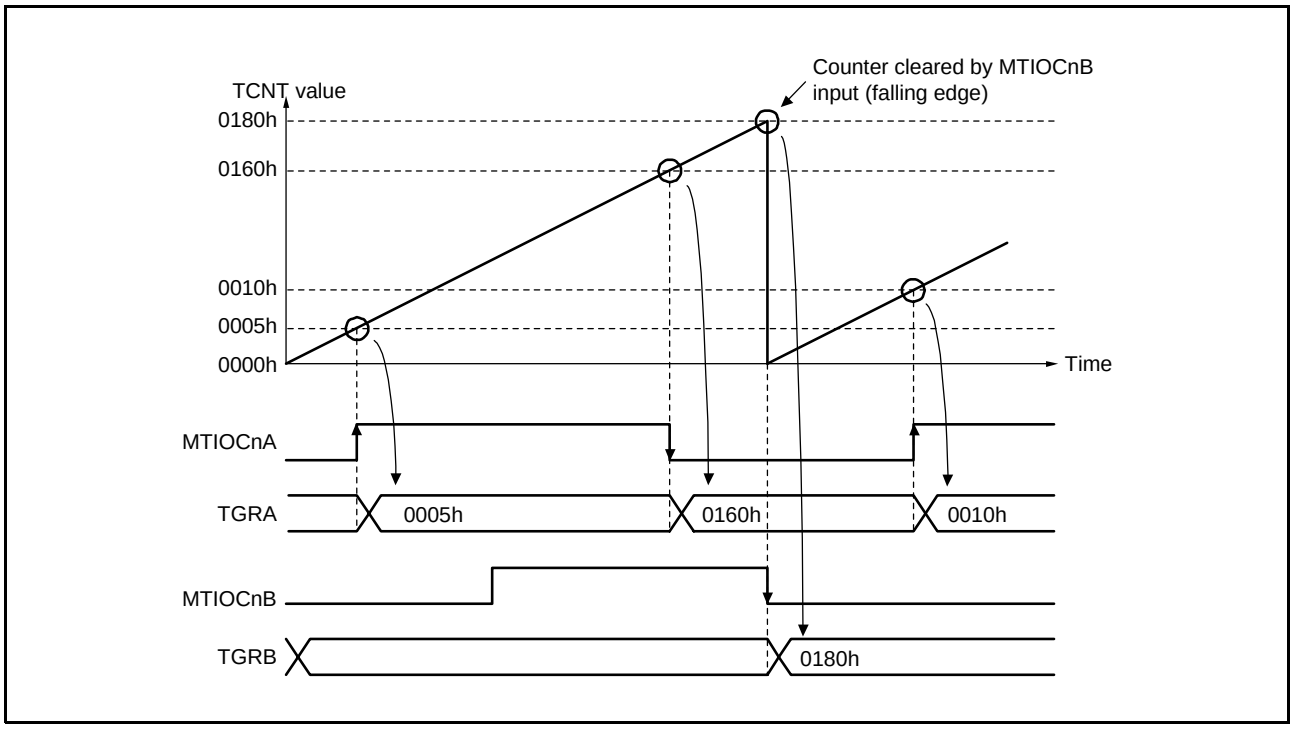


Figure 22.12 Example of Input Capture Operation (n = 0 to 4, 6, 7, 8)

22.3.2 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 22.13 shows an example of the synchronous operation setting procedure.

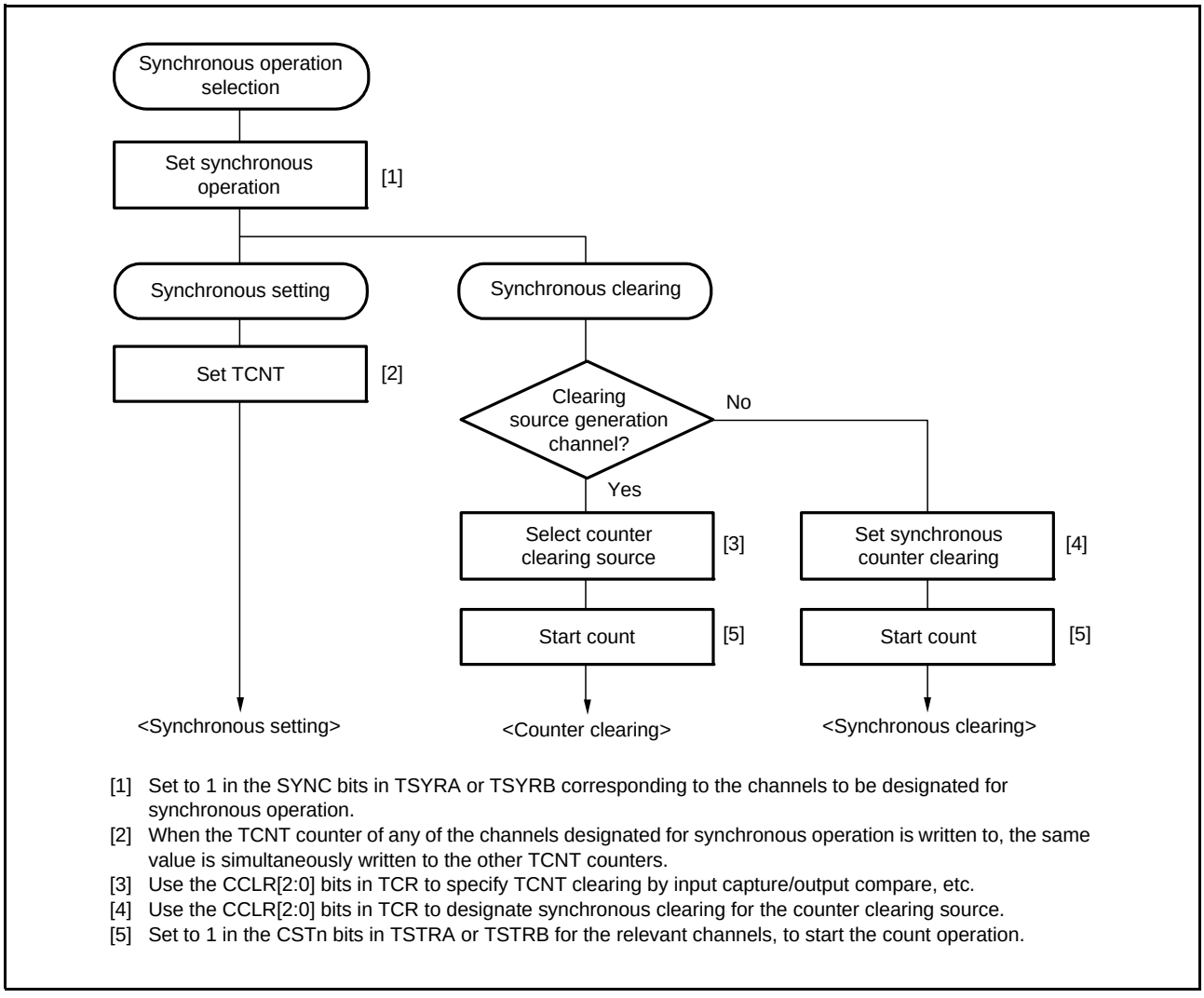


Figure 22.13 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 22.14 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 22.3.5, PWM Modes.

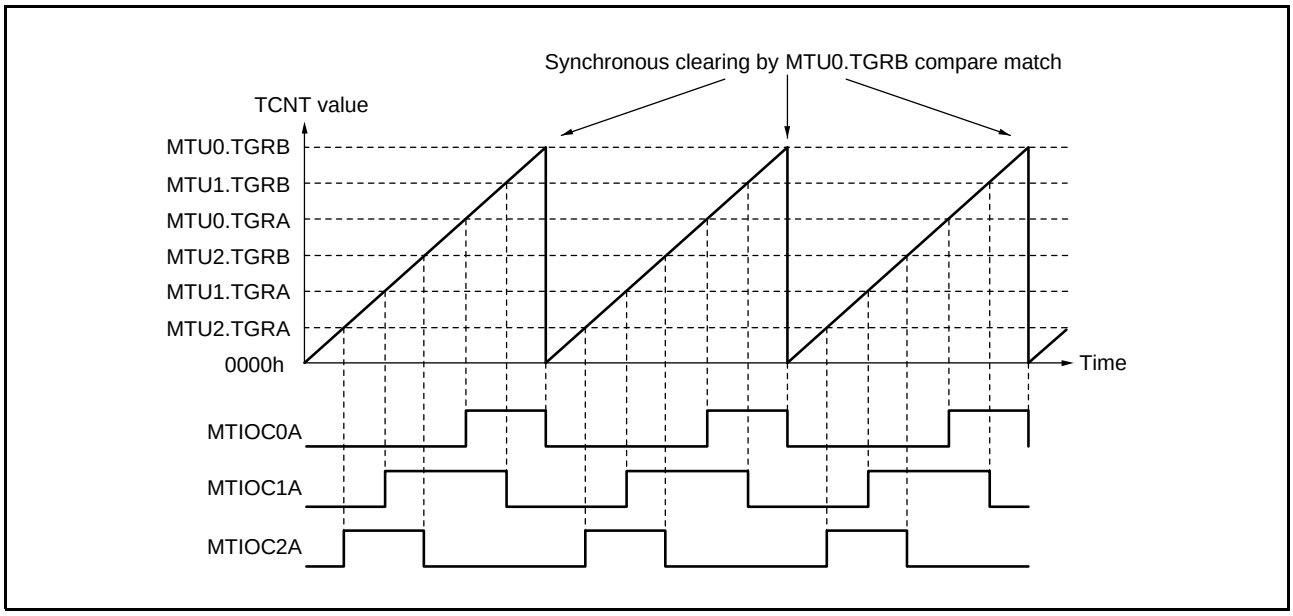


Figure 22.14 Example of Synchronous Operation

22.3.3 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 22.61 shows the register combinations used in buffer operation.

Table 22.61 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 22.15.

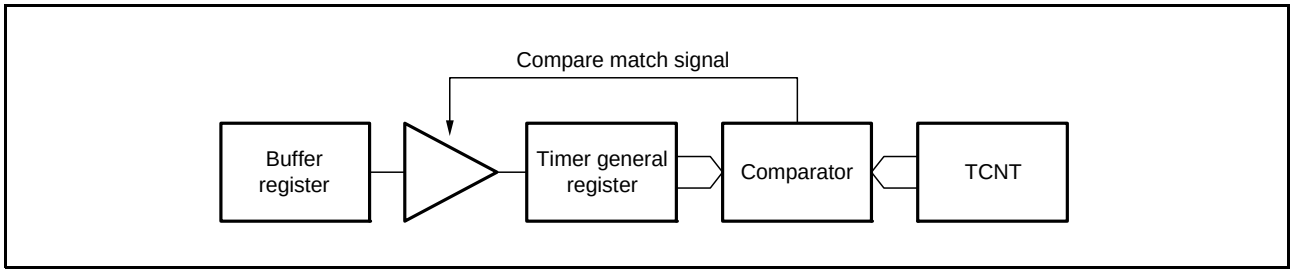


Figure 22.15 Compare Match Buffer Operation

- 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 22.16.

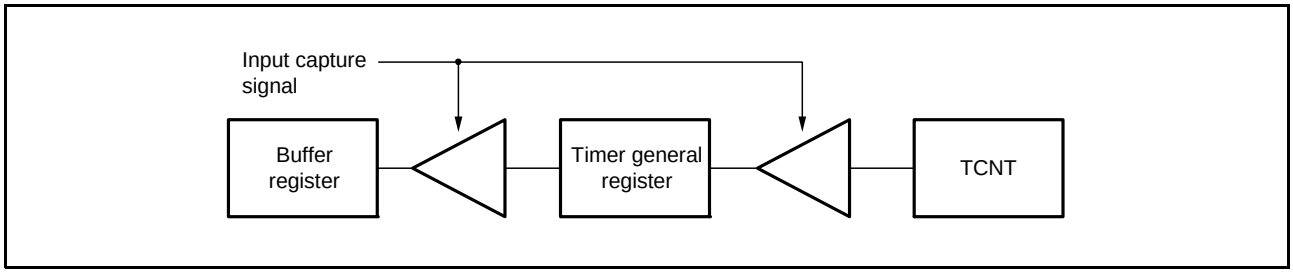


Figure 22.16 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 22.17 shows an example of the buffer operation setting procedure.

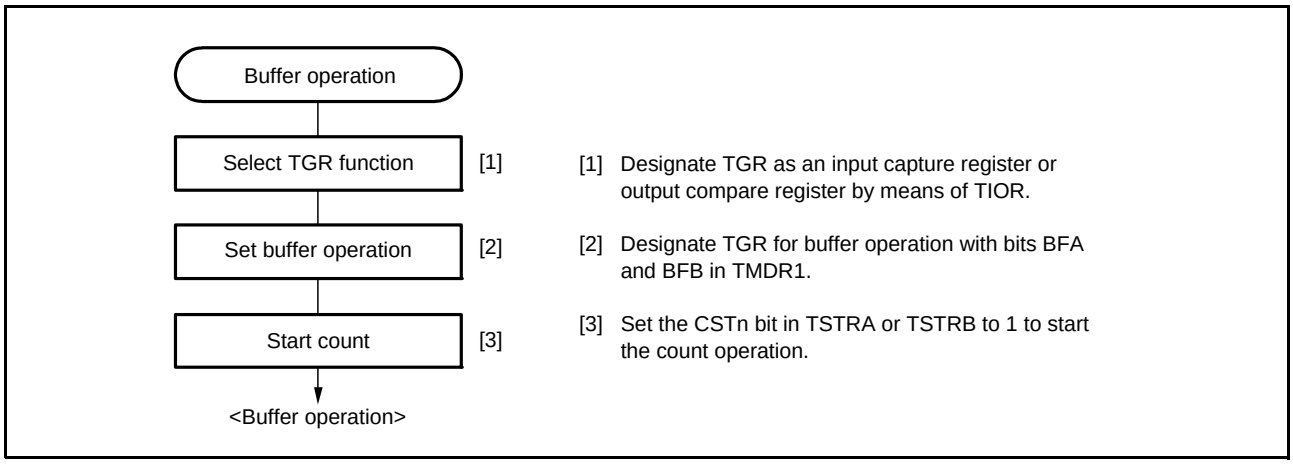


Figure 22.17 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 22.18 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 22.3.5, PWM Modes.

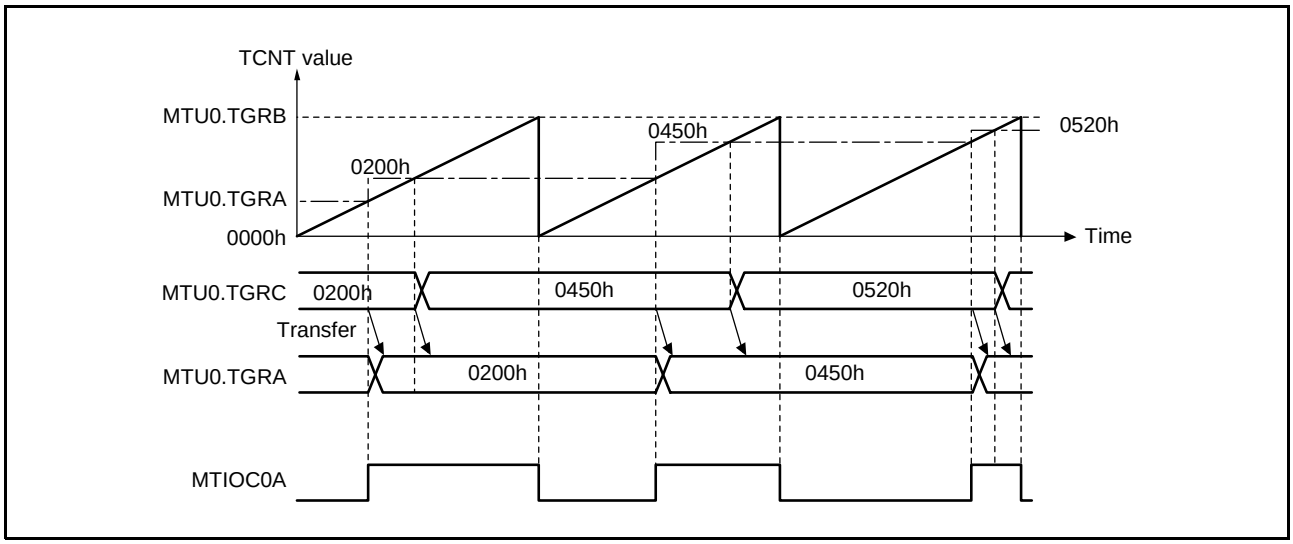


Figure 22.18 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 22.19 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.

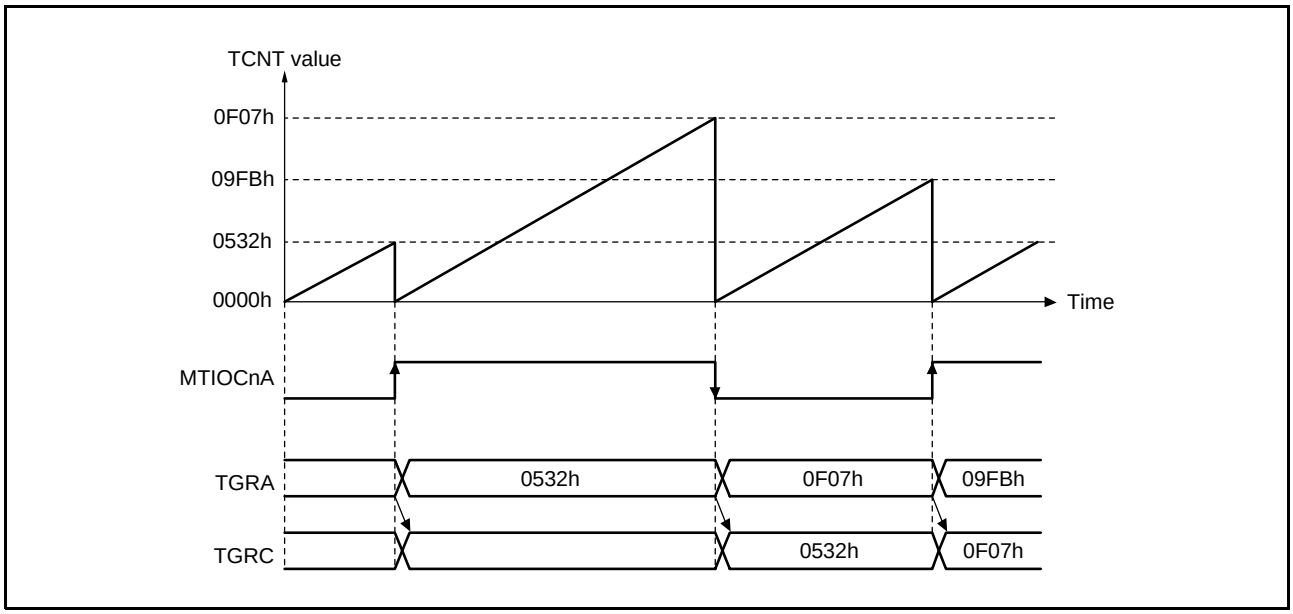


Figure 22.19 Example of Buffer Operation (2) (n = 0 to 4, 6, 7, 8)

(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 22.20 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.

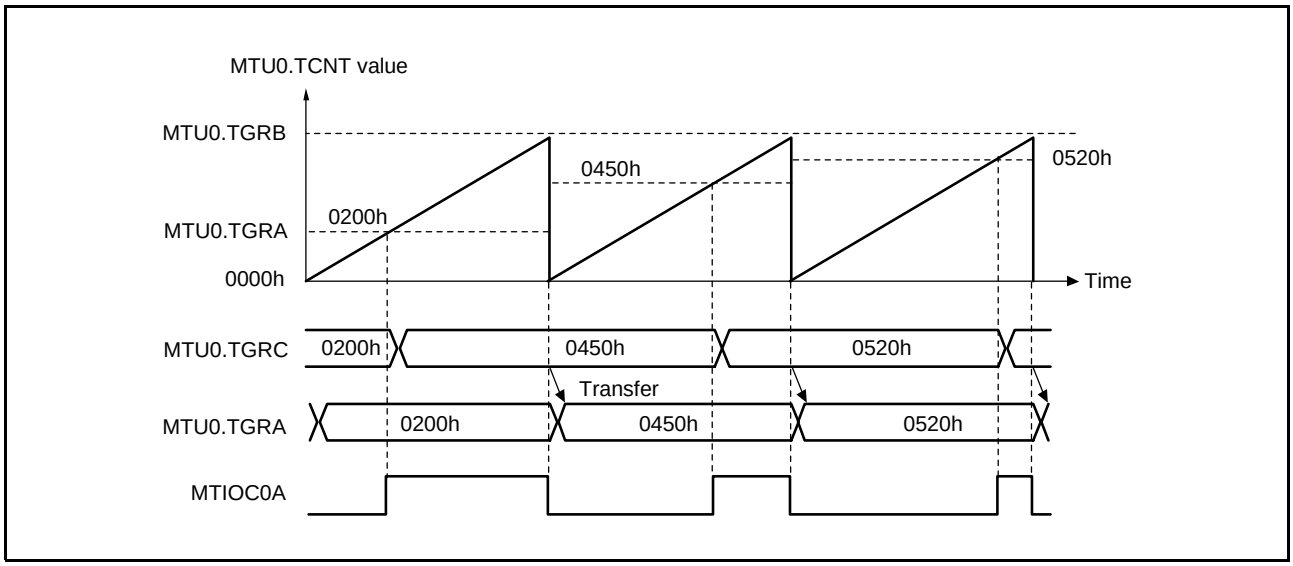


Figure 22.20 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

22.3.4 Cascaded Operation

In cascaded operation, 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 22.3.6.2, 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 22.62 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 22.62 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 22.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 22.63 shows the TICCR setting and input capture input pins.

Table 22.63 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 22.21 shows an example of the cascaded operation setting procedure.

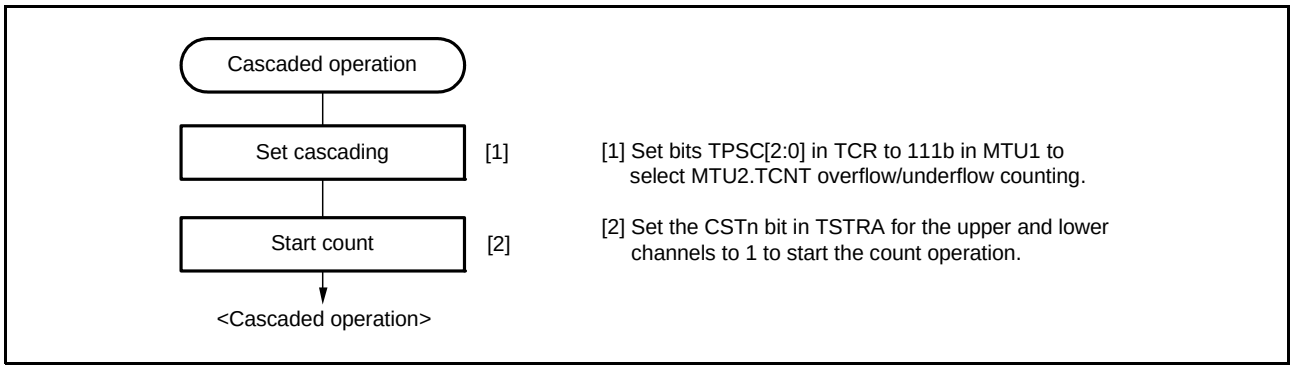


Figure 22.21 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 22.22 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.

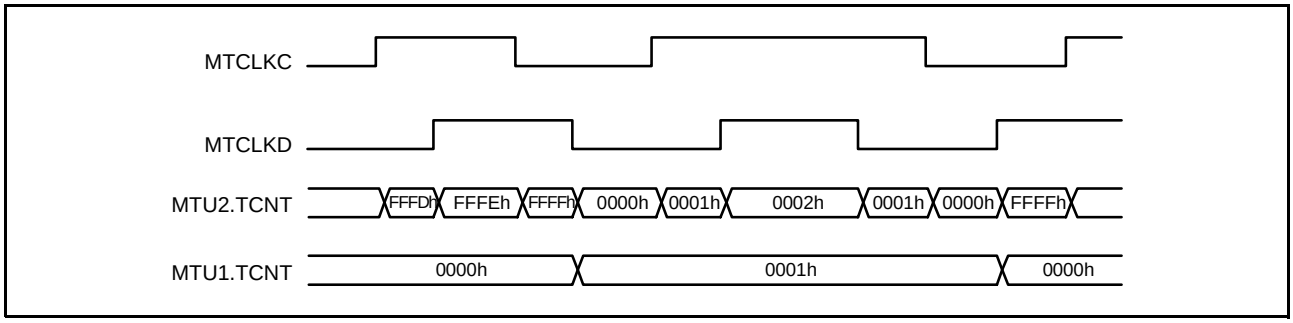


Figure 22.22 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 22.23 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.

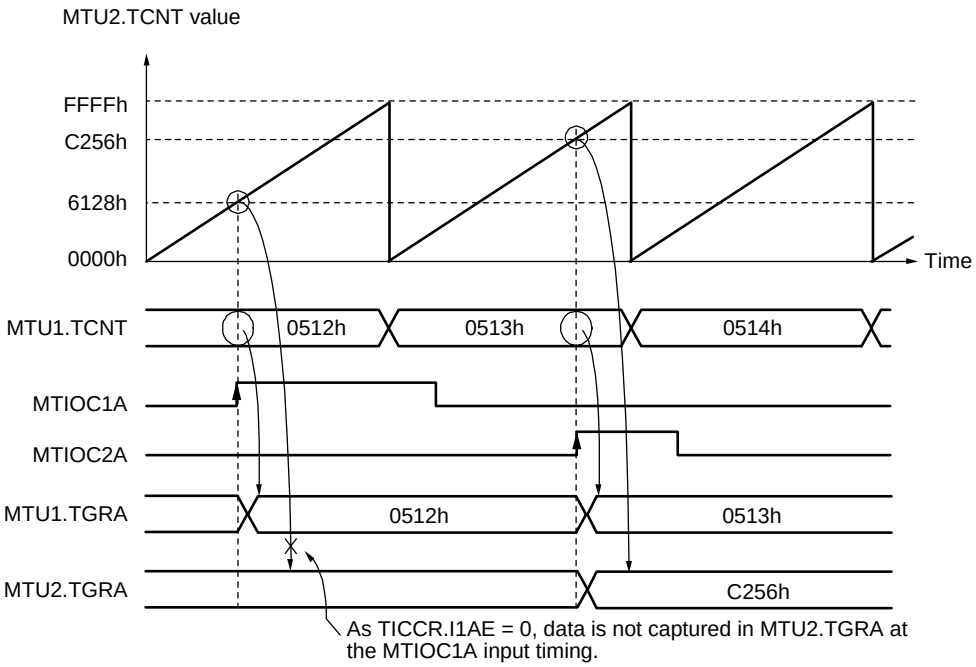
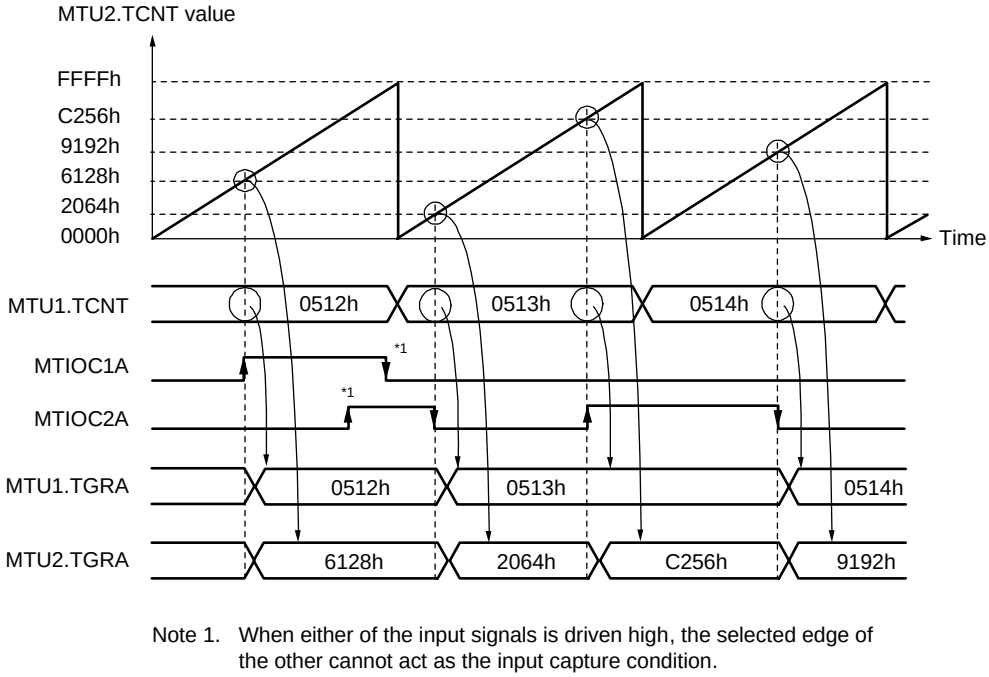


Figure 22.23 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 22.24 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 22.24 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 22.25 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.

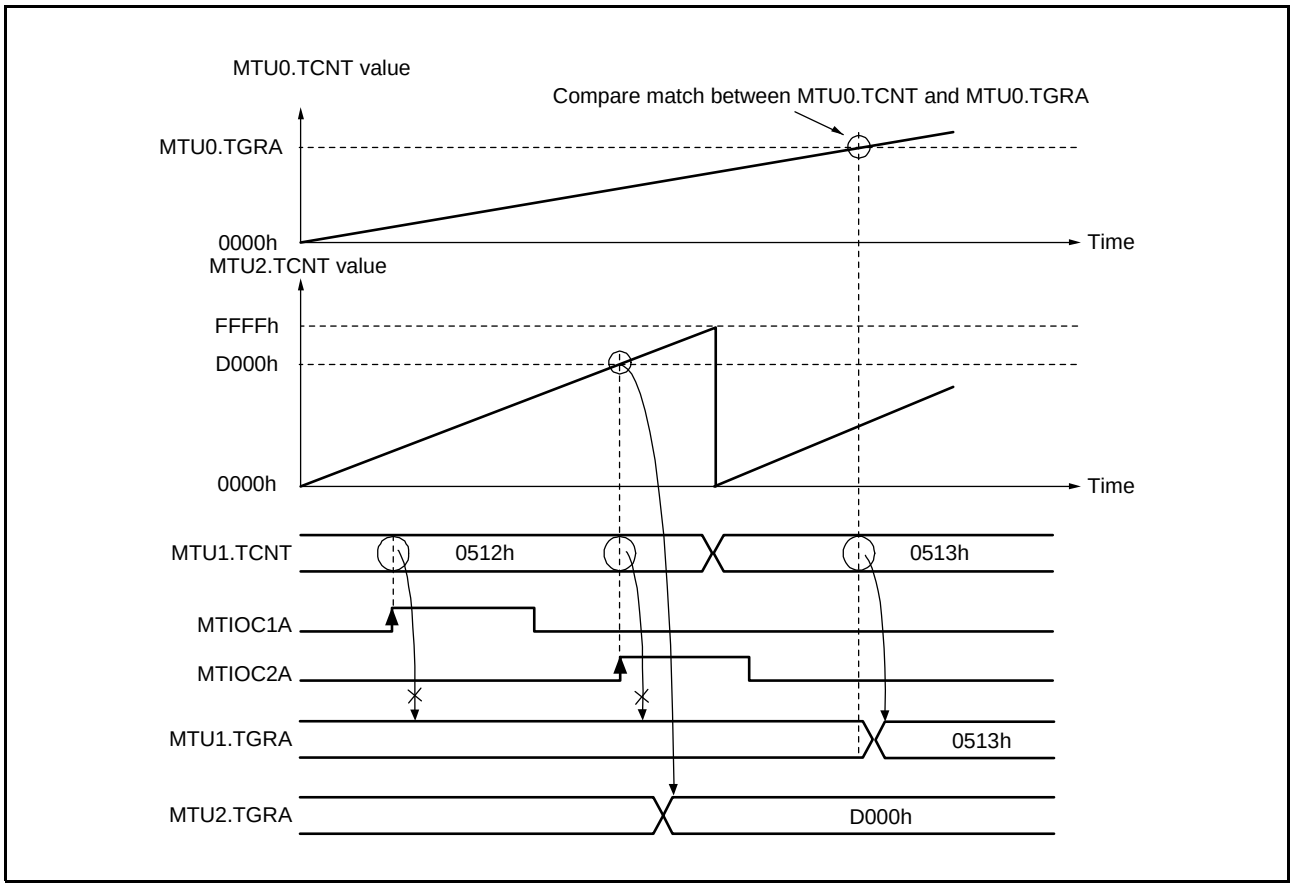


Figure 22.25 Cascaded Operation Example (d)

22.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM 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 22.64.

Table 22.64 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 22.26 shows an example of the PWM mode setting procedure.

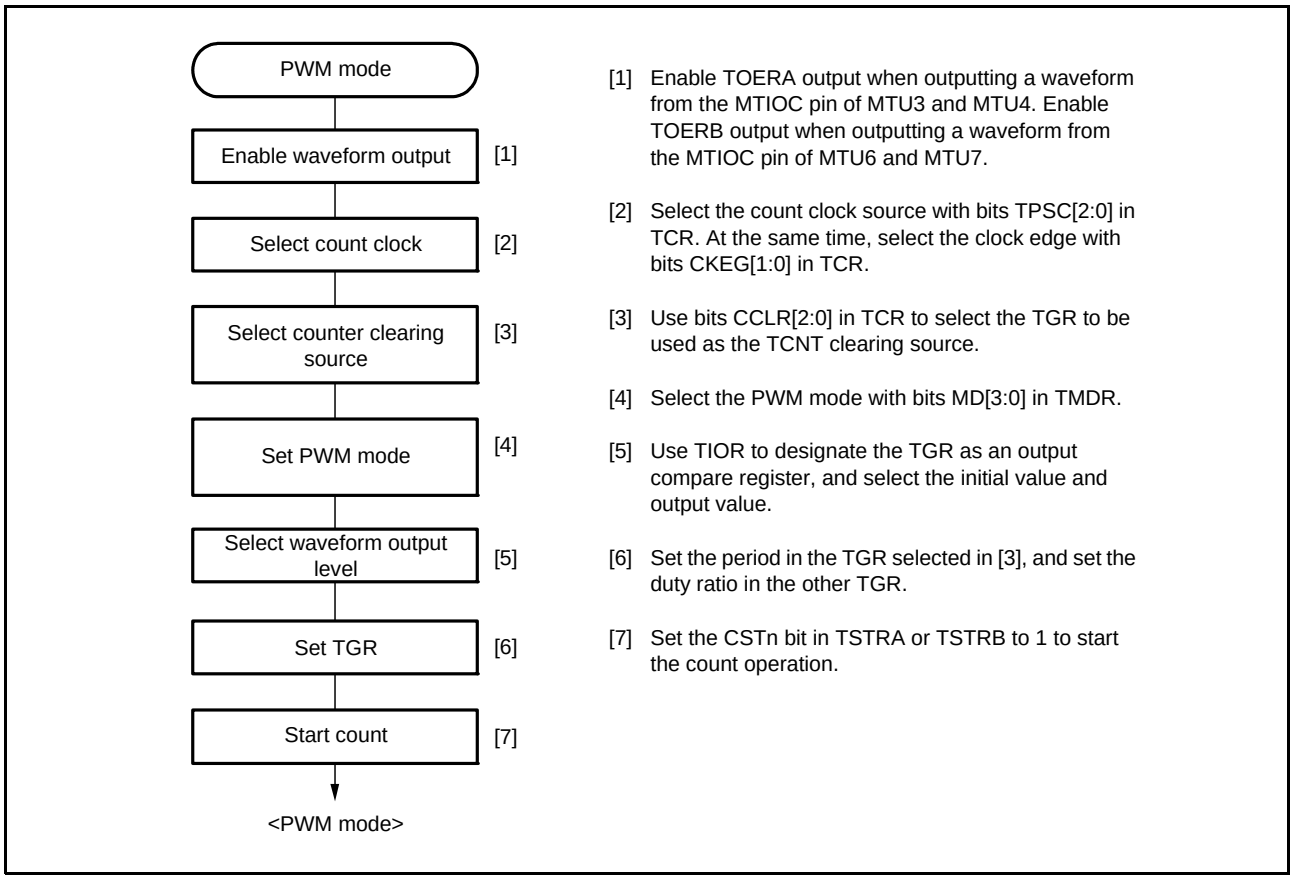


Figure 22.26 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 22.27 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.

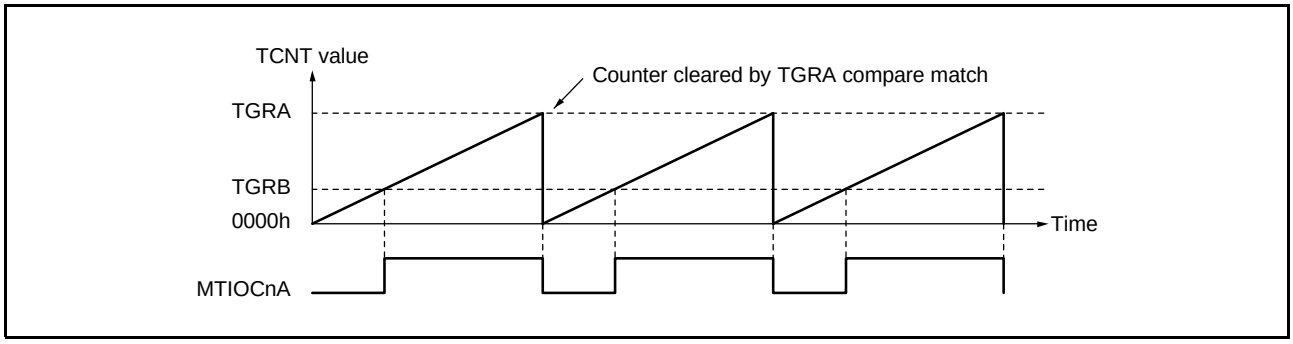


Figure 22.27 Example of PWM Mode 1 Operation (n = 0 to 4, 6, 7)

Figure 22.28 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.

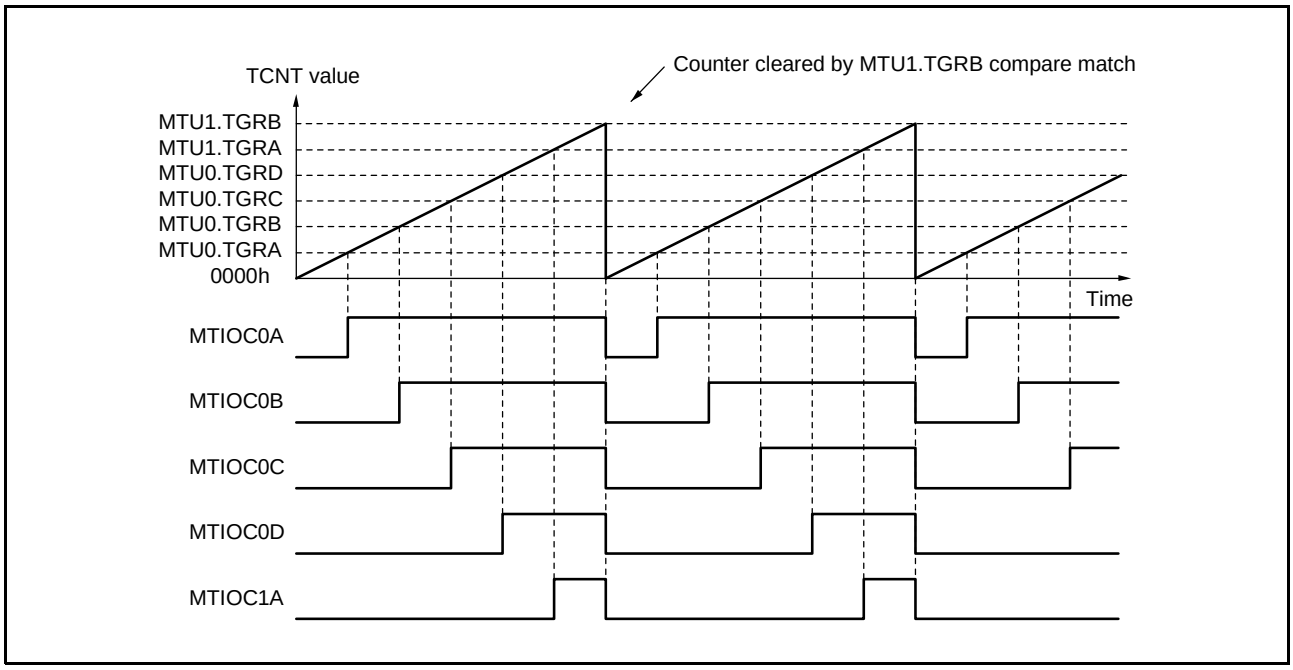


Figure 22.28 Example of PWM Mode 2 Operation

Figure 22.29 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.

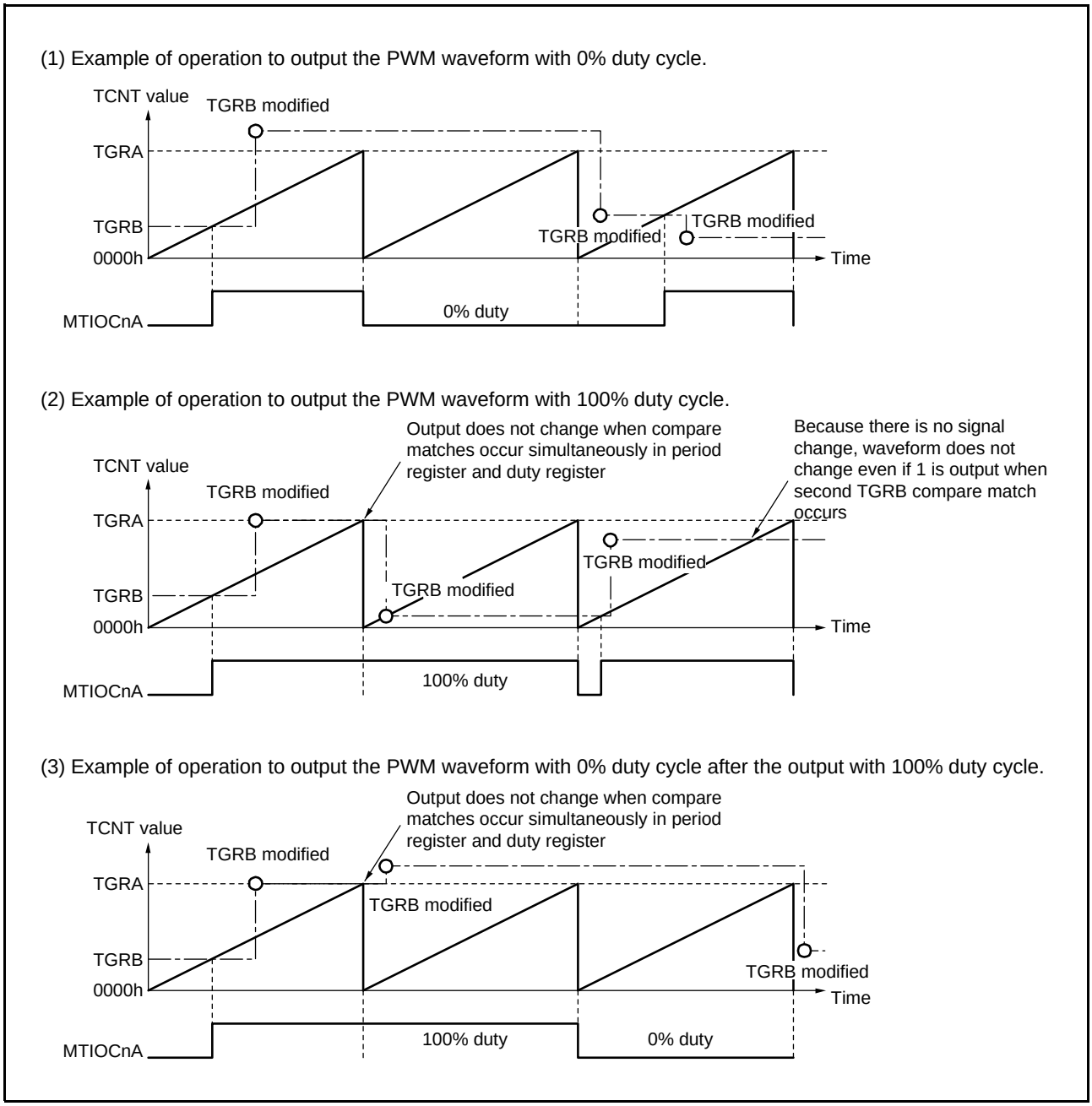


Figure 22.29 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4, 6, 7)

22.3.6 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 22.65 lists the external clock input pins to be connected in each phase counting mode.

Table 22.65 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

22.3.6.1 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 22.30 shows an example of the phase counting mode setting procedure.

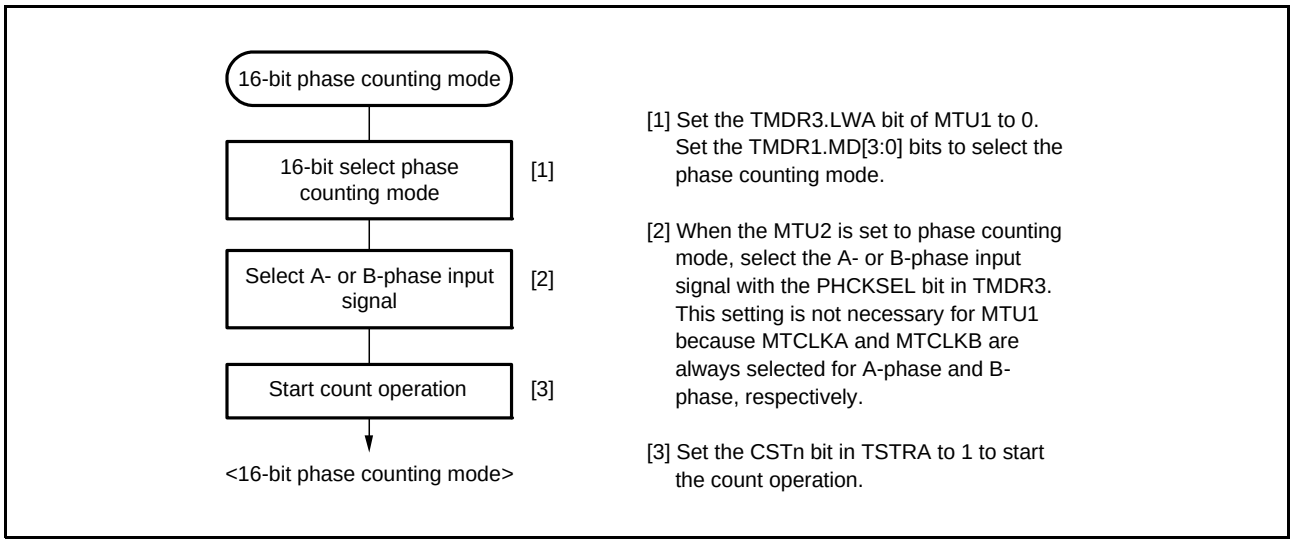


Figure 22.30 Example of 16-Bit Phase Counting Mode Setting Procedure

(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 22.31 shows an example of operation in phase counting mode 1, and Table 22.66 summarizes the TCNT up-counting and down-counting conditions.

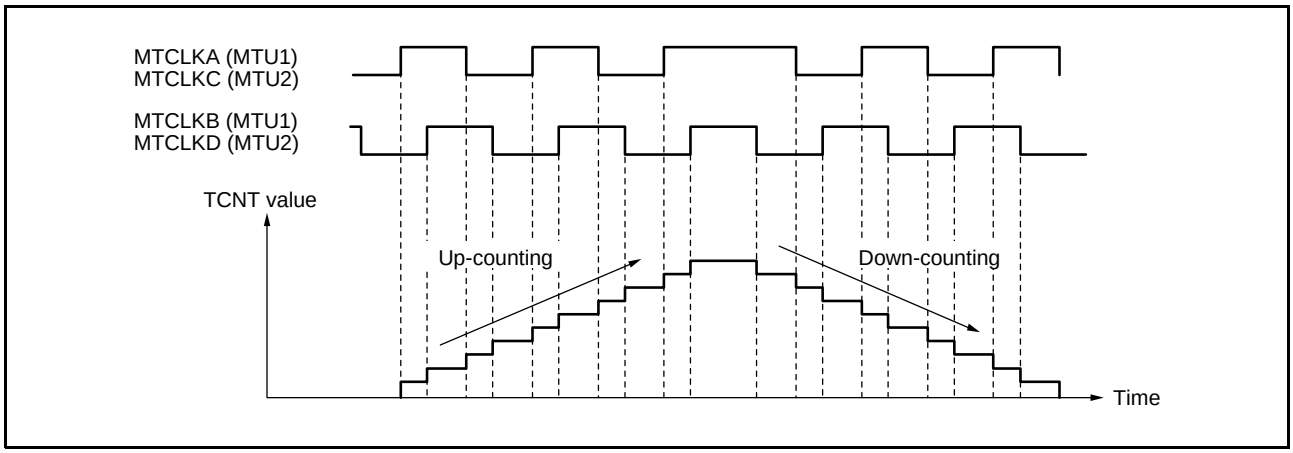


Figure 22.31 Example of Operation in Phase Counting Mode 1

Table 22.66 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 22.32 to Figure 22.34 show the examples of operation in phase counting mode 2 and Table 22.67 summarizes the TCNT up-counting and down-counting conditions.

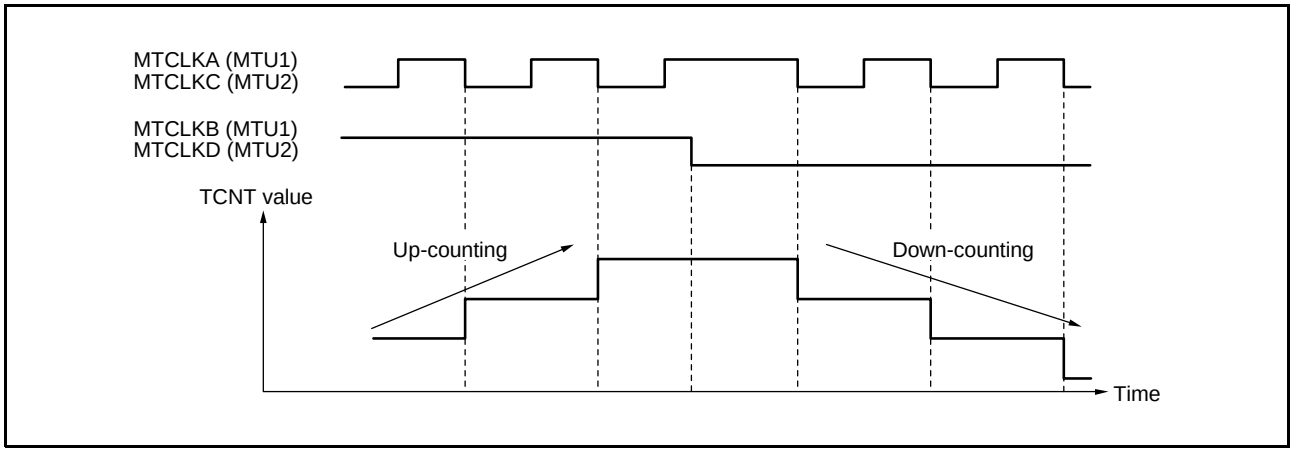


Figure 22.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

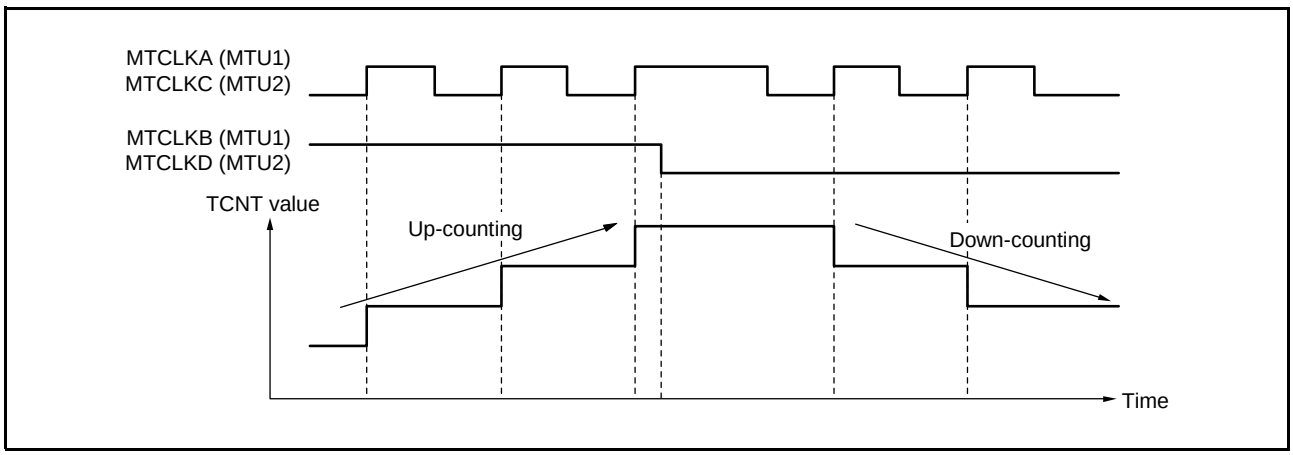


Figure 22.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

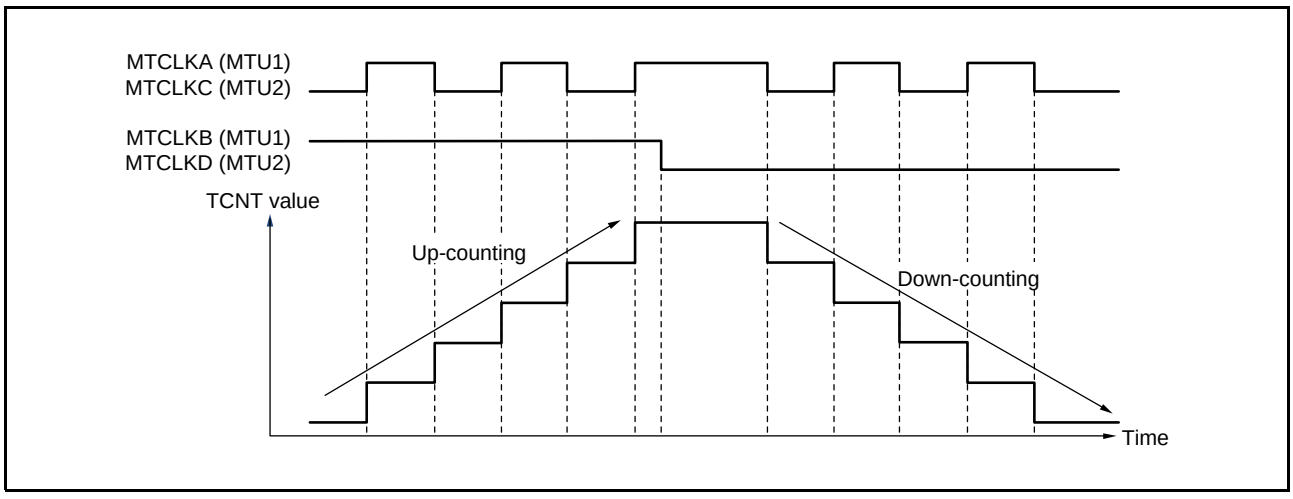


Figure 22.34 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 22.67 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 22.35 to Figure 22.37 show the examples of operation in phase counting mode 3 and Table 22.68 summarizes the TCNT up-counting and down-counting conditions.

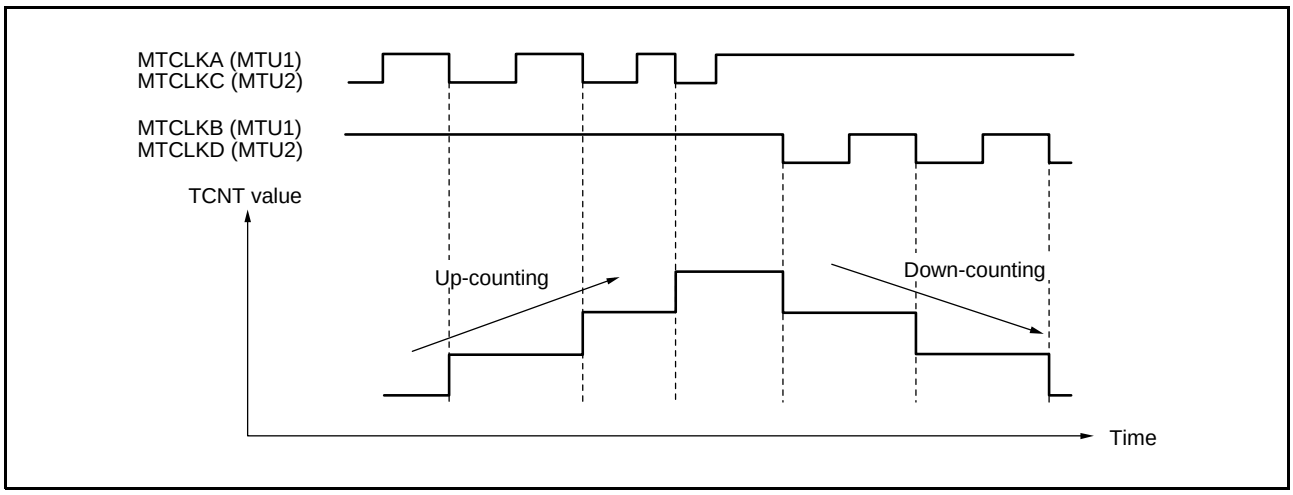


Figure 22.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

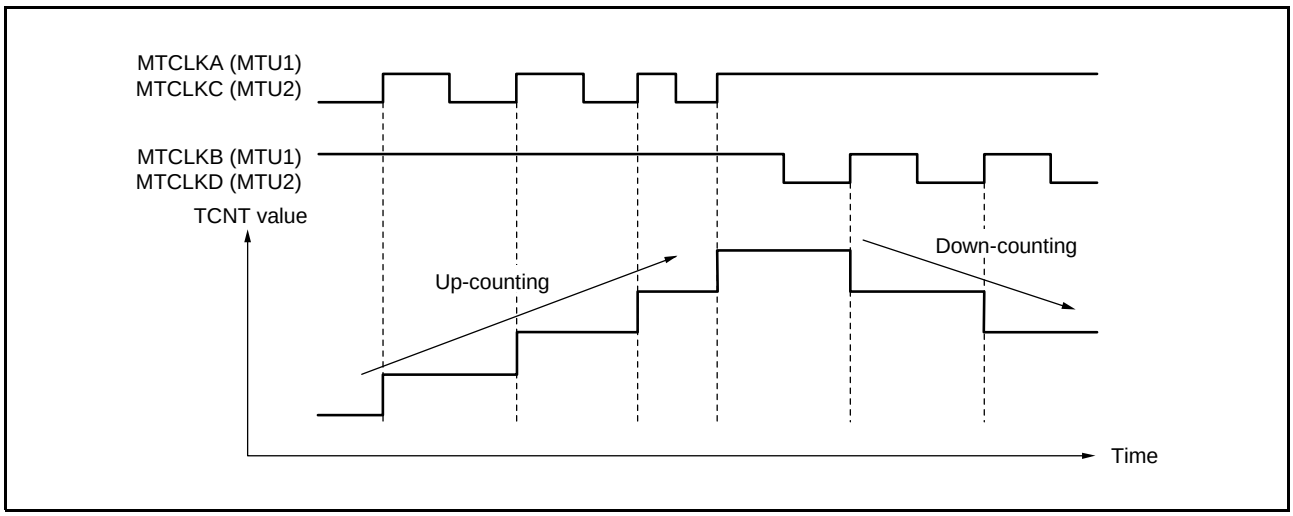


Figure 22.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

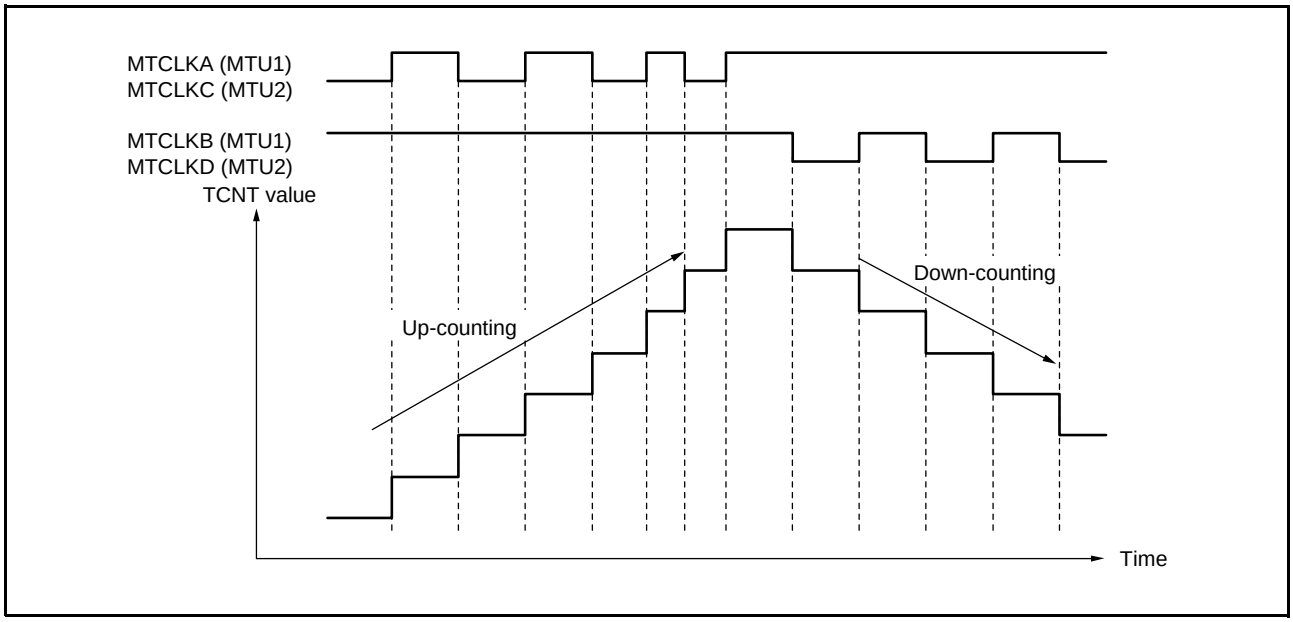


Figure 22.37 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 22.68 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 22.38 shows an example of operation in phase counting mode 4, and Table 22.69 summarizes the TCNT up-counting and down-counting conditions.

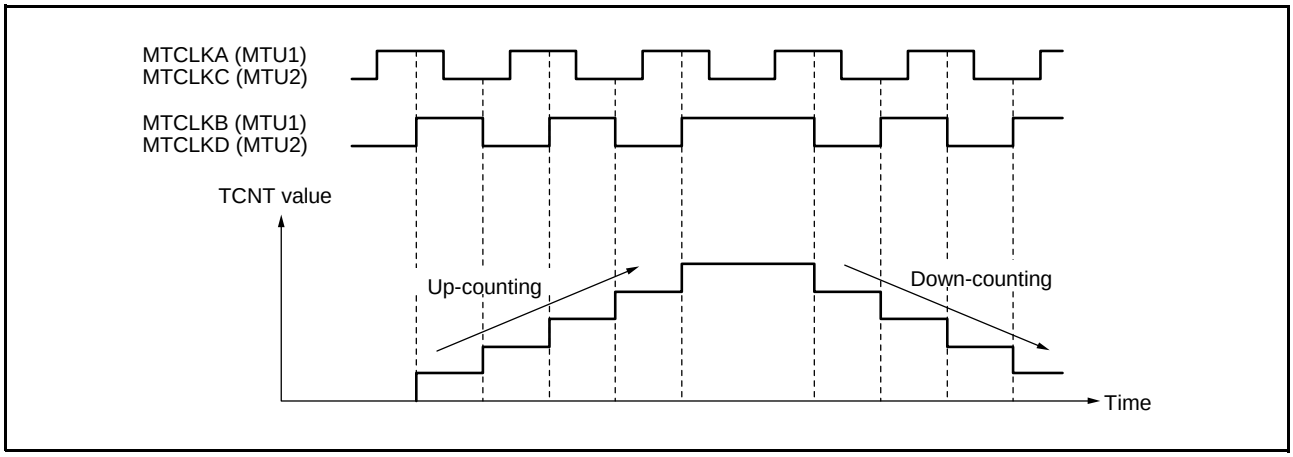


Figure 22.38 Example of Operation in Phase Counting Mode 4

Table 22.69 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 22.39 and Figure 22.40 show the examples of operation in phase counting mode 5 and Table 22.70 summarizes the TCNT up-counting and down-counting conditions.

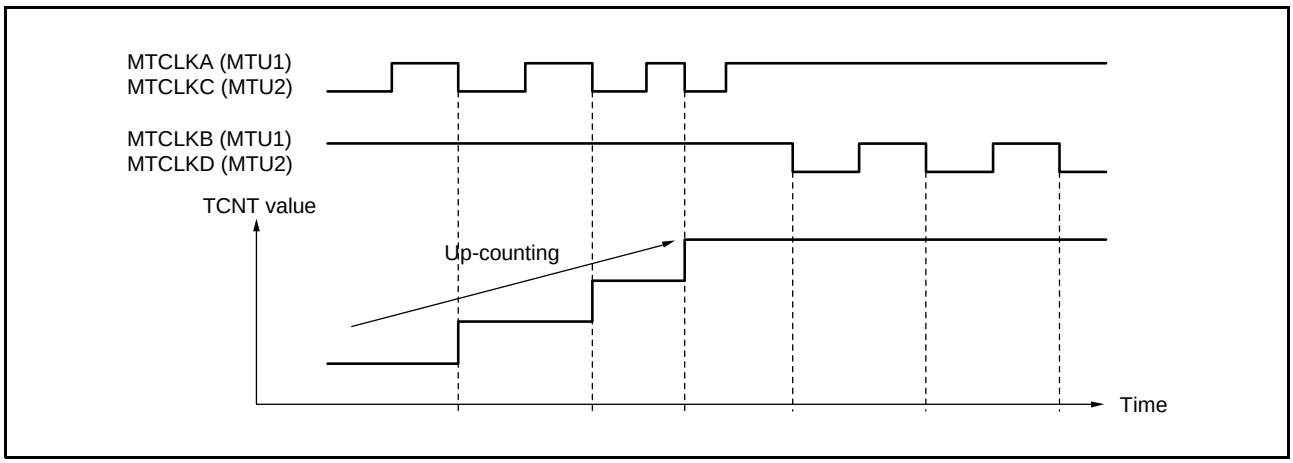


Figure 22.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

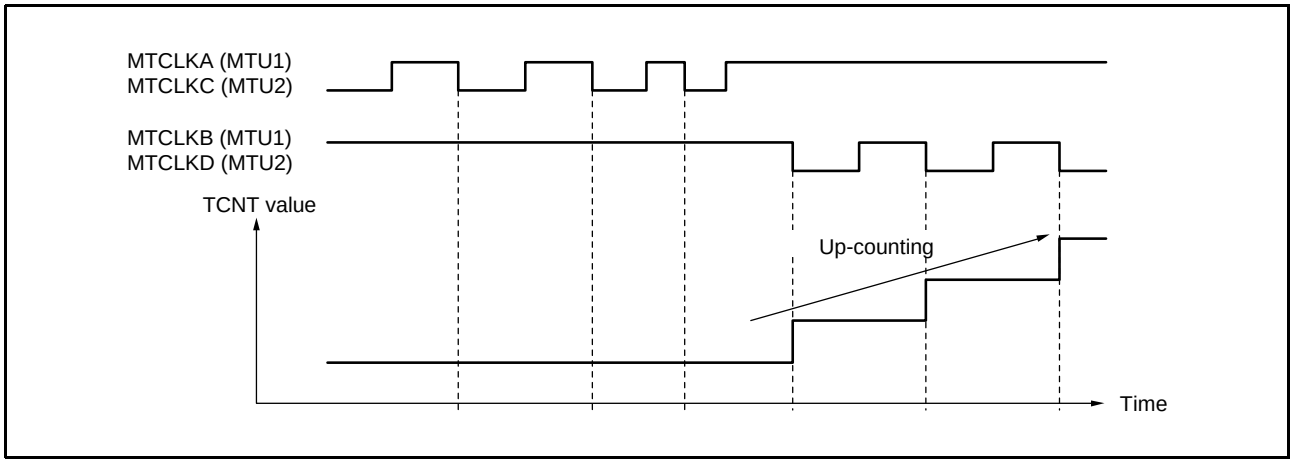


Figure 22.40 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 22.70 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 22.41 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.

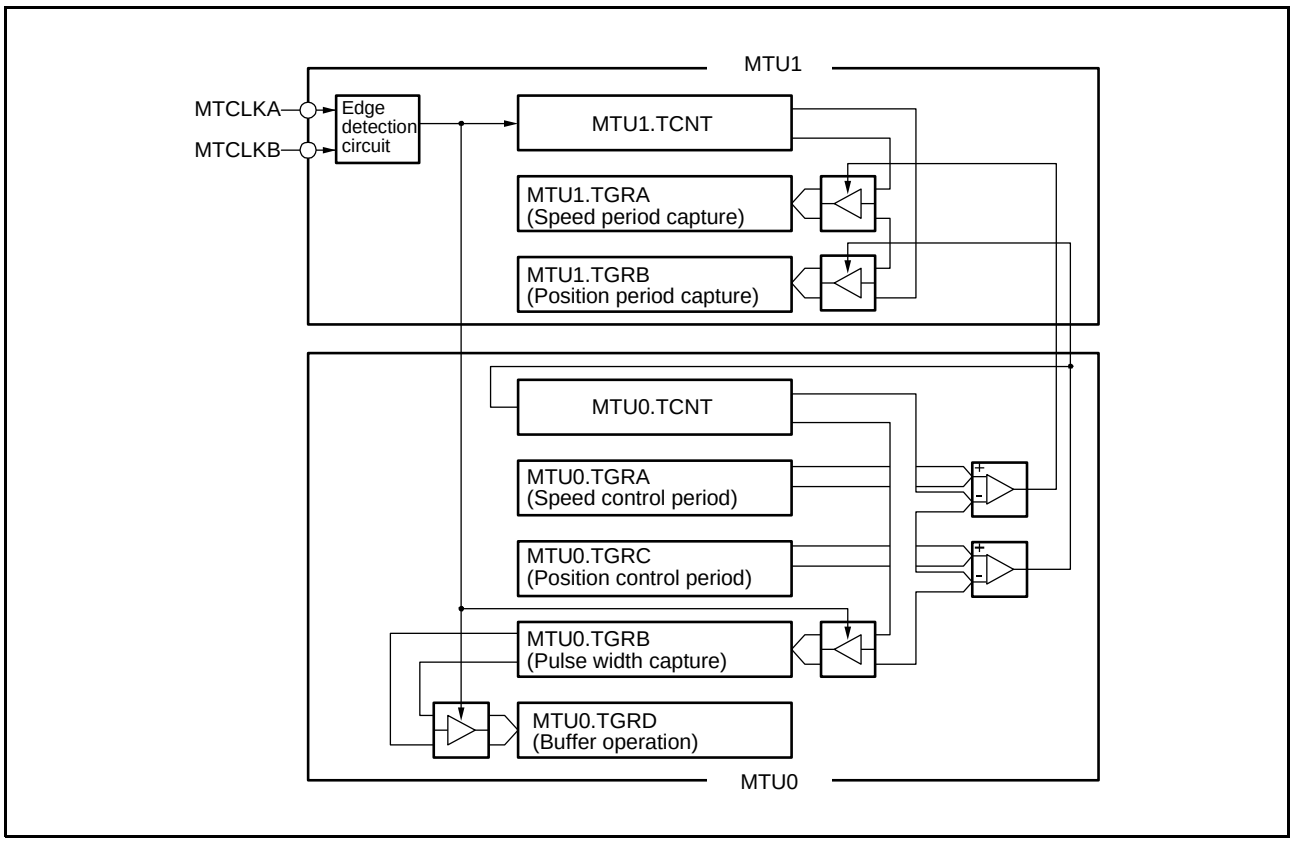


Figure 22.41 16-Bit Phase Counting Mode Application Example

22.3.6.2 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 22.42. 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 22.43 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 22.69 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 22.3.4, 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.

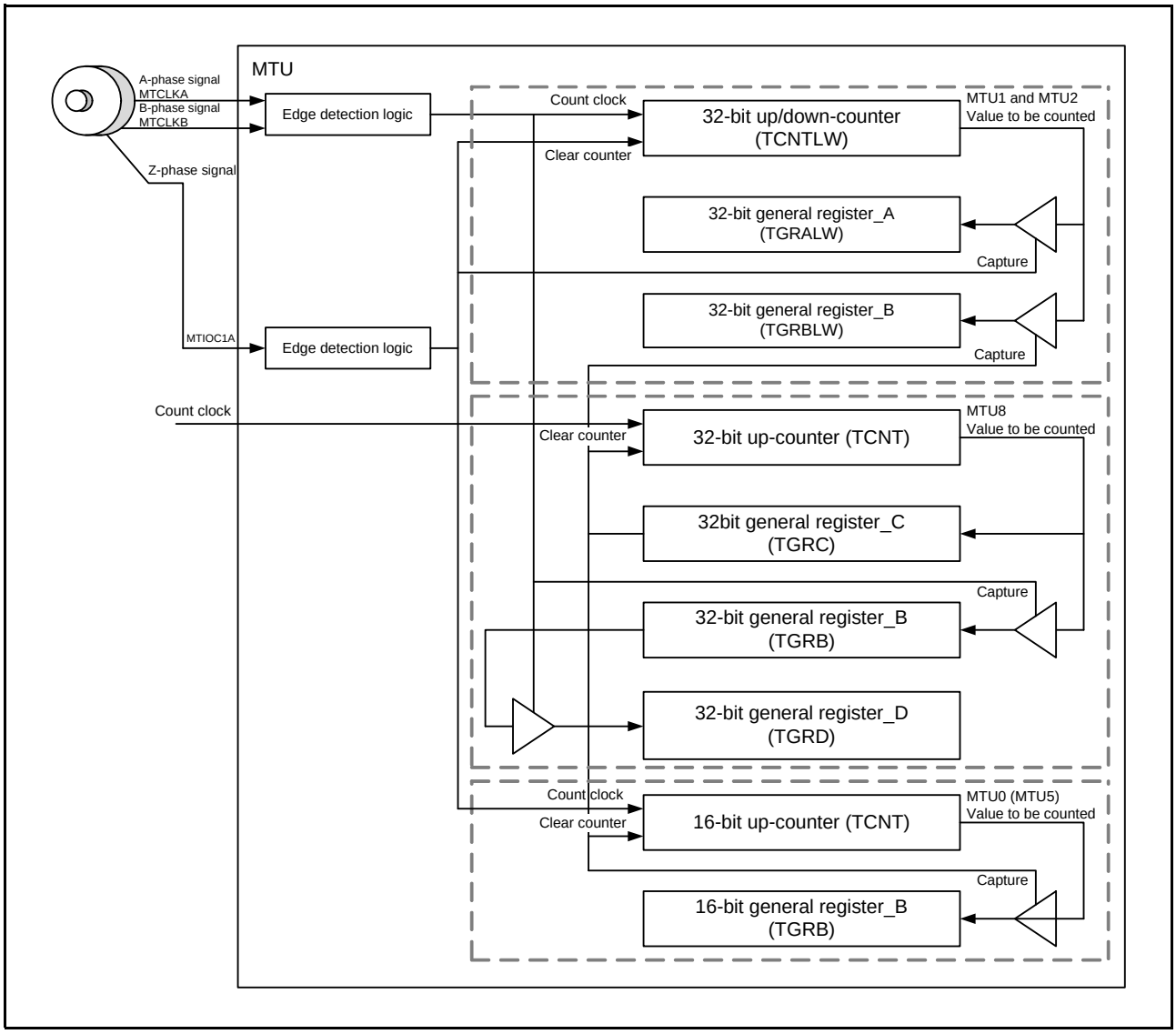


Figure 22.42 Block Diagram for Operation in Cascade Connection 32-Bit Phase Counting Mode

(1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 22.43 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.

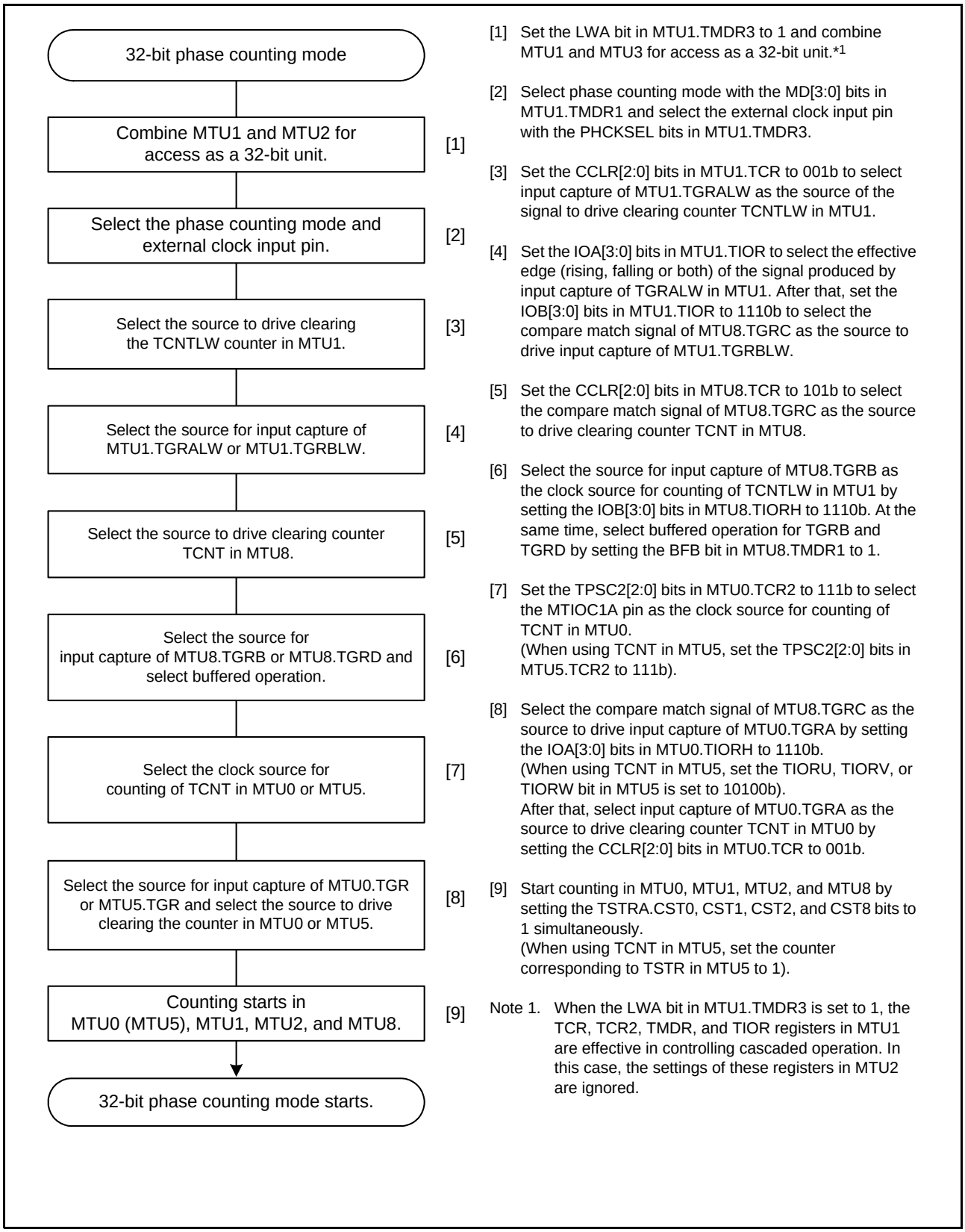


Figure 22.43 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

22.3.7 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 22.71 shows the PWM output pins used. Table 22.72 shows the settings of the registers.

Table 22.71 Output Pins for Reset-Synchronized PWM Mode

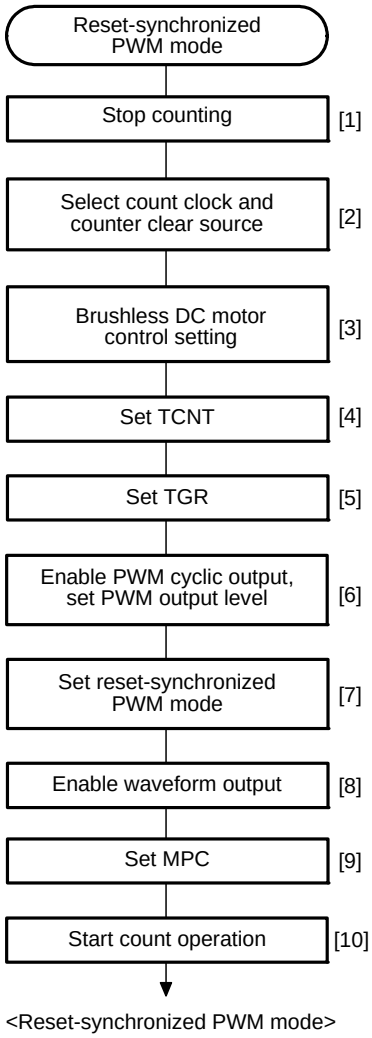
Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	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 22.72 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 22.44 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).

$X \leq$ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM 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 22.44 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 22.45 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.

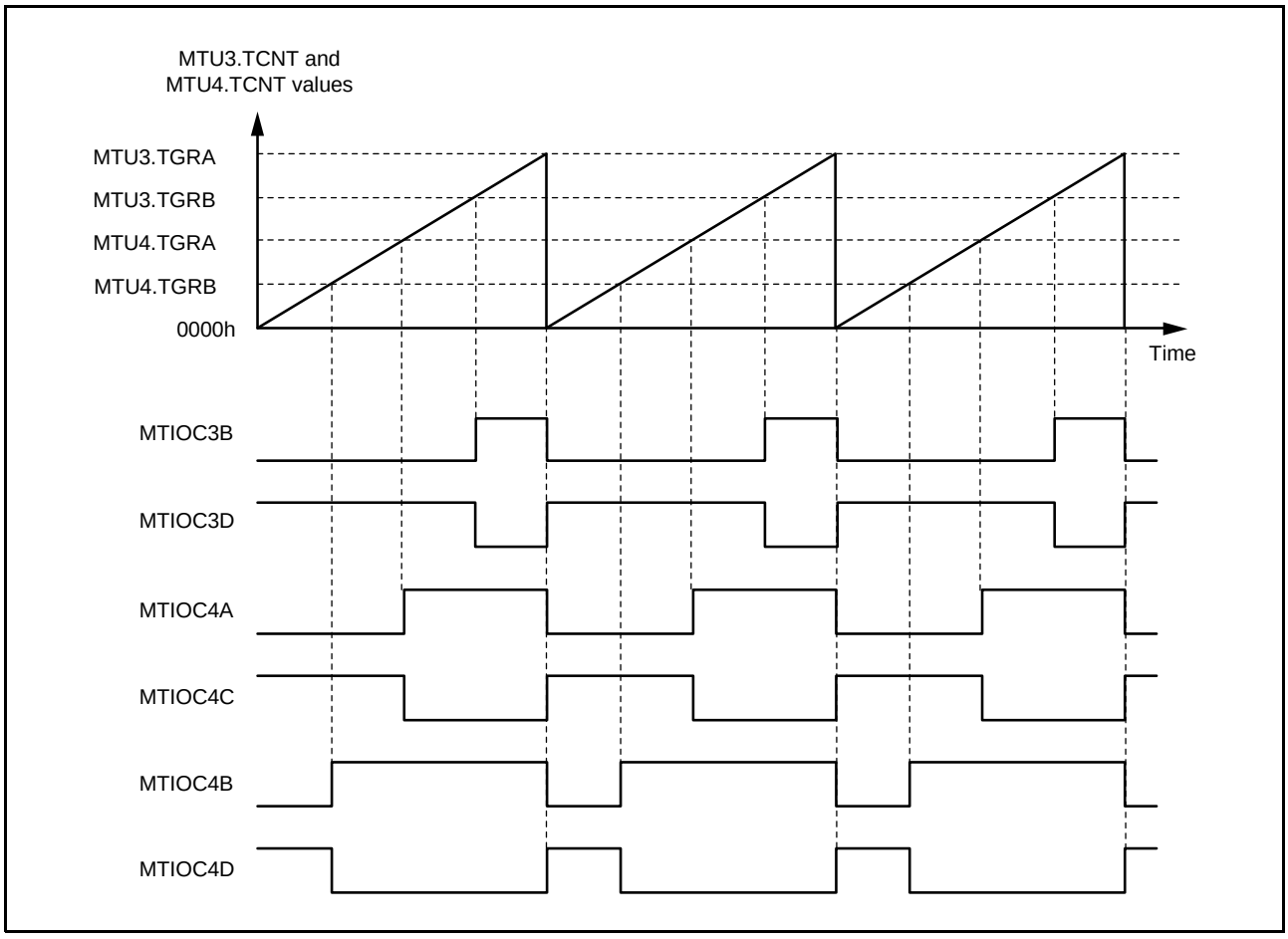


Figure 22.45 Example of Reset-Synchronized PWM Mode Operation (When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

22.3.8 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 22.73 shows the PWM output pins used. Table 22.74 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 22.73 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 22.74 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 22.75 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).

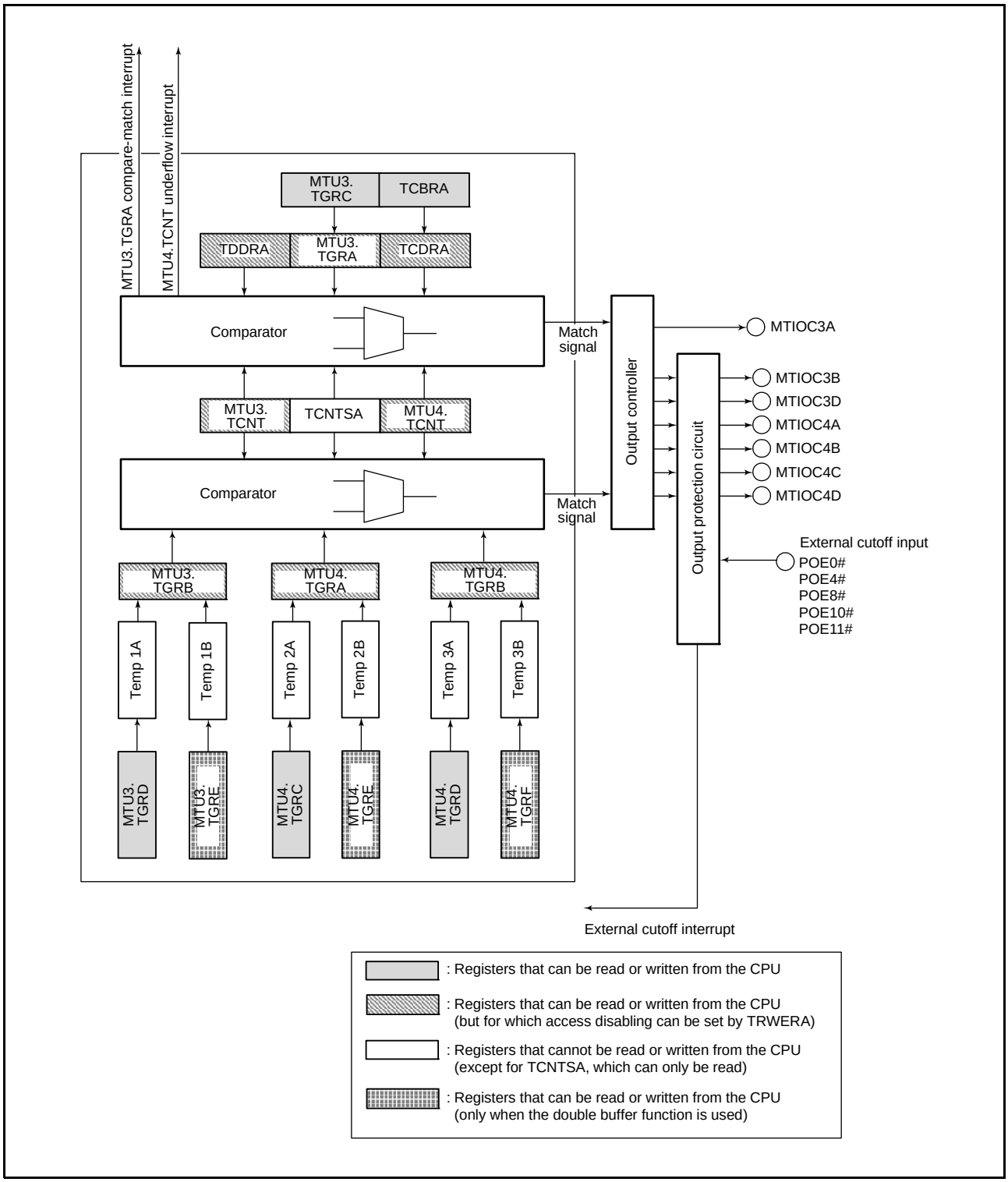


Figure 22.46 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

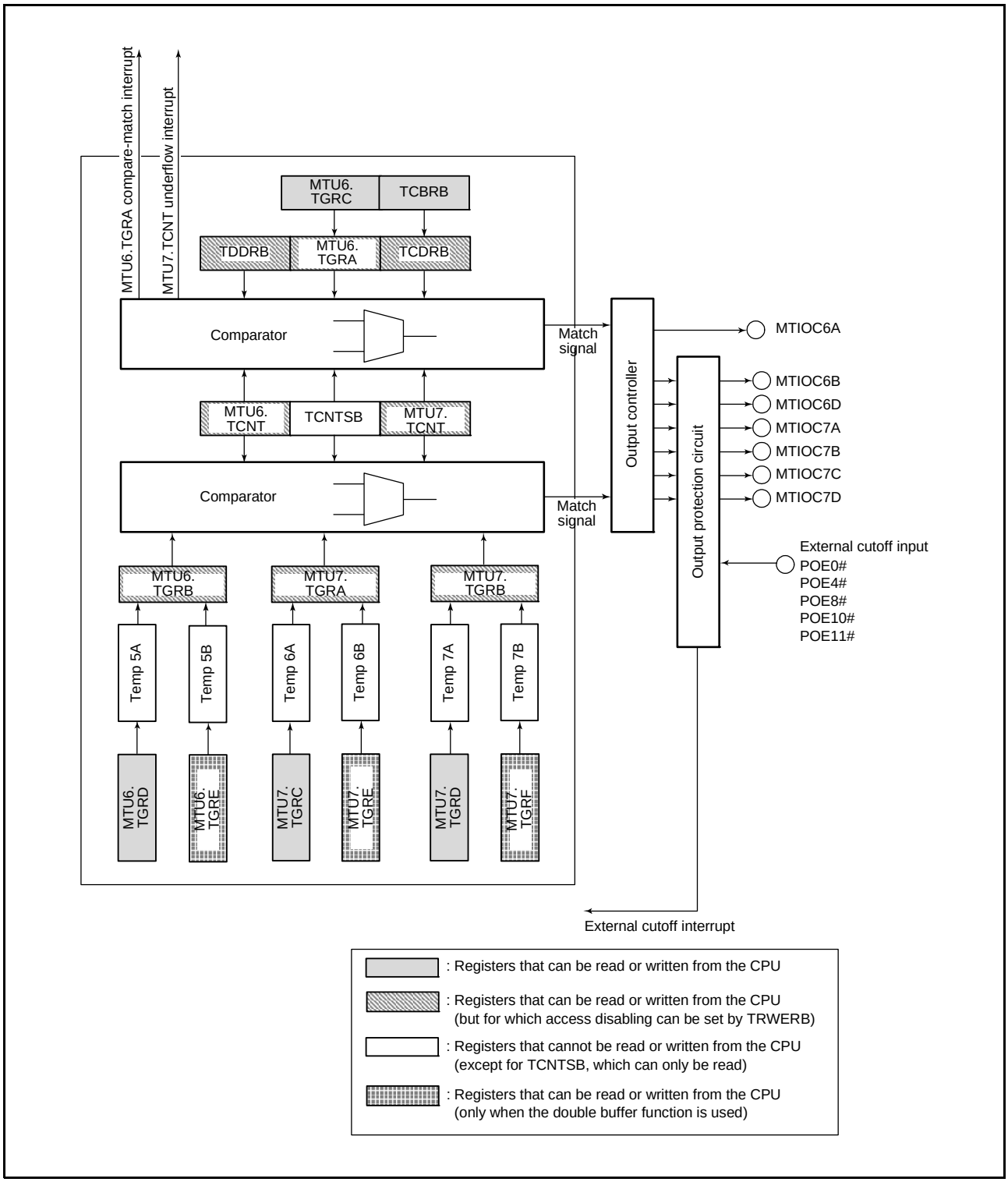
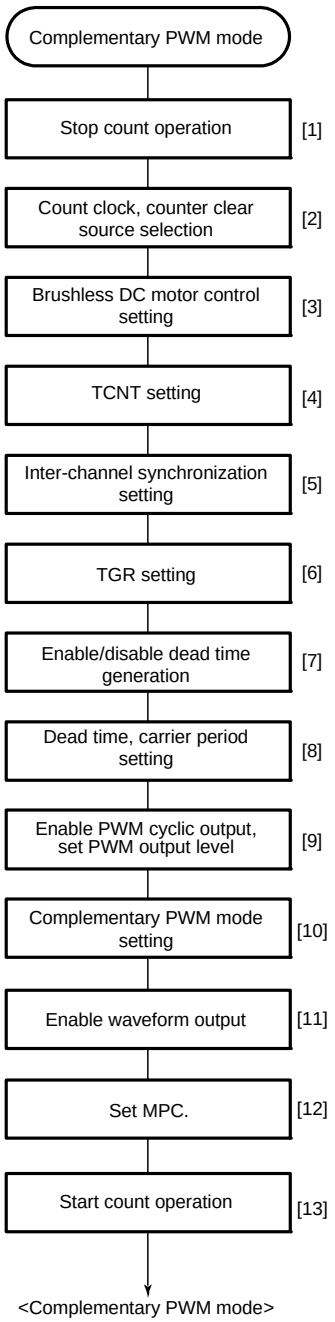


Figure 22.47 Block Diagram of MTU6 and MTU7 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 22.48 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 22.48 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 22.49 illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 22.50 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.

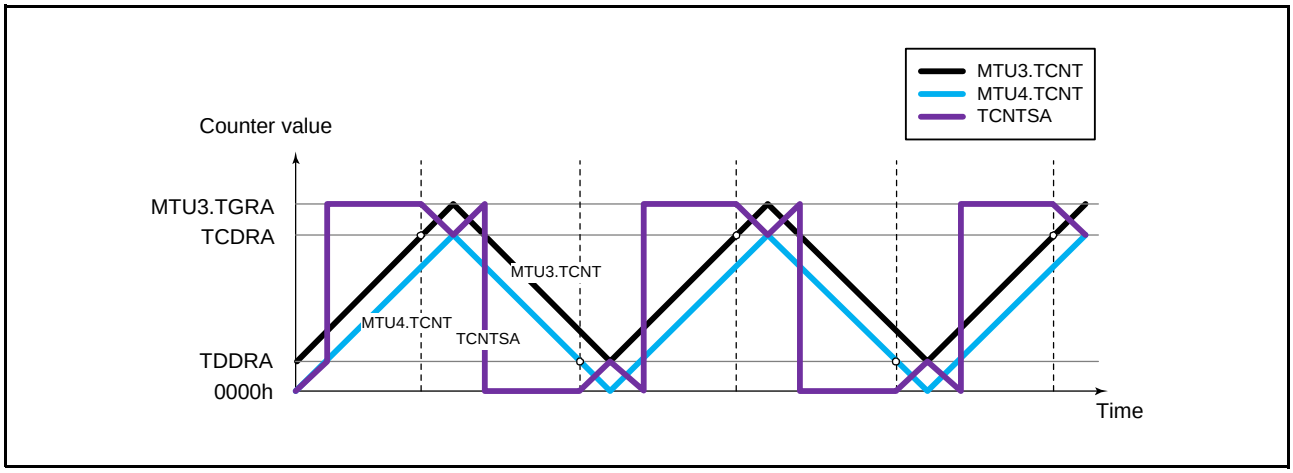


Figure 22.49 Count Operation in Complementary PWM Mode (MTU3 and MTU4)

(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 22.50 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 22.3.8 (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 22.50 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 22.50), 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.

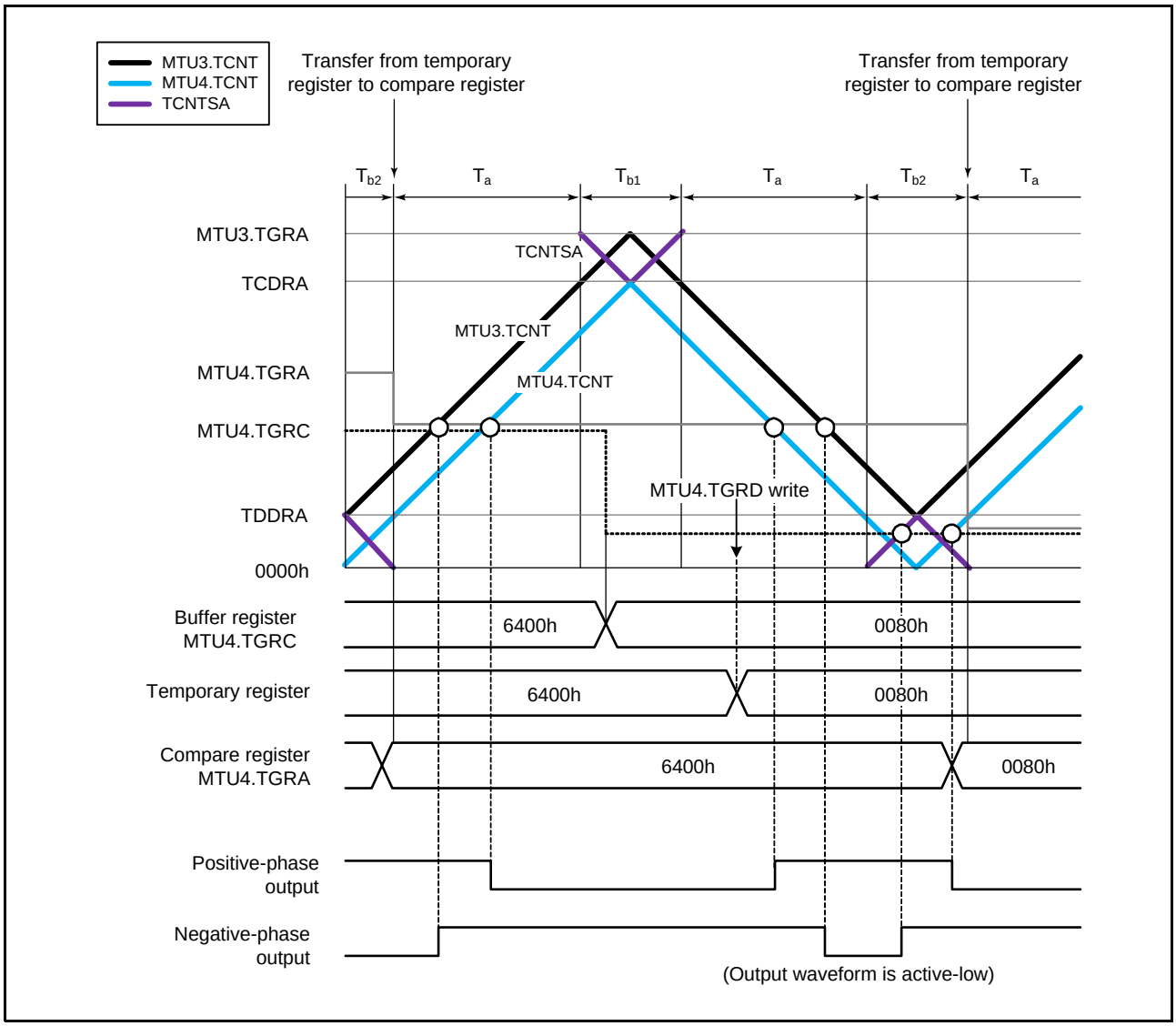


Figure 22.50 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

(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 22.76 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 22.51 shows an example of operation without dead time (MTU3 and MTU4).

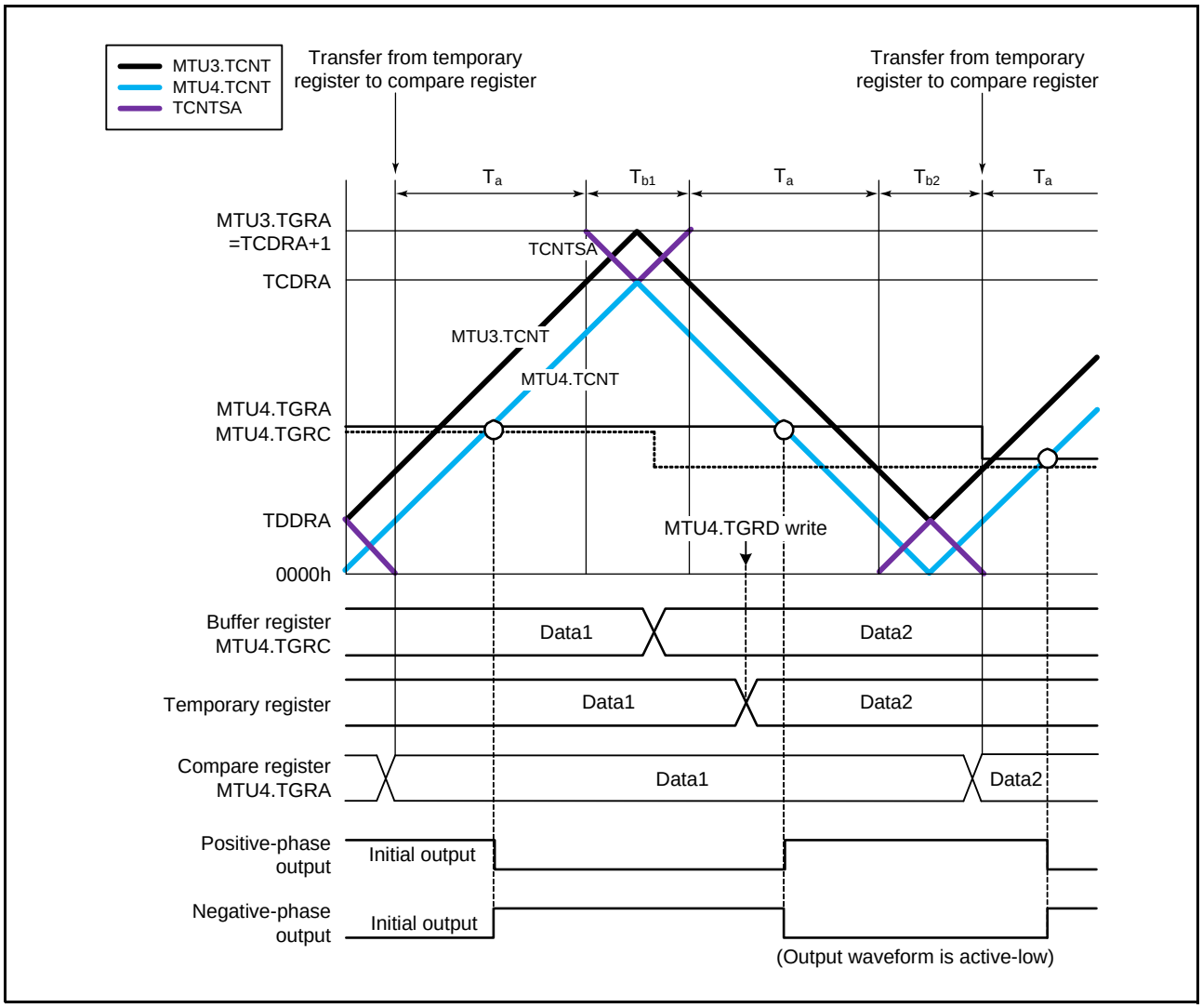


Figure 22.51 Example of Operation without Dead Time (MTU3 and MTU4)

(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 22.52 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.

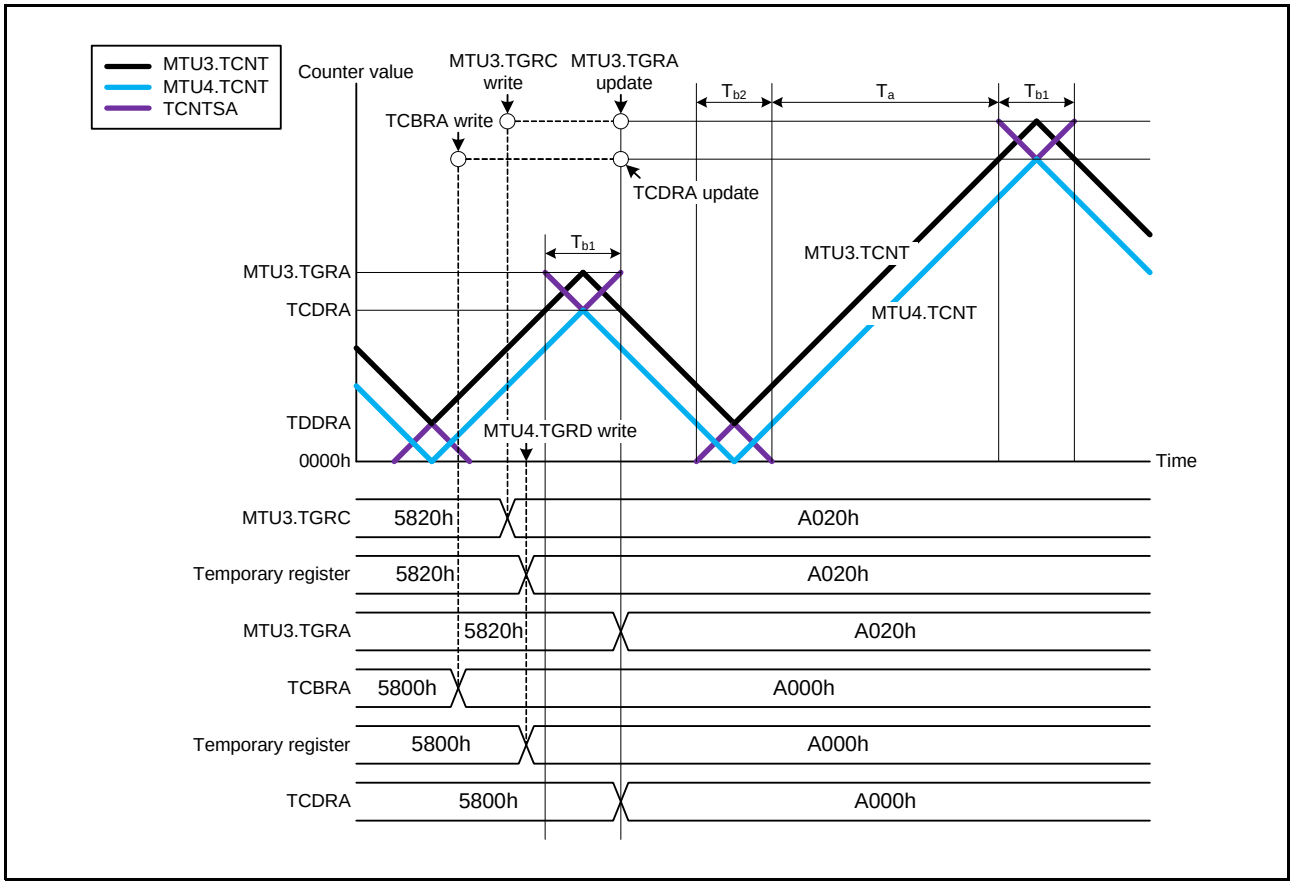


Figure 22.52 Example of PWM Period Updating (MTU3 and MTU4)

(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 22.53 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 22.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

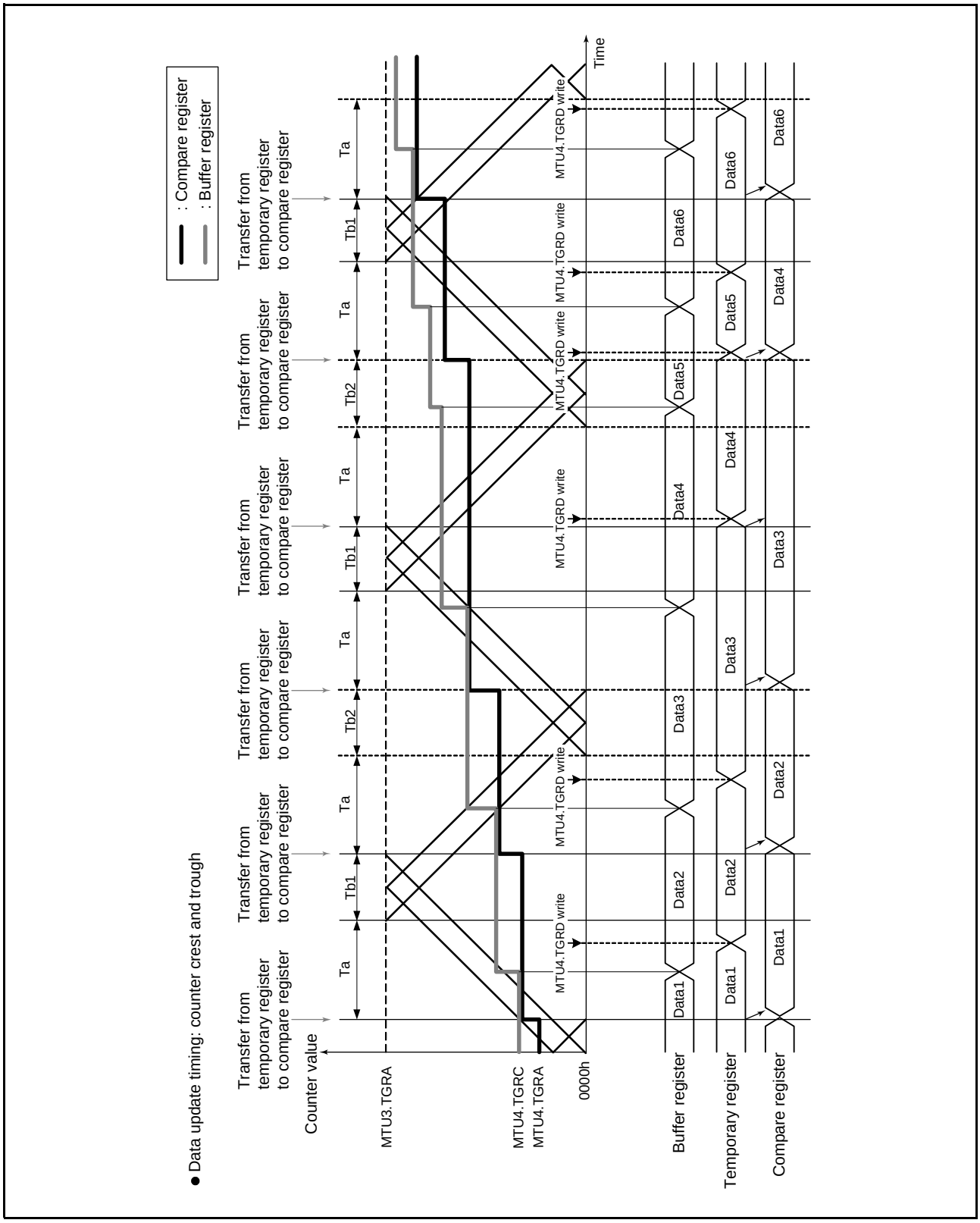


Figure 22.53 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

(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 22.54 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 22.55.

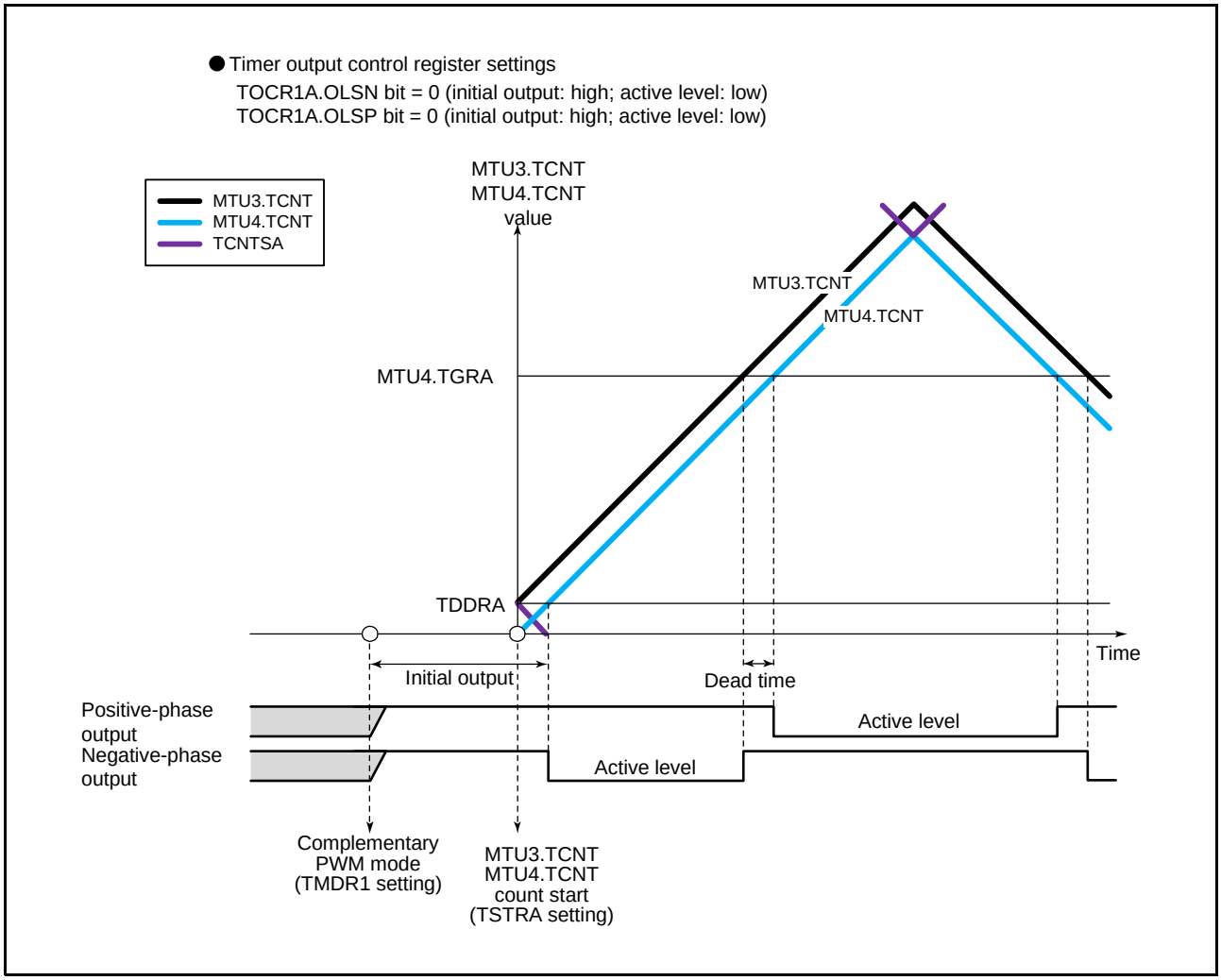


Figure 22.54 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1))

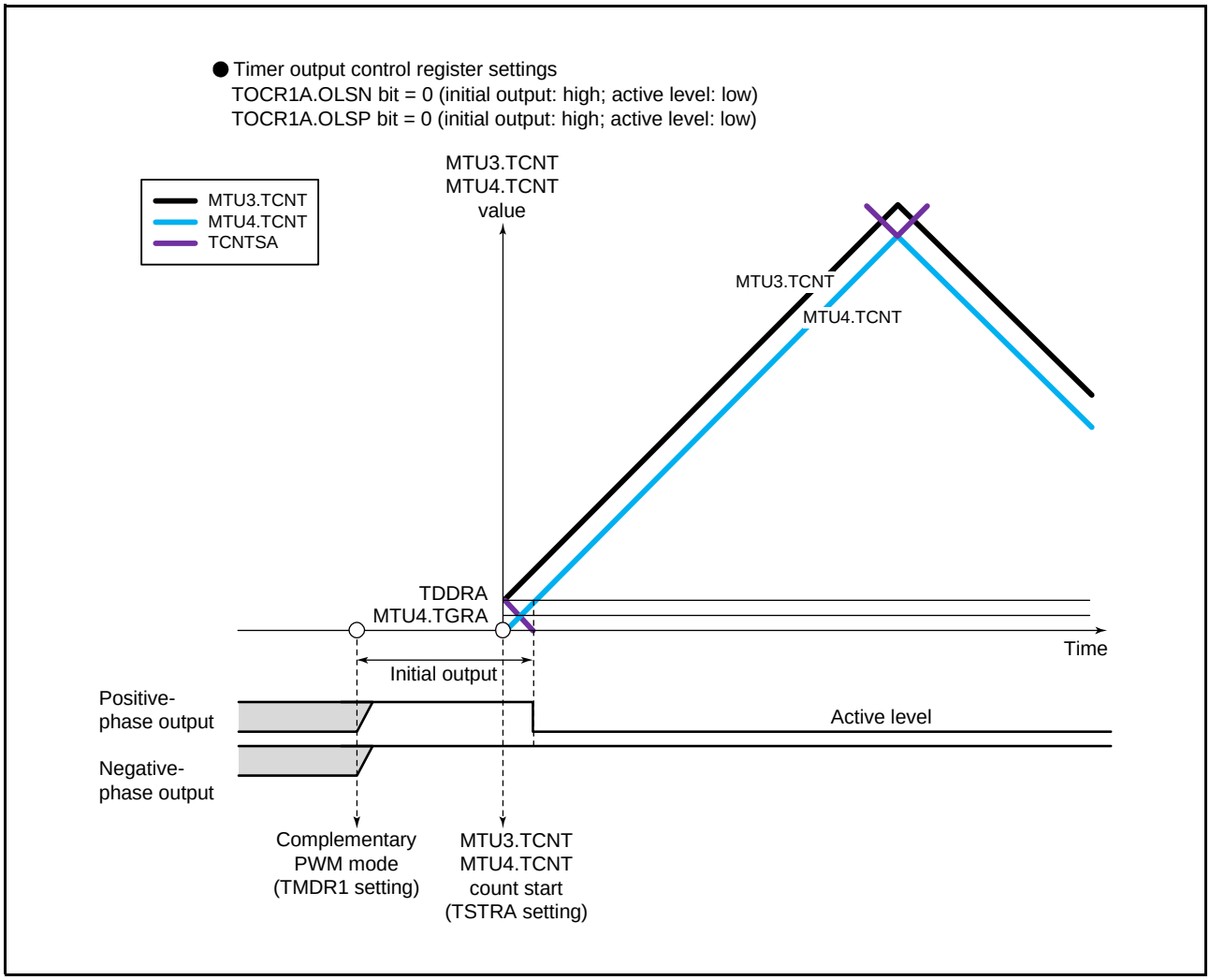


Figure 22.55 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

(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 22.56 to Figure 22.58 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 22.56.

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 22.57, 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 22.58, 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.

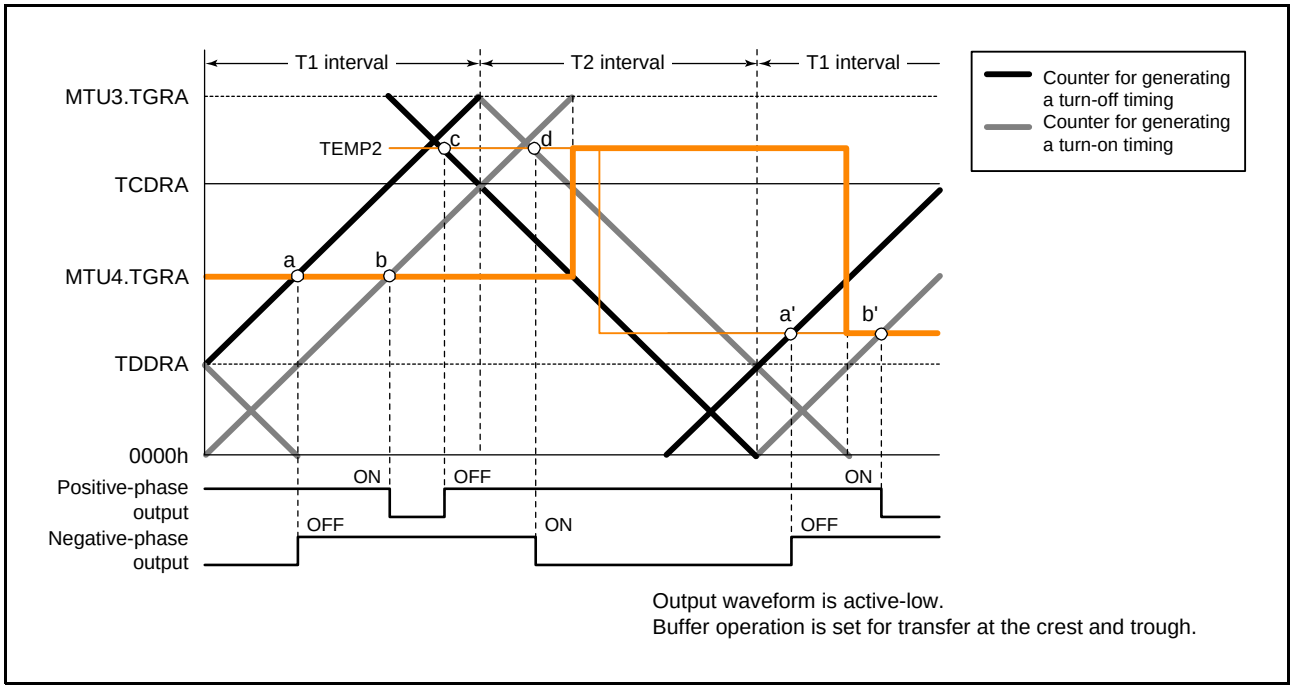


Figure 22.56 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

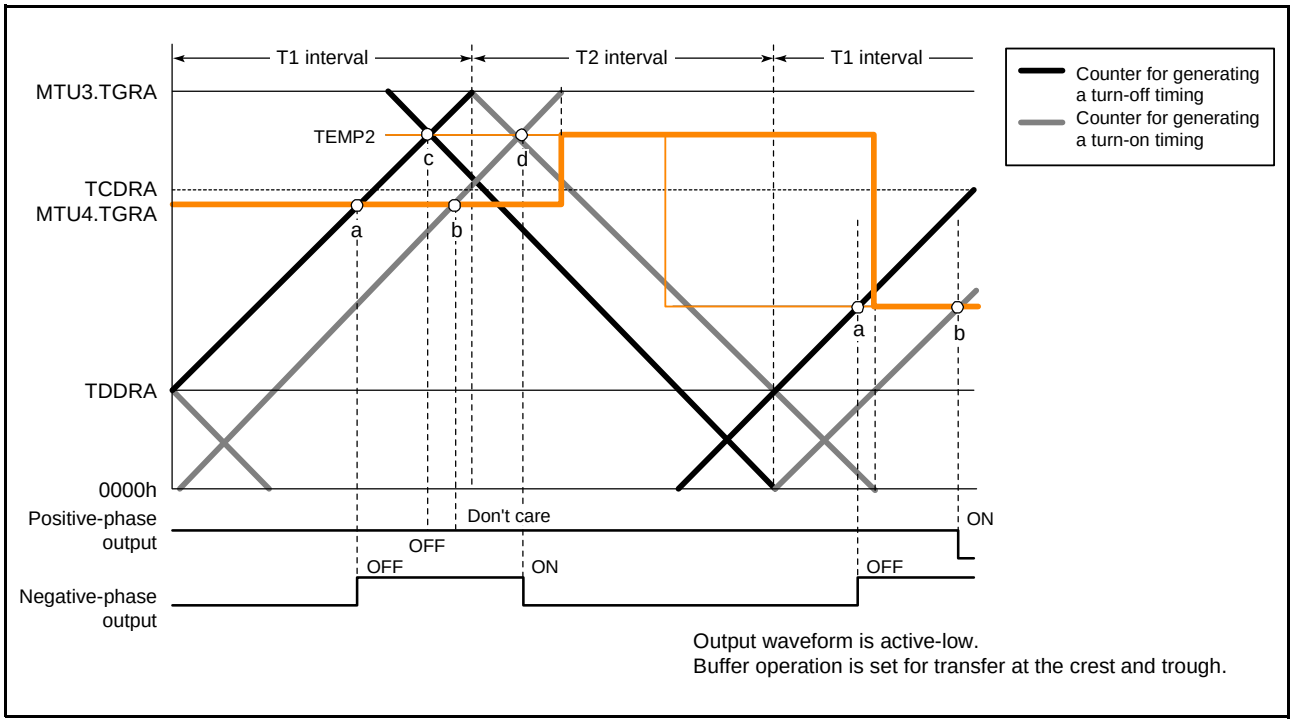


Figure 22.57 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

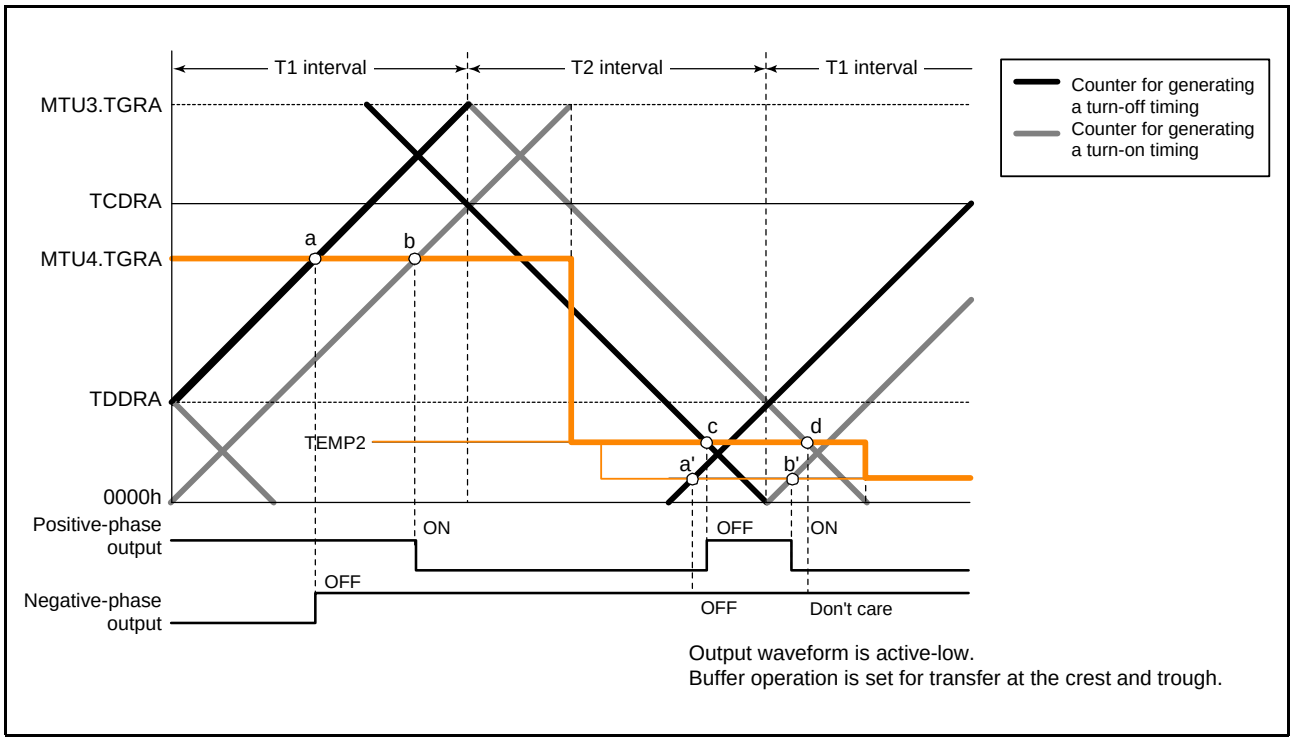


Figure 22.58 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

(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 22.59 to Figure 22.63 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.

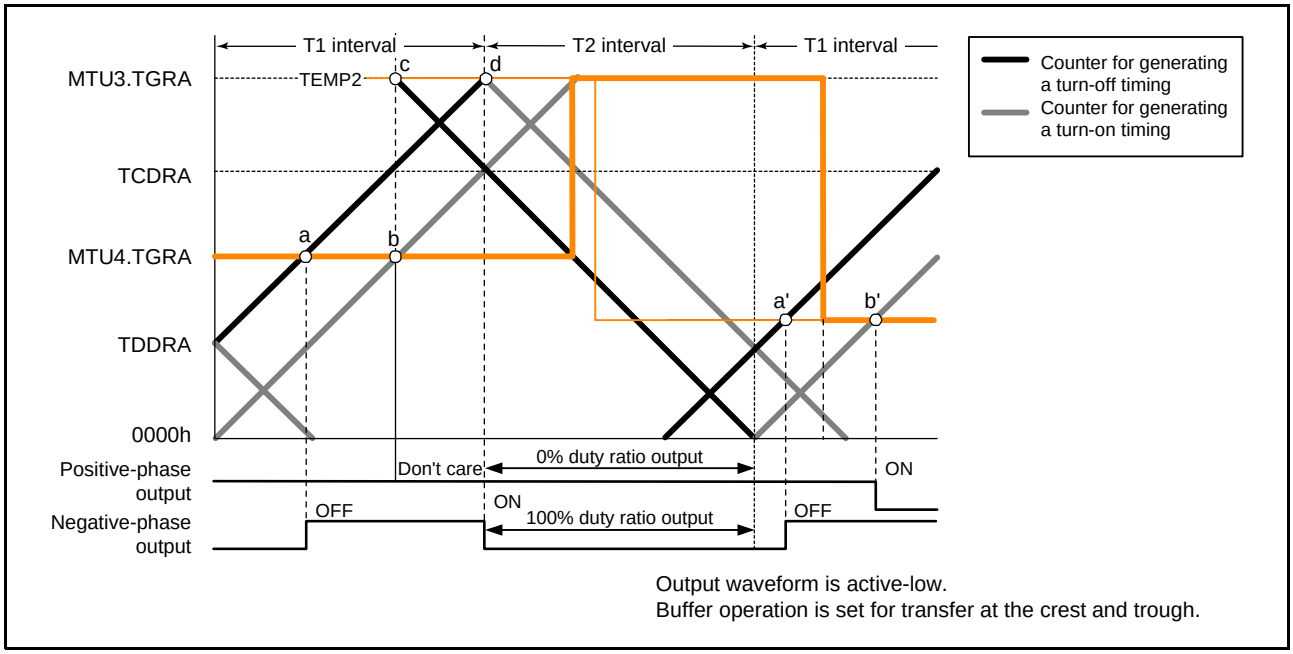


Figure 22.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

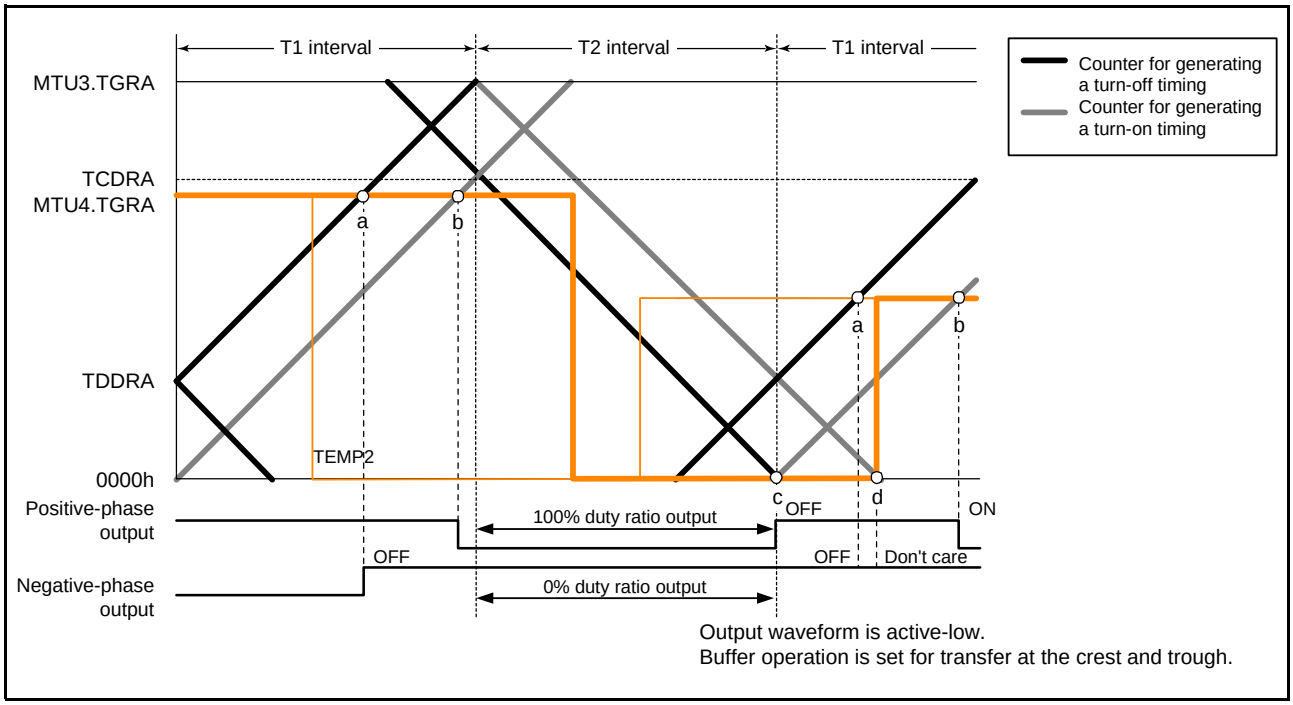


Figure 22.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

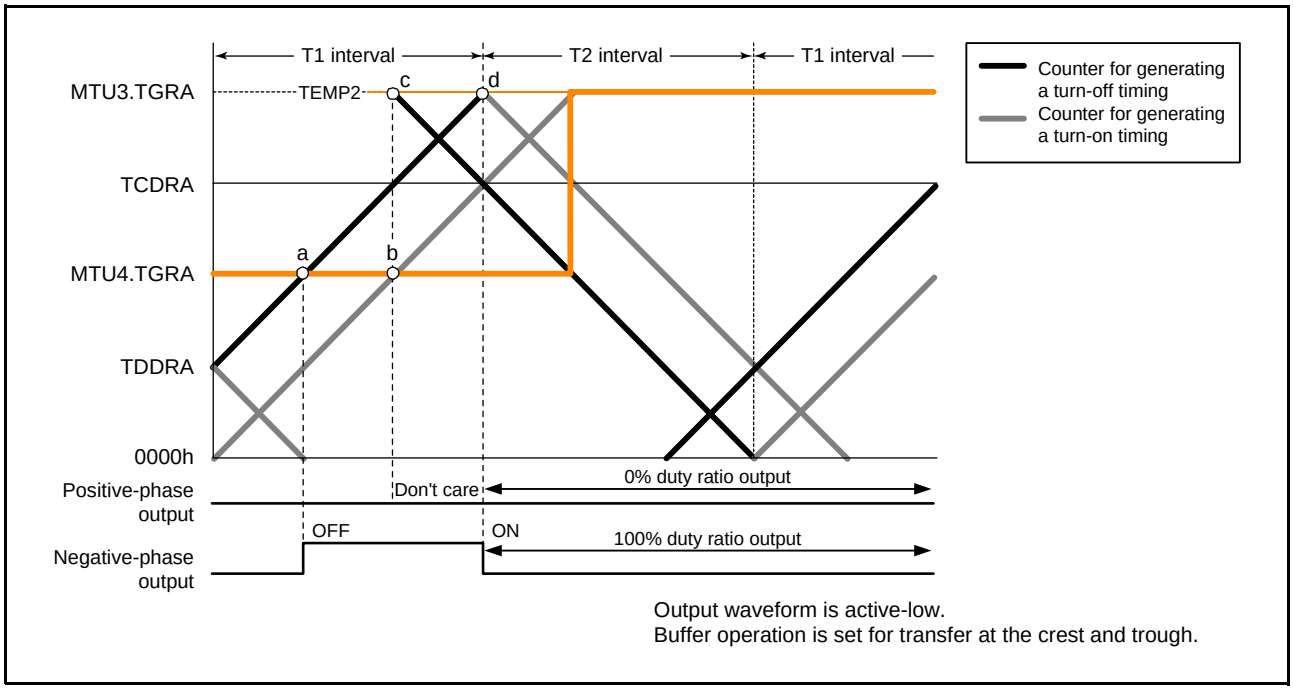


Figure 22.61 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

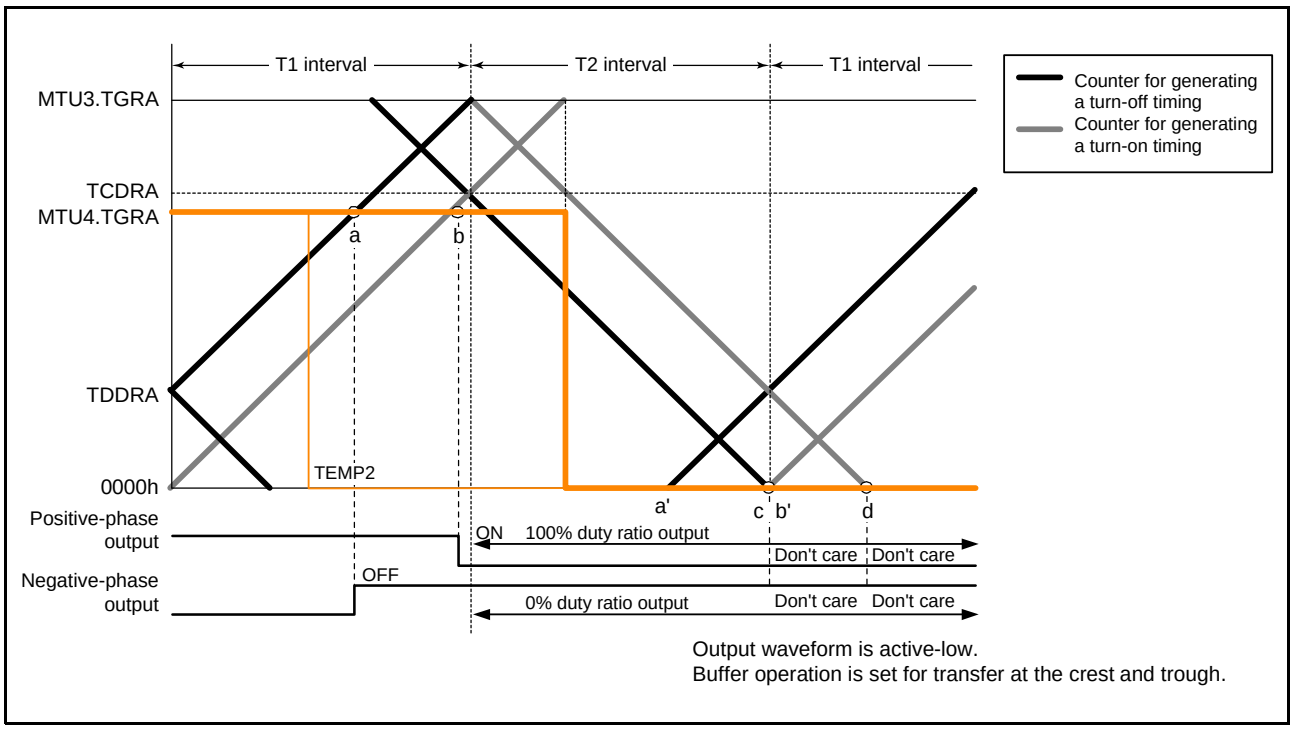


Figure 22.62 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

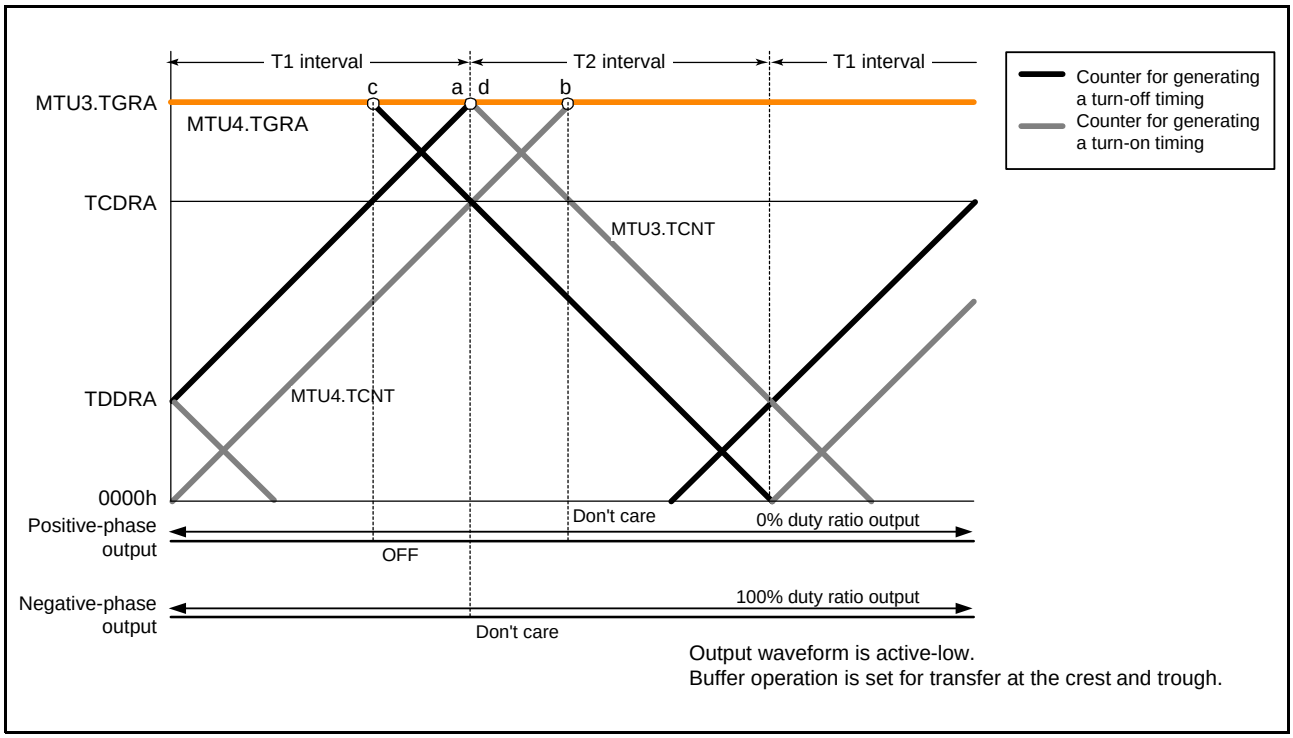


Figure 22.63 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

(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 22.64.

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.

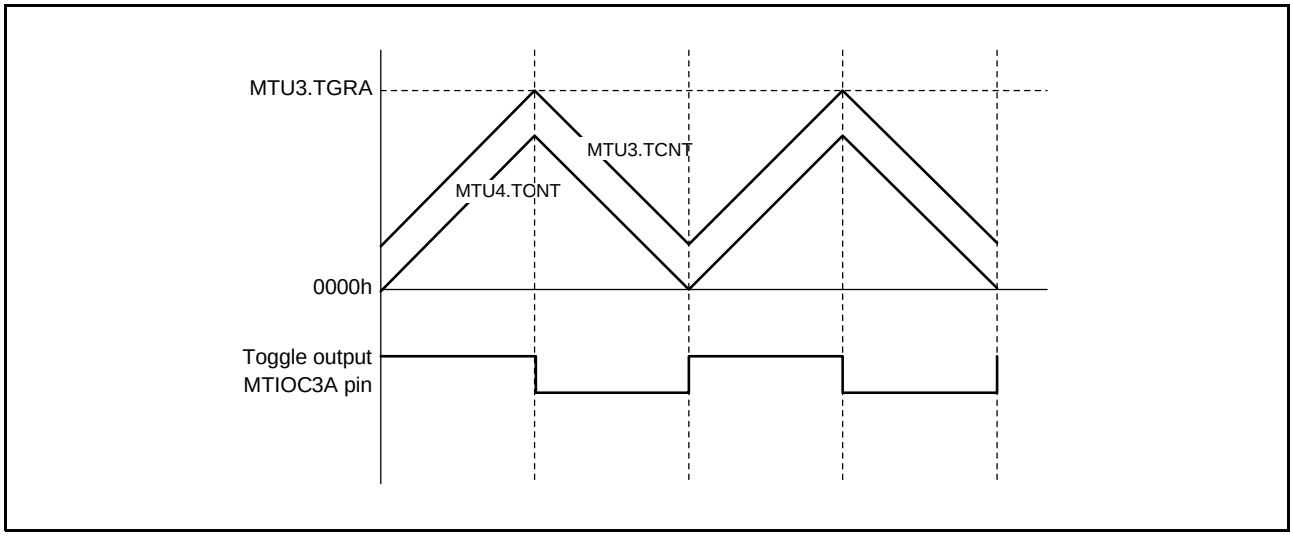


Figure 22.64 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

(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 22.65 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

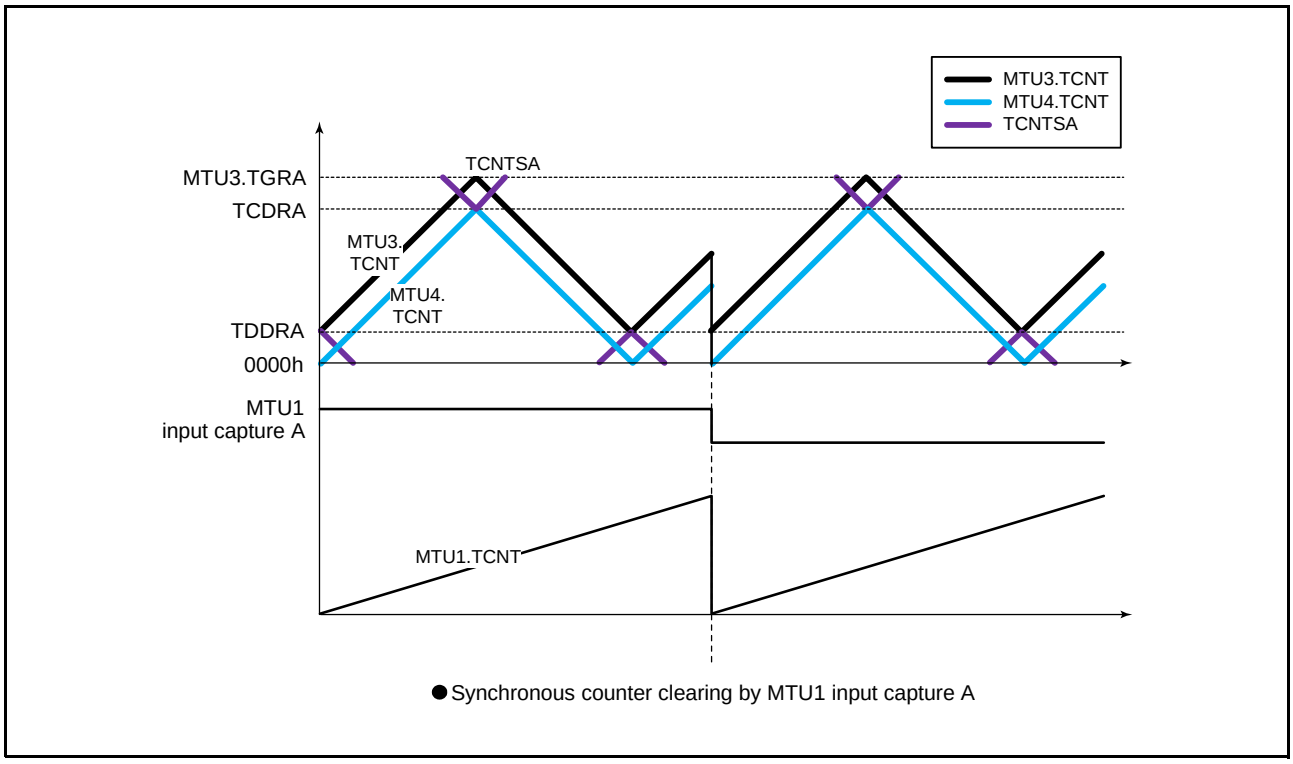


Figure 22.65 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

(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 22.66. 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 22.66) 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.

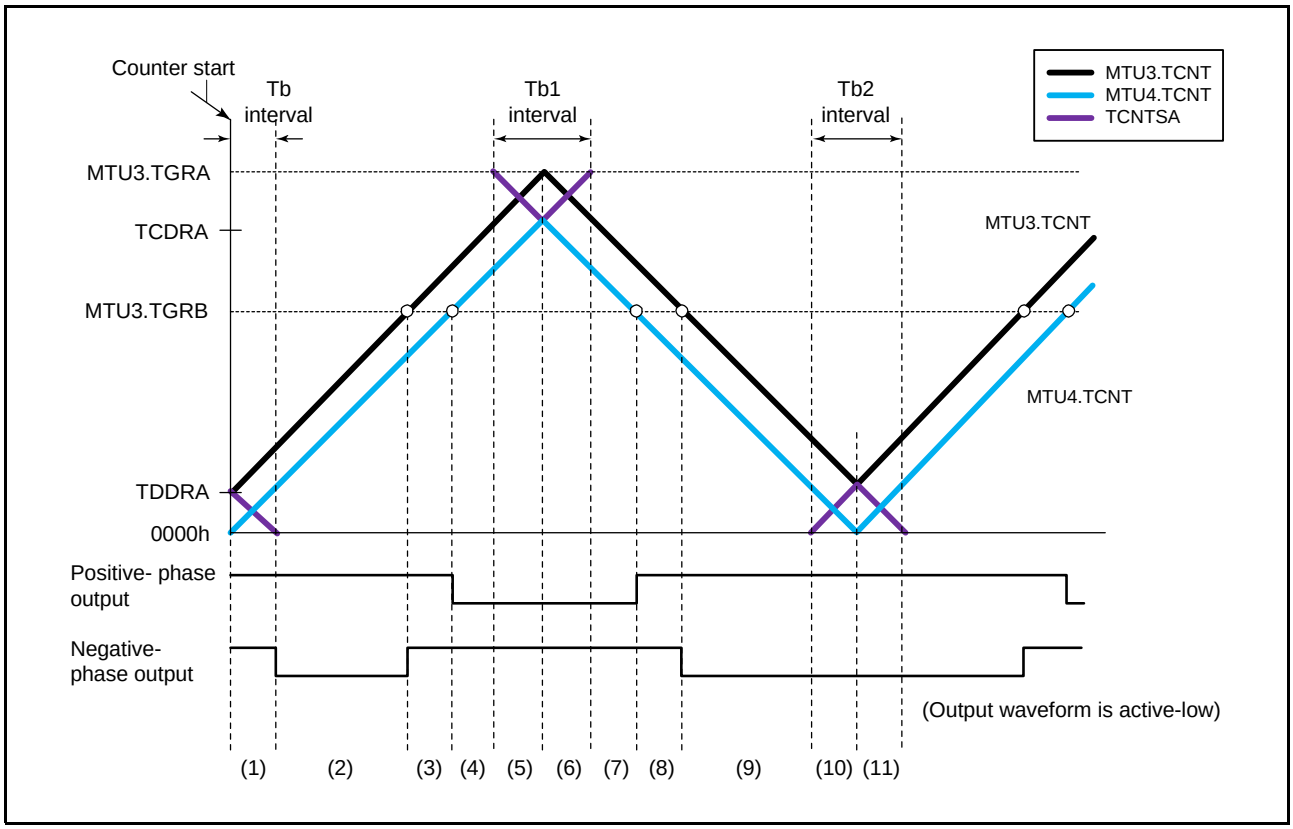


Figure 22.66 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- 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 22.67.

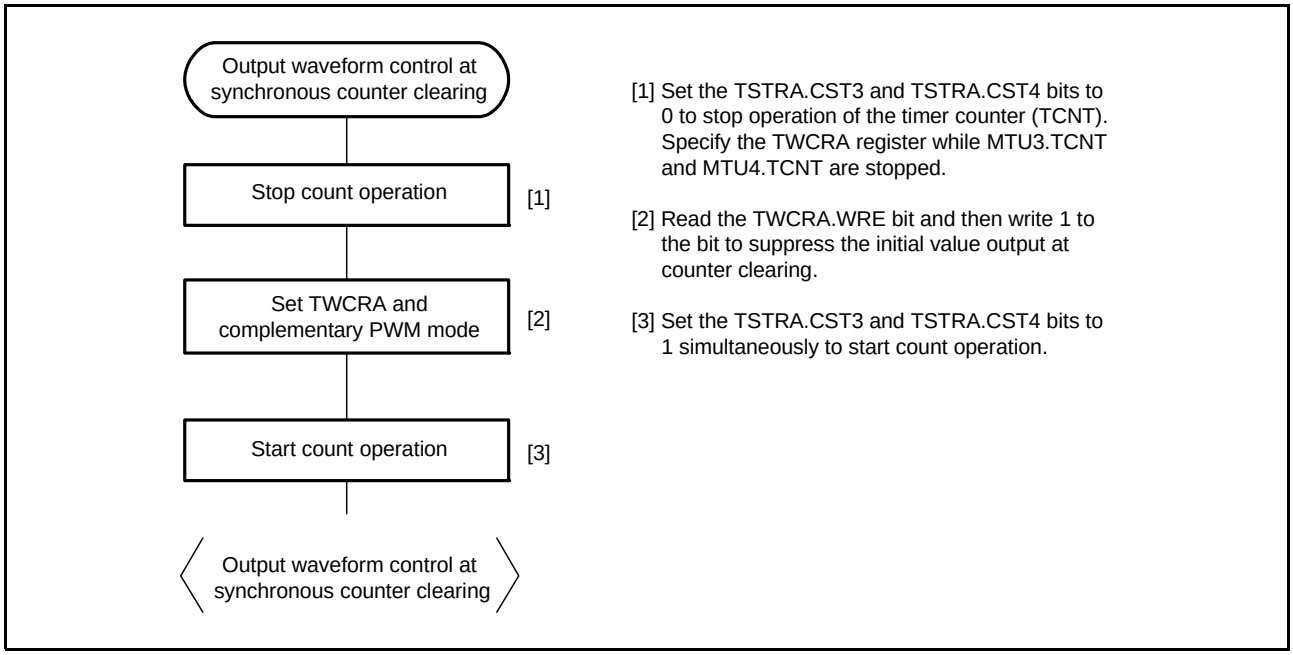


Figure 22.67 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 22.68 to Figure 22.71 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 22.68 to Figure 22.71, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.66, 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.

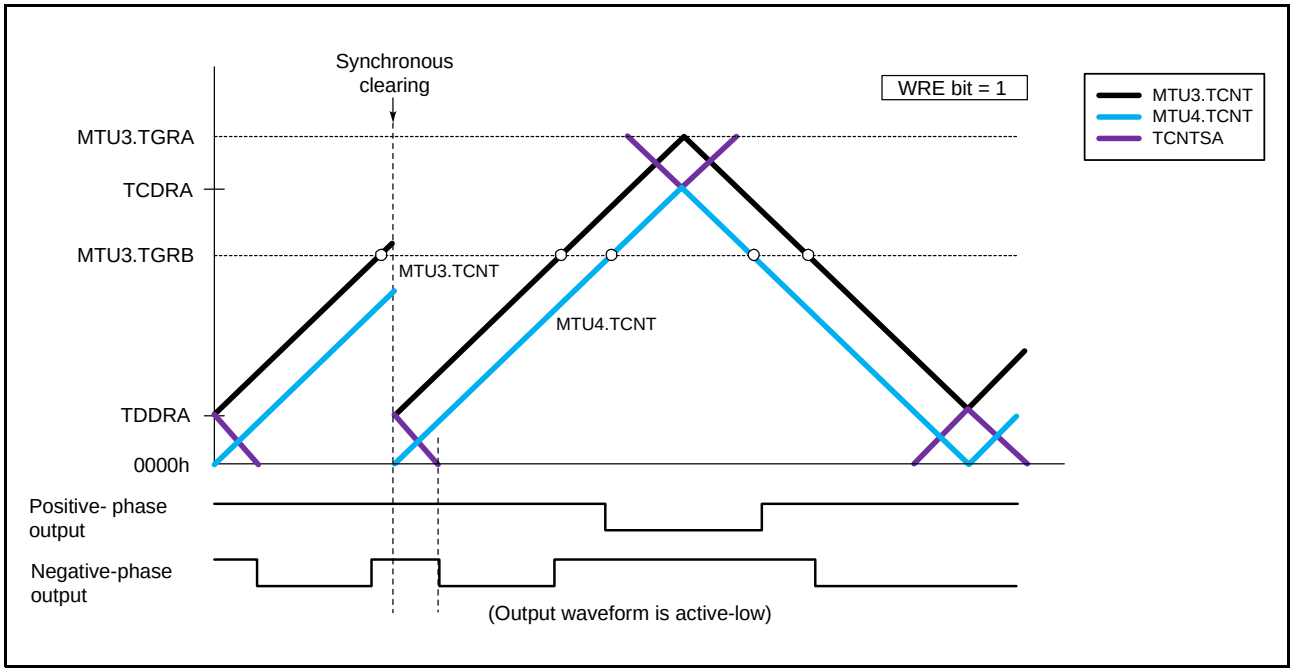


Figure 22.68 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.66; TWCRA.WRE Bit is 1)

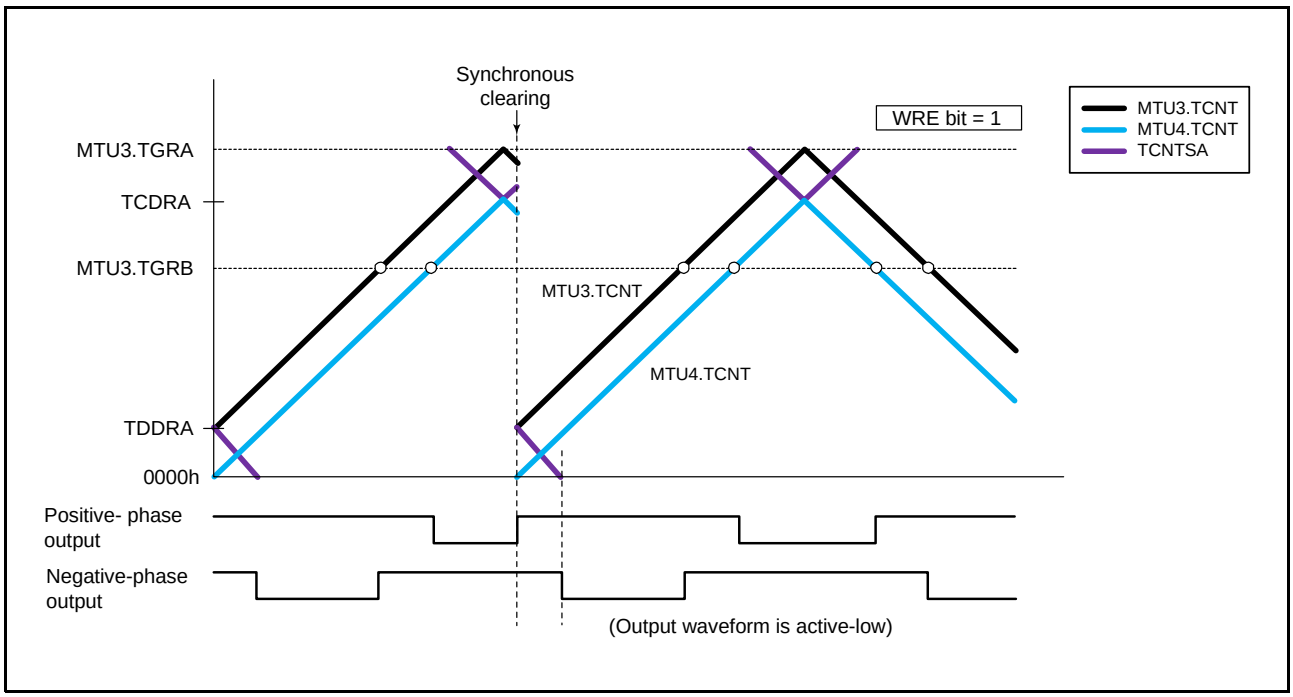


Figure 22.69 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 22.66; TWCRA.WRE Bit is 1)

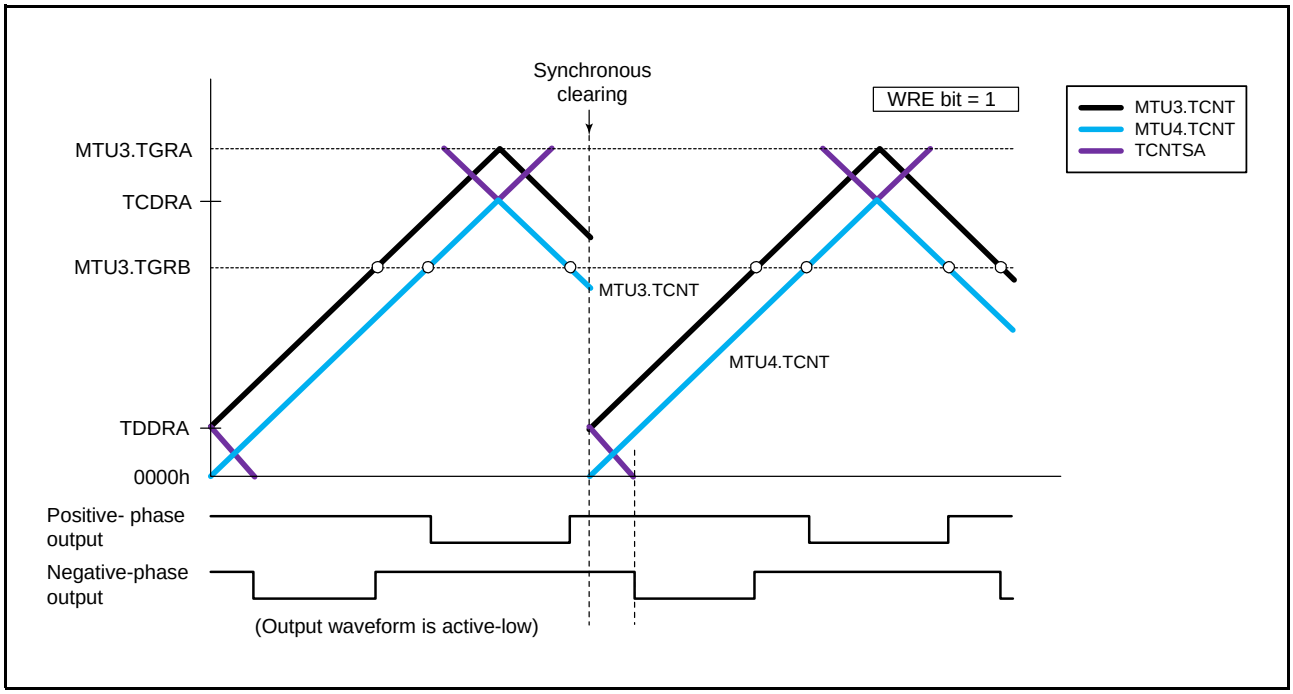


Figure 22.70 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.66; TWCRA.WRE Bit is 1)

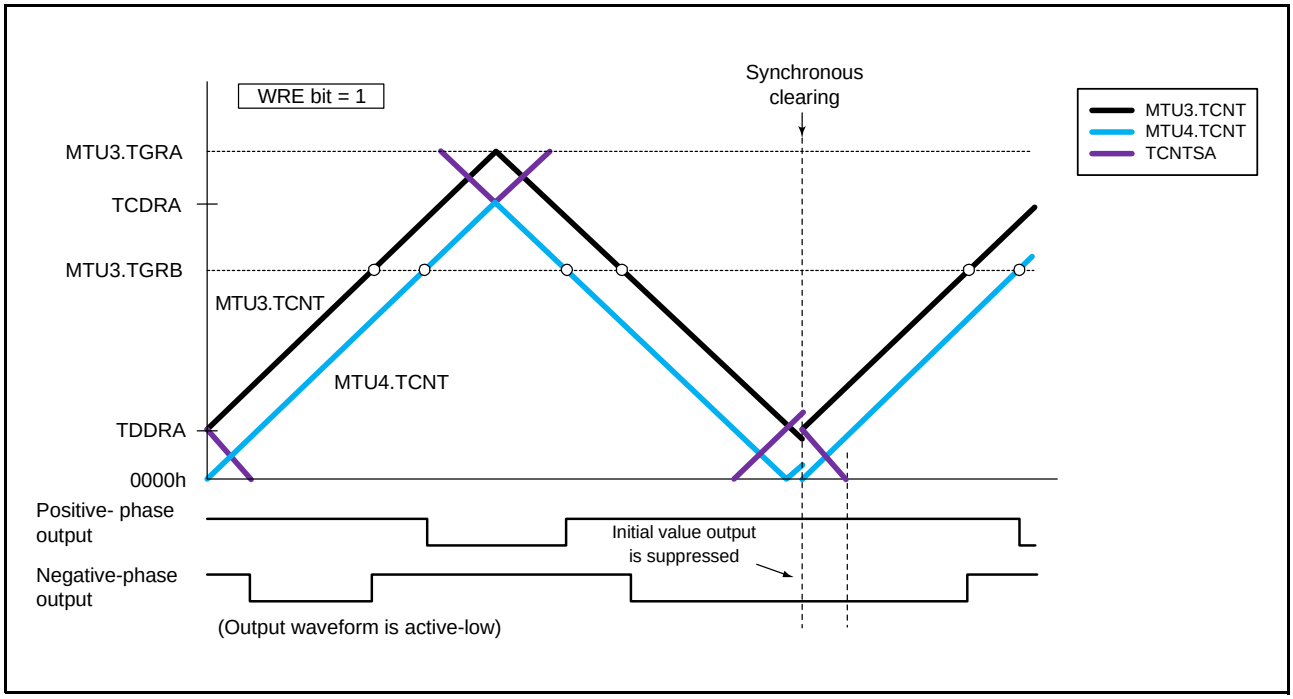


Figure 22.71 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 22.66; TWCRA.WRE Bit is 1)

(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 22.72. 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 22.3.10 (2), Synchronous Counter Clearing for MTU6 and MTU7.

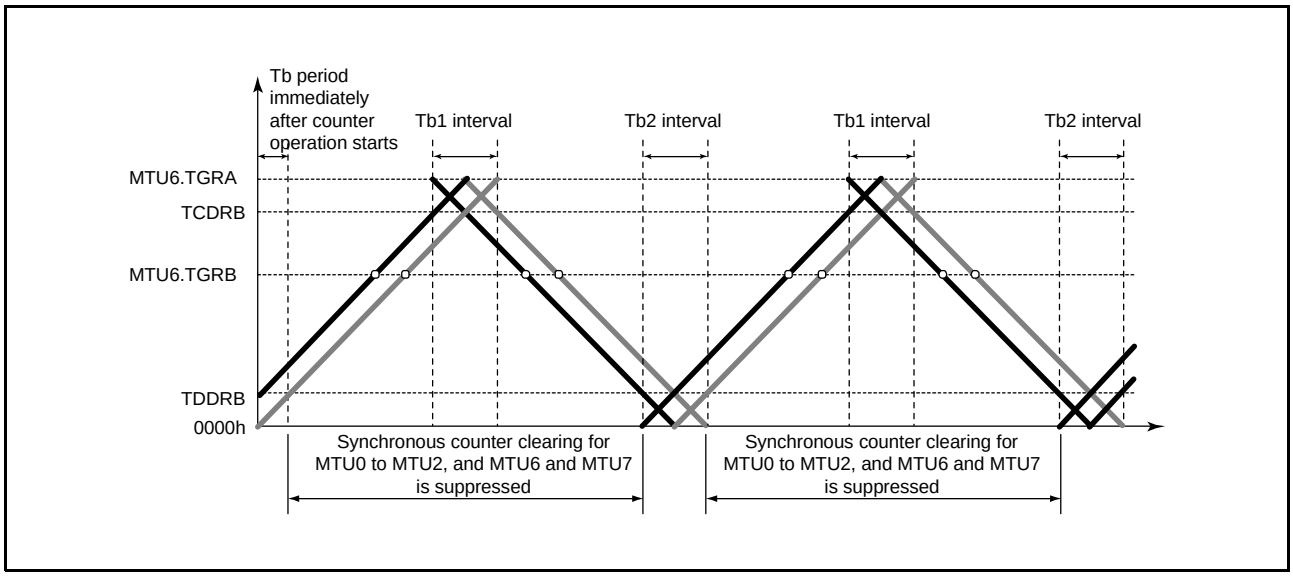


Figure 22.72 Synchronous Clearing-Suppressed Interval Specified by TWCRB.SCC Bit for MTU0 to MTU2, and MTU6 and MTU7

- 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 22.73.

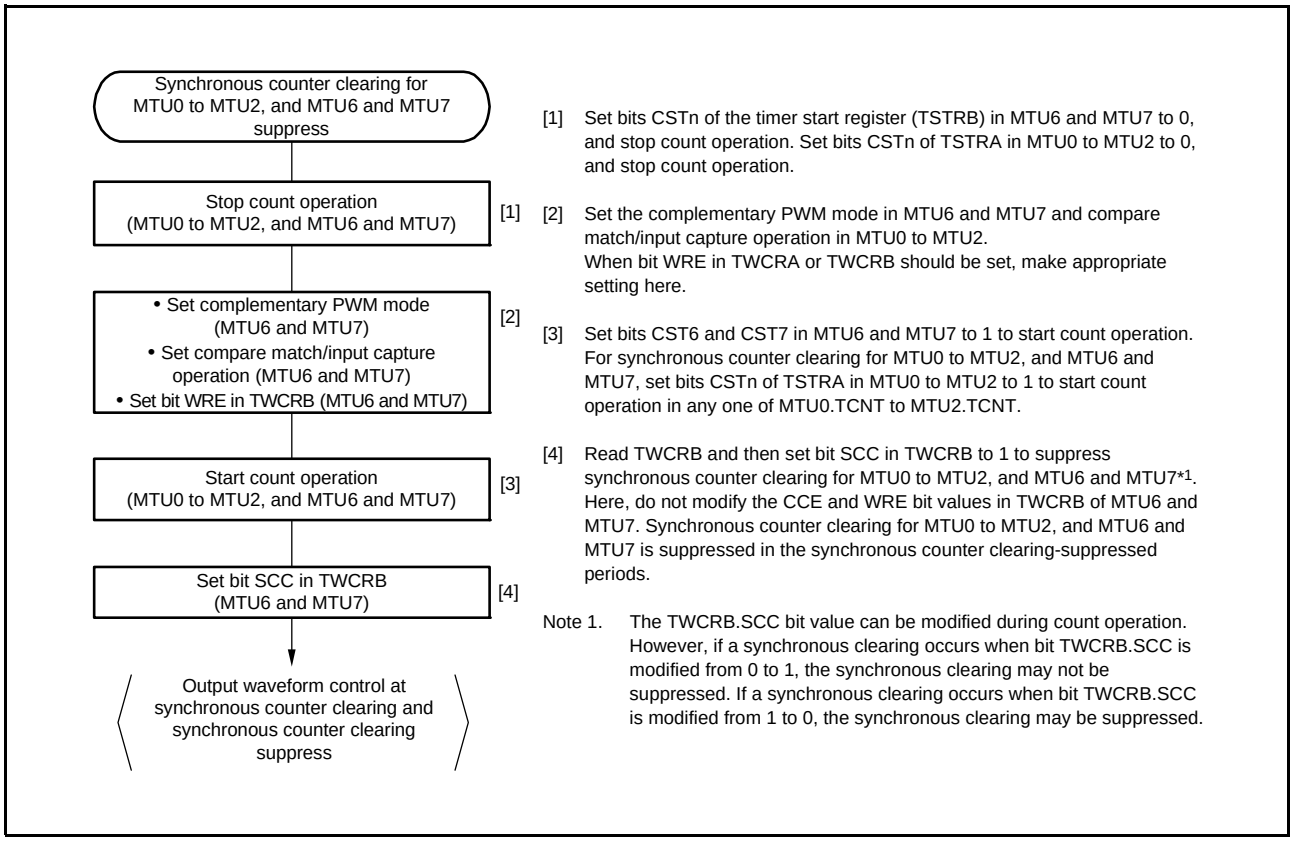


Figure 22.73 Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

- Examples of Suppression of Synchronous Counter Clearing for MTU 0 to MTU2, and MTU6 and MTU7

Figure 22.74 to Figure 22.77 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 22.74 to Figure 22.77, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.66, respectively. In these examples, the WRE bit in TWCRB in MTU6 and MTU7 is set to 1.

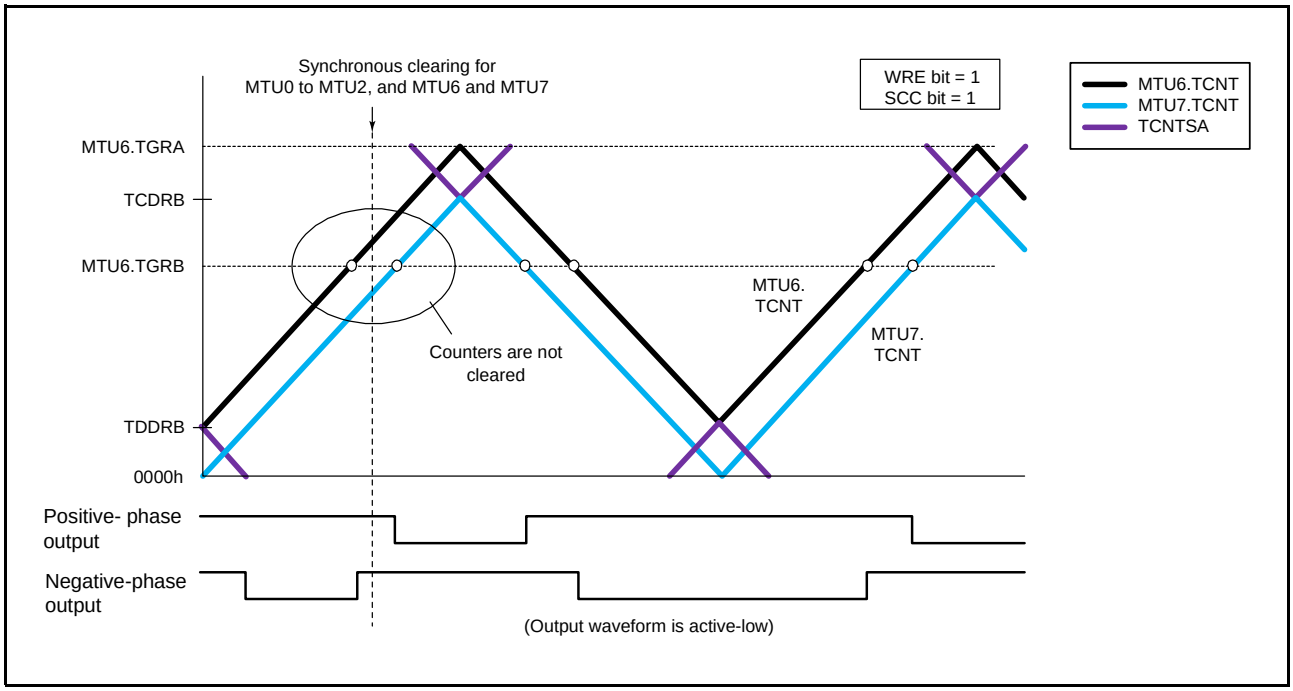


Figure 22.74 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

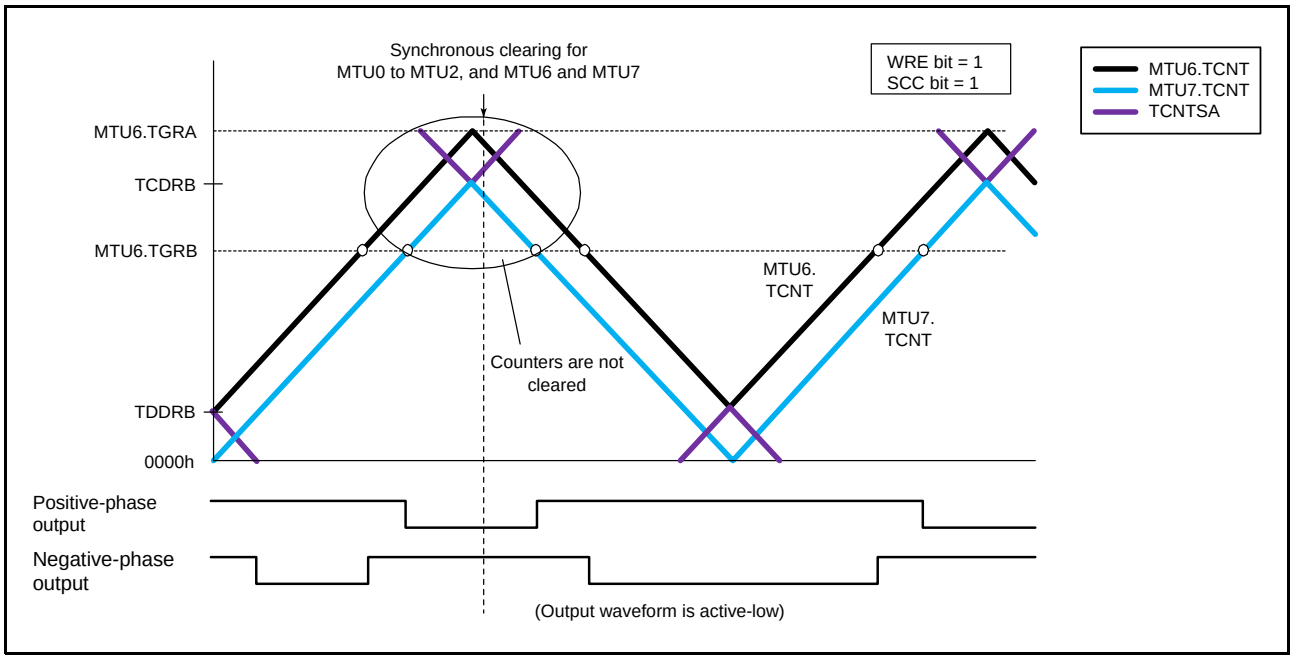


Figure 22.75 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 22.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

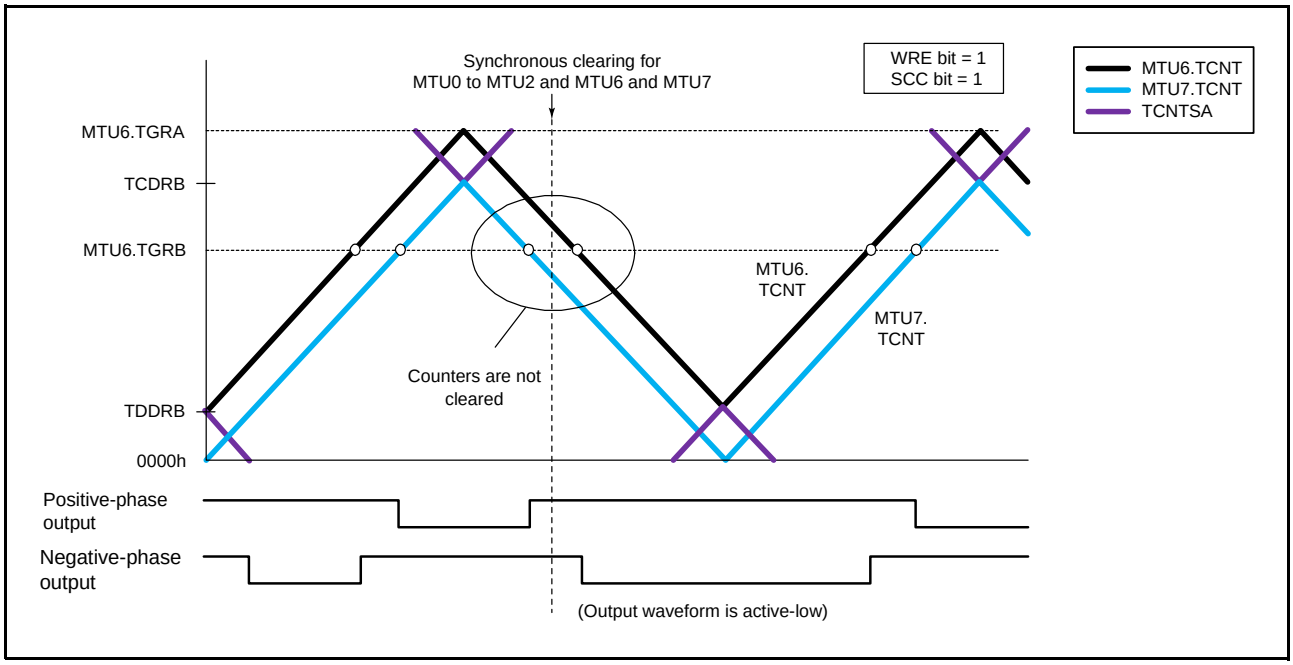


Figure 22.76 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

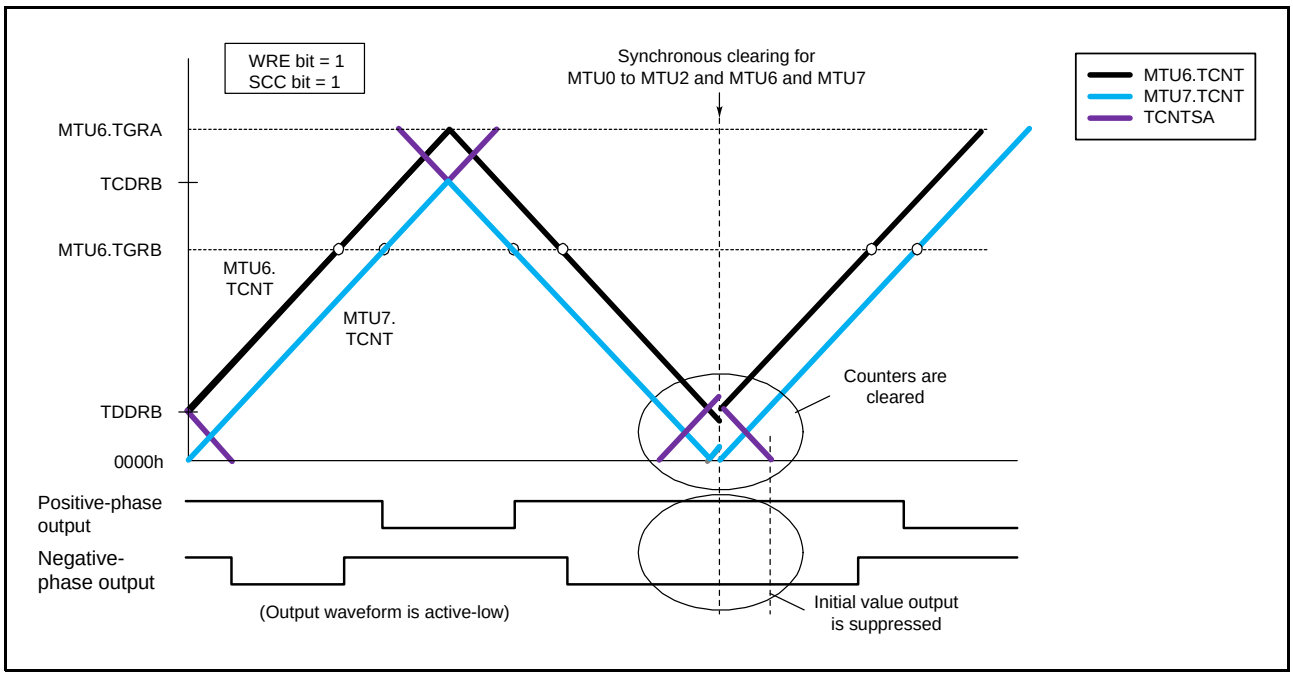


Figure 22.77 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 22.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

(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 22.78 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.

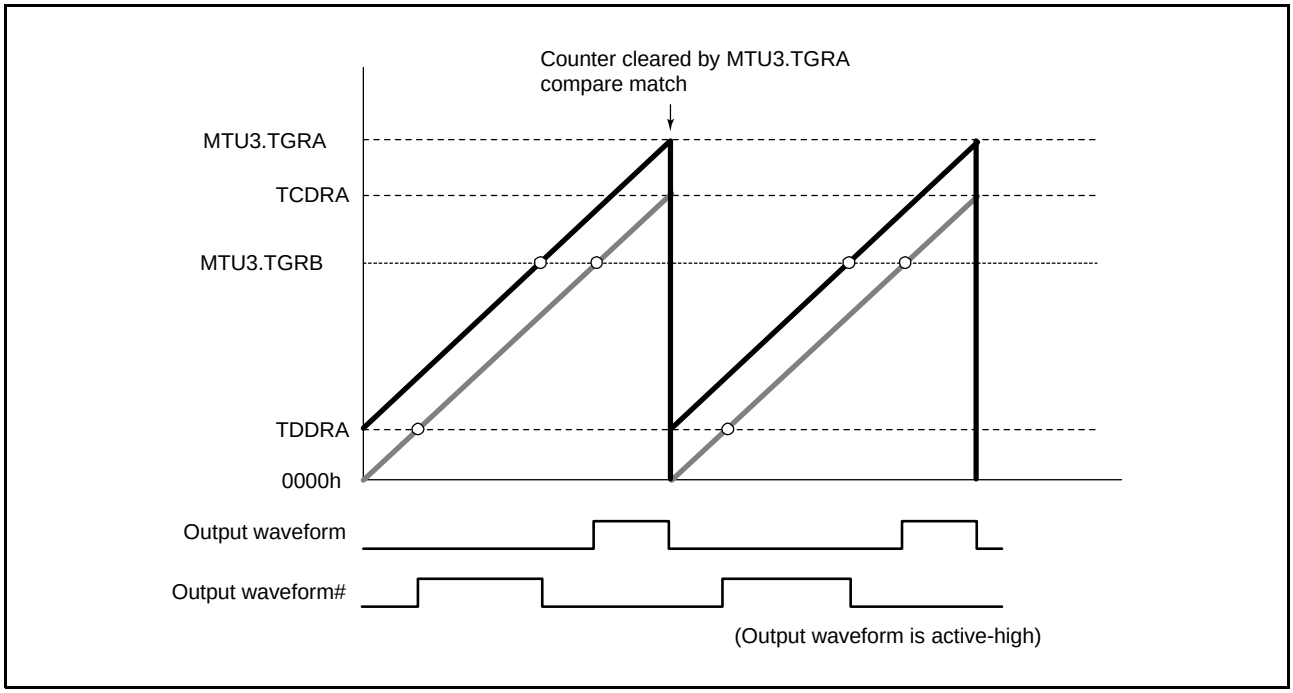


Figure 22.78 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(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 22.79 to Figure 22.82 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.

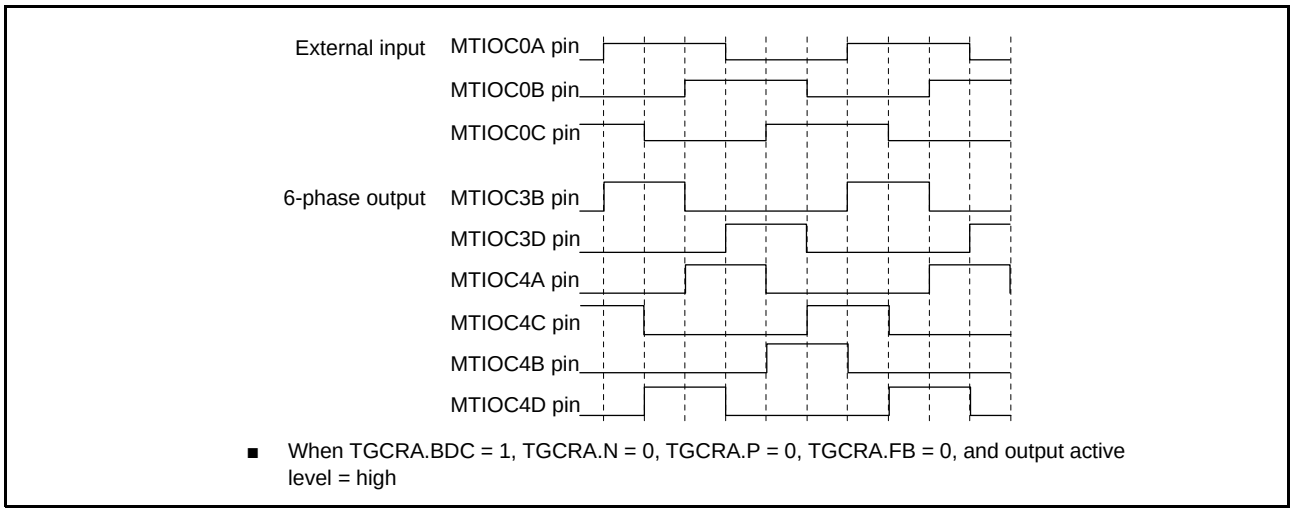


Figure 22.79 Example of Output Phase Switching by External Input (1)

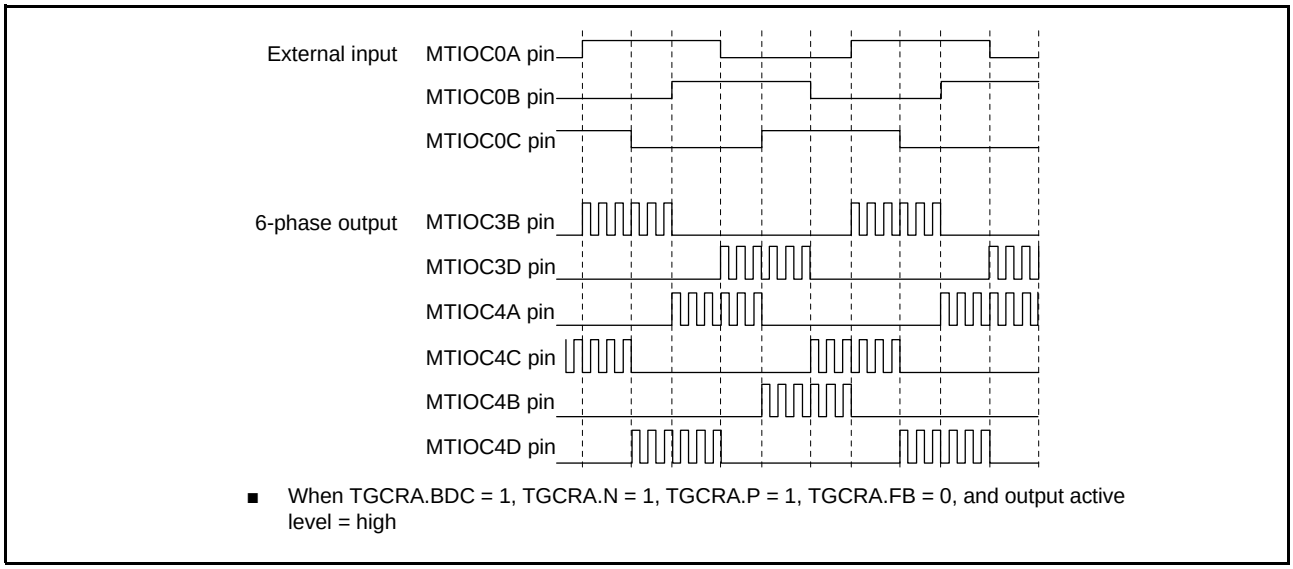
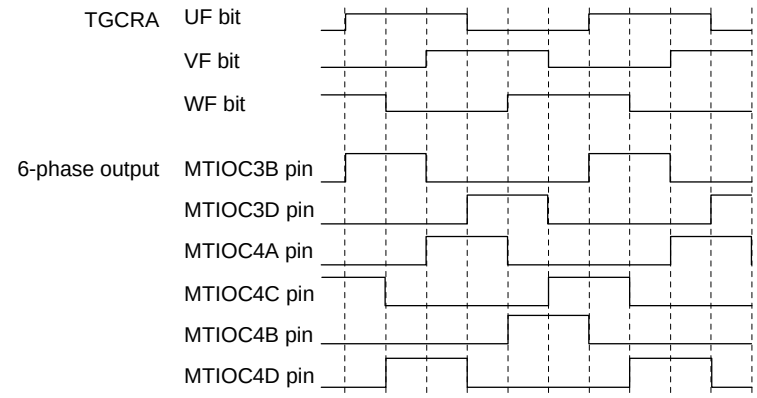
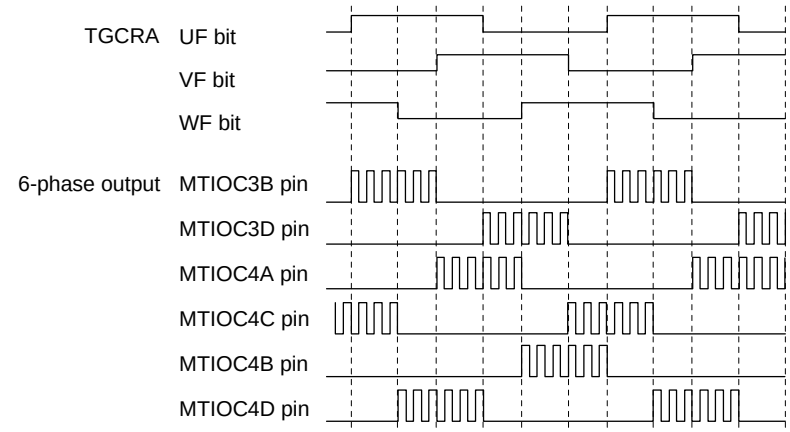


Figure 22.80 Example of Output Phase Switching by External Input (2)



- When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 1, and output active level = high

Figure 22.81 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 22.82 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 22.3.8 (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 22.83 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.

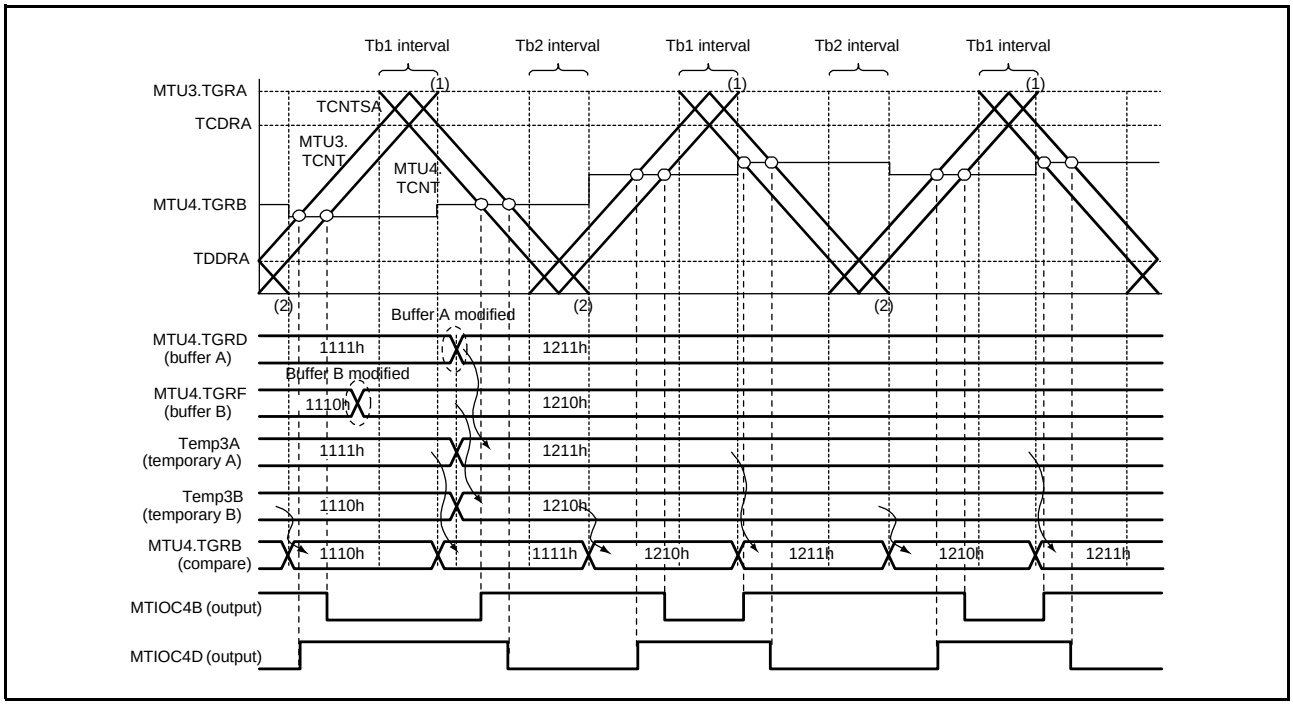


Figure 22.83 Example of Double Buffer Operation

Figure 22.84 shows an example when the buffer write value is smaller than the TDDRA (TDDRB) value, and Figure 22.85 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.

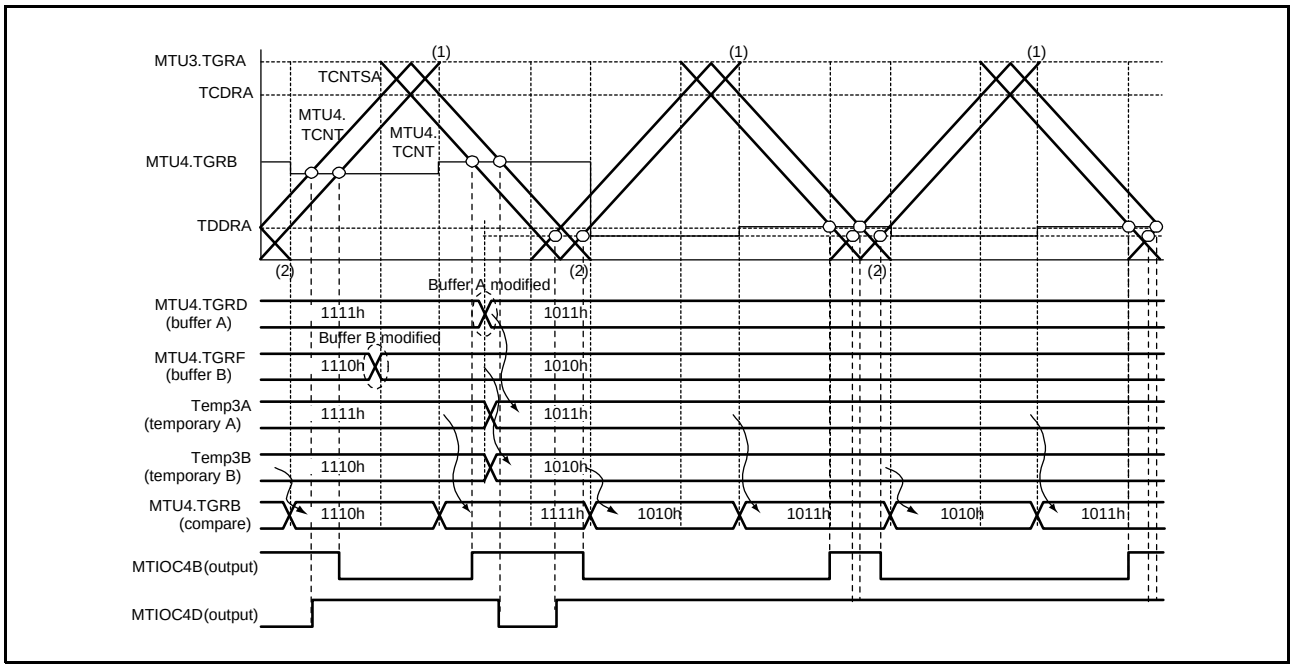


Figure 22.84 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

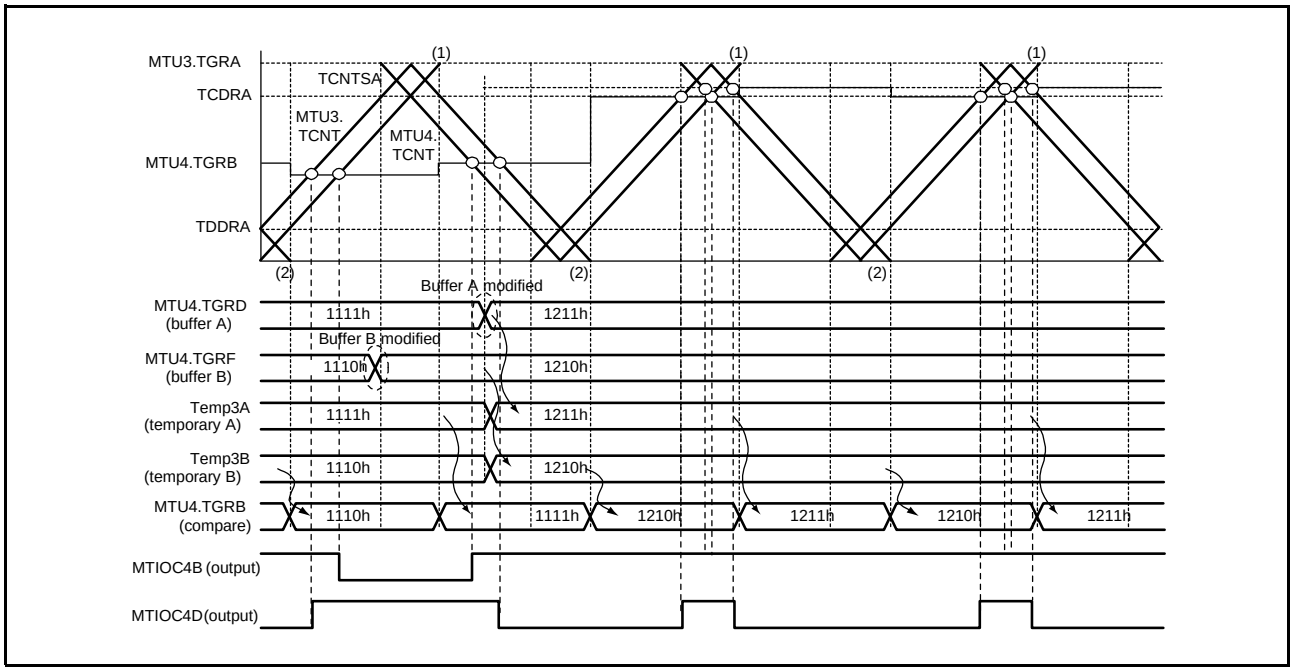


Figure 22.85 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

(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 22.3.9, 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 22.86 shows an example of the interrupt skipping function 1 setting procedure. Figure 22.87 shows the periods during which interrupt skipping count can be changed.

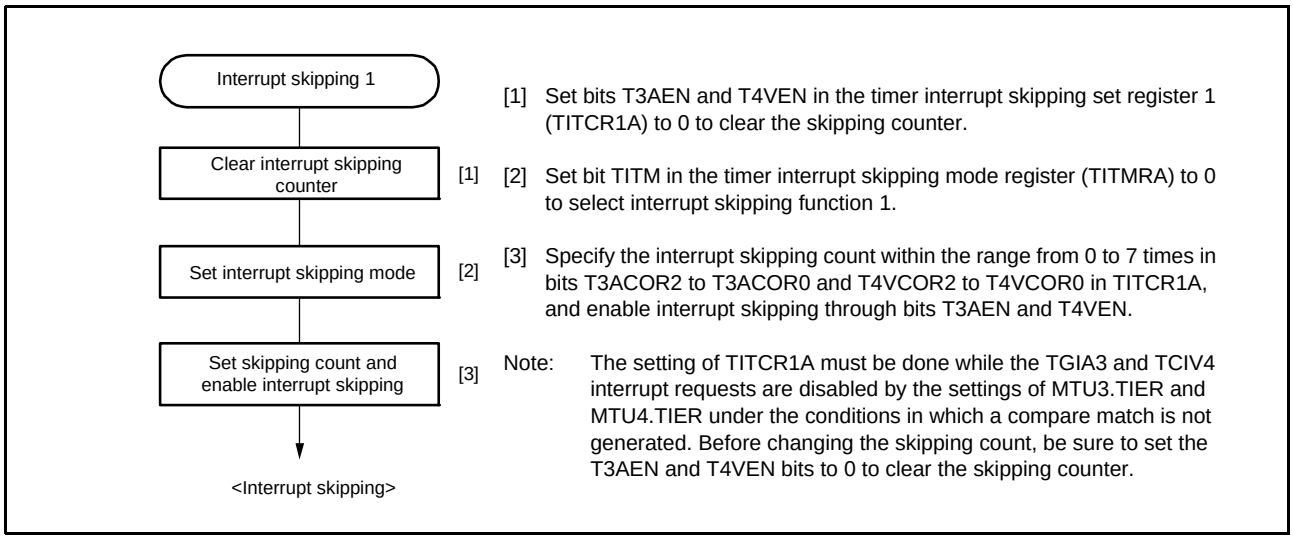


Figure 22.86 Example of Interrupt Skipping Function 1 Setting Procedure

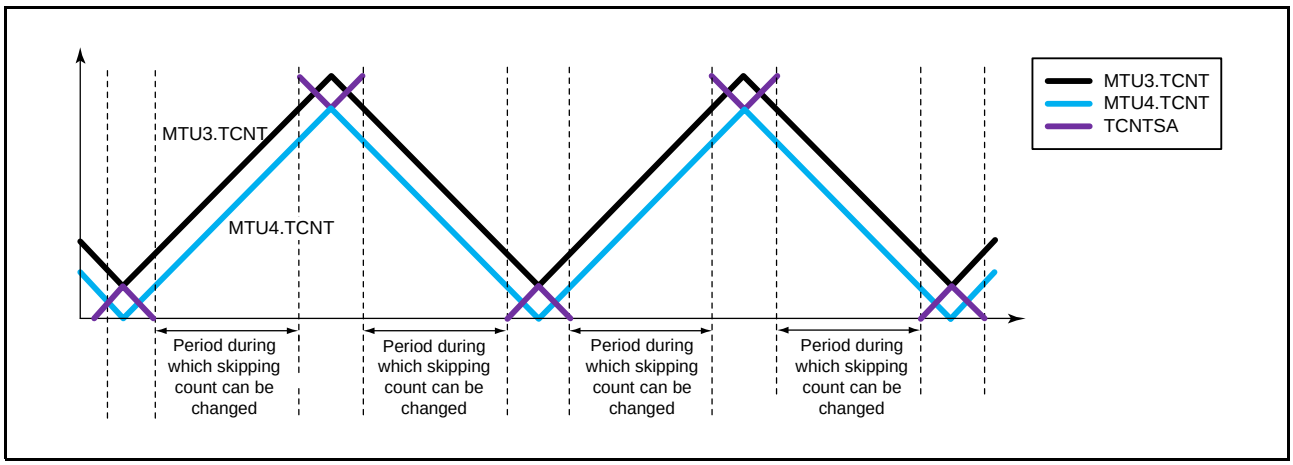


Figure 22.87 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Function 1

Figure 22.88 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.

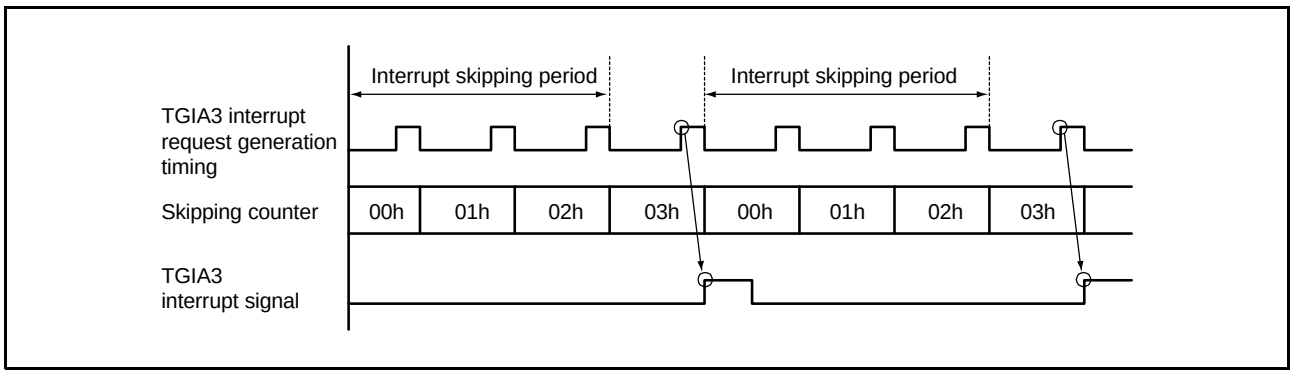
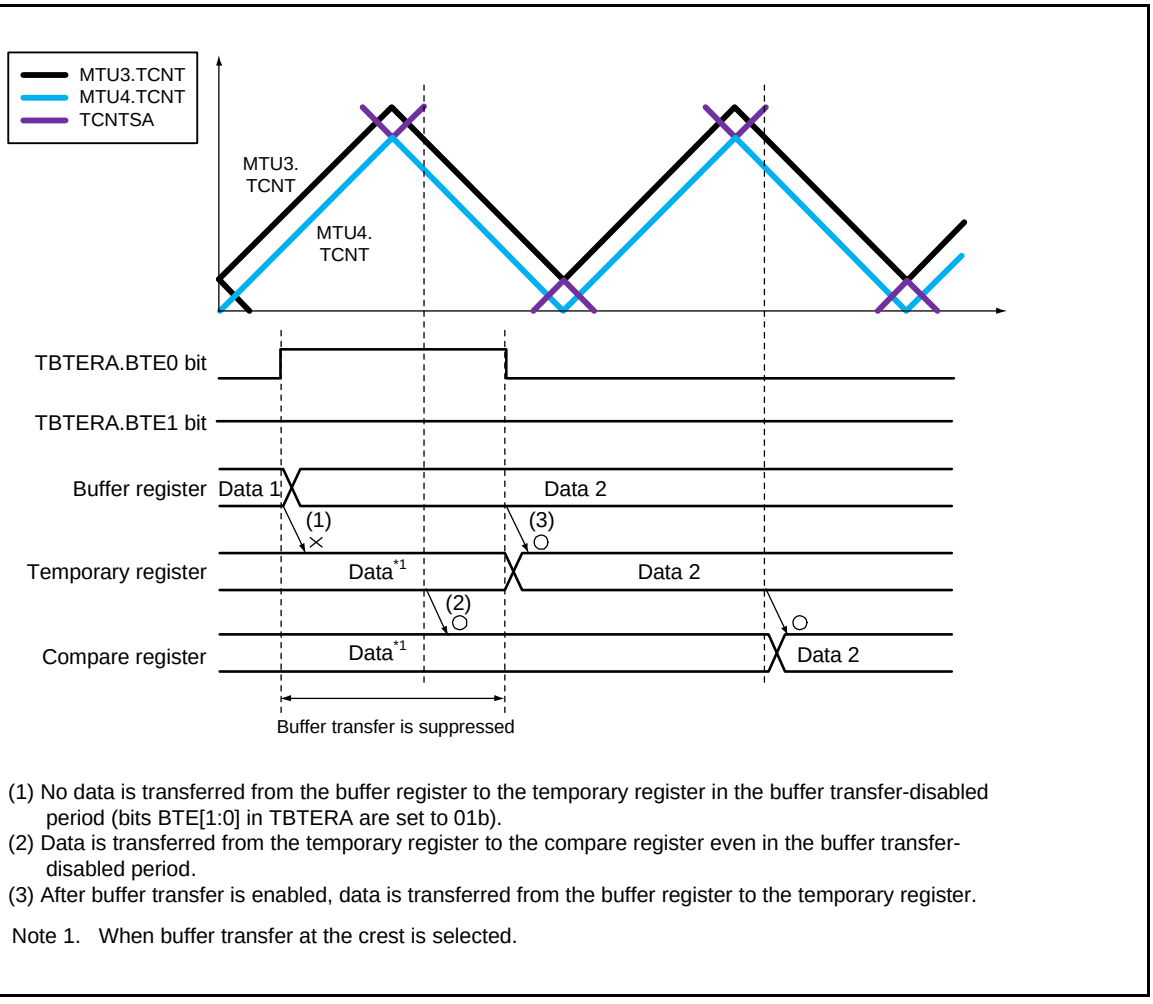


Figure 22.88 Example of Interrupt Skipping Function 1

(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 22.89 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 22.90 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 22.91 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 22.89 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)

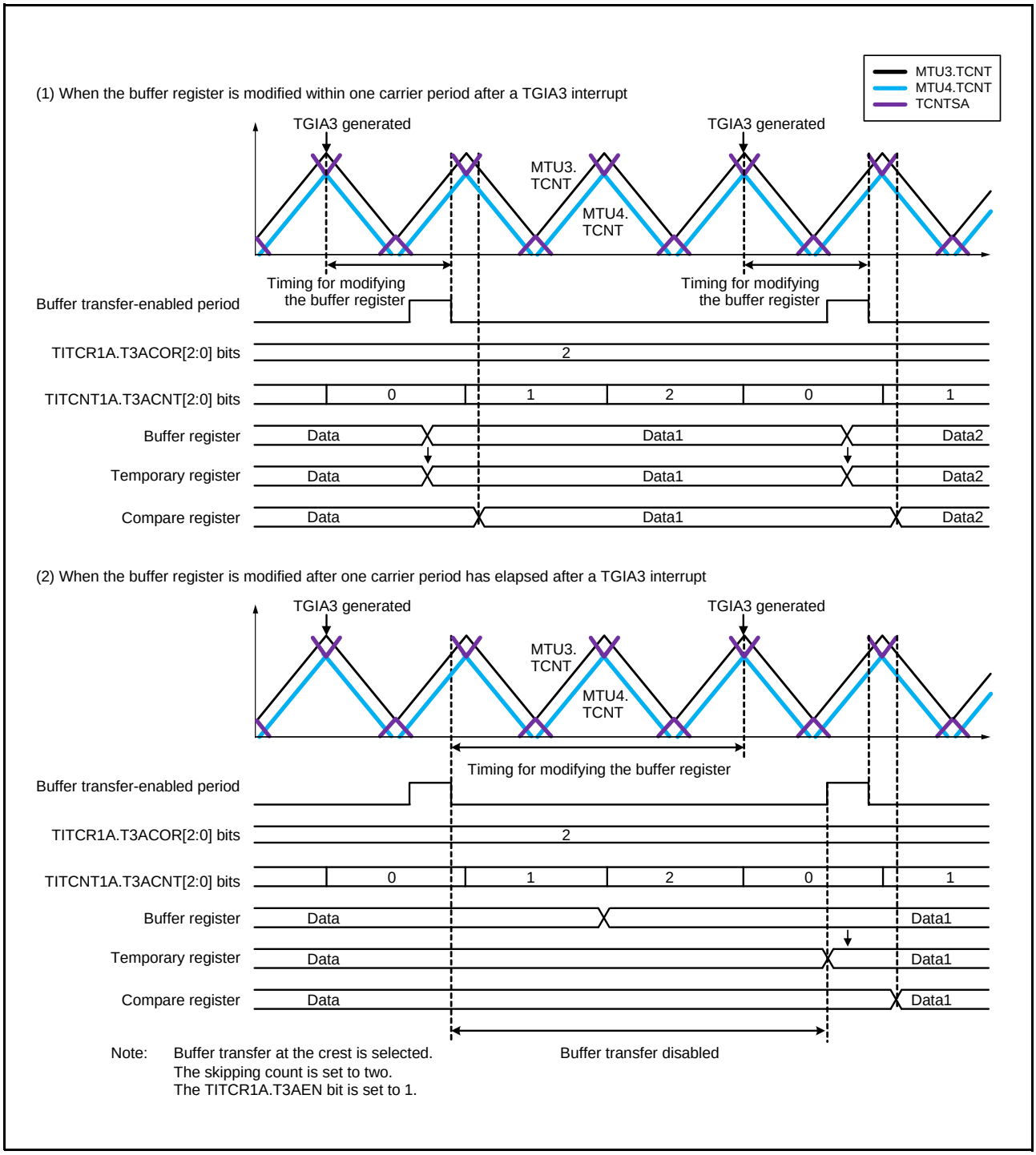


Figure 22.90 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

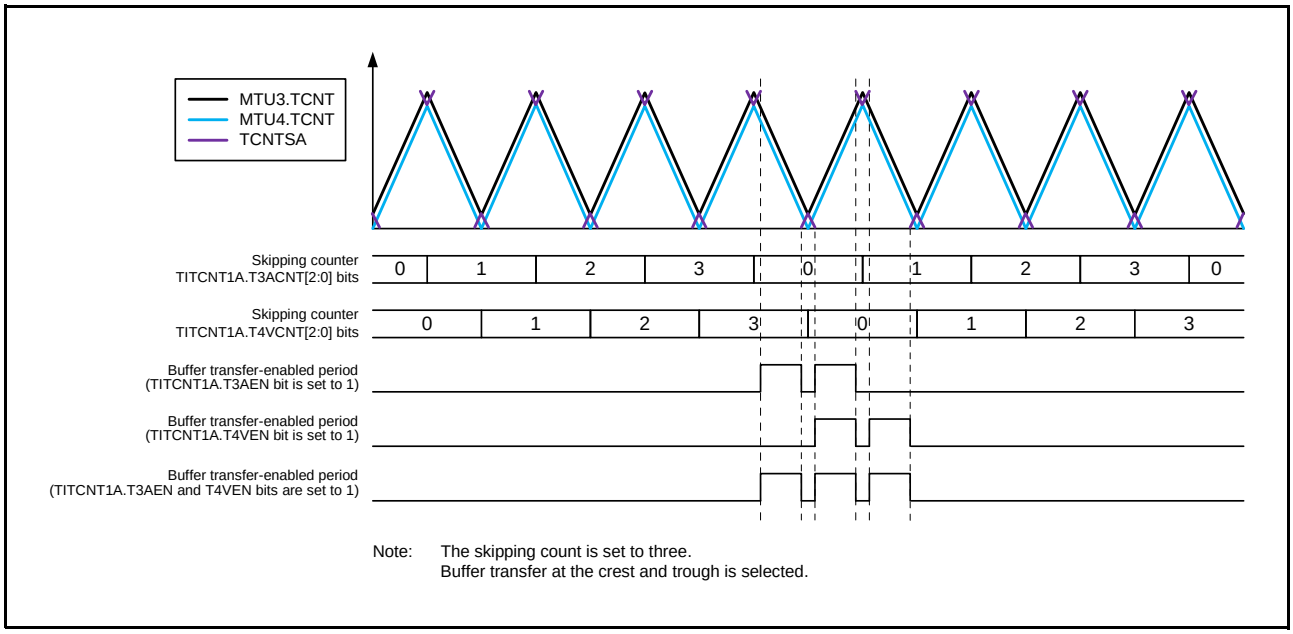


Figure 22.91 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

(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 23, Port Output Enable 3 (POE3a), for details.

22.3.9 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 22.92 shows an example of procedure for specifying the A/D conversion start request delaying function.

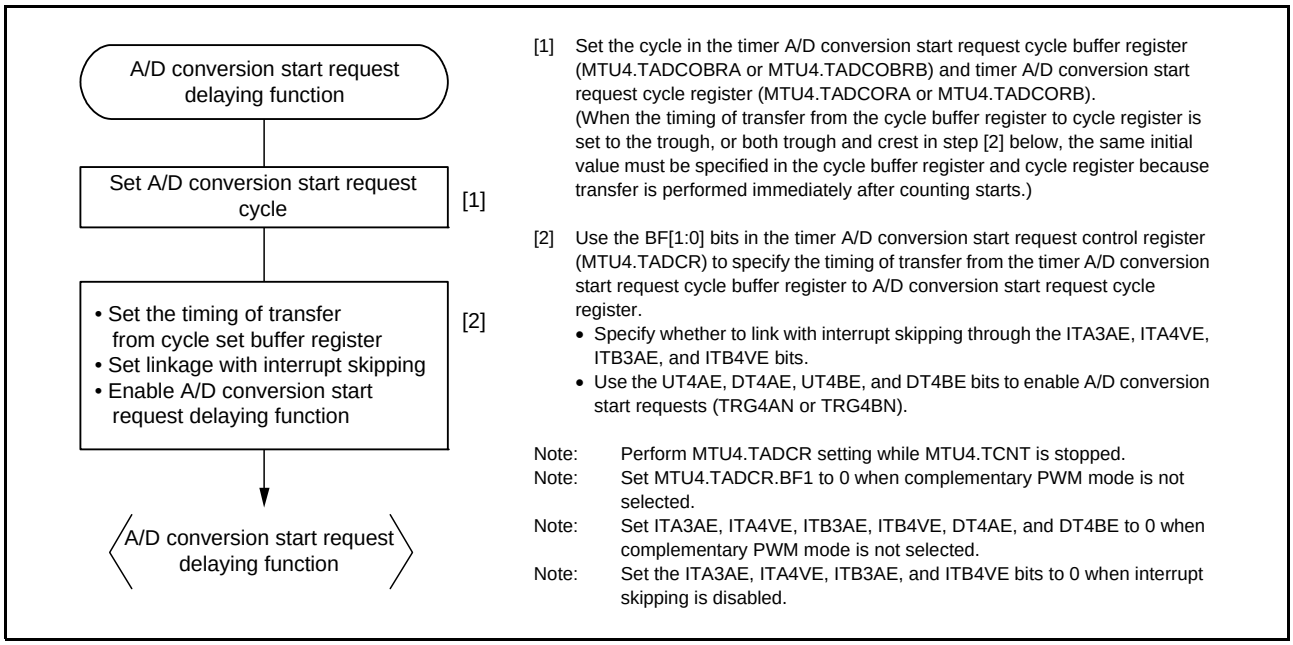
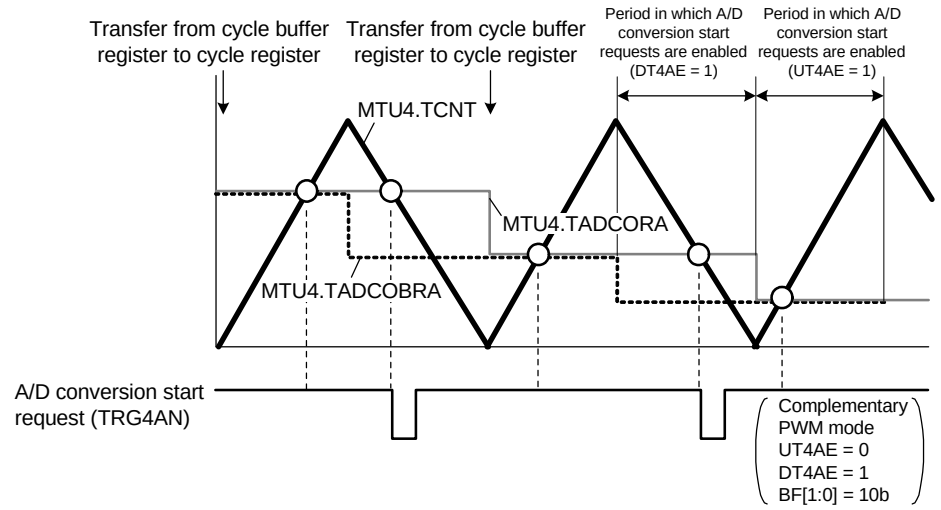


Figure 22.92 Example of Procedure for Specifying A/D Conversion Start Request Delaying Function (MTU3 and MTU4)

(2) Basic Example of A/D Conversion Start Request Delaying Function Operation

Figure 22.93 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 22.93 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 22.93.

(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 22.6.28, 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.

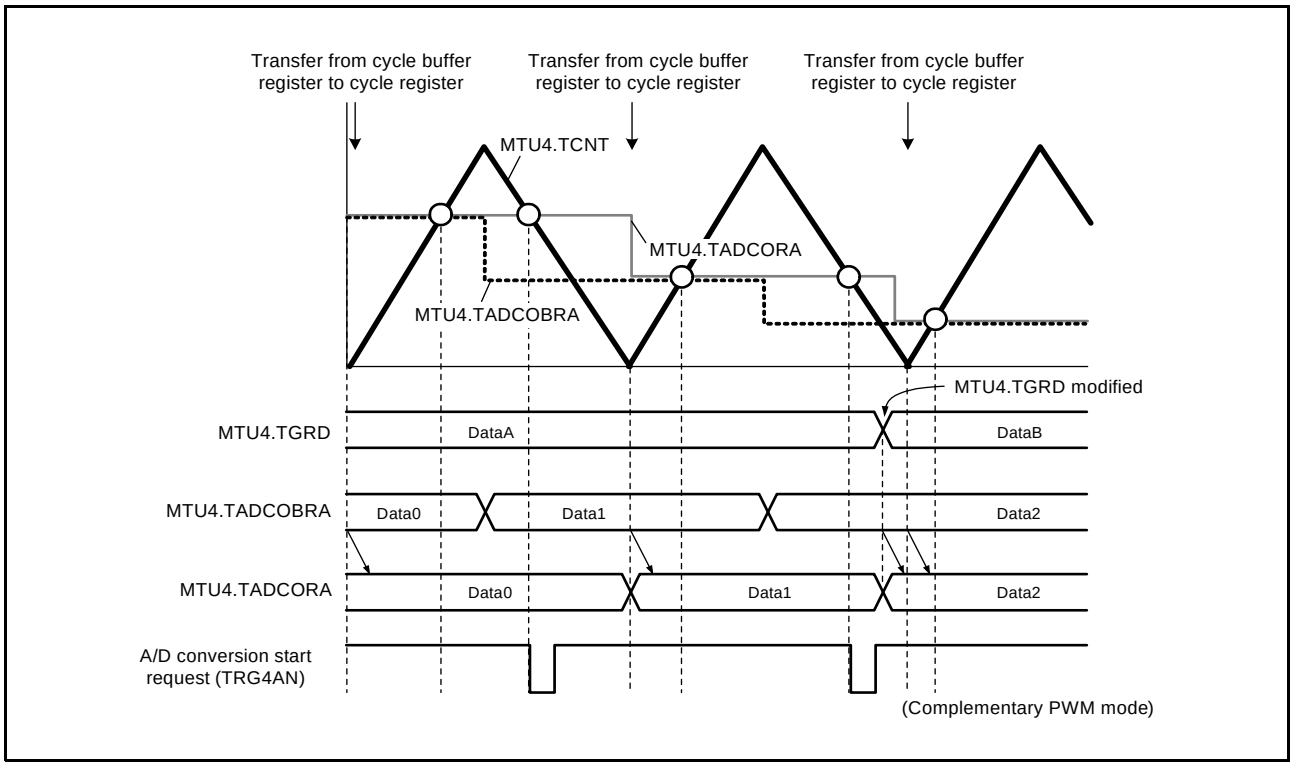


Figure 22.94 Example of A/D Conversion Start Request Signal (TRG4AN) and Buffer Transfer Operation

(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 22.95 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 22.96 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).

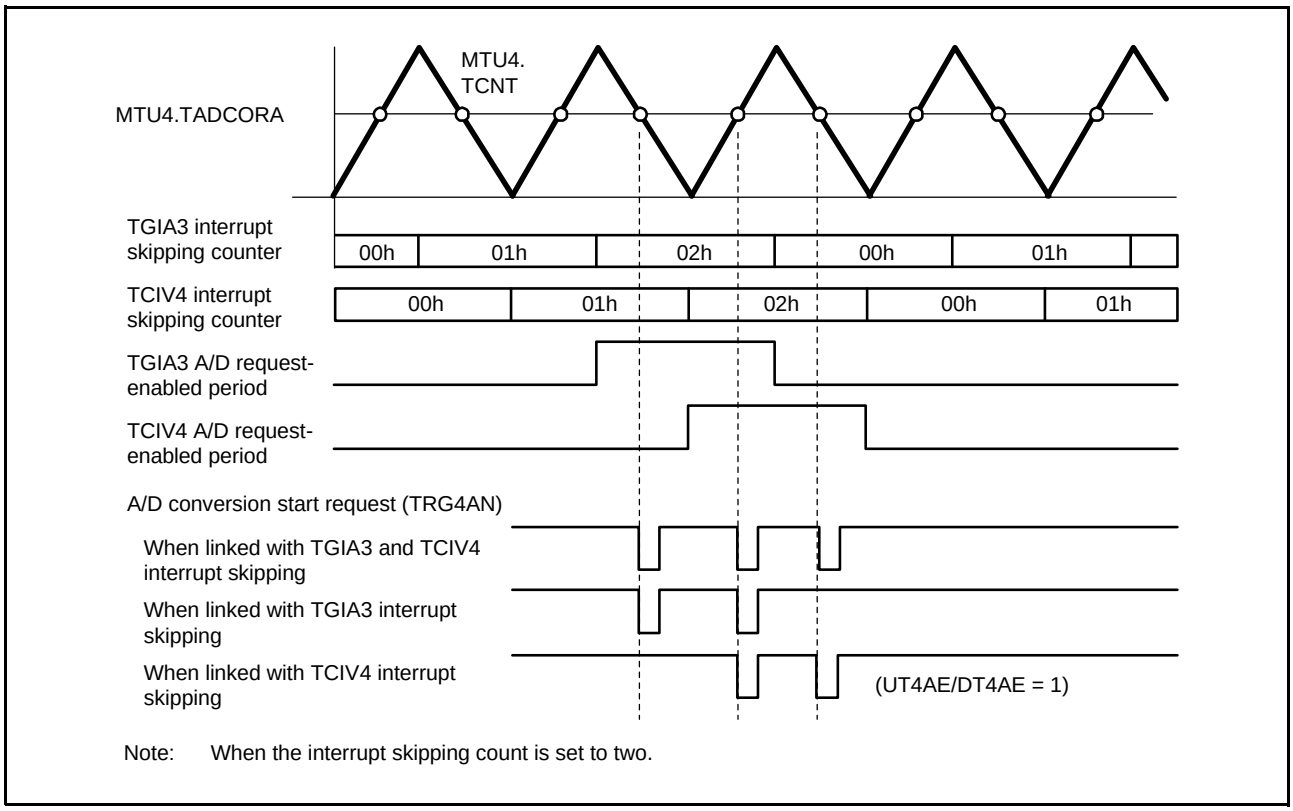


Figure 22.95 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)

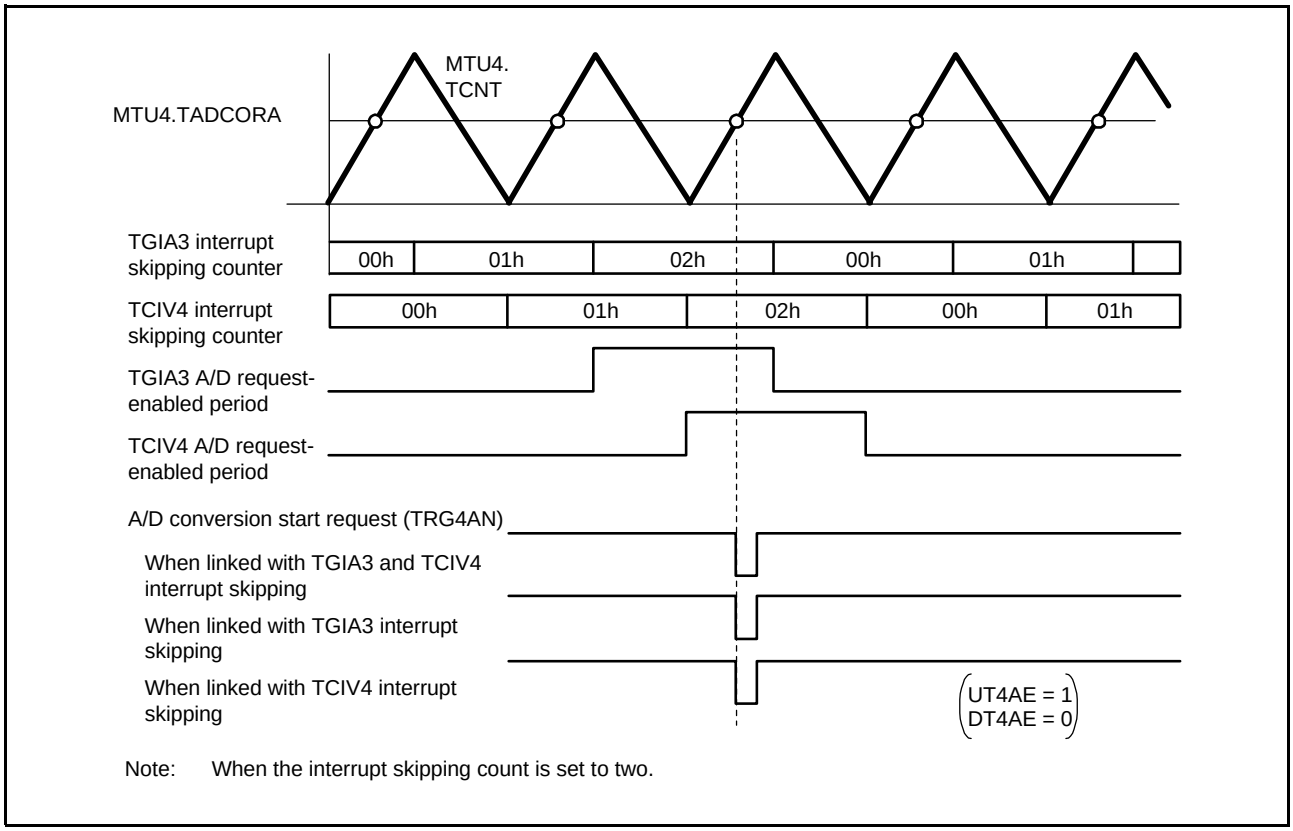


Figure 22.96 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

(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 22.97 shows an example of procedure for setting interrupt skipping function 2.

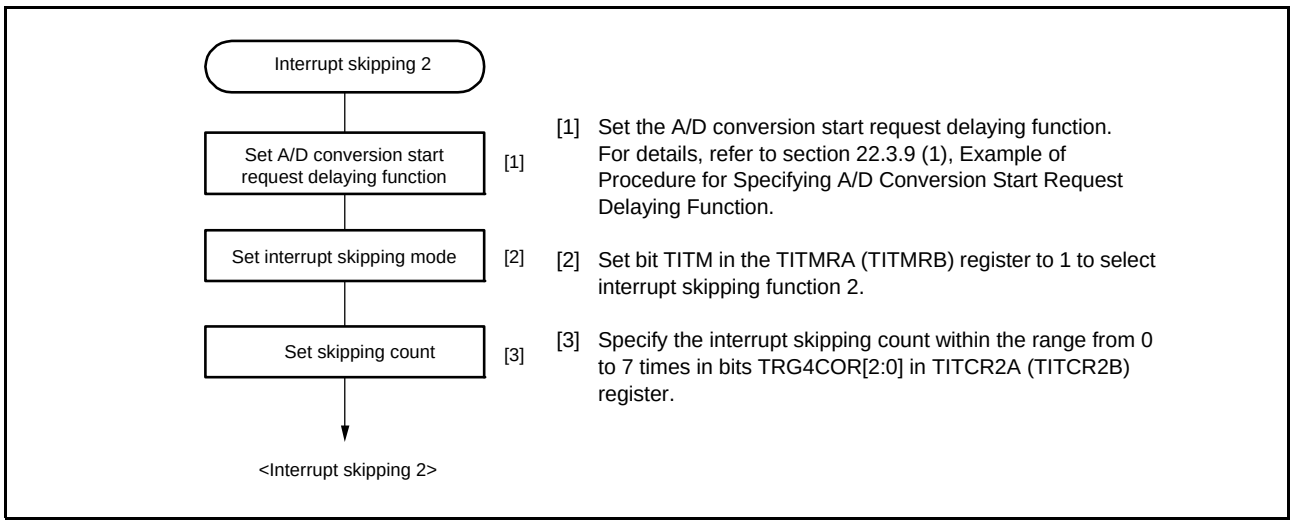


Figure 22.97 Example of Procedure for Setting Interrupt Skipping Function 2

(b) Example of Interrupt Skipping Function 2 Operation

Figure 22.98 shows an example of interrupt skipping function 2 operation.

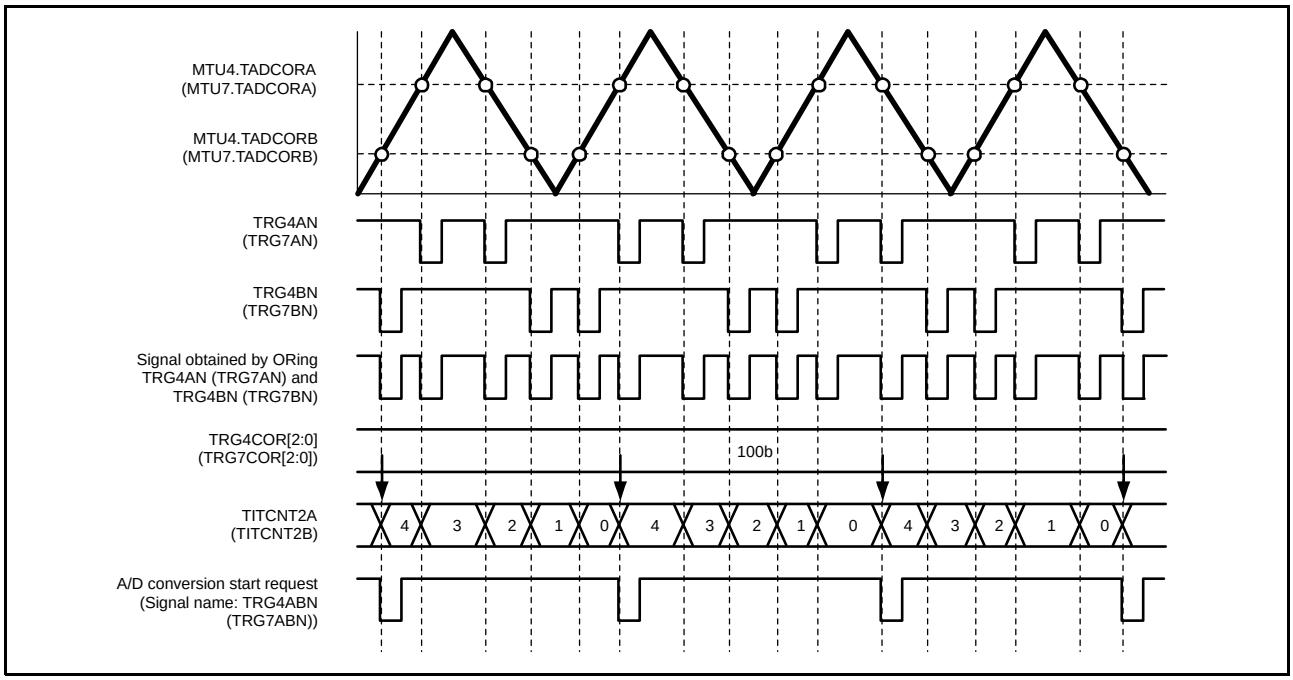


Figure 22.98 Example of Interrupt Skipping Function 2 Operation (Skipping Count is Set to Four)

22.3.10 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 22.99 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7.

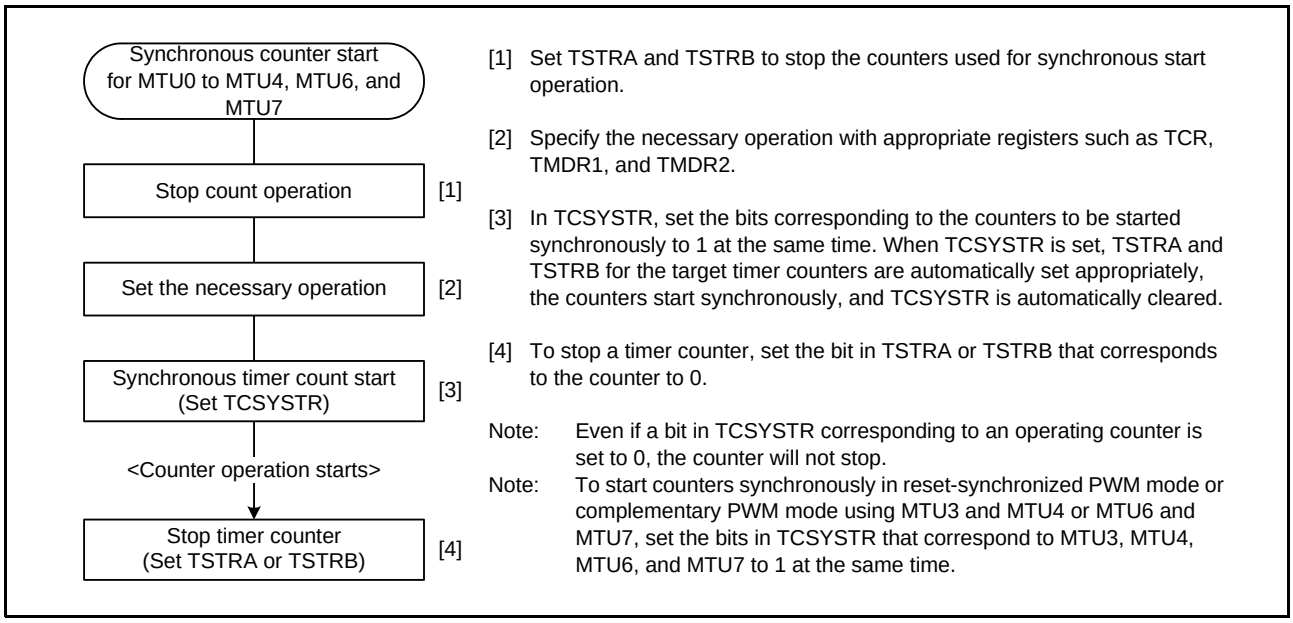


Figure 22.99 Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

(b) Examples of Synchronous Counter Start Operation

Figure 22.100 shows an examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7.

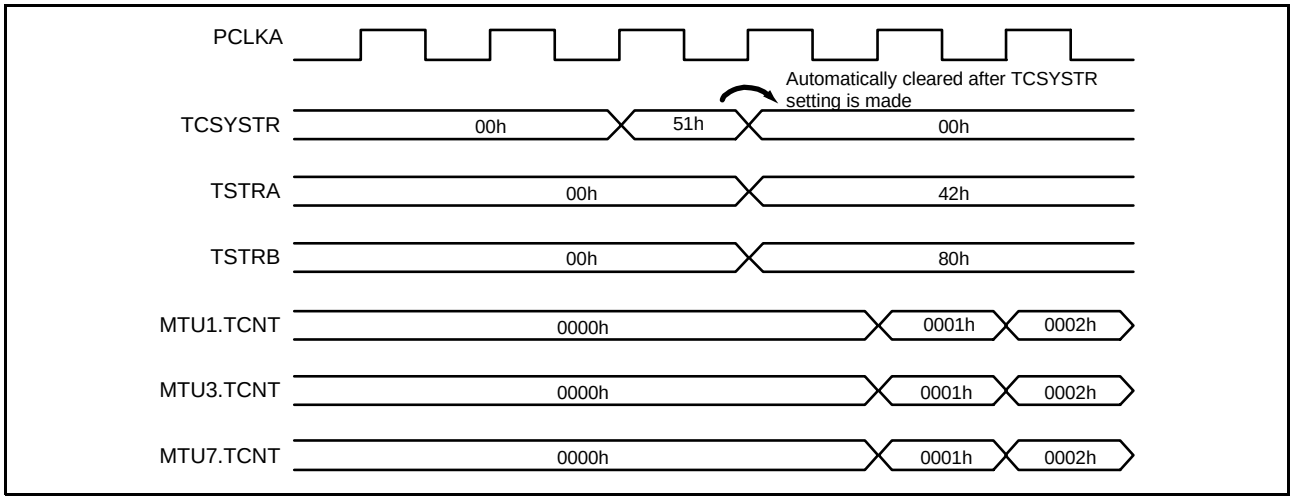


Figure 22.100 Examples of Synchronous Counter Start Operation for MTU0 to MTU4, MTU6, and MTU7

(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 22.101 shows an example of procedure for specifying synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

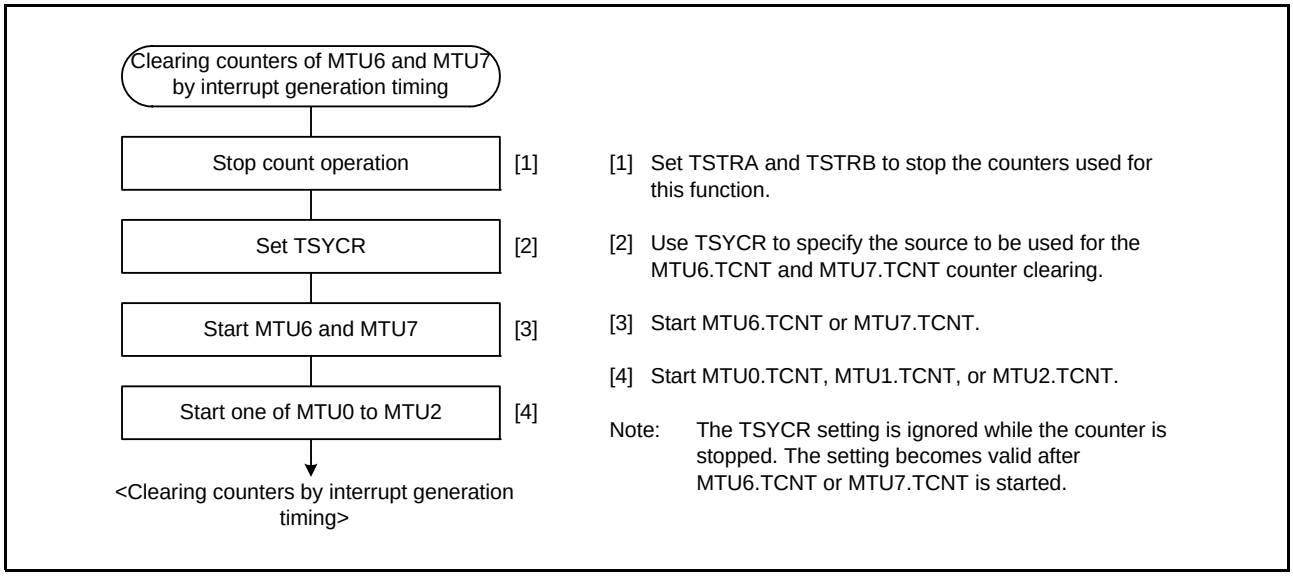


Figure 22.101 Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

(b) Examples of Synchronous Counter Clearing for MTU6 and MTU7

Figure 22.102 and Figure 22.103 show examples of synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

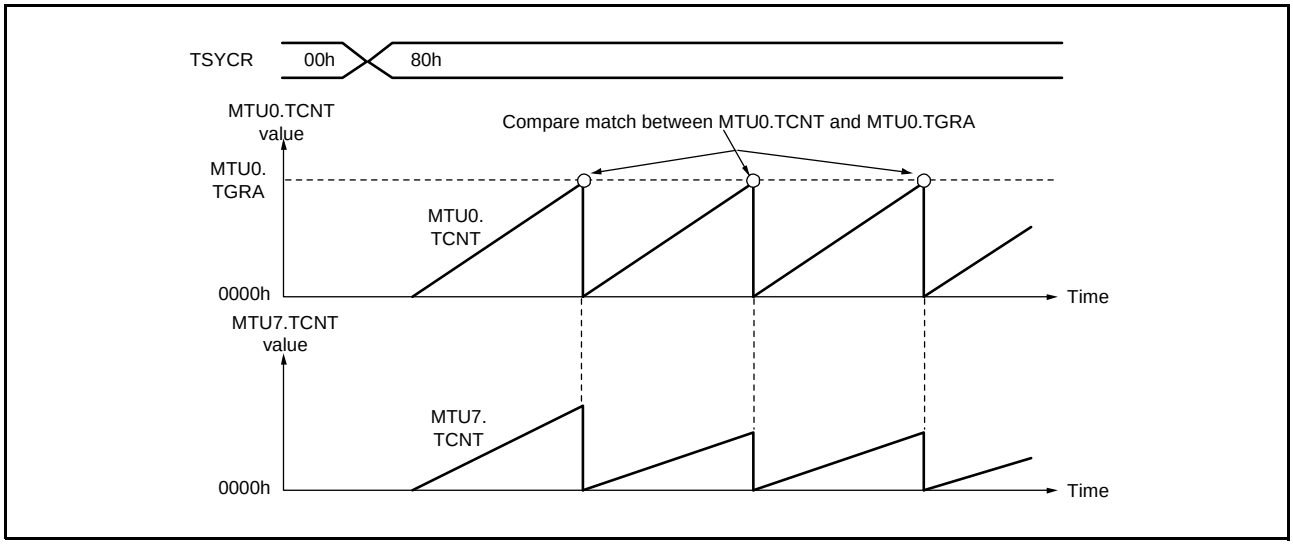


Figure 22.102 Example of Synchronous Counter Clearing for MTU6 and MTU7 (1)

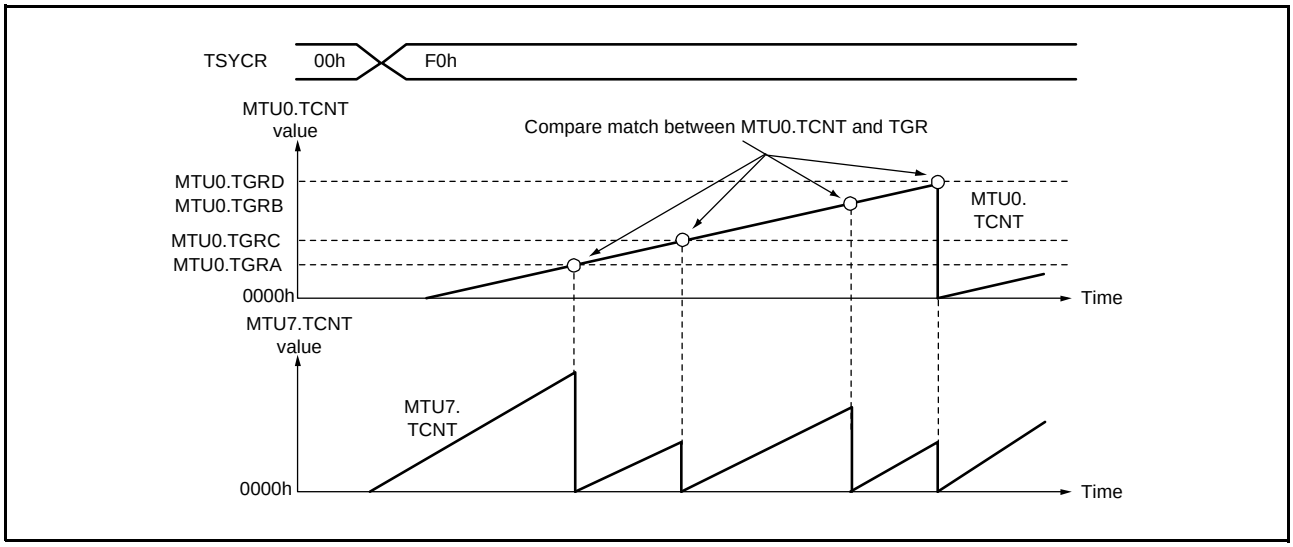


Figure 22.103 Example of Synchronous Counter Clearing for MTU6 and MTU7 (2)

22.3.11 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 22.104 shows an example of setting external pulse width measurement, and Figure 22.105 an example of external pulse width measurement.

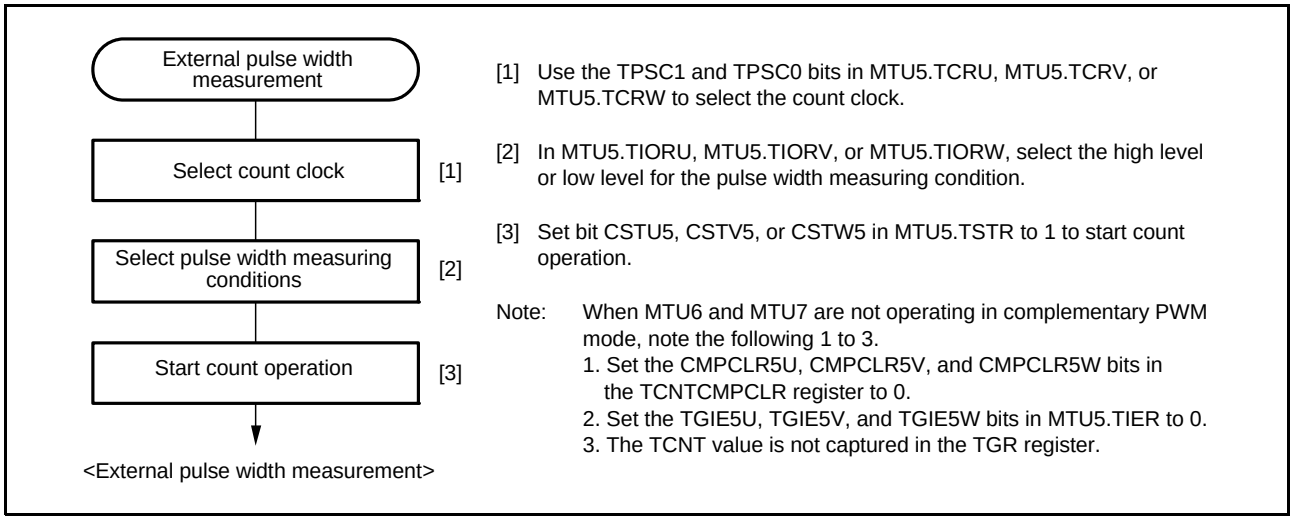


Figure 22.104 Example of External Pulse Width Measurement Setting Procedure

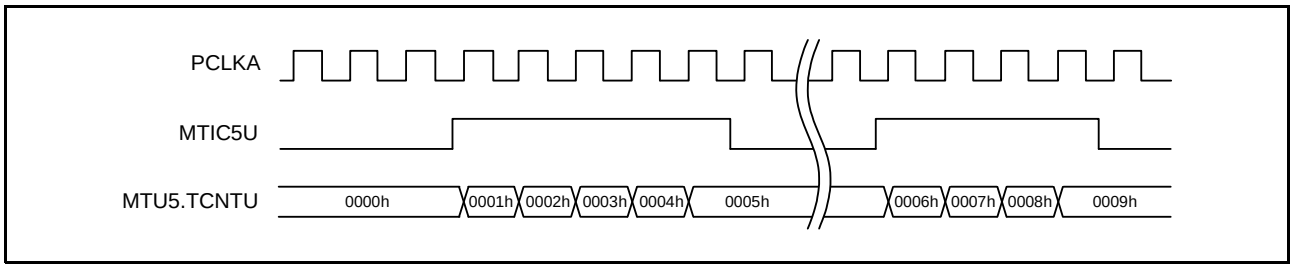


Figure 22.105 Example of External Pulse Width Measurement (Measuring High Pulse Width)

22.3.12 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 22.106 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 22.107). Figure 22.108 shows the procedure for setting dead time compensation using MTU5 to MTU7. For details on MTU5 operation at this time, refer to section 22.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

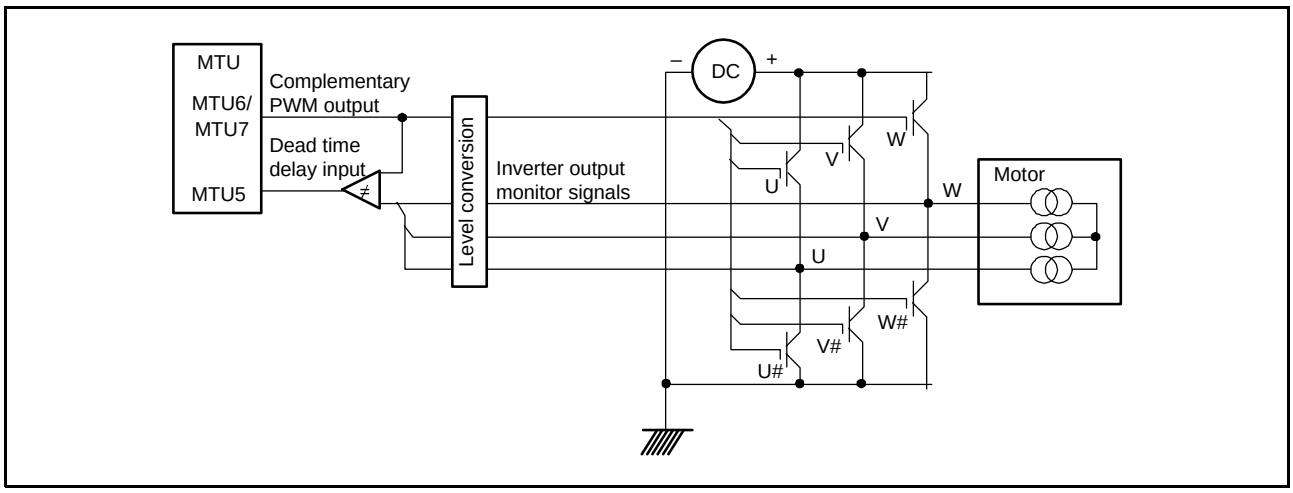


Figure 22.106 Motor Control Circuit Example

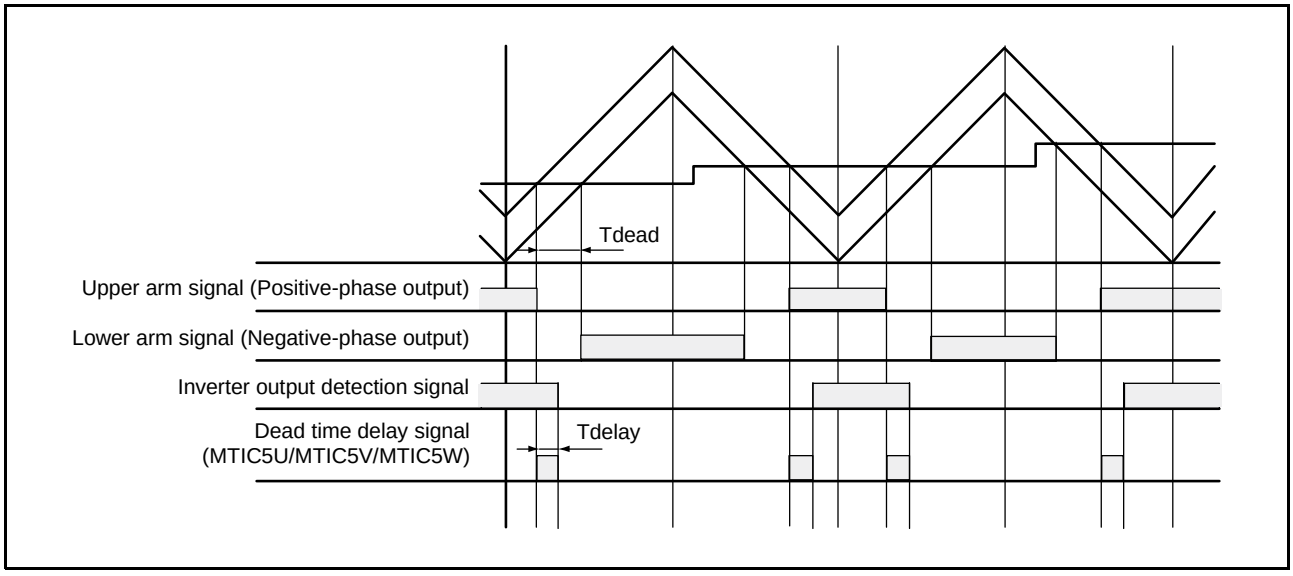
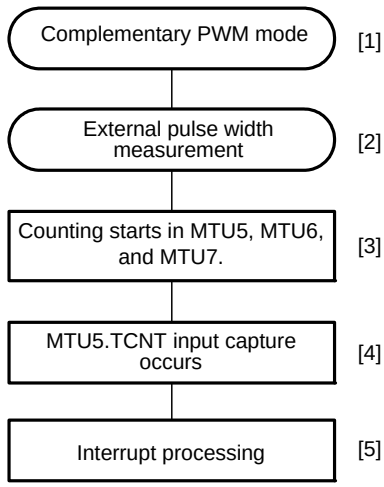


Figure 22.107 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 22.108 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 22.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section 22.3.11, 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 22.108 Example of Dead Time Compensation Setting Procedure

22.3.13 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 22.109 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.

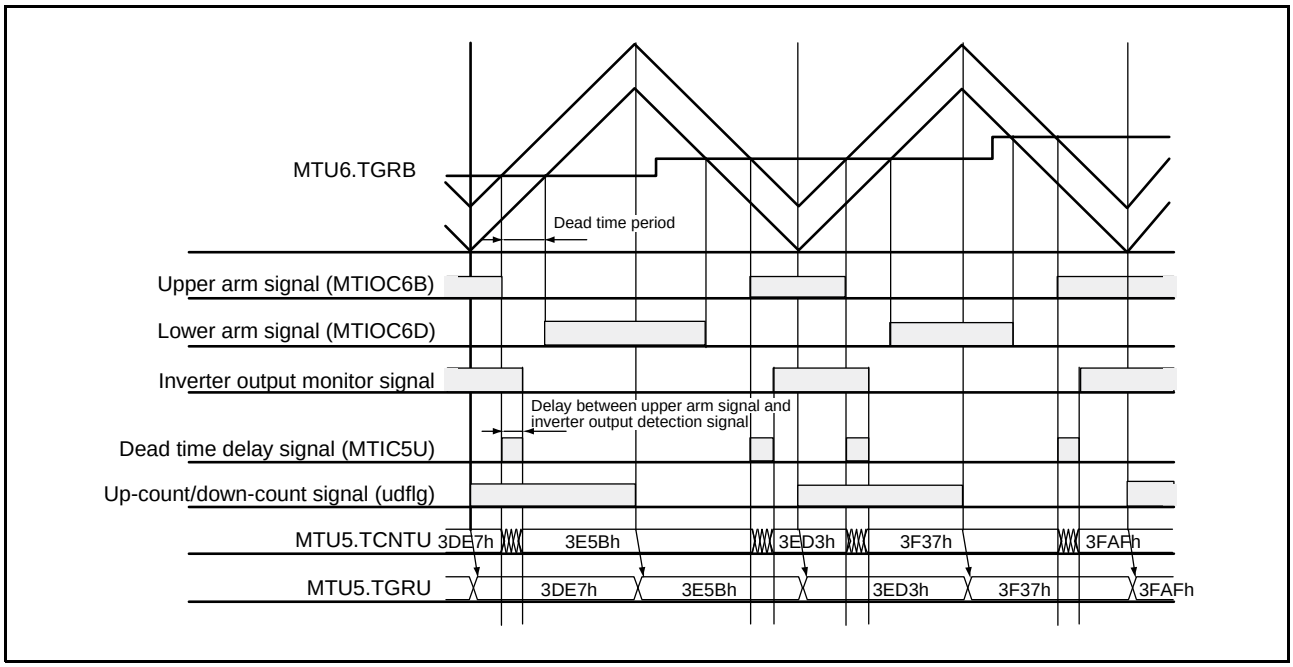


Figure 22.109 TCNTU Capture at Crest and Trough in Complementary PWM Operation

22.3.14 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 22.110 shows the timing of noise filtering.

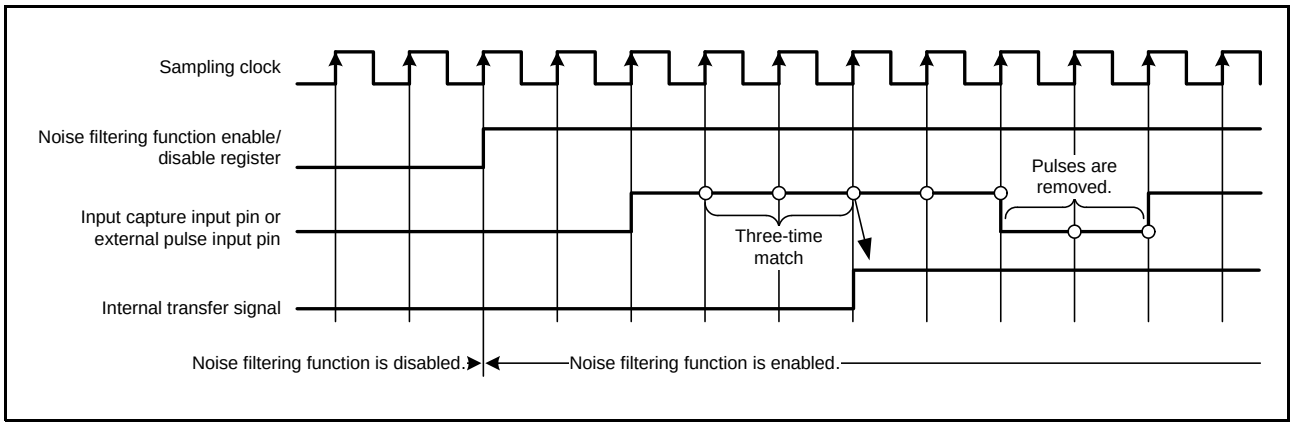


Figure 22.110 Timing of Noise Filtering

22.4 Interrupt Sources

22.4.1 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 14, Interrupt Controller (ICUF). Table 22.77 lists the MTU interrupt sources.

Table 22.77 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).

22.4.2 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 18, 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 17, DMA Controller (DMACAa).
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.

22.4.3 A/D Converter Trigger Sources

The A/D converter can be triggered by one of the following three methods in the MTU. Table 22.78 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 22.3.9, 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 22.78 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).

22.5 Operation Timing

22.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 22.111 and Figure 22.112 show the TCNT count timing in internal clock operation, Figure 22.113 shows the TCNT count timing in external clock operation (normal mode), and Figure 22.114 shows the TCNT count timing in external clock operation (phase counting mode).

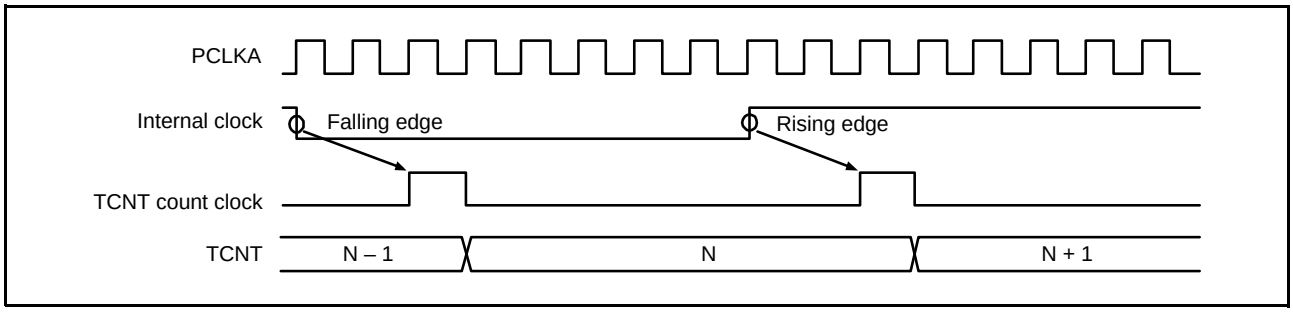


Figure 22.111 Count Timing in Internal Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

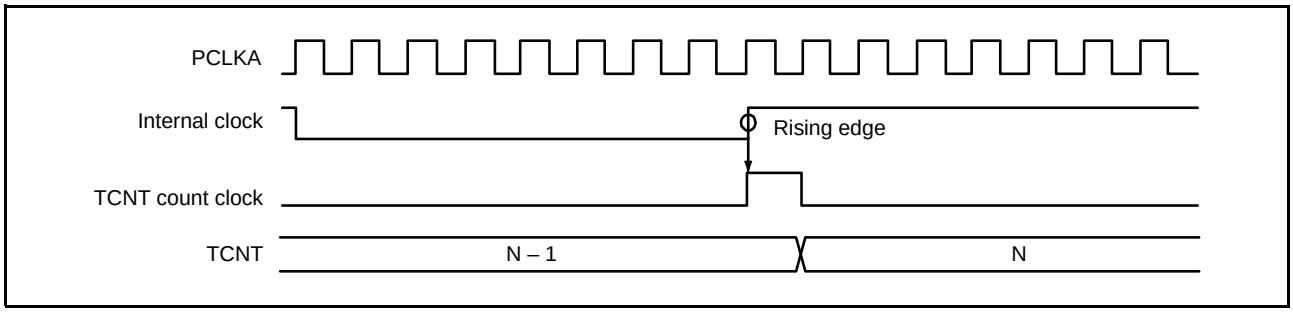


Figure 22.112 Count Timing in Internal Clock Operation (MTU5)

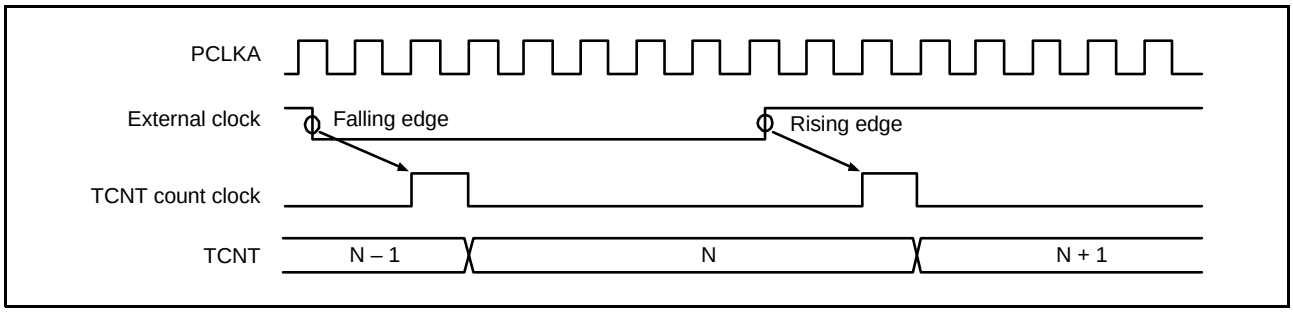


Figure 22.113 Count Timing in External Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

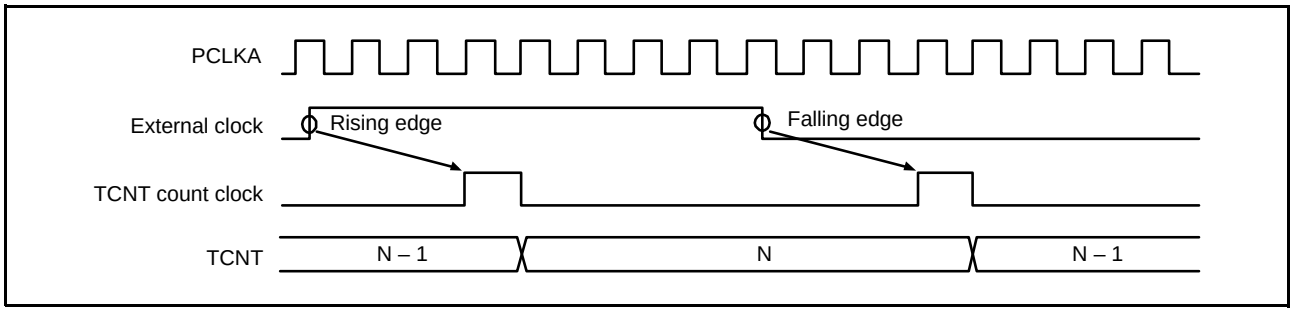


Figure 22.114 Count Timing in External Clock Operation (Phase Counting Mode)

(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 22.115 shows the output compare output timing (normal mode or PWM mode) and Figure 22.116 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

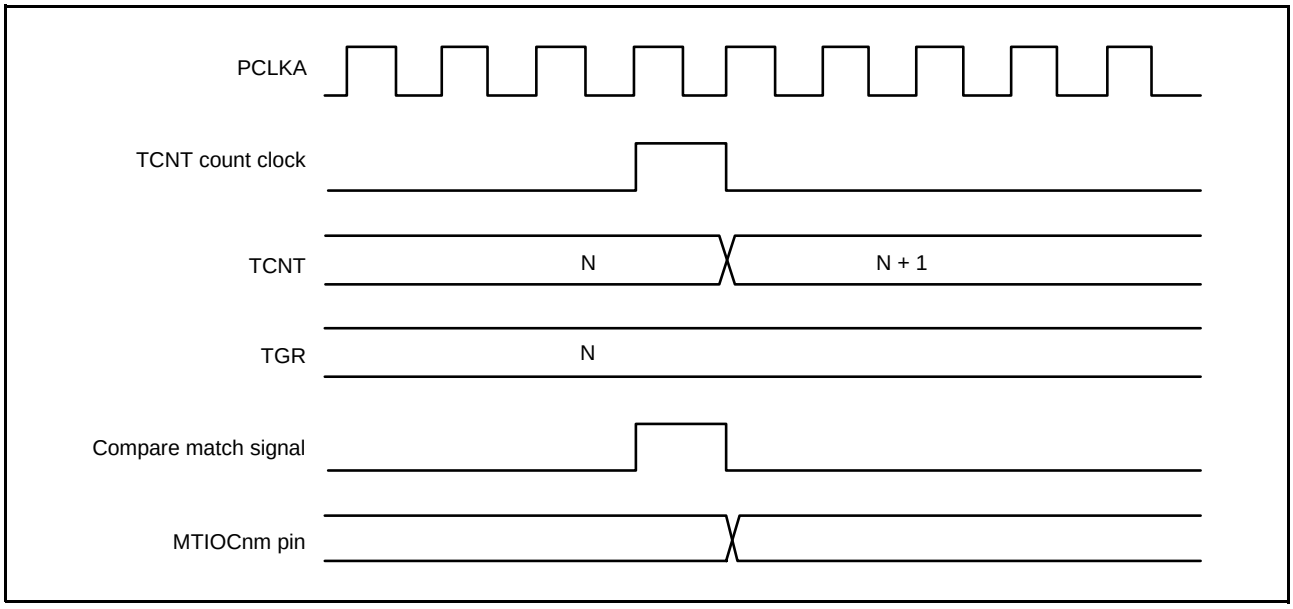


Figure 22.115 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

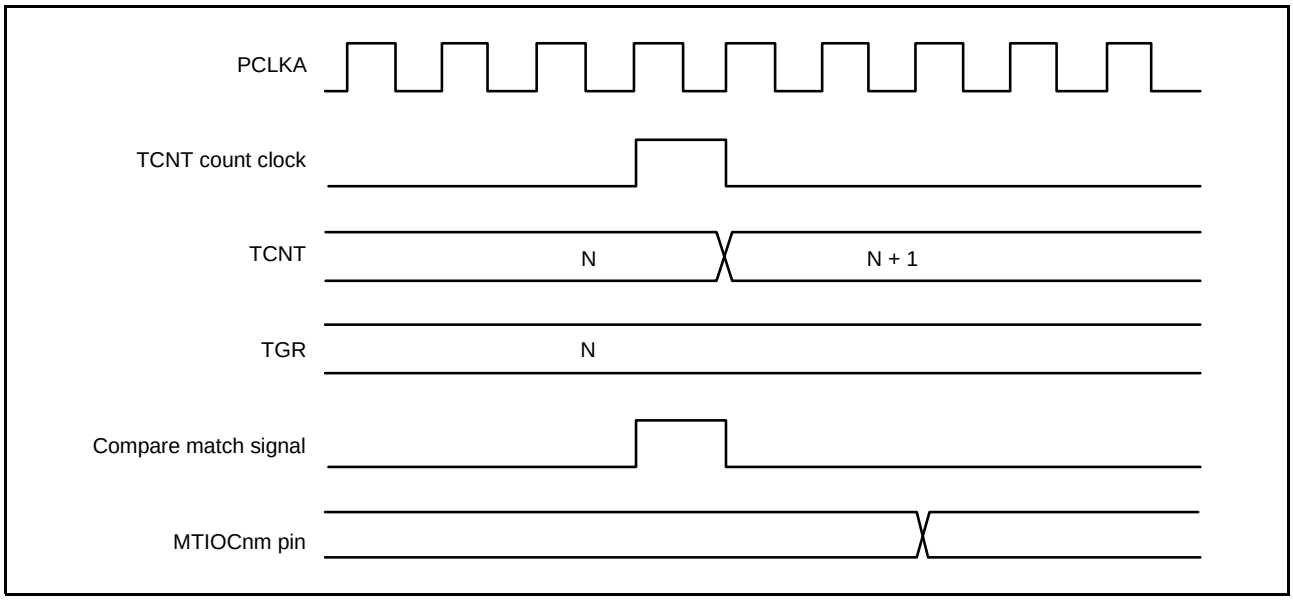


Figure 22.116 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

(3) Input Capture Signal Timing

Figure 22.117 shows the input capture signal timing.

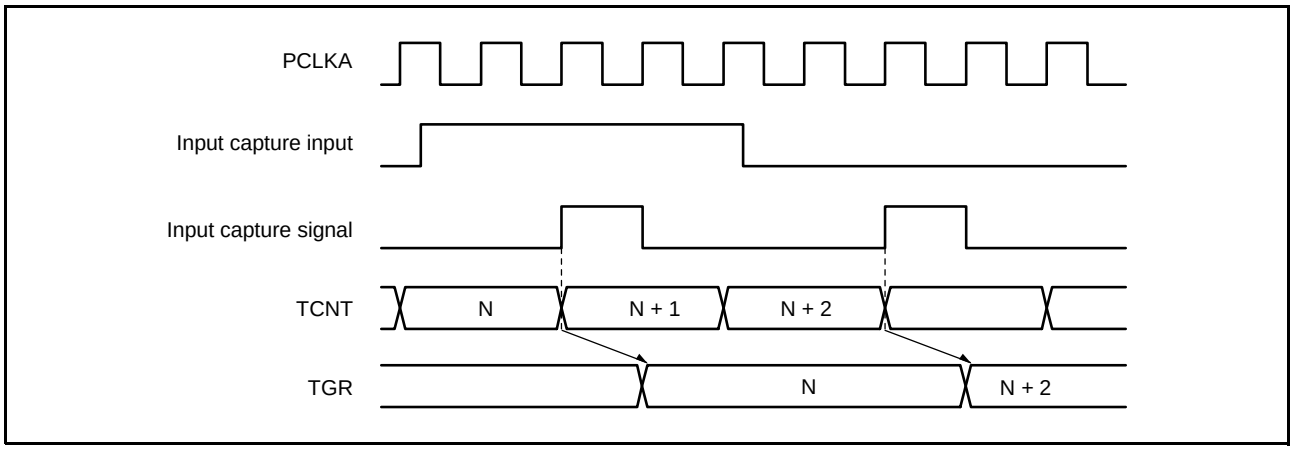


Figure 22.117 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

FFigure 22.118 and Figure 22.119 show the timing when counter clearing on compare match is specified, and Figure 22.120 shows the timing when counter clearing on input capture is specified.

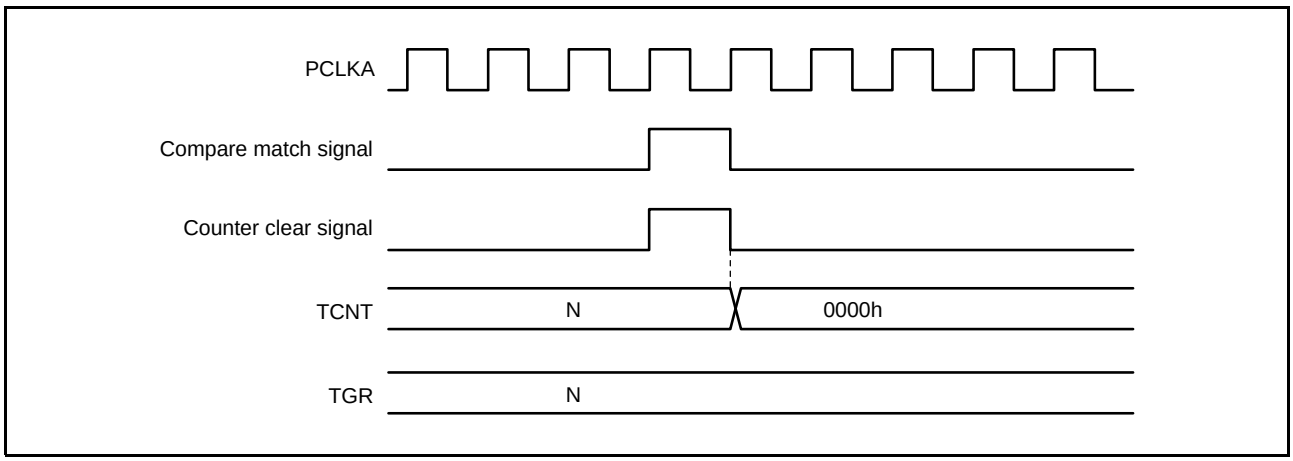


Figure 22.118 Counter Clear Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

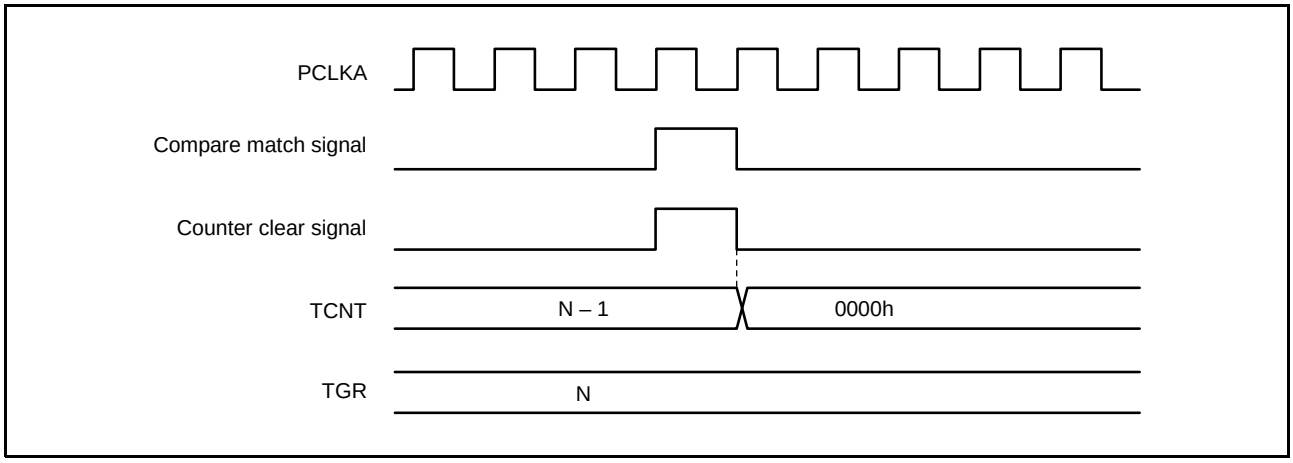


Figure 22.119 Counter Clear Timing (Compare Match) (MTU5)

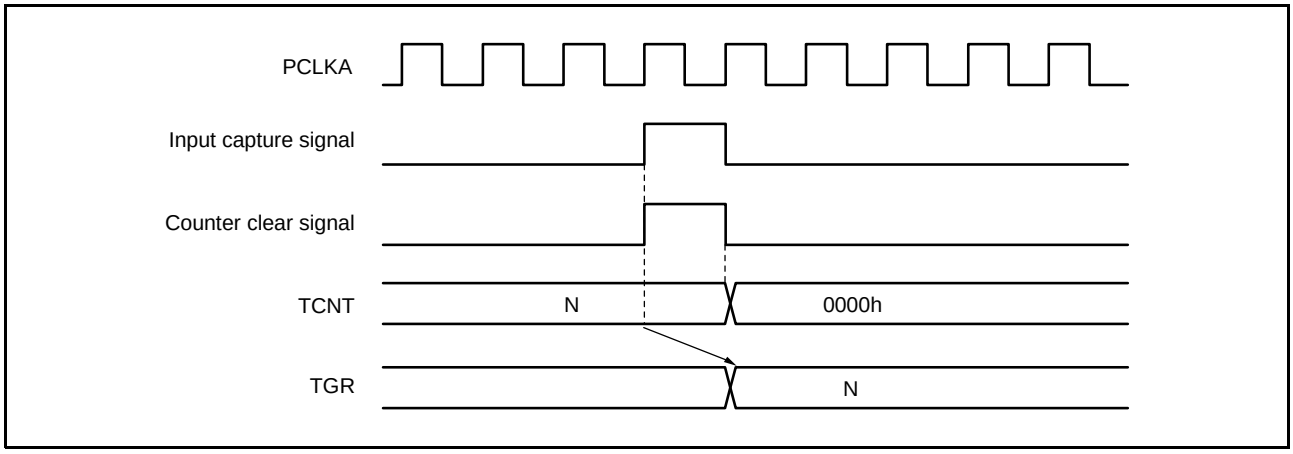


Figure 22.120 Counter Clear Timing (Input Capture) (MTU0 to MTU8)

(5) Buffer Operation Timing

Figure 22.121 to Figure 22.123 show the timing in buffer operation.

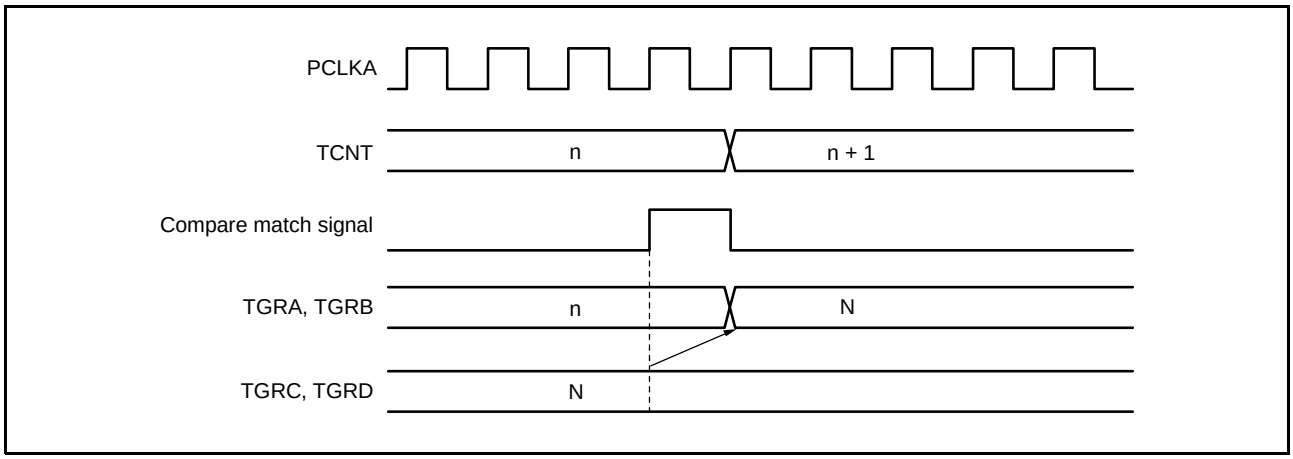


Figure 22.121 Buffer Operation Timing (Compare Match)

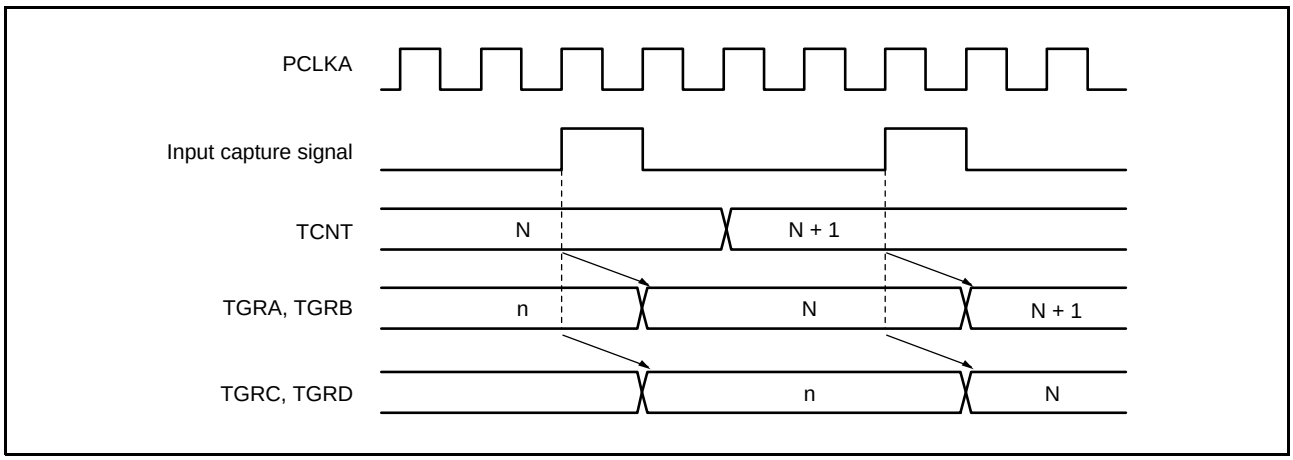


Figure 22.122 Buffer Operation Timing (Input Capture)

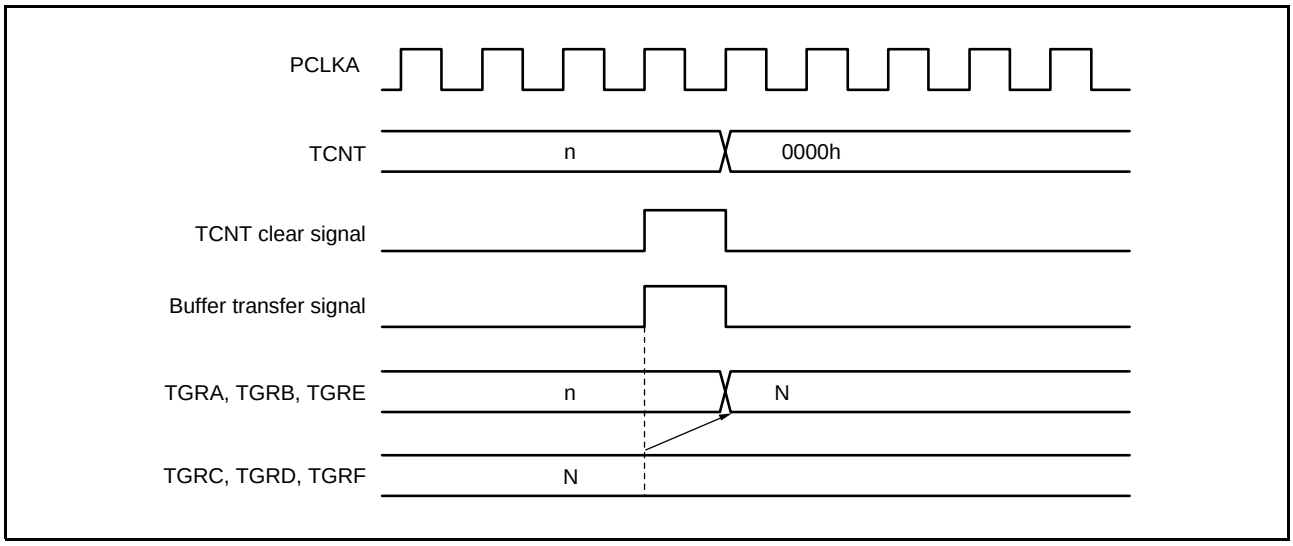


Figure 22.123 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 22.124 to Figure 22.126 show the buffer transfer timing in complementary PWM mode.

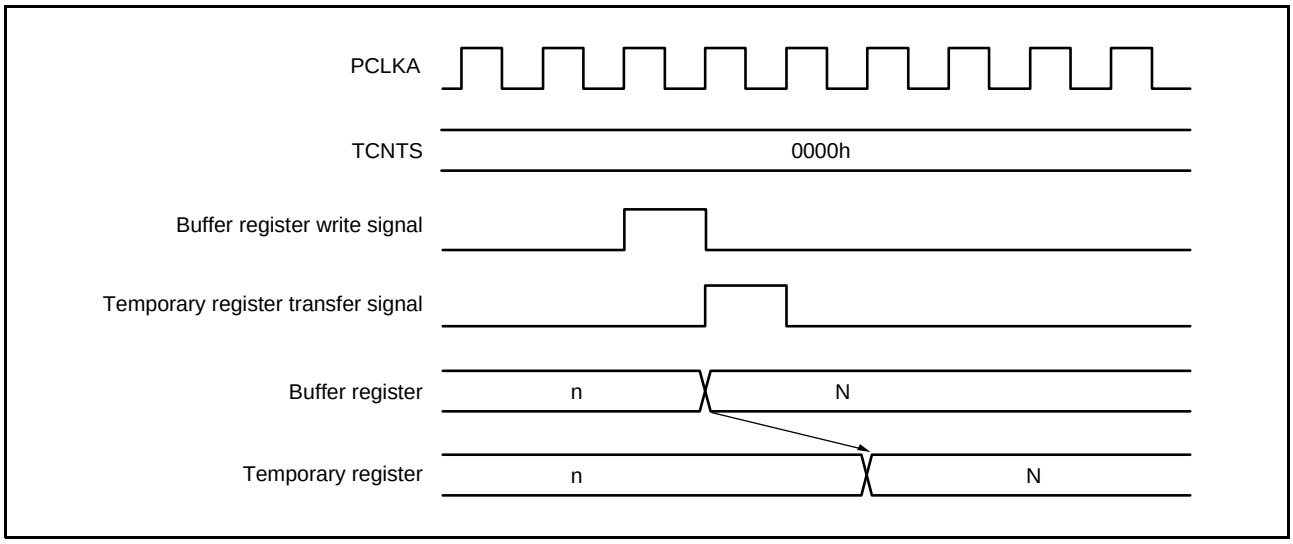


Figure 22.124 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

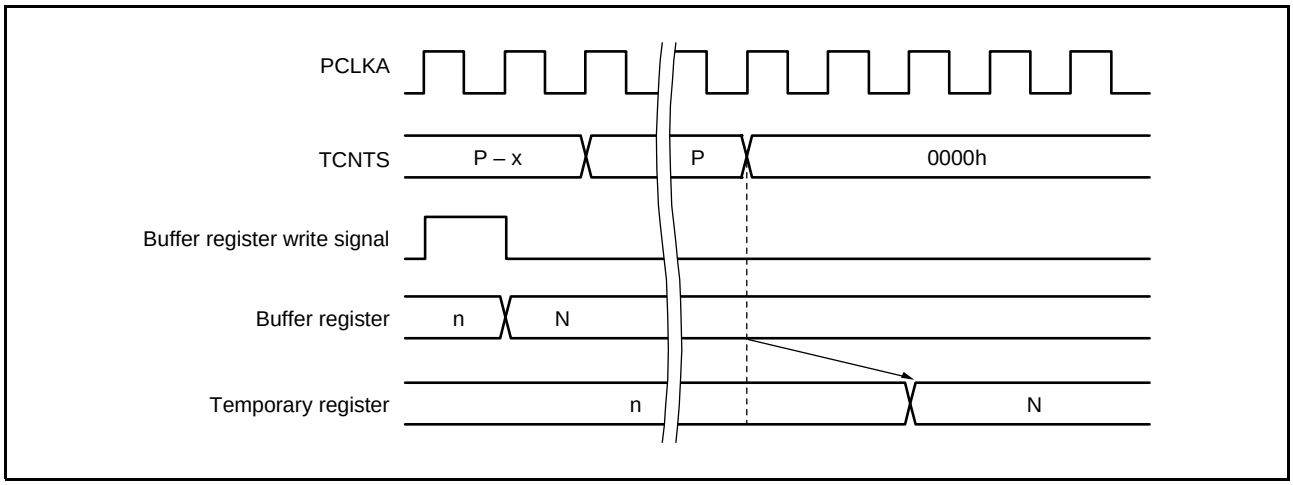


Figure 22.125 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

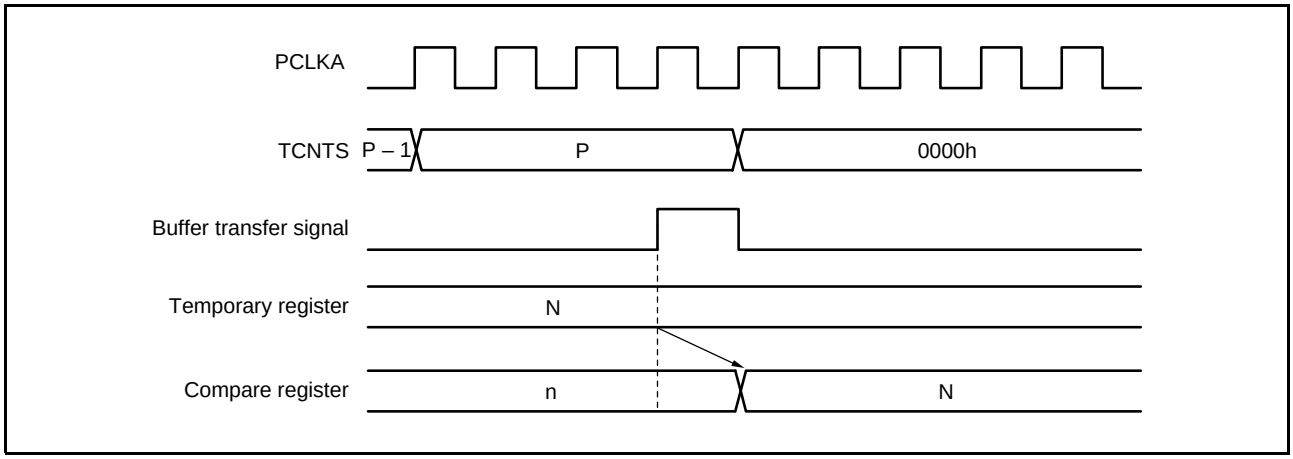


Figure 22.126 Transfer Timing from Temporary Register to Compare Register

22.5.2 Interrupt Signal Timing

(1) TGI Interrupt Timing by Compare Match

Figure 22.127 and Figure 22.128 show the TGI interrupt request signal timing when a compare match occurs.

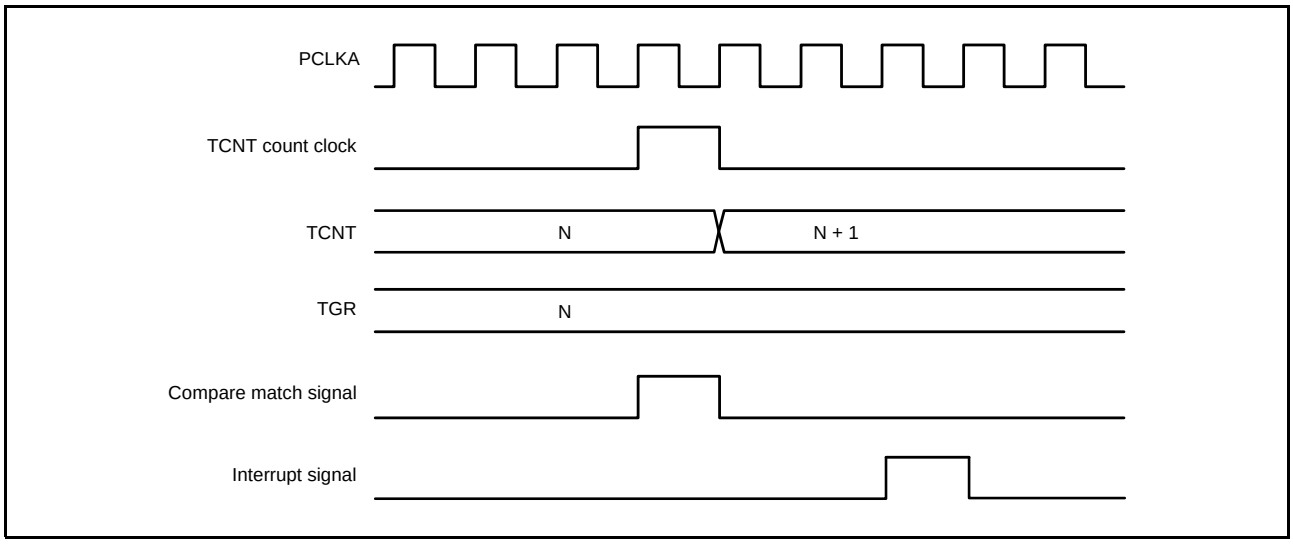


Figure 22.127 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

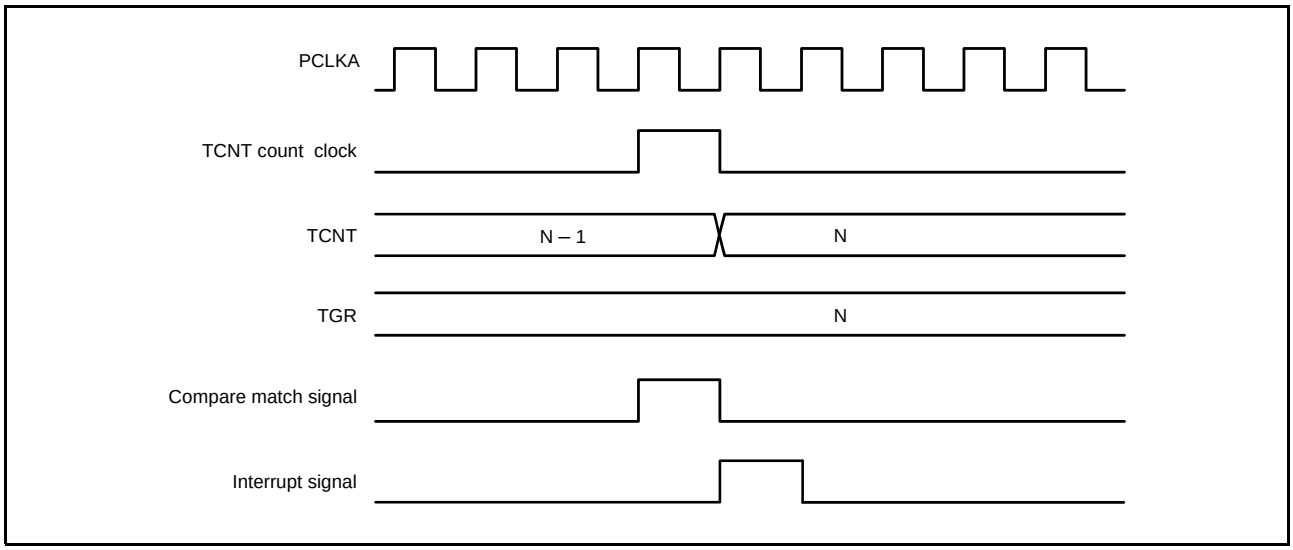


Figure 22.128 TGI Interrupt Timing (Compare Match) (MTU5)

(2) TGI Interrupt Timing by Input Capture

Figure 22.129 and Figure 22.130 show the TGI interrupt request signal timing when an input capture occurs.

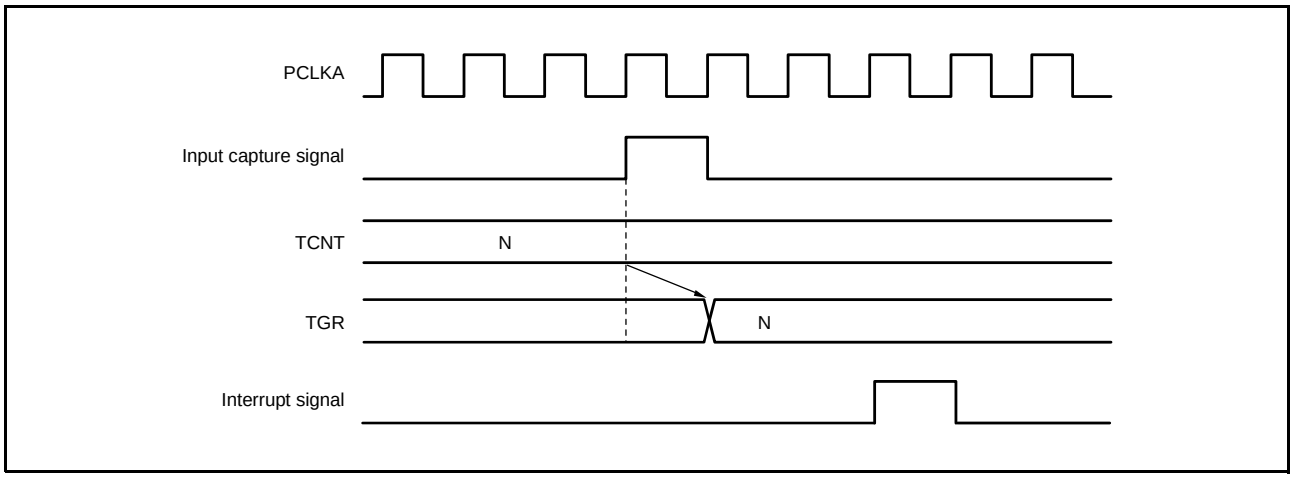


Figure 22.129 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4 and MTU6 to MTU8)

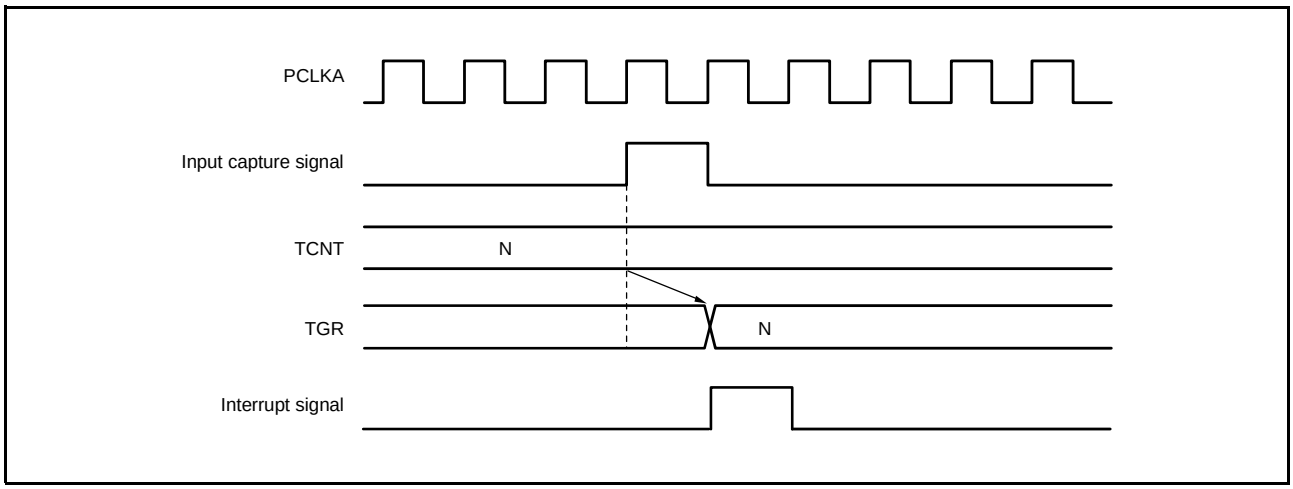


Figure 22.130 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 22.131 shows the TCIV interrupt request signal timing when an overflow is generated.
Figure 22.132 shows the TCIU interrupt request signal timing when an underflow is generated.

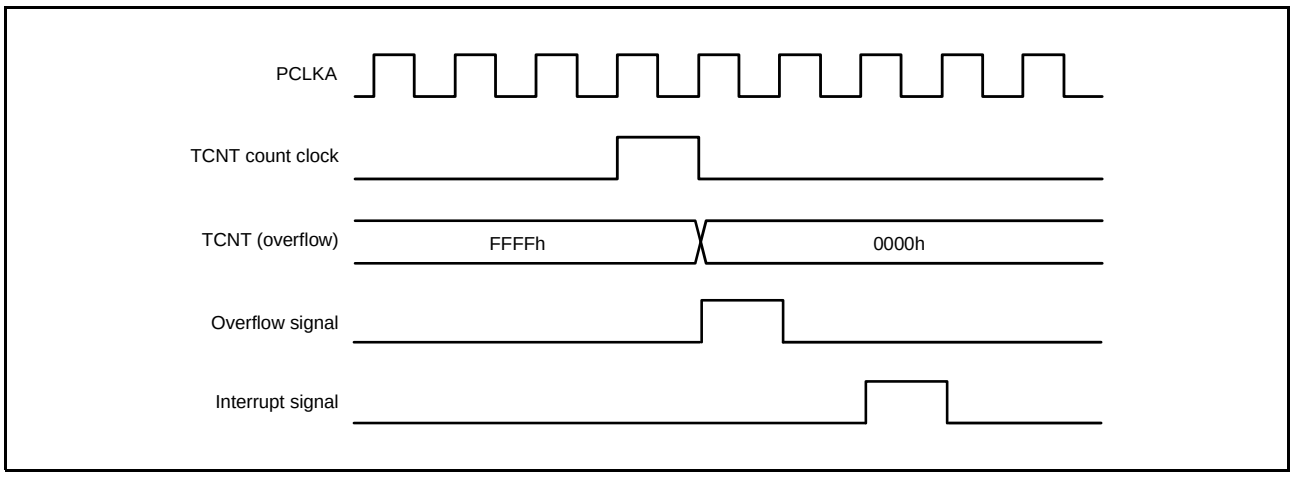


Figure 22.131 TCIV Interrupt Timing

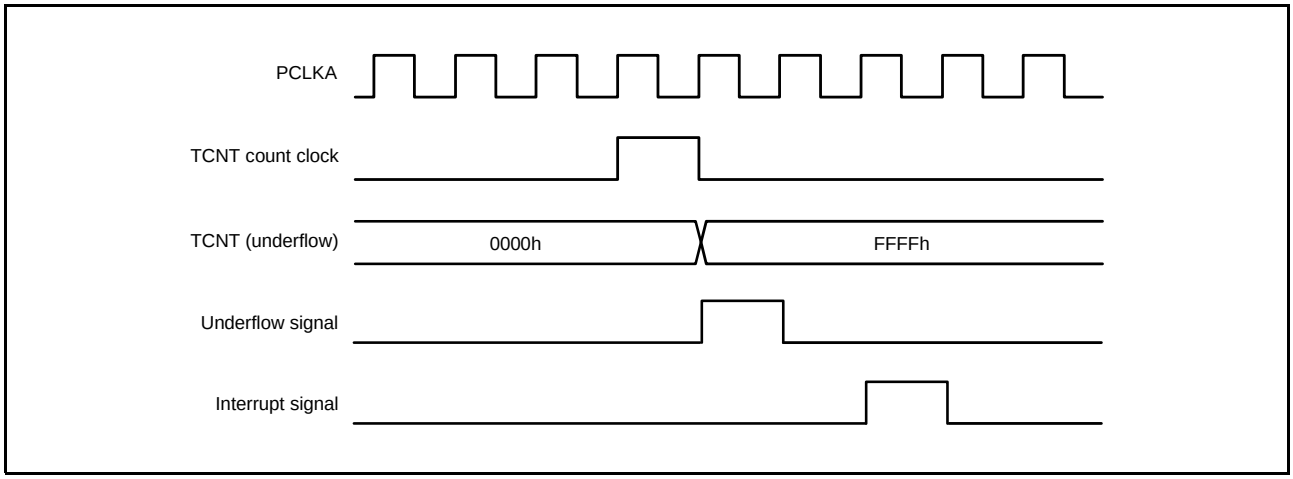


Figure 22.132 TCIU Interrupt Timing

22.6 Usage Notes

22.6.1 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.

22.6.2 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 22.133 shows the input clock conditions in phase counting mode.

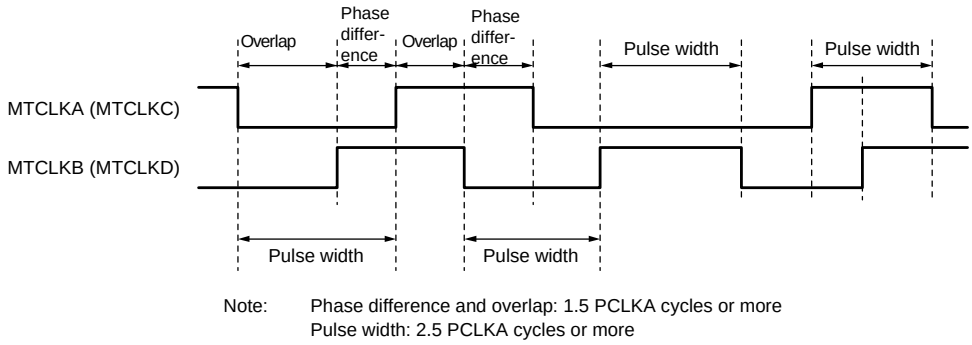


Figure 22.133 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

22.6.3 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

22.6.4 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 22.134 shows the timing in this case.

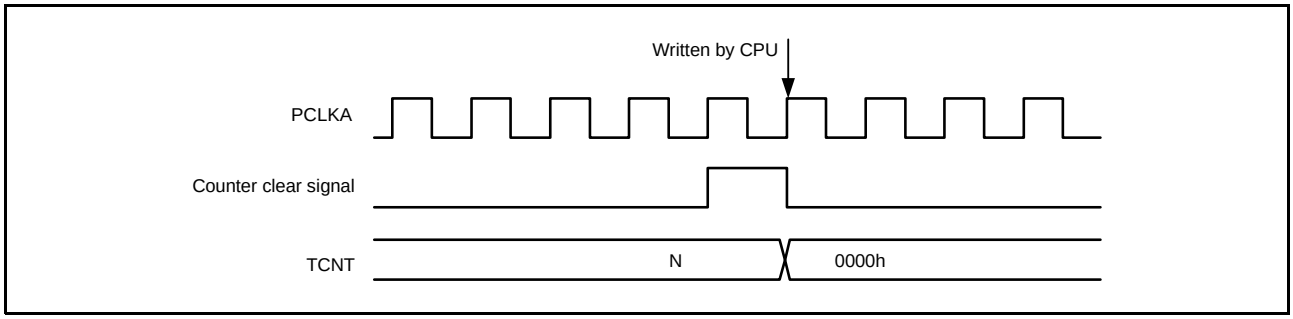


Figure 22.134 Contention between TCNT Write and Clear Operations

22.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.
Figure 22.135 shows the timing in this case.

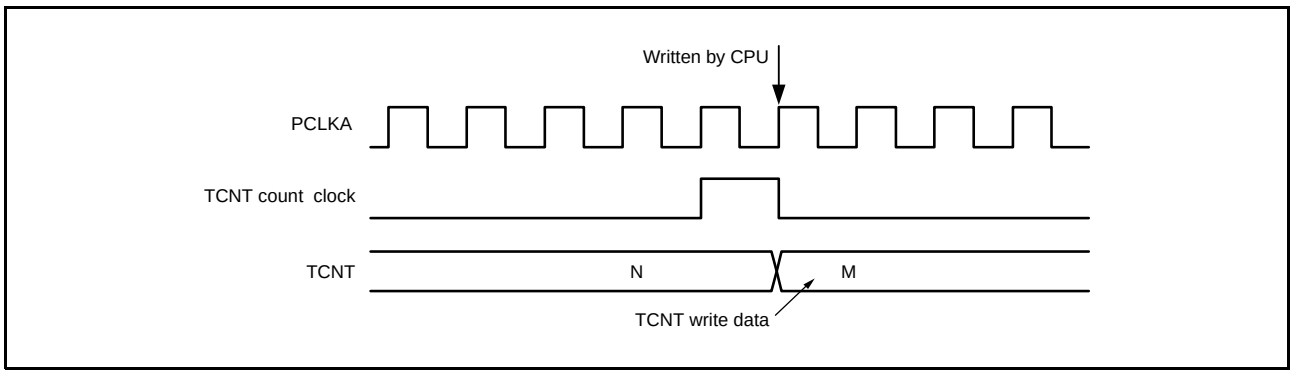


Figure 22.135 Contention between TCNT Write and Increment Operations

22.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.
Figure 22.136 shows the timing in this case.

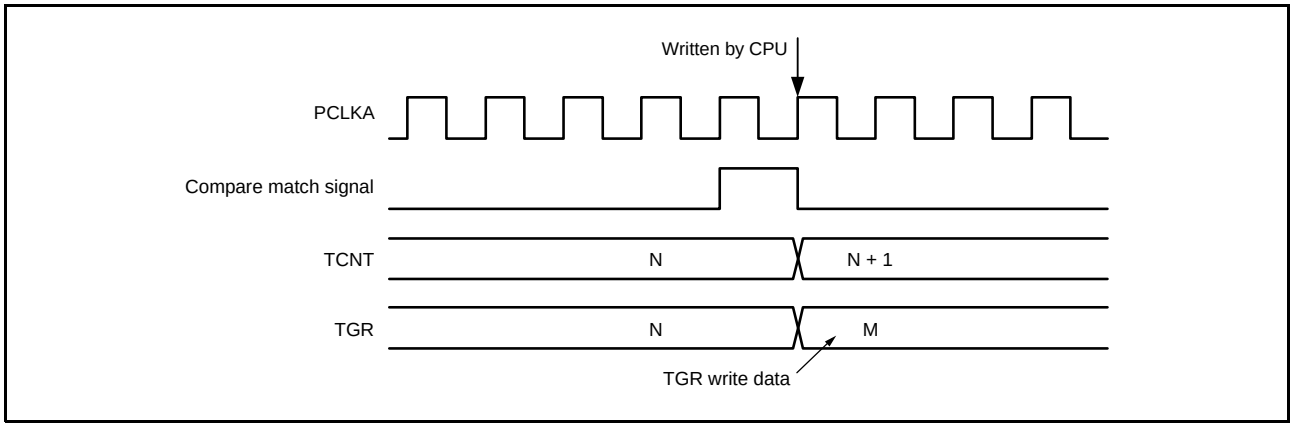


Figure 22.136 Contention between TGR Write Operation and Compare Match

22.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 22.137 shows the timing in this case.

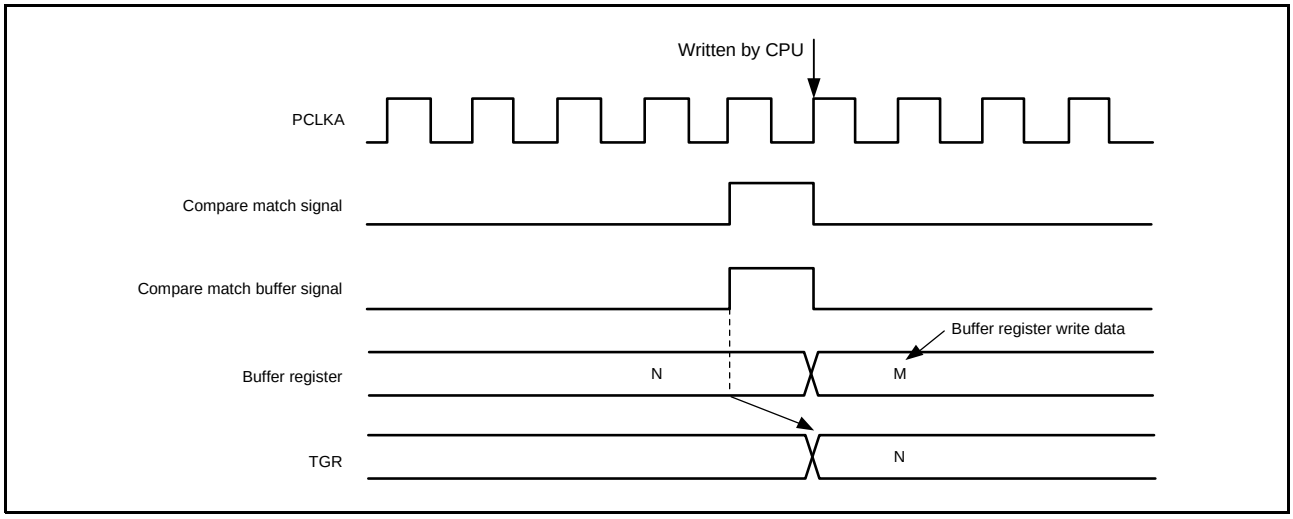


Figure 22.137 Contention between Buffer Register Write Operation and Compare Match

22.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the 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 22.138 shows the timing in this case.

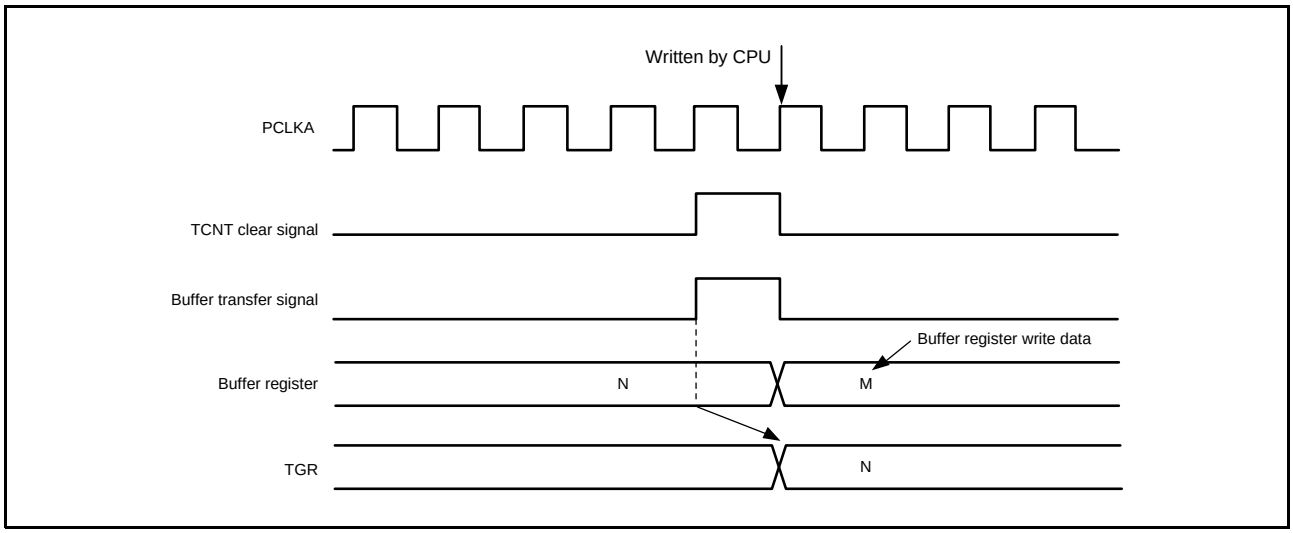


Figure 22.138 Contention between Buffer Register Write and TCNT Clear Operations

22.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 22.139 shows the timing in this case.

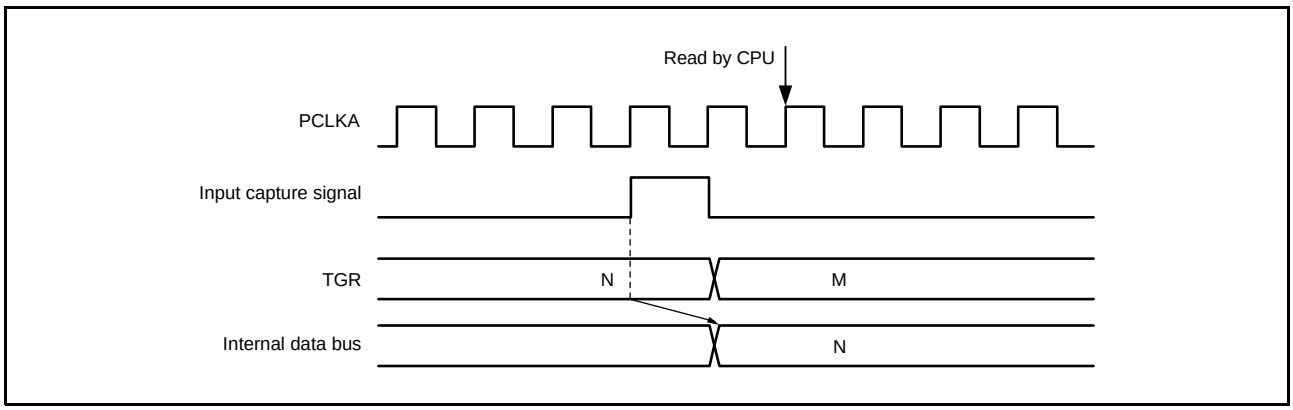


Figure 22.139 Contention between TGR Read Operation and Input Capture (MTU0 to MTU8)

22.6.10 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 22.140 and Figure 22.141 show the timing in this case.

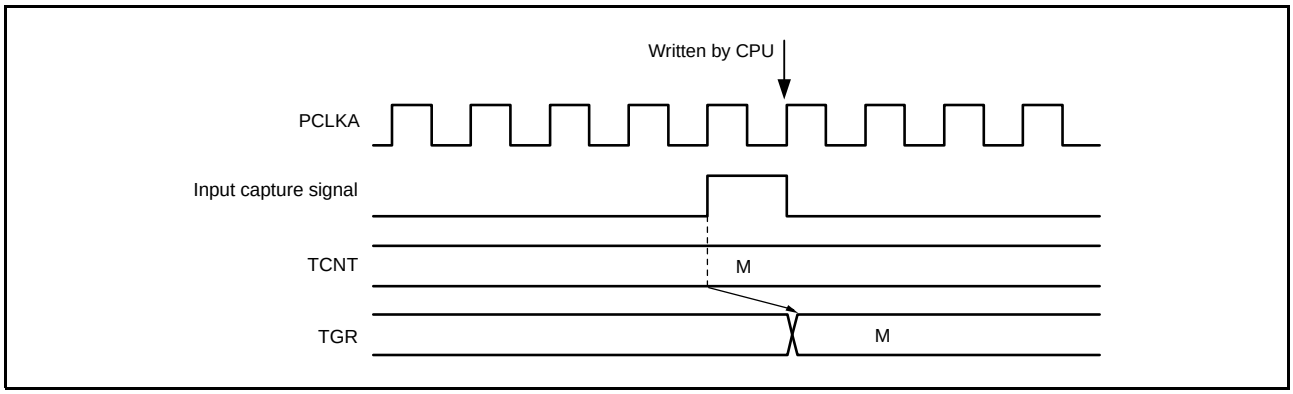


Figure 22.140 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4 and MTU6 to MTU8)

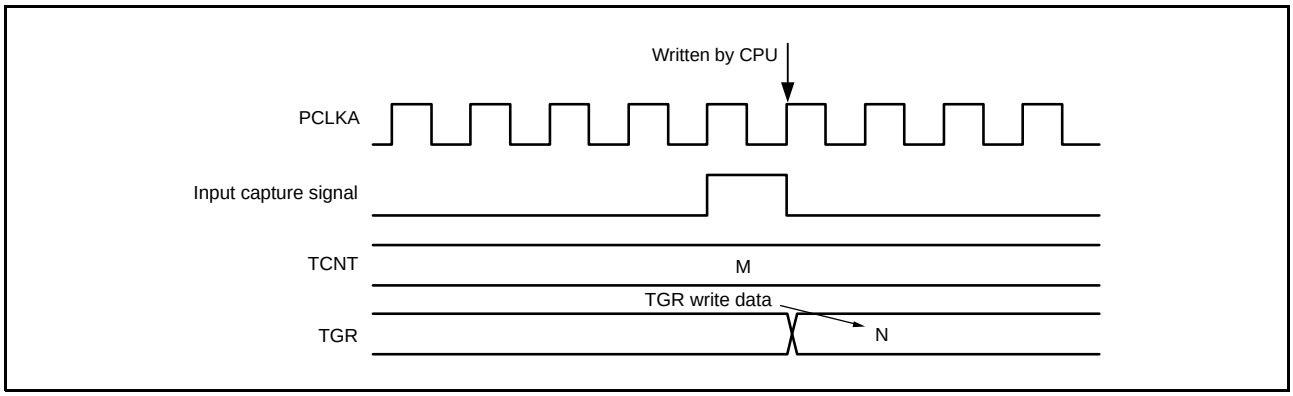


Figure 22.141 Contention between TGR Write Operation and Input Capture (MTU5)

22.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 22.142 shows the timing in this case.

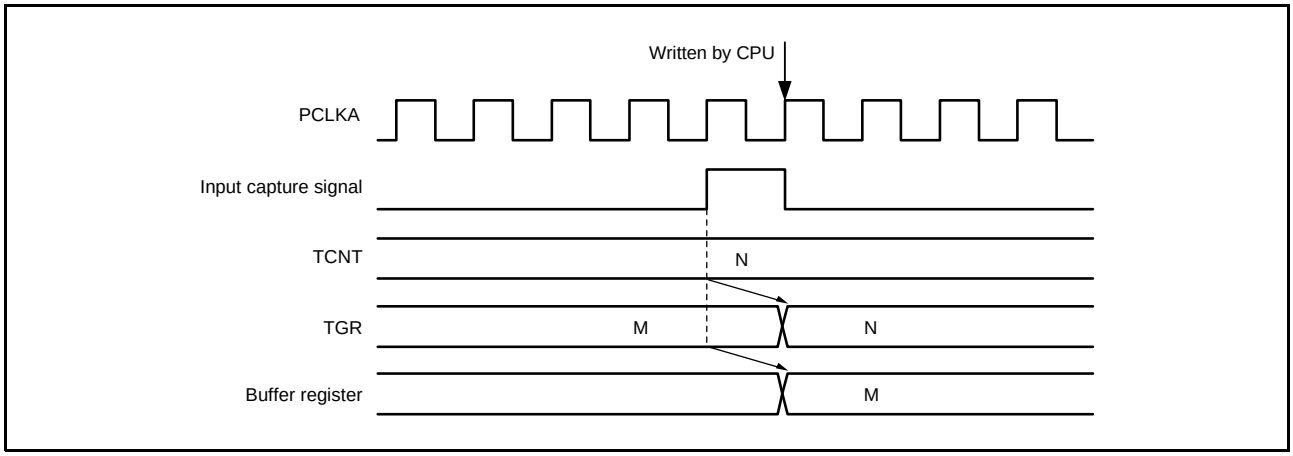


Figure 22.142 Contention between Buffer Register Write Operation and Input Capture

22.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write 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 22.143 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

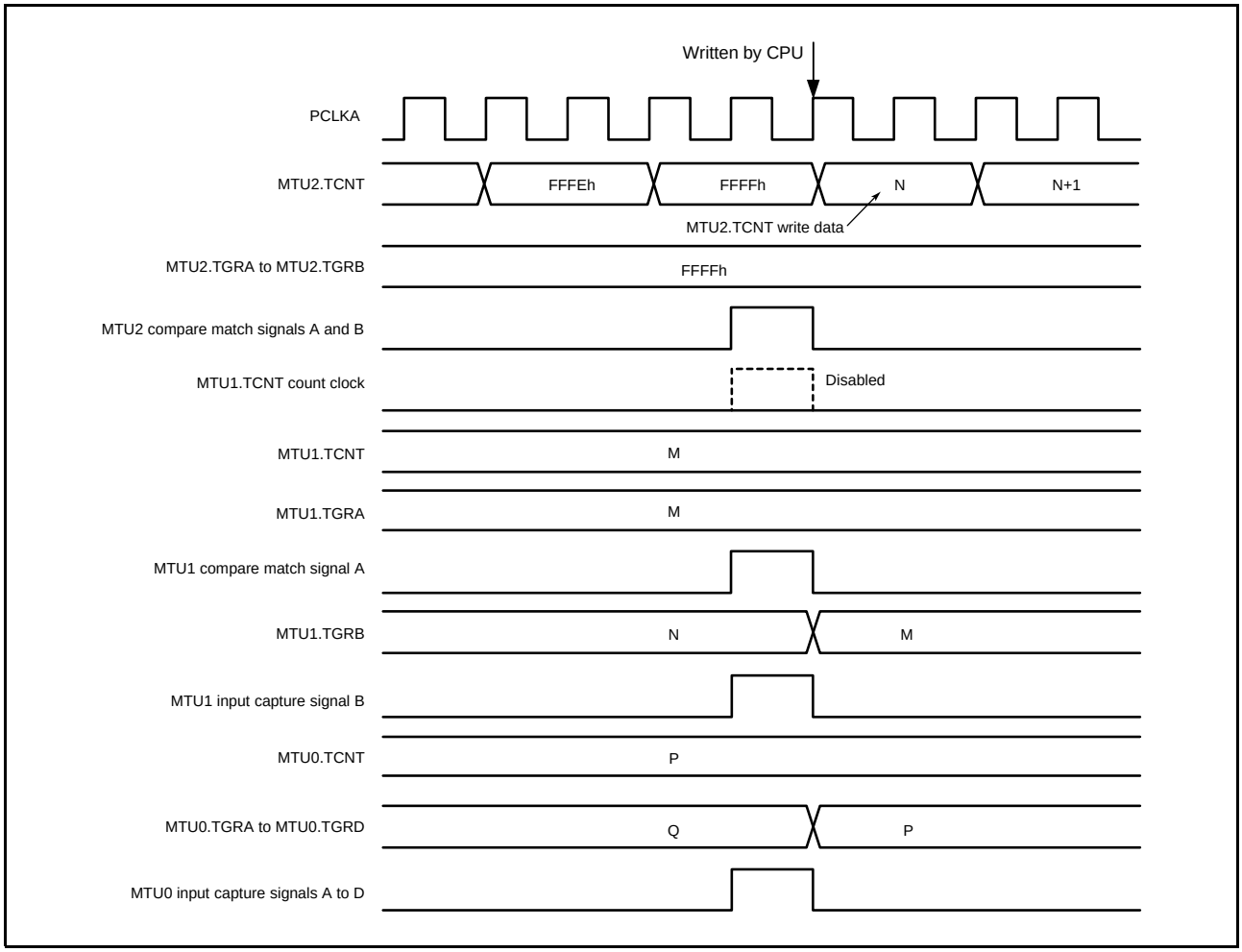


Figure 22.143 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

22.6.13 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 22.144 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).

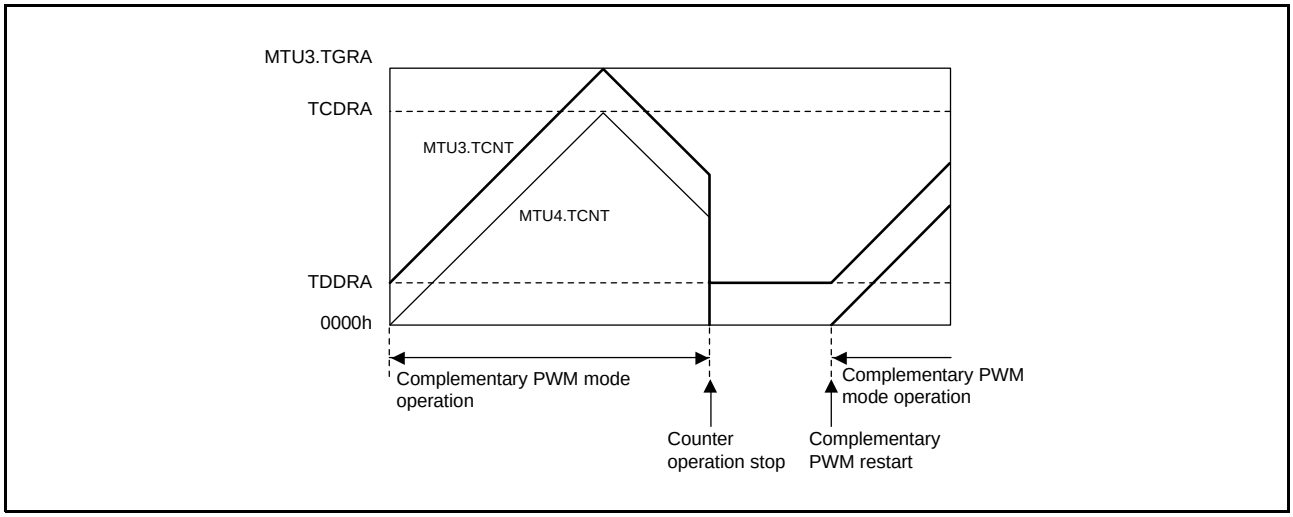


Figure 22.144 Counter Value When Stopped in Complementary PWM Mode

22.6.14 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).

22.6.15 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 22.145 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.

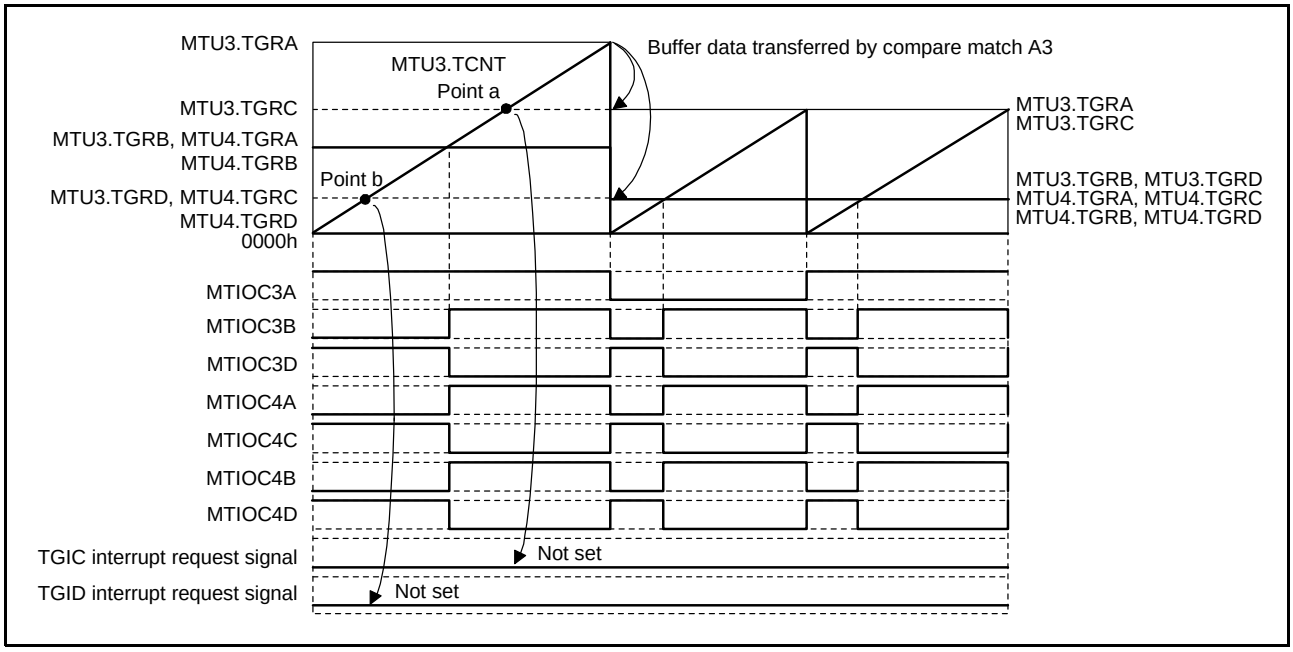


Figure 22.145 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

22.6.16 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 22.146 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.

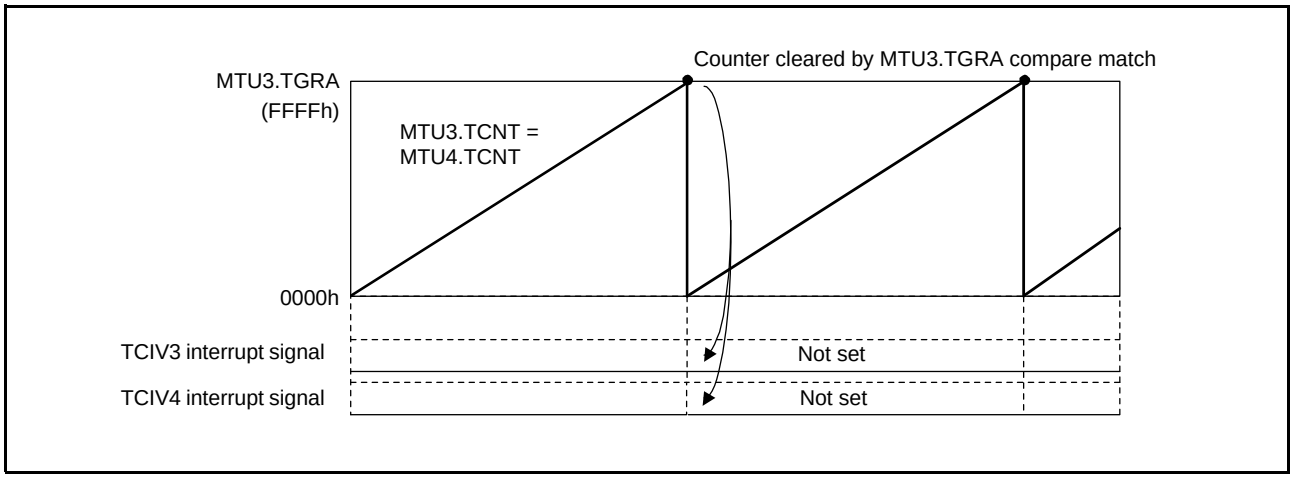


Figure 22.146 Overflow in Reset-Synchronized PWM Mode

22.6.17 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 22.147 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

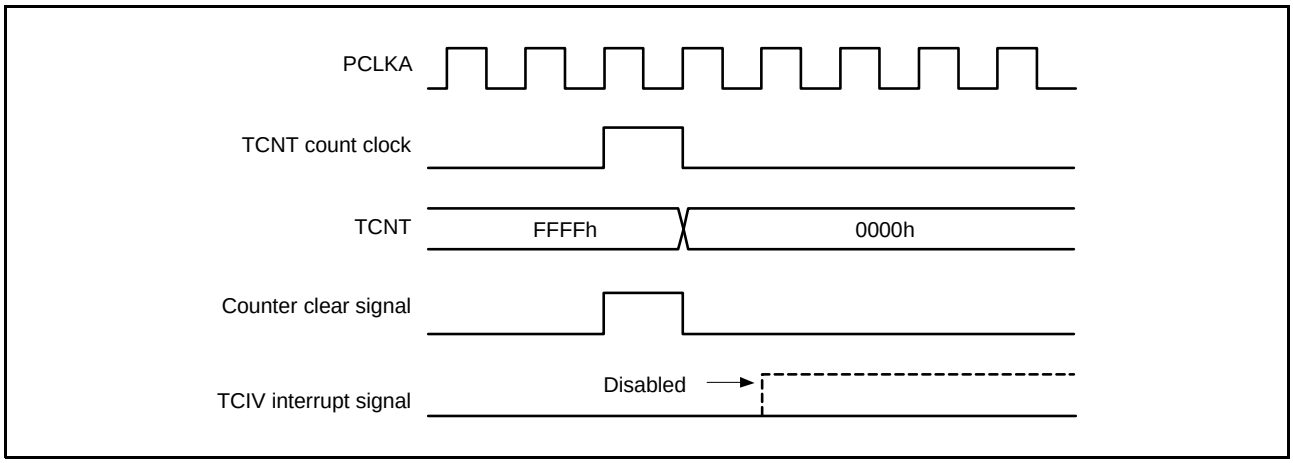


Figure 22.147 Contention between Overflow and Counter Clearing

22.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated.
Figure 22.148 shows the operation timing when there is contention between TCNT write operation and overflow.

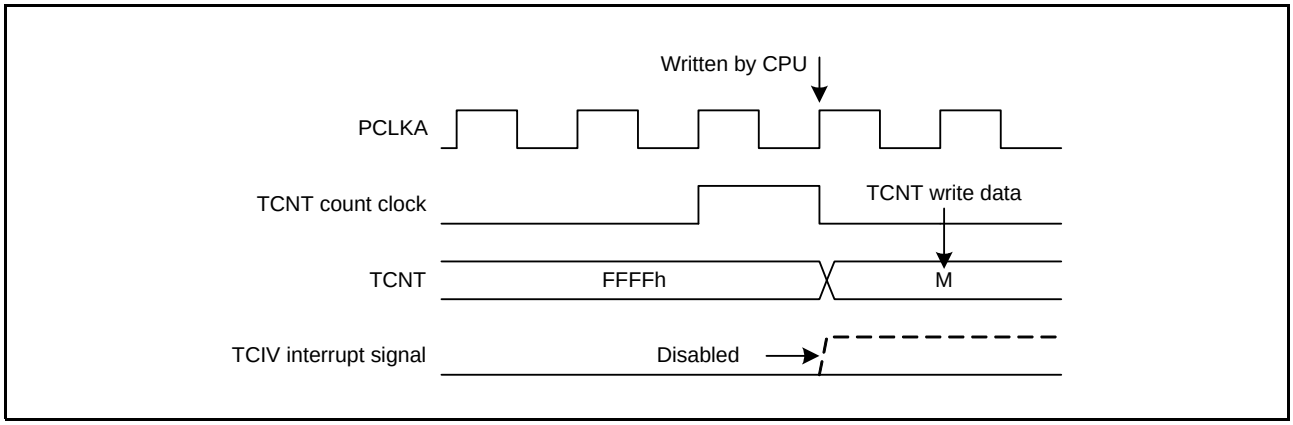


Figure 22.148 Contention between TCNT Write Operation and Overflow

22.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (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.

22.6.20 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.

22.6.21 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 22.2.11, Timer Input Capture Control Register (TICCR).

22.6.22 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).

22.6.23 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 23, Port Output Enable 3 (POE3a).

22.6.24 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.

22.6.25 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 22.149, 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 22.150, 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).

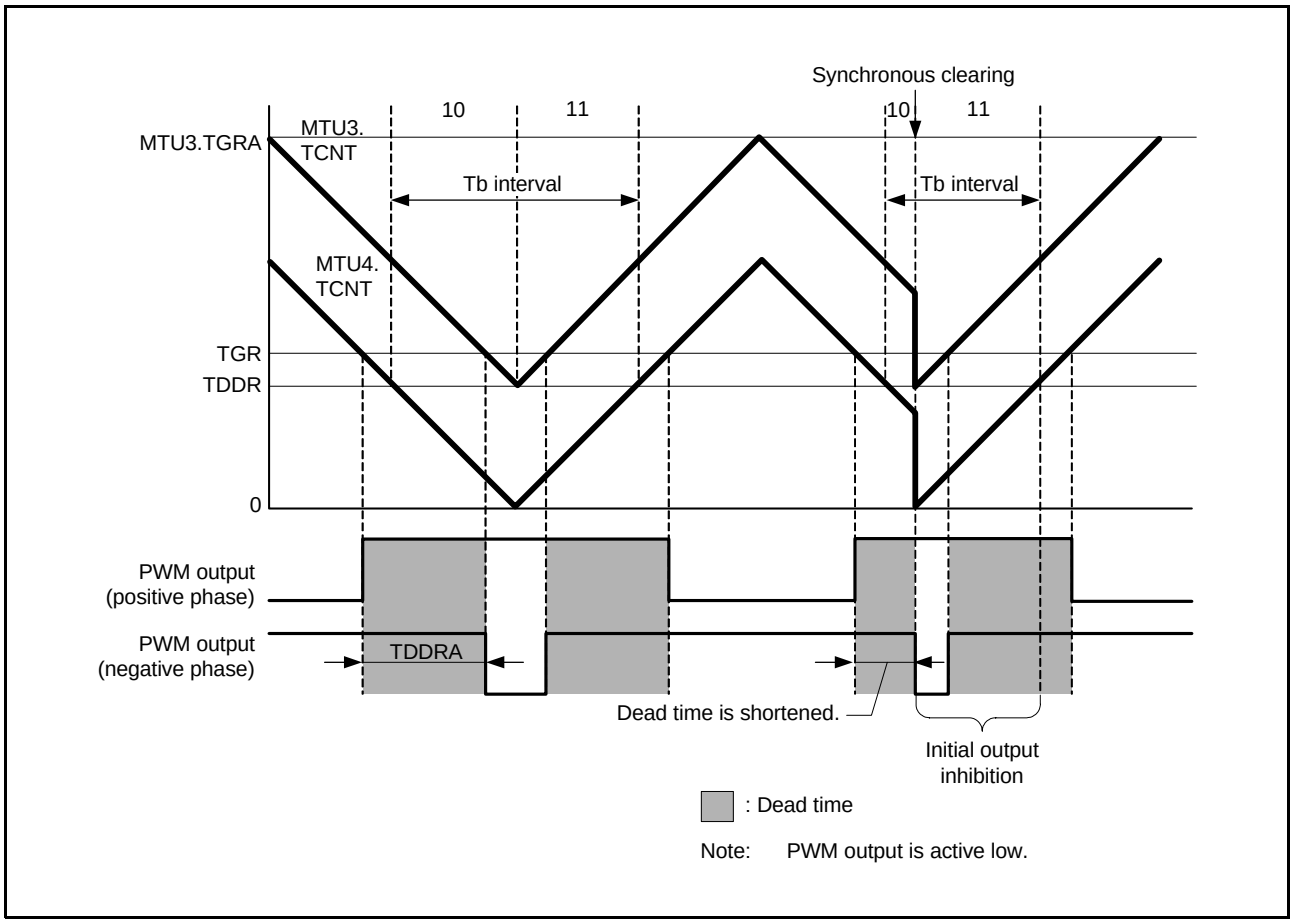


Figure 22.149 Example of Synchronous Clearing (When Condition 1 Applies)

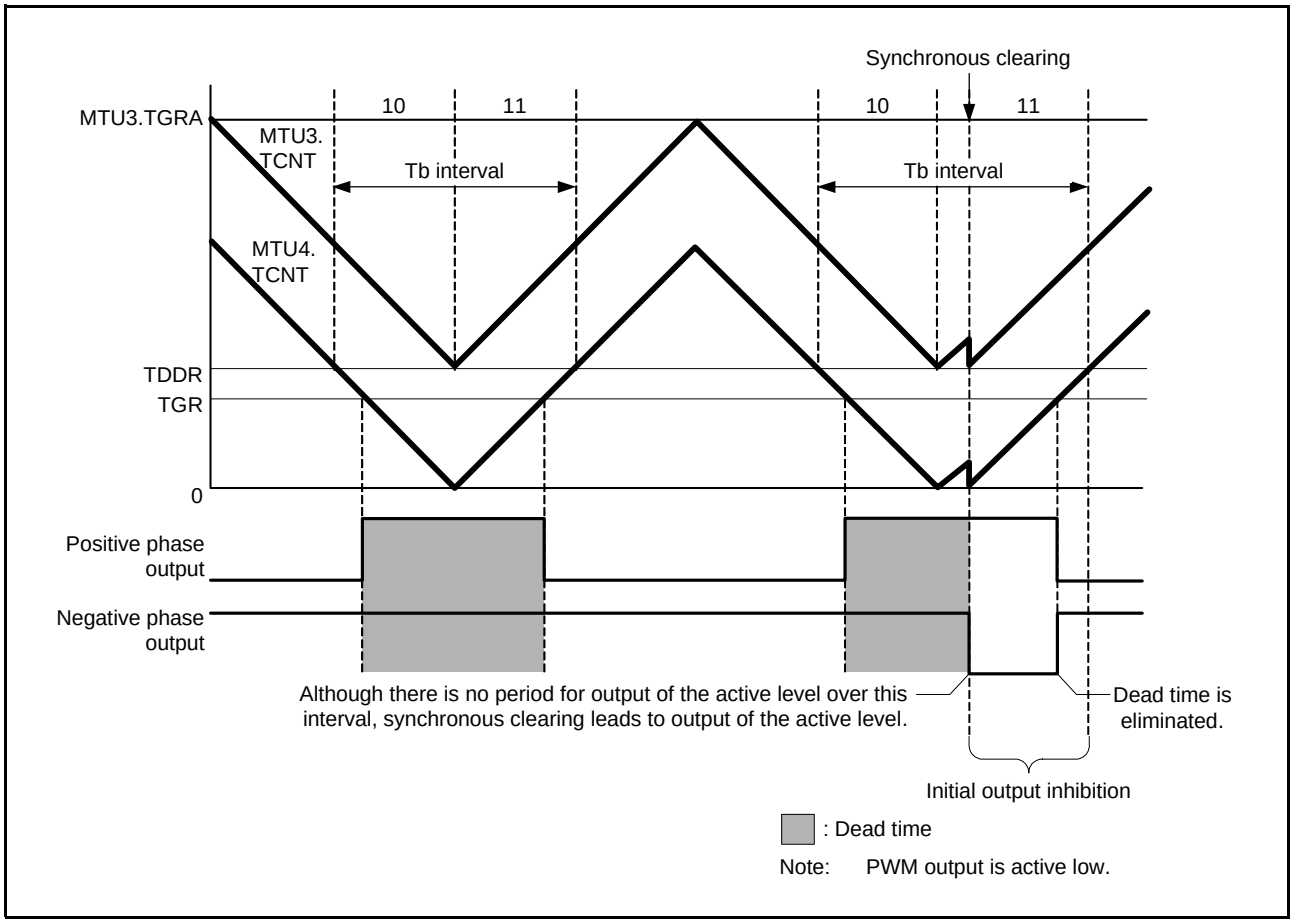


Figure 22.150 Example of Synchronous Clearing (When Condition 2 Applies)

22.6.26 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).

22.6.27 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 22.151 shows the timing for continuous output of the interrupt signal in response to a compare match.

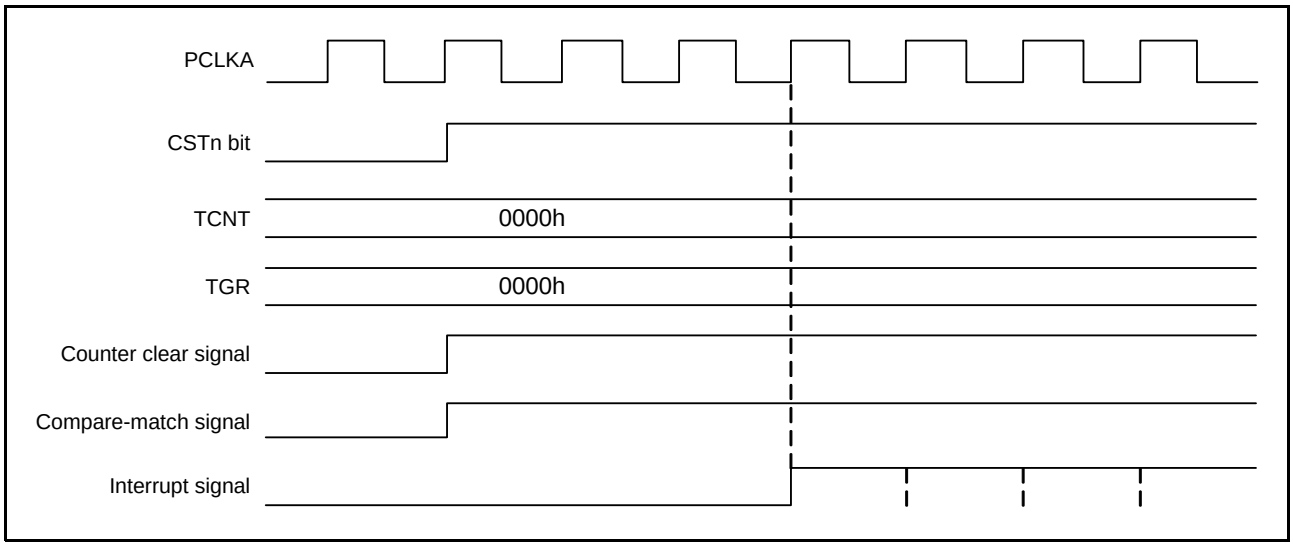


Figure 22.151 Continuous Output of Interrupt Signal in Response to a Compare Match

22.6.28 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 22.152.
- 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 22.153.
- 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).

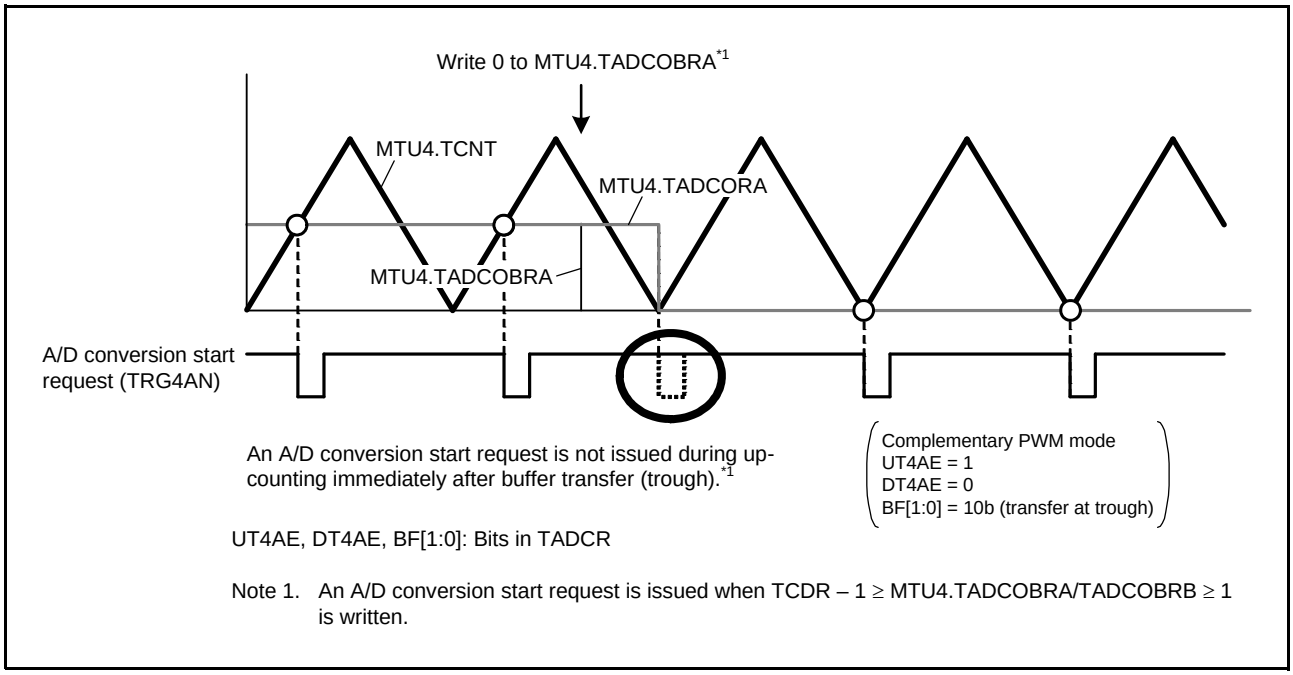


Figure 22.152 A/D Conversion Start Request When 0 is Written to MTU4.TADCOBRA

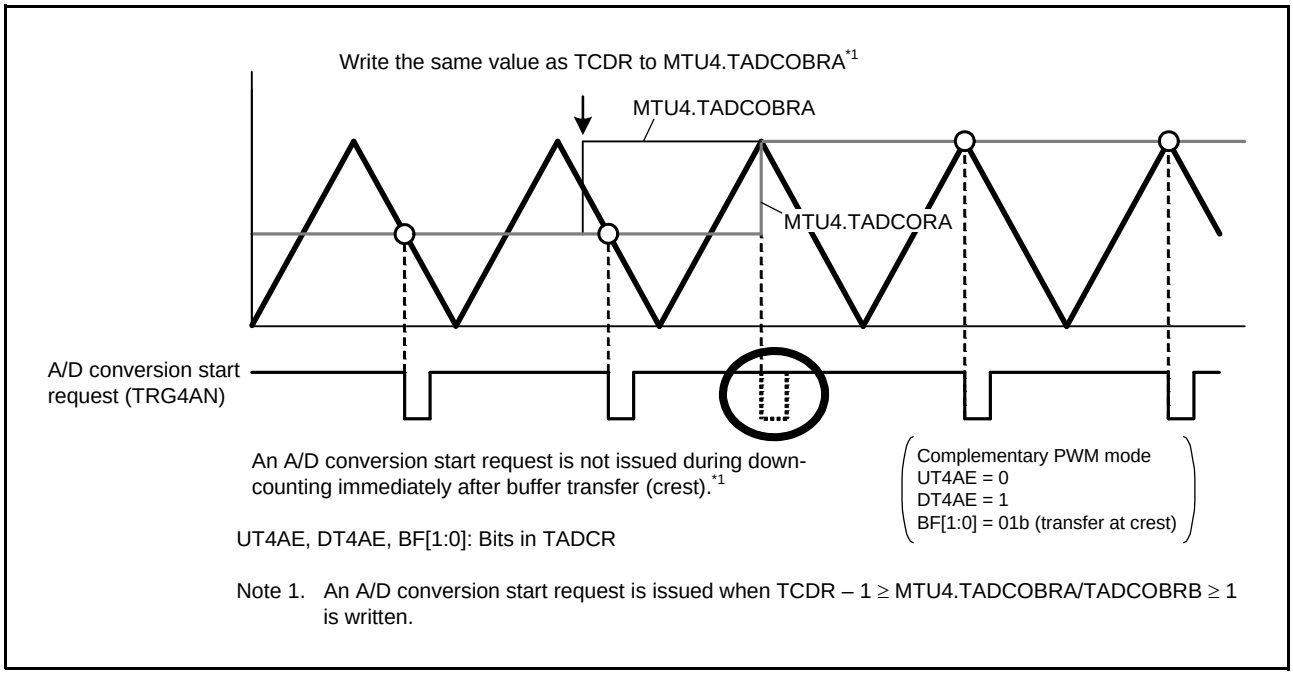


Figure 22.153 A/D Conversion Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

22.7 MTU Output Pin Initialization

22.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4 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.

22.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. 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 22.79.

Table 22.79 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

22.7.3 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 22.79. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 22.154 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

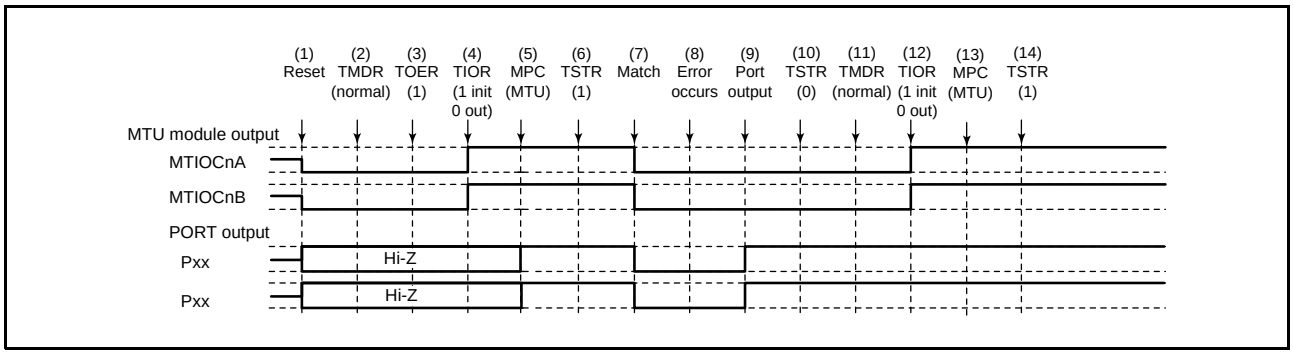


Figure 22.154 Error Occurrence in Normal Mode, Recovery in Normal Mode

(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 22.155 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

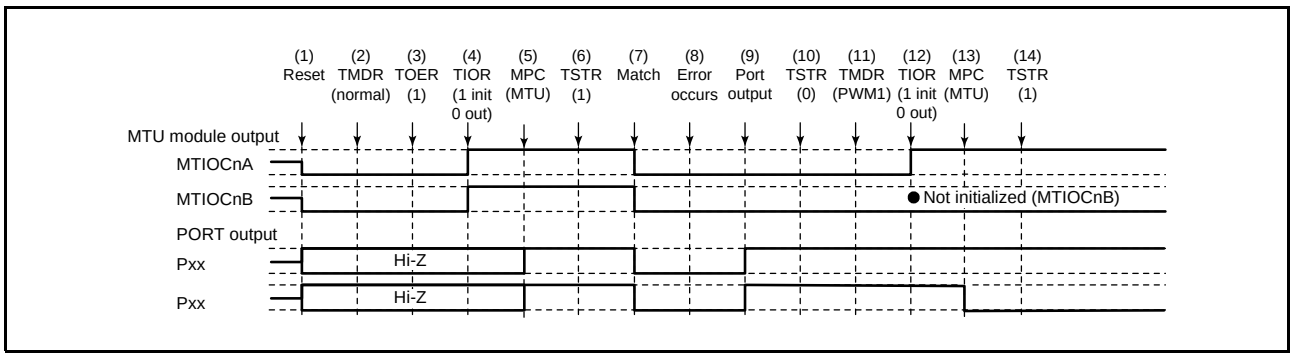


Figure 22.155 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.154.
(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 22.156 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

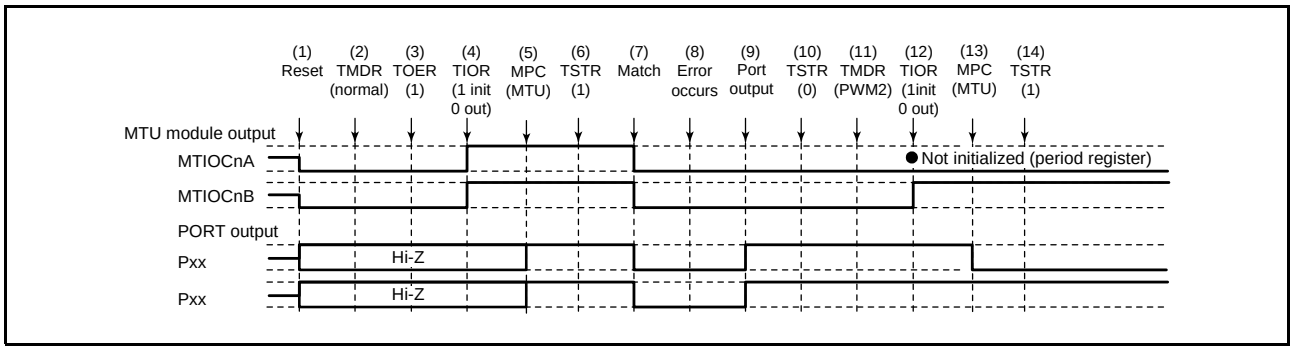


Figure 22.156 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 22.154.
(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 22.157 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

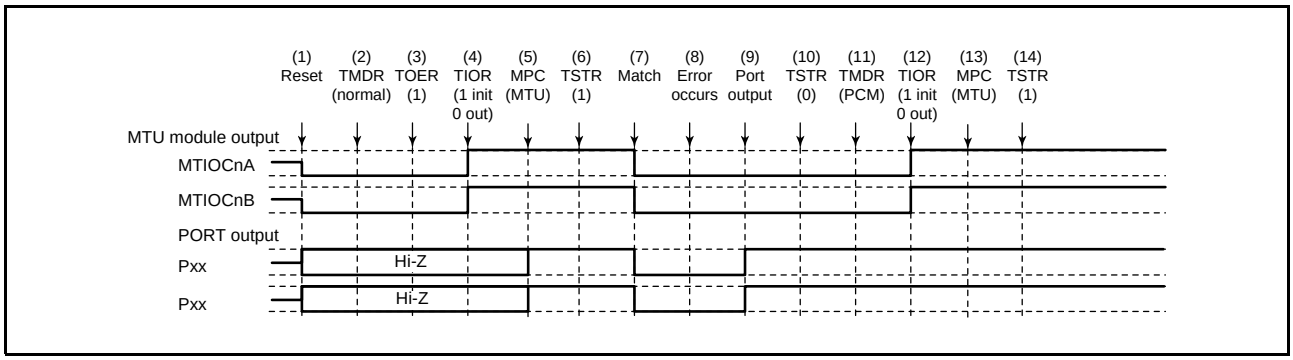


Figure 22.157 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 22.154.
(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 22.158 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

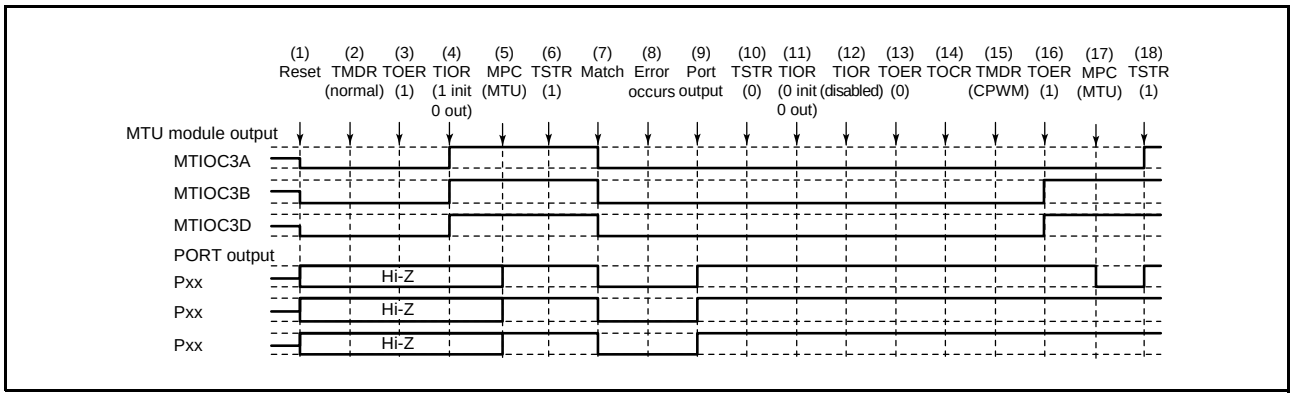


Figure 22.158 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 22.154.
(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 22.159 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

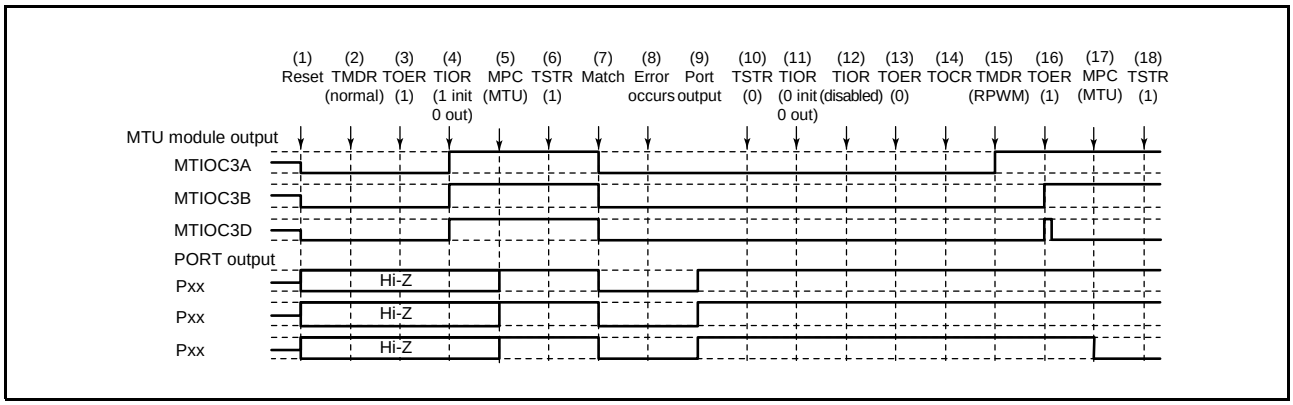


Figure 22.159 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 22.158.

(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 22.160 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

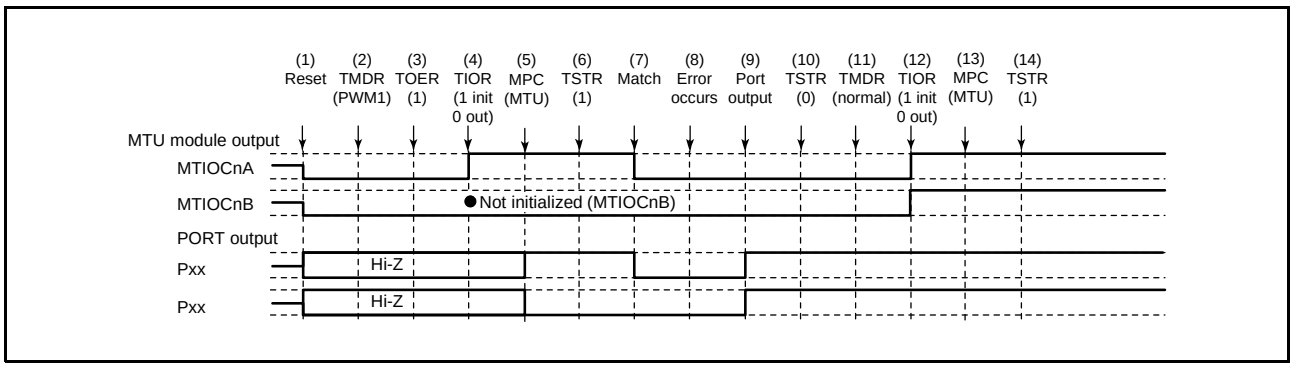


Figure 22.160 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(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 22.161 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

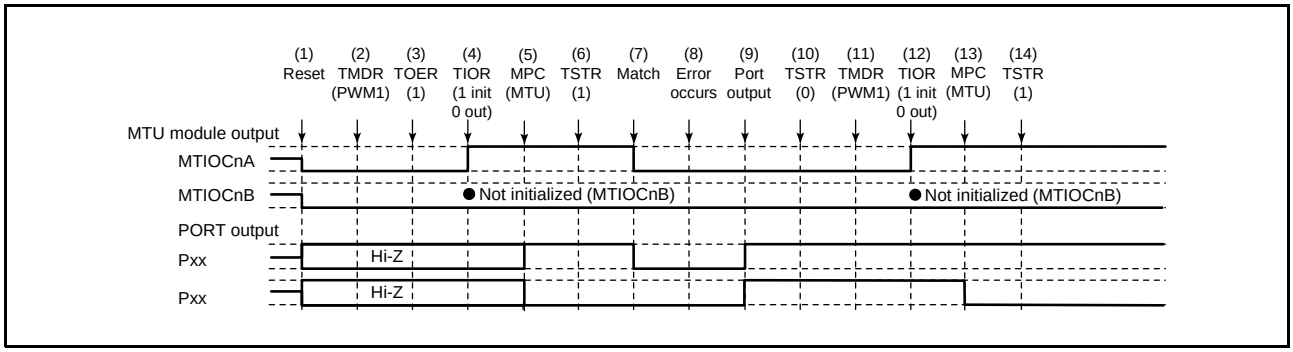


Figure 22.161 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.160.
(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 22.162 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

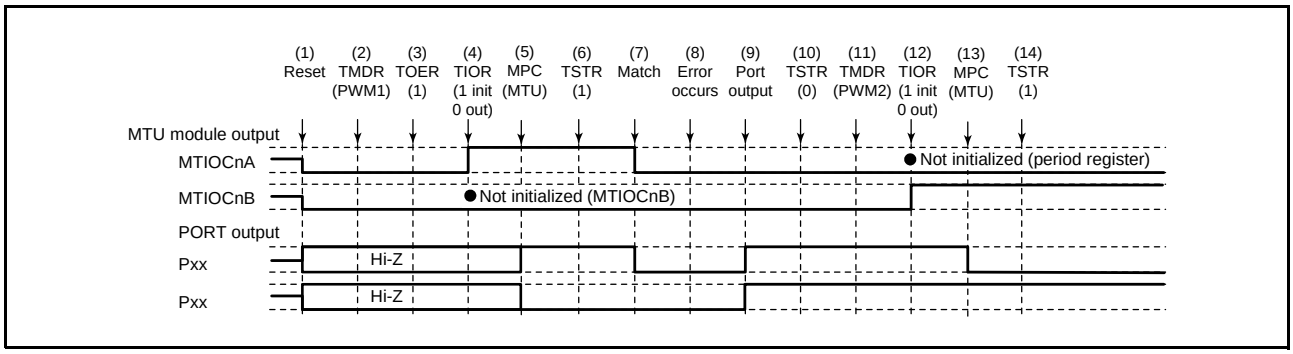


Figure 22.162 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 22.160.
(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 22.163 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

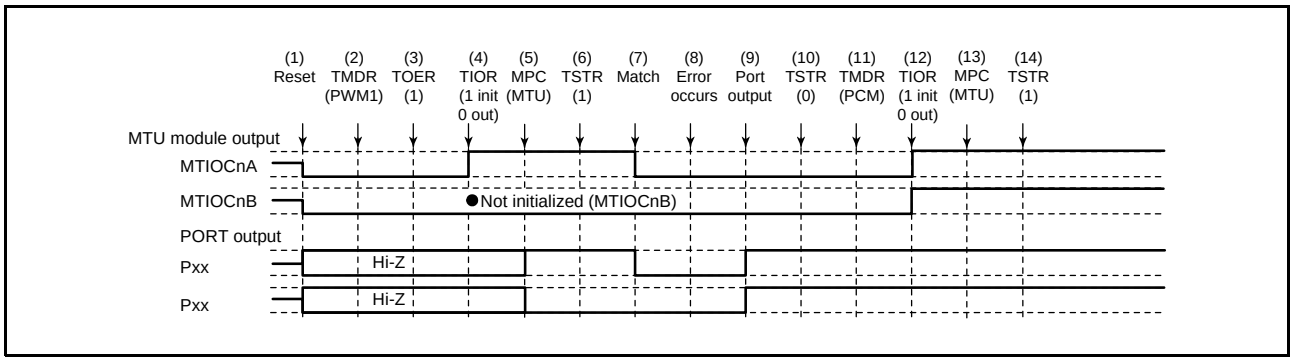


Figure 22.163 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 22.160.
(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 22.164 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

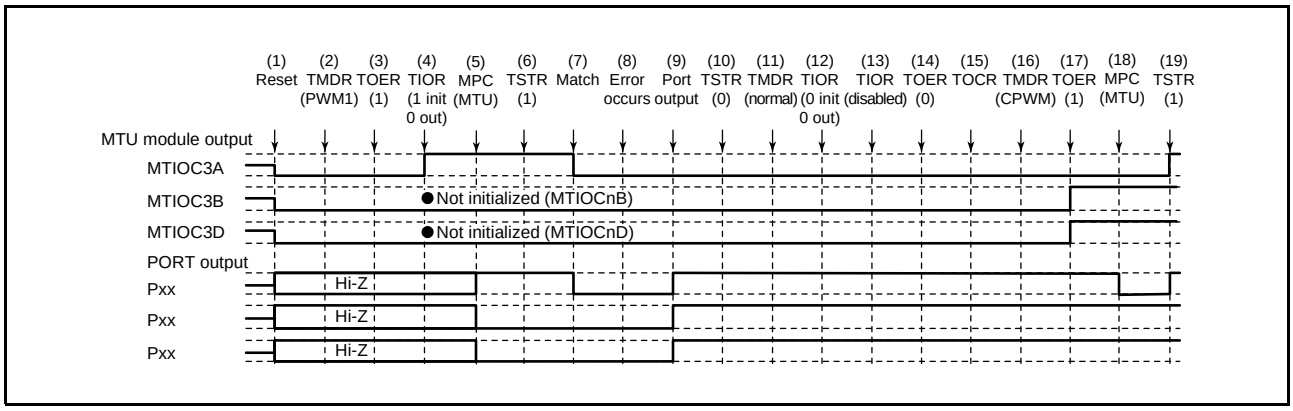


Figure 22.164 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 22.160.
(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 22.165 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

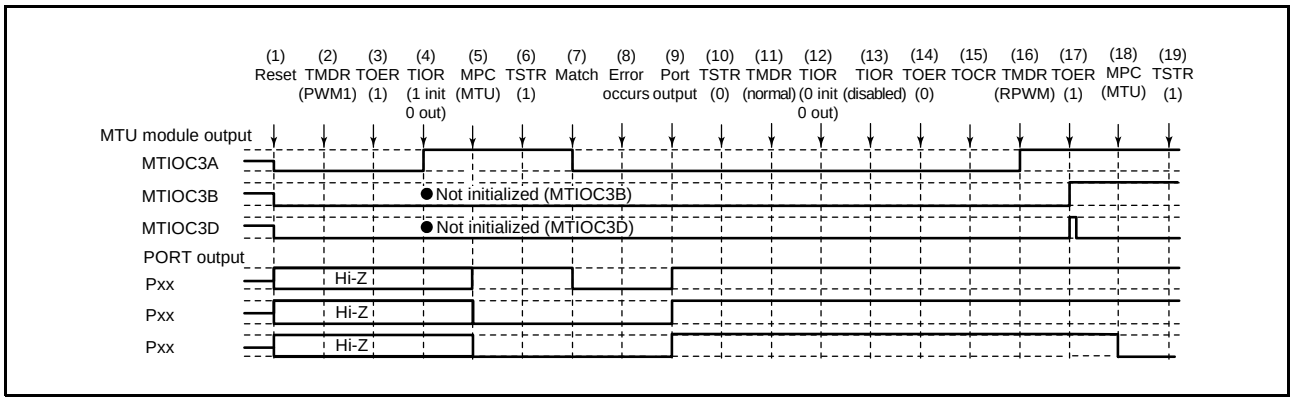


Figure 22.165 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 22.164.

(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 22.166 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

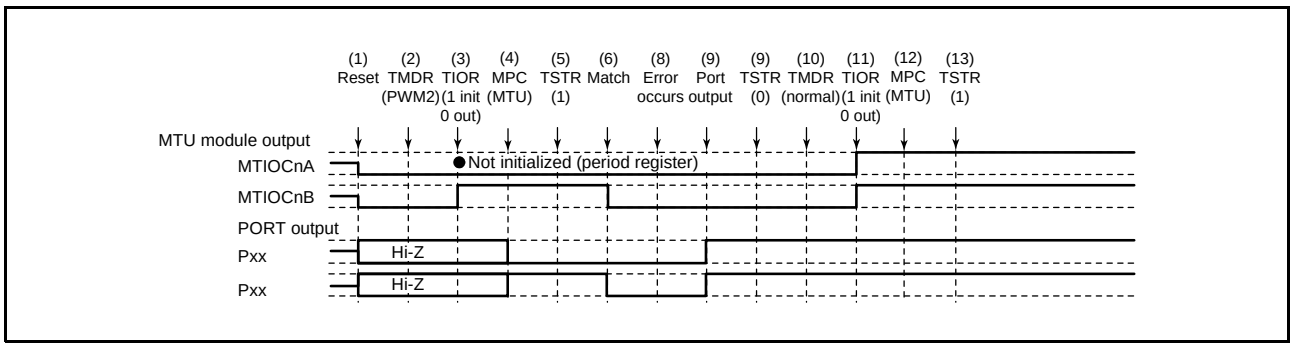


Figure 22.166 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(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 22.167 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

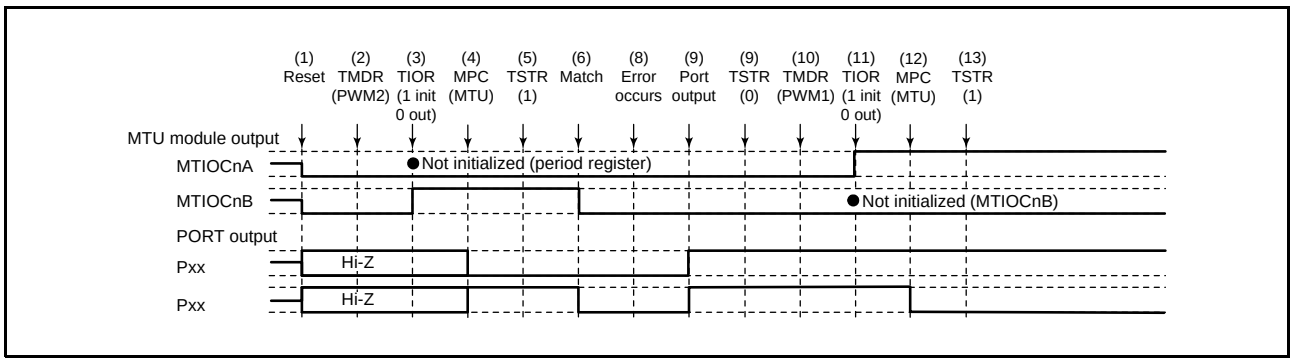


Figure 22.167 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.166.
(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 22.168 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

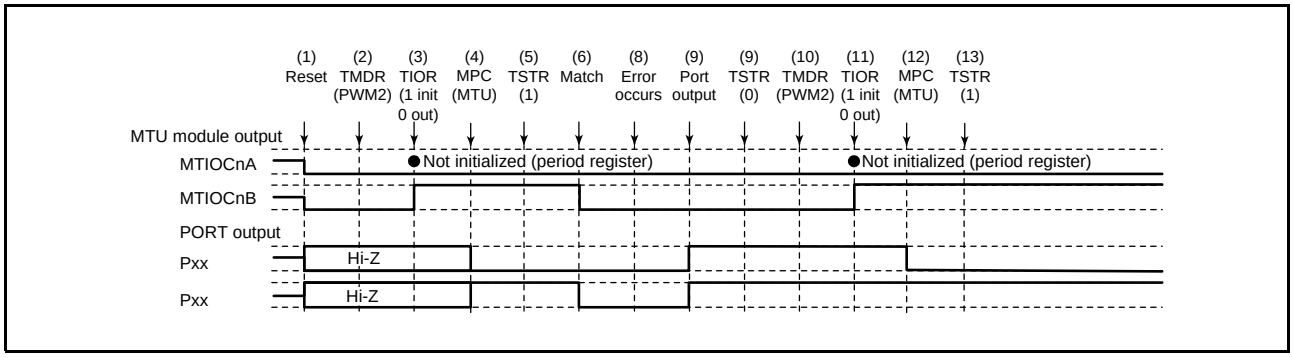


Figure 22.168 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 22.166.
(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 22.169 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

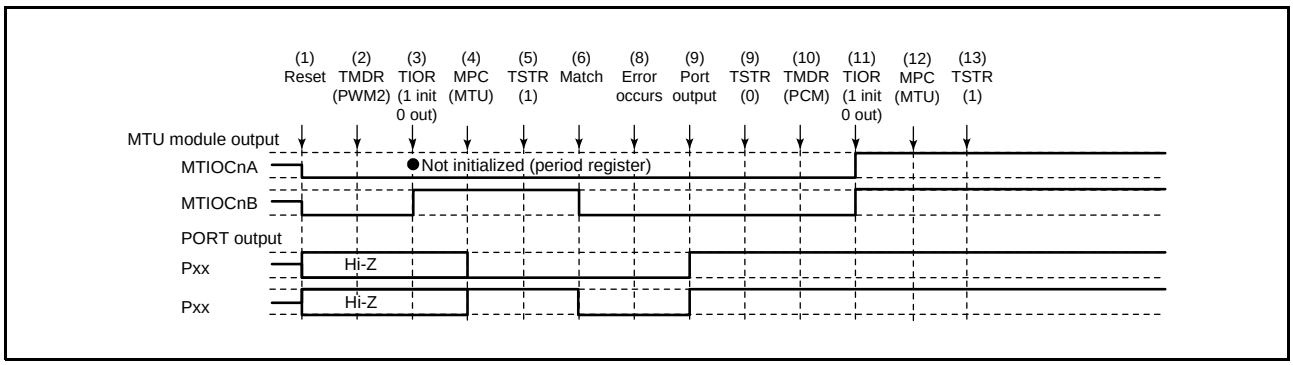


Figure 22.169 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 22.166.
(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 22.170 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

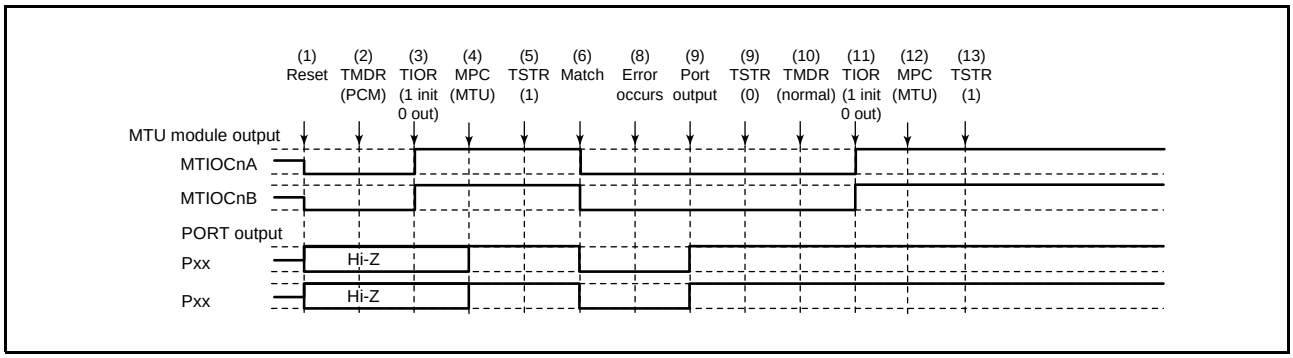


Figure 22.170 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(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 22.171 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

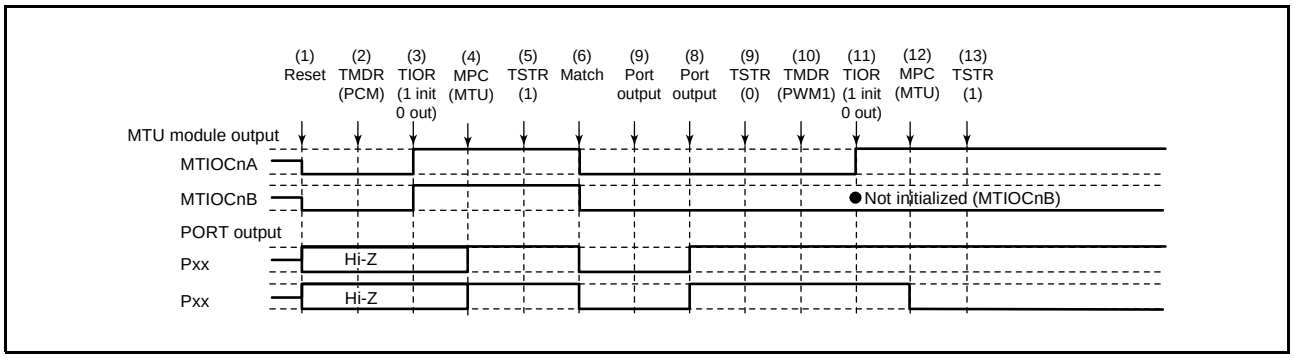


Figure 22.171 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.170.
(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 22.172 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

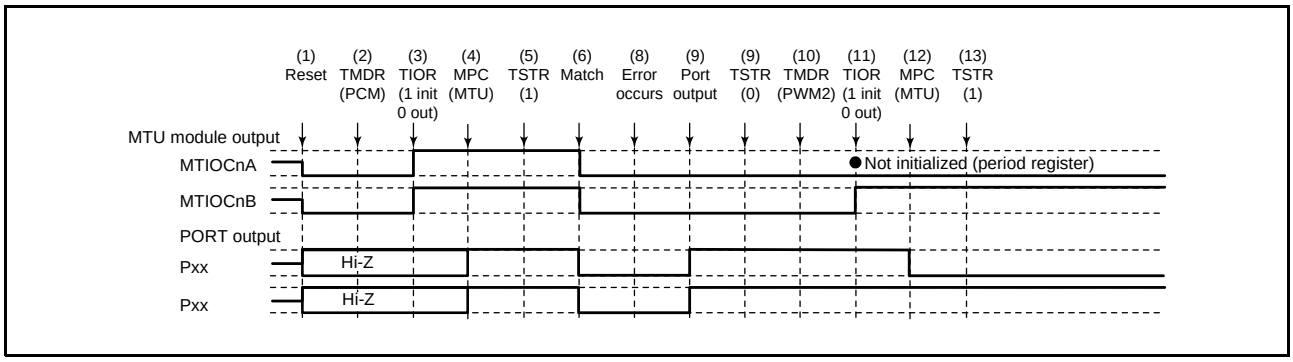


Figure 22.172 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 22.170.
(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 22.173 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

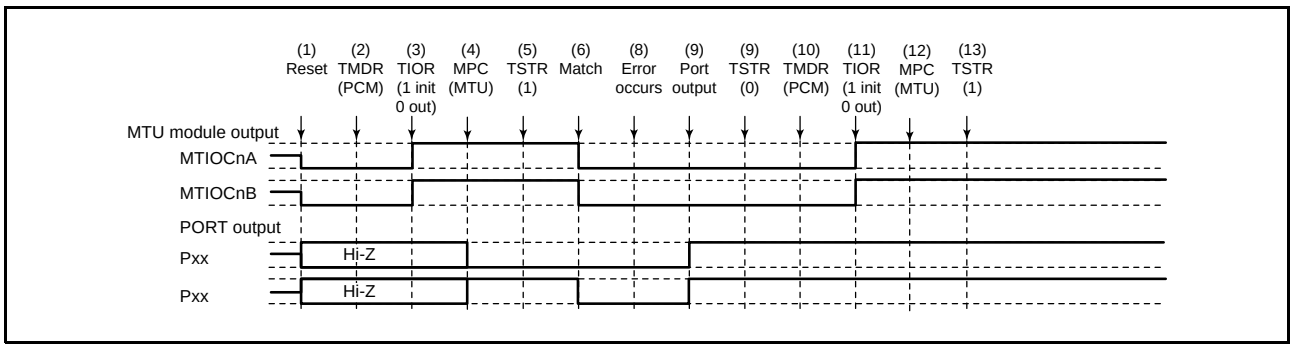


Figure 22.173 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 22.170.
(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 22.174 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

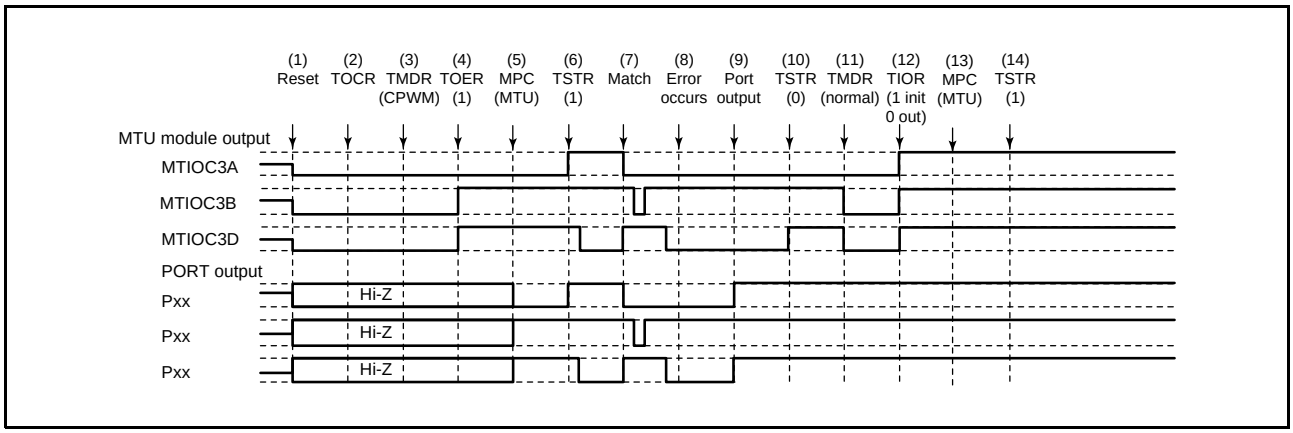


Figure 22.174 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(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 22.175 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

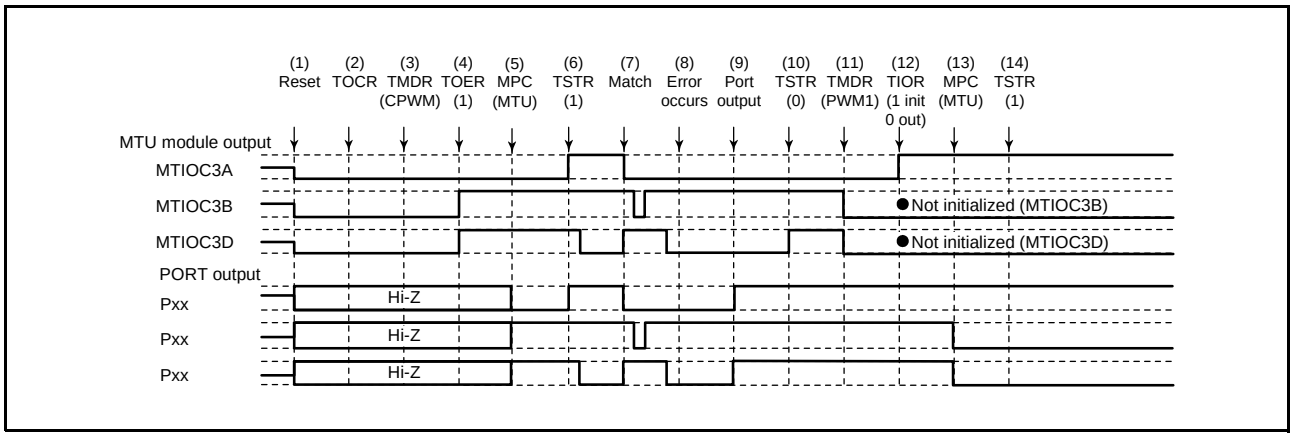


Figure 22.175 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.174.
(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 22.176 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).

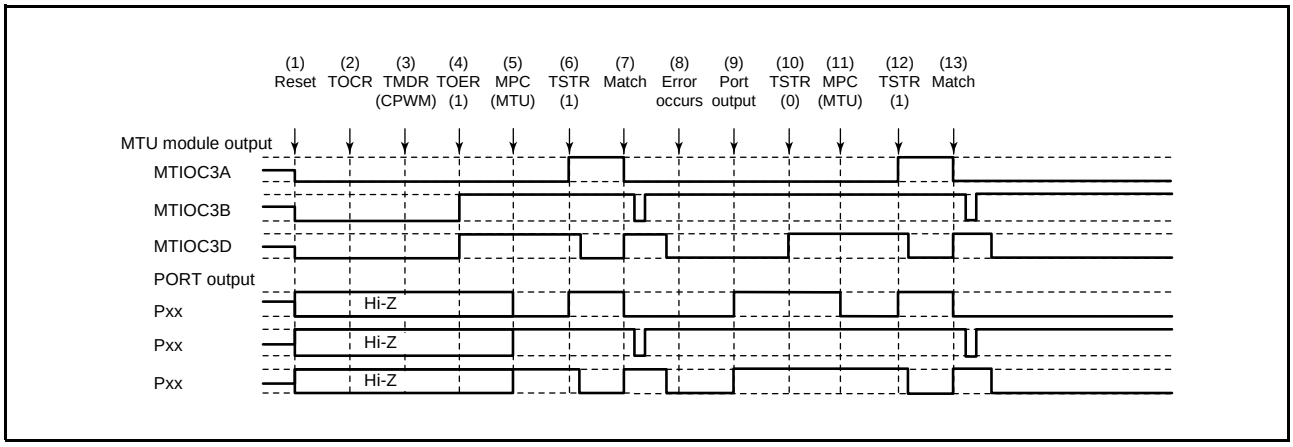


Figure 22.176 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in Figure 22.174.
(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 22.177 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).

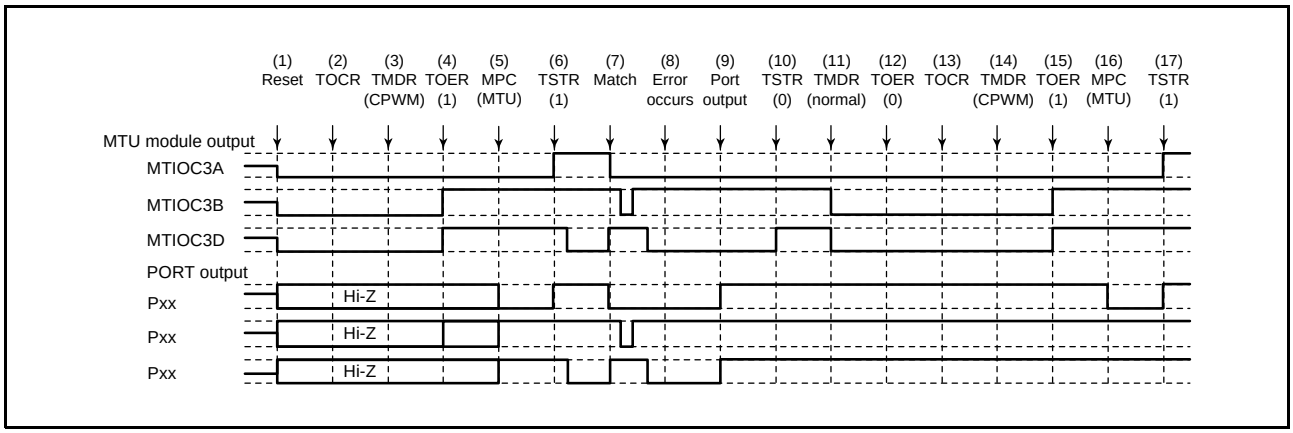


Figure 22.177 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in Figure 22.174.
(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 22.178 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

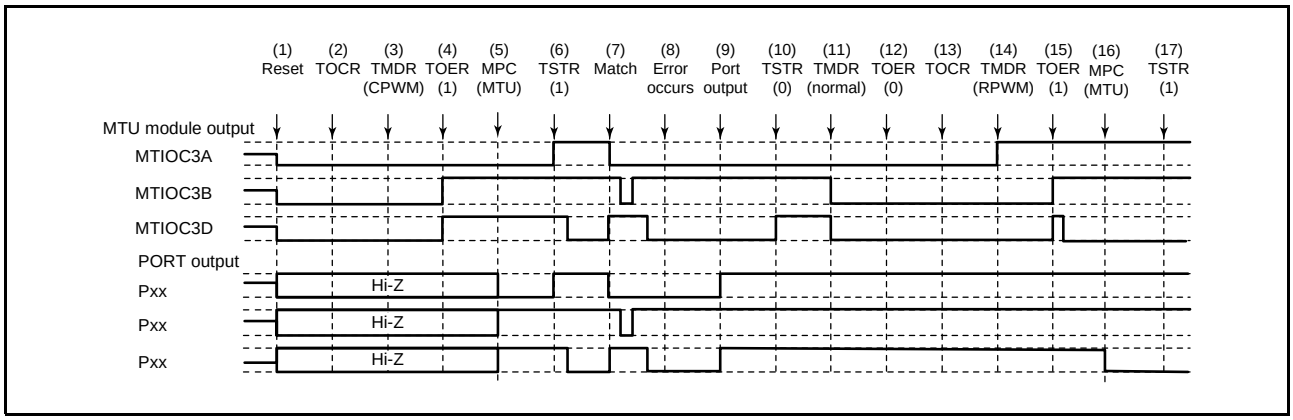


Figure 22.178 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 22.174.
(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 22.179 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

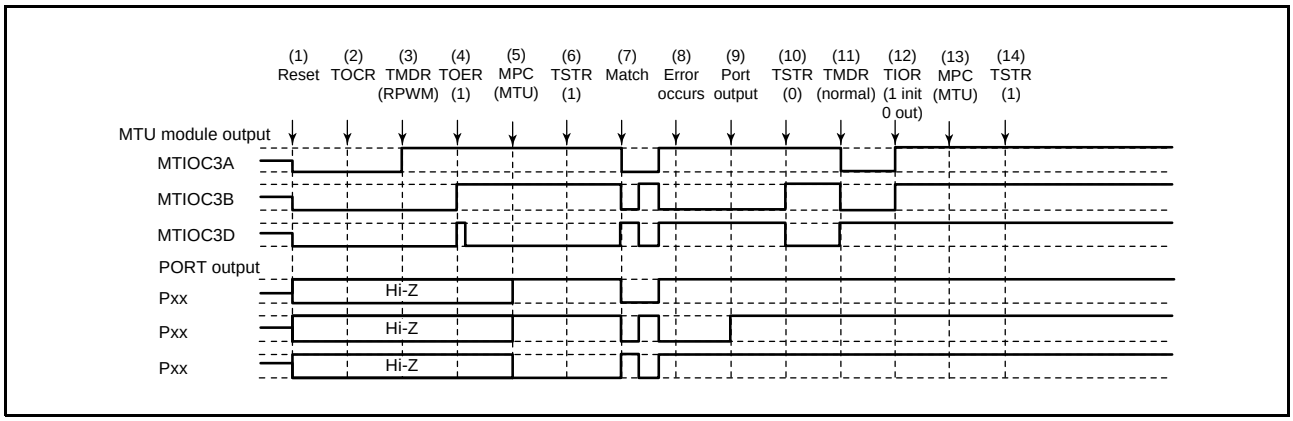


Figure 22.179 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(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 22.180 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

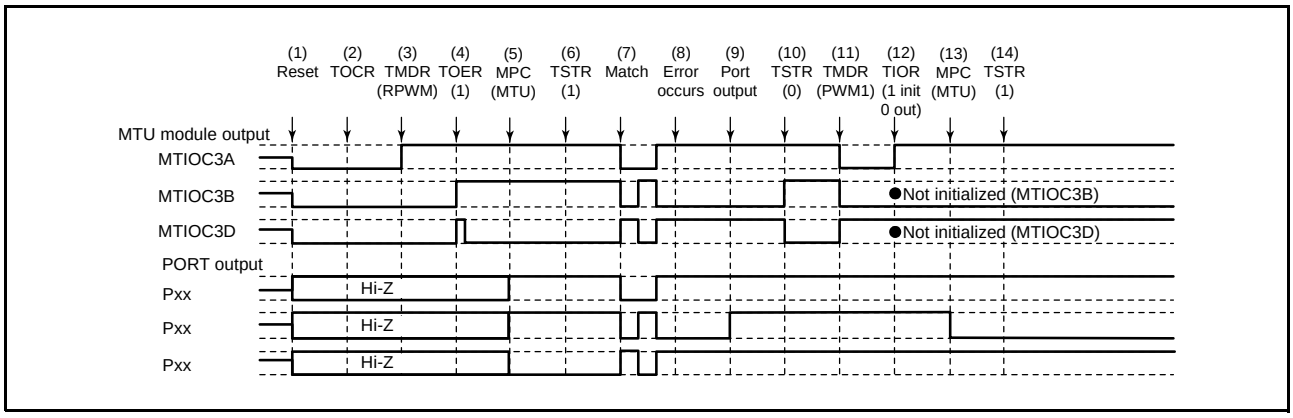


Figure 22.180 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.179.

(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 22.181 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

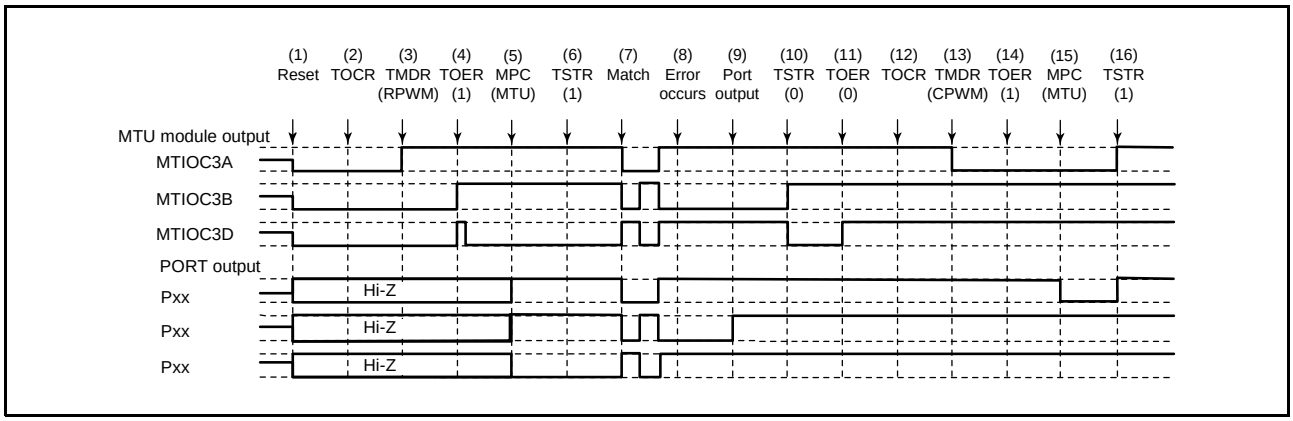


Figure 22.181 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 22.179.
(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 22.182 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.

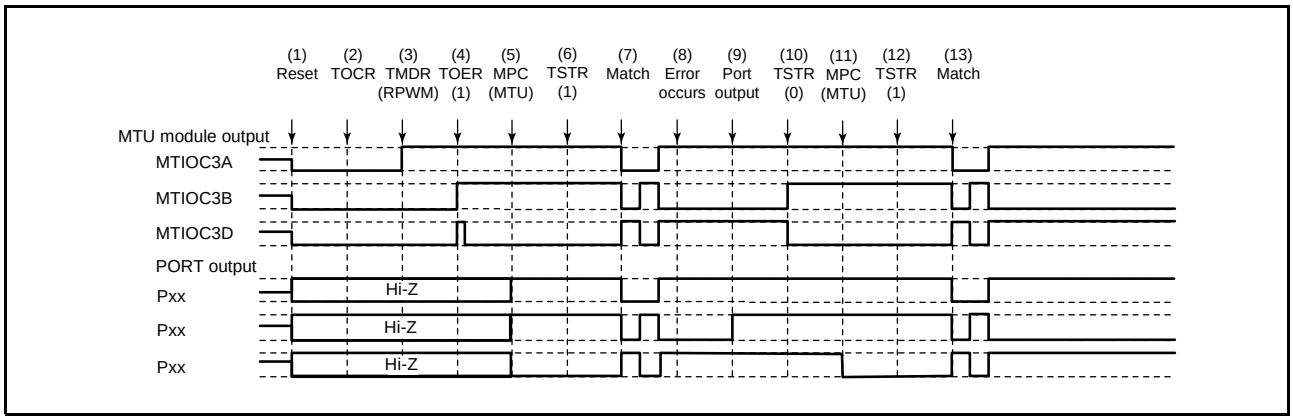


Figure 22.182 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 22.179.
(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.

22.8 Operations Linked by the ELC

22.8.1 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.

22.8.2 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, ELOPB, or ELOPE register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3, the ELOPB register controls the operation of MTU4, and the ELOPE register controls the operation of MTU6, MTU7, and MTU8. When the event specified in the ELSRn register occurs, the CSTn bit in the TSTRA/TSTRB register shown in Table 22.80 is set to 1, and the MTU counter starts.
However, when the specified event is generated while the CSTn bit in the TSTRA/TSTRB register has already been set to 1, the event has no effect. Table 22.80 lists the TSTRA/TSTRB register bits used for each channel.

Table 22.80 Counter Start Bit Set by the ELC

Channel No.	Counter Start Bit
MTU0	TSTRA.CST0 bit
MTU3	TSTRA.CST3 bit
MTU4	TSTRA.CST4 bit
MTU6	TSTRB.CST6 bit
MTU7	TSTRB.CST7 bit
MTU8	TSTRA.CST8 bit

(2) Input Capture Operation

Select “input capture” as the operation of the MTU by setting the ELOPA, ELOPB, or ELOPE register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3, the ELOPB register controls the operation of MTU4, and the ELOPE register controls the operation of MTU6, MTU7, and MTU8. 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/TSTRB 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 22.81 lists the timer general register and I/O control bit used for each channel in input capture operations in response to the ELC.

Table 22.81 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
MTU6	MTU6.TGRA	MTU6.TIORH.IOA[3:0] bits
MTU7	MTU7.TGRA	MTU7.TIORH.IOA[3:0] bits
MTU8	MTU8.TGRA	MTU8.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, ELOPB, or ELOPE register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3, the ELOPB register controls the operation of MTU4, and the ELOPE register controls the operation of MTU6, MTU7, and MTU8. When the event specified in the ELSRn register occurs, the TCNT register is cleared. If the corresponding CSTn bit in the TSTRA/TSTRB register is set to 1, counting continues. For the CSTn bits in the TSTRA/TSTRB register, refer to Table 22.80.

22.8.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC

The following notes on usage apply when the MTU is used in event link operation.

(1) Start Counting

If the event specified in the ELSRn register occurs during a cycle of writing to a CSTn bit in the TSTRA/TSTRB register, writing to the CSTn bit in the TSTRA/TSTRB register does not proceed because setting of the bit to 1 due to the event takes priority.

(2) Restart 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 or MTU6 and MTU7 in complementary PWM mode, do not use the counter restarting by the ELC.

23. 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.

23.1 Overview

Table 23.1 lists the specifications of the POE3, and Figure 23.1 shows a block diagram of the POE3.

Table 23.1 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 23.1.

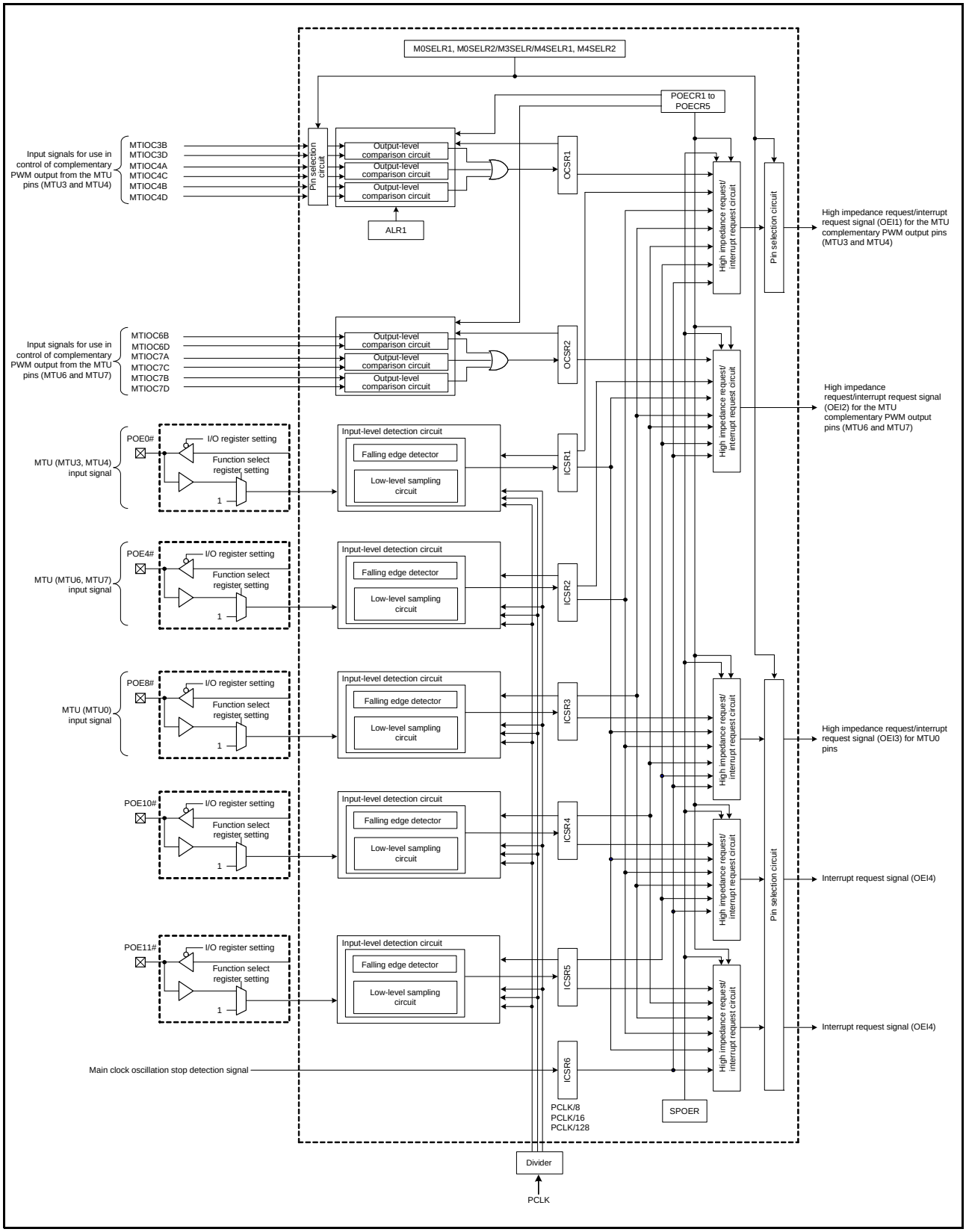


Figure 23.1 POE3 Block Diagram

Table 23.2 shows I/O pins to be used by the POE3.

Table 23.2 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 23.3 shows output-level comparisons with pin combinations.

Table 23.3 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	

23.2 Register Descriptions

The POE3 registers are initialized by a reset.

23.2.1 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 23.3.7, Recover from High-Impedance State.

23.2.2 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 23.3.7, Recover from High-Impedance State.

23.2.3 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 23.3.7, Recover from High-Impedance State.

23.2.4 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 23.3.7, Recover from High-Impedance State.

23.2.5 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 23.3.7, Recover from High-Impedance State.

23.2.6 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

23.2.7 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 23.2.9, 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 23.3.7, Recover from High-Impedance State.

23.2.8 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 22, 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 23.3.7, Recover from High-Impedance State.

23.2.9 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.

23.2.10 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

23.2.11 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.

23.2.12 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.

23.2.13 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).

23.2.14 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).

23.2.15 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.*3 0010: Controls the high-impedance state of PB3 on the assumption that the MTIOC0A pin is assigned to PB3. 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.*3 0001: Controls the high-impedance state of P15 on the assumption that the MTIOC0B pin is assigned to P15. 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 80 pins. Set these bits to 0010b in products with fewer pins.
Note 3. This selection is only usable in products with at least 80 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)

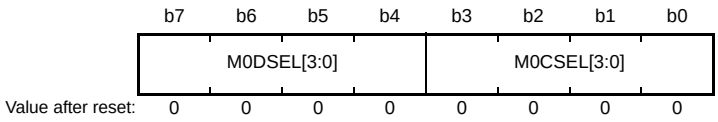
These bits select the target MTIOC0A pin for high-impedance control.

M0BSEL[3:0] Bits (MTU0-B (MTIOC0B) Pin Select)

These bits select the target MTIOC0B pin for high-impedance control.

23.2.16 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.*3 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*4	b7 b4 0000: Controls the high-impedance state of P33 on the assumption that the MTIOC0D pin is assigned to P33.*5 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 64 pins. Set these bits to 0010b in products with 48 pins.
Note 3. This selection is only usable in products with at least 64 pins.
Note 4. These bits are only effective in products with at least 100 pins. Set these bits to 0010b in products with fewer pins.
Note 5. This selection is only usable in products with at least 100 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)

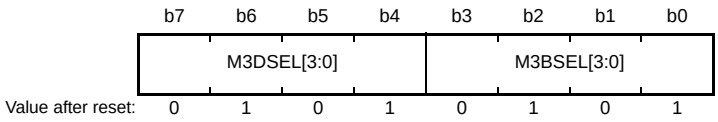
These bits select the target MTIOC0C pin for high-impedance control.

M0DSEL[3:0] Bits (MTU0-D (MTIOC0D) Pin Select)

These bits select the target MTIOC0D pin for high-impedance control.

23.2.17 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. 0001: Controls the high-impedance state of P22 on the assumption that the MTIOC3B pin is assigned to P22.*2 0010: Controls the high-impedance state of P80 on the assumption that the MTIOC3B pin is assigned to P80.*3 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.*4 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.*4 0001: Controls the high-impedance state of P23 on the assumption that the MTIOC3D pin is assigned to P23.*2 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.*3 0100: Controls the high-impedance state of PB6 on the assumption that the MTIOC3D pin is assigned to PB6.*4 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 100 pins.
Note 3. This selection is only usable in products with 144 pins.
Note 4. This selection is only usable in products with at least 64 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)

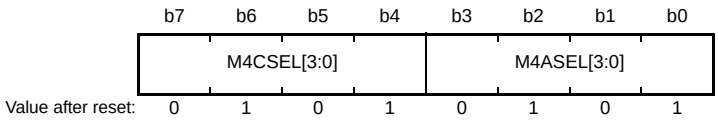
These bits select the target MTIOC3B pin for high-impedance control.

M3DSEL[3:0] Bits (MTU3-D (MTIOC3D) Pin Select)

These bits select the target MTIOC3D pin for high-impedance control.

23.2.18 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	b3 b0 0000: Controls the high-impedance state of PE2 on the assumption that the MTIOC4A pin is assigned to PE2. 0001: Controls the high-impedance state of P21 on the assumption that the MTIOC4A pin is assigned to P21.*2 0010: Controls the high-impedance state of PB3 on the assumption that the MTIOC4A pin is assigned to PB3. 0011: Controls the high-impedance state of P82 on the assumption that the MTIOC4A pin is assigned to P82.*3 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.*5 Settings other than above are prohibited.	R/W*1
b7 to b4	M4CSEL[3:0]	MTU4-C (MTIOC4C) Pin Select	b7 b4 0000: Controls the high-impedance state of PE5 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.*3 0010: Controls the high-impedance state of PB1 on the assumption that the MTIOC4C pin is assigned to PB1. 0011: Controls the high-impedance state of P83 on the assumption that the MTIOC4C pin is assigned to P83.*3 0100: Controls the high-impedance state of PE1 on the assumption that the MTIOC4C pin is assigned to PE1. 0101: Controls the high-impedance state of P25 on the assumption that the MTIOC4C pin is assigned to P25.*5 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 80 pins.
Note 3. This selection is only usable in products with 144 pins.
Note 4. This selection is only usable in products with at least 64 pins.
Note 5. This selection is only usable in products with at least 100 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)

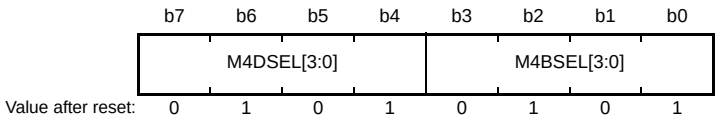
These bits select the target MTIOC4A pin for high-impedance control.

M4CSEL[3:0] Bits (MTU4-C (MTIOC4C) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

23.2.19 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. 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.*3 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. 0001: Controls the high-impedance state of P86 on the assumption that the MTIOC4D pin is assigned to P86.*4 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.*3 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 64 pins.
Note 3. This selection is only usable in products with at least 80 pins.
Note 4. This selection is only usable in products with 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.

23.3 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.

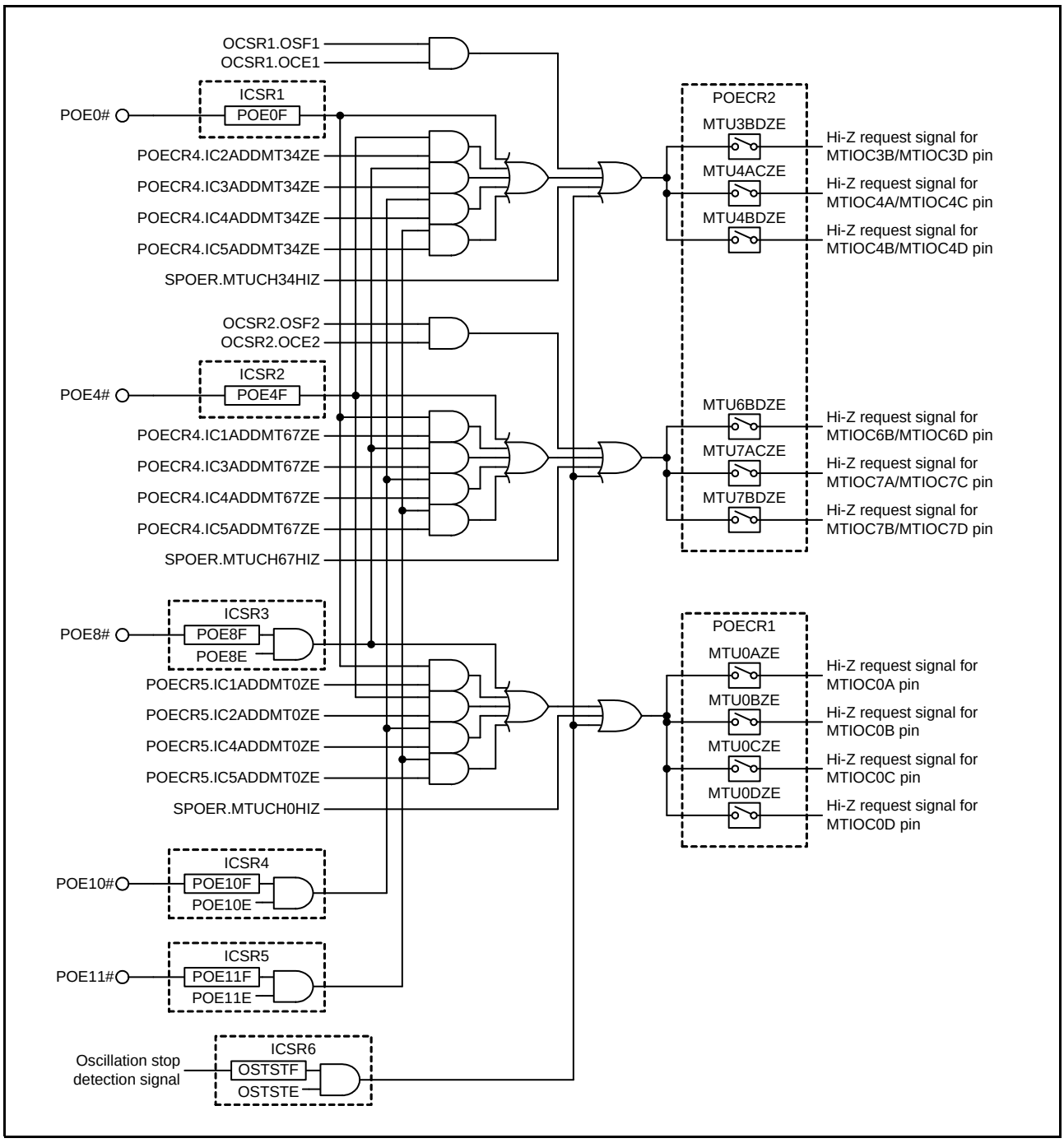


Figure 23.2 Target Pins and Conditions for High-Impedance Control

23.3.1 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 23.4 shows the correspondence between MTU pins and select registers. The high-impedance control cannot be performed for pins not listed in this table.
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 23.4 Correspondence between MTU Pins

MTU Pin Functions	Corresponding Ports	Select Registers
MTIOC0A	P34*1	M0SELR1
	PB3	
MTIOC0B	P13*1	
	P15	
	PA1	
MTIOC0C	P32*2	M0SELR2
	PB1	
MTIOC0D	P33*3	
	PA3	
MTIOC3B	PE1	M3SELR
	P22*3	
	P80*4	
	PC5	
	PB7*2	
	P17	
MTIOC3D	PE0*2	
	P23*3	
	PC4	
	P81*4	
	PB6*2	
	P16	
MTIOC4A	PE2	M4SELR1
	P21*1	
	PB3	
	P82*4	
	PA0*2	
	P24*3	
MTIOC4C	PE5*2	
	P87*4	
	PB1	
	P83*4	
	PE1	
	P25*3	
MTIOC4B	PE3	M4SELR2
	P17	
	P54*2	
	PC2*2	
	PD1*1	
	P30	
MTIOC4D	PE4	
	P86*4	
	P55*2	
	PC3*2	
	PD2*1	
	P31	
MTIOC6B	PA5*1	—
MTIOC6D*2	PA0*2	
MTIOC7A	PA2*1	—
MTIOC7C	P67*4	
MTIOC7B	PA1	
MTIOC7D	P66*4	

Note 1. This is only present in products with at least 80 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 100 pins.
Note 4. This is only present in products with 144 pins.

23.3.2 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 23.3 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.

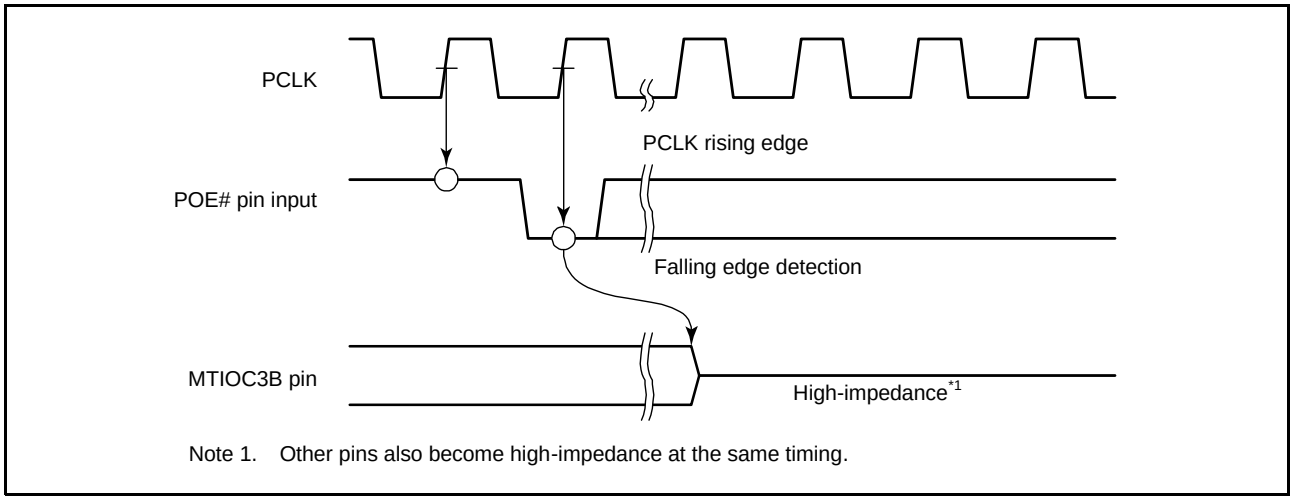
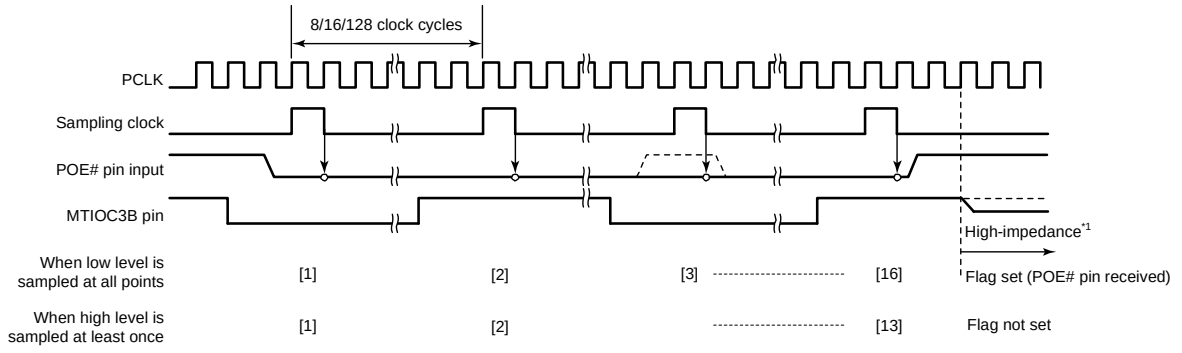


Figure 23.3 Operation when A Falling Edge Detection is Selected

(2) Low-Level Detection

Figure 23.4 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.

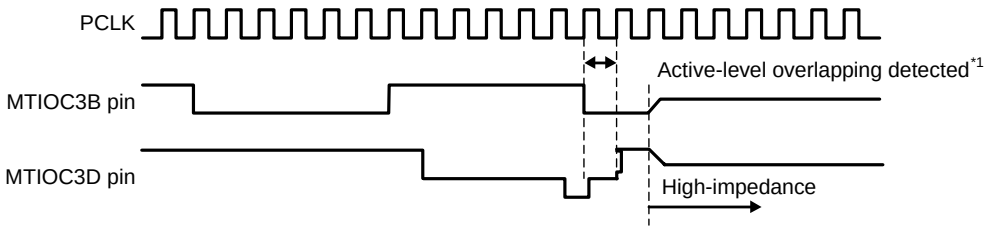


Note 1. Other pins also become high-impedance at the same timing.

Figure 23.4 Operation when A Low-Level Detection is Selected

23.3.3 Output-Level Compare Operation

Figure 23.5 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 23.5 Output-Level Compare Operation

23.3.4 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.

23.3.5 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.

23.3.6 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

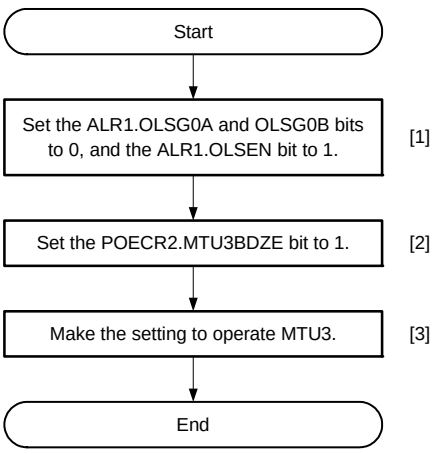
The high-impedance control of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5 registers.

23.3.7 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.

23.4 POE3 Setting Procedure

Figure 23.6 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.



Set the MTU3 pins (P22 and P23) for the peripheral function by the PMR register of the I/O port before making the following settings.

[1] Set the ALR1.OLSG0A and OLSG0B bits to 0 and the ALR1.OLSEN bit to 1 to set MTIOC3B and MTIOC3D pins as active low.

[2] Set the POECR2.MTU3BDZE bit to 1 to enable high-impedance control on the MTIOC3B and MTIOC3D pins.

[3] Make the setting to operate MTU3 pin.

After the above settings, if MTIOC3B (P22) and MTIOC3D (P23) pins are simultaneously at the low level, then the pins become high-impedance and the OCSR1.OSF1 flag is set to 1.
For the method for releasing high-impedance, refer to section 23.3.7, Recover from High-Impedance State.

Figure 23.6 Procedure for Setting the POE3

23.5 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 23.5 shows the interrupt sources and their conditions.

Table 23.5 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

23.6 Usage Notes

23.6.1 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.

23.6.2 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.

23.6.3 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.

24. 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.

24.1 Overview

Table 24.1 lists the specifications of the TMR. Table 24.2 lists the TMR functions.
Figure 24.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 24.2 shows that of the 8-bit timer module (unit 1).

Table 24.1 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 30, Serial Communications Interface (SCIk, SCIm, SCIh).
Note 2. For details, refer to section 36, Remote Control Signal Receiver (REMCa).

Table 24.2 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 38, 12-Bit A/D Converter (S12ADH).
Note 2. For details, refer to section 30, Serial Communications Interface (SCIk, SCIm, SCIh).
Note 3. For details, refer to section 36, Remote Control Signal Receiver (REMCa).
Note 4. For details, refer to section 11, Low Power Consumption.

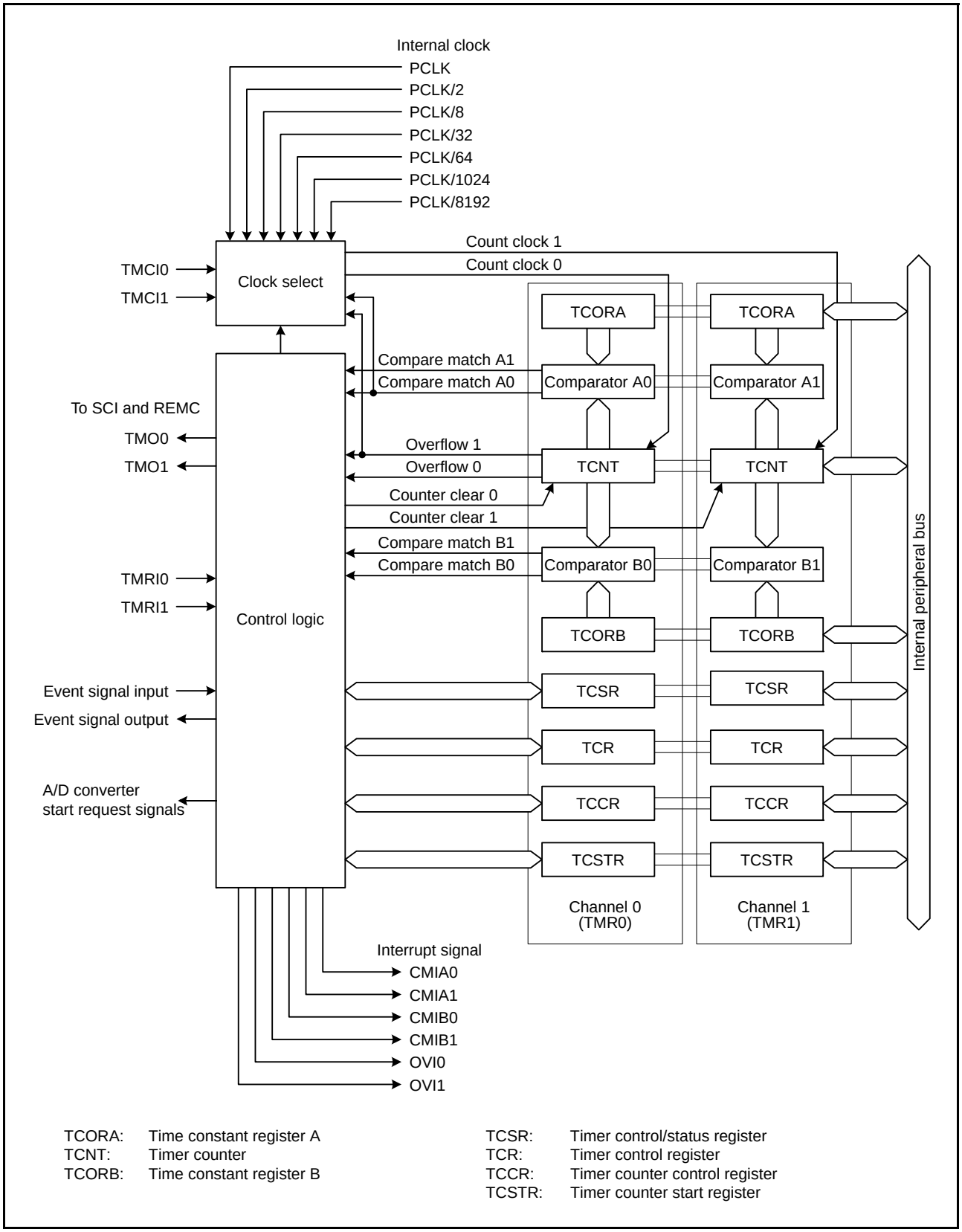


Figure 24.1 Block Diagram of TMR (Unit 0)

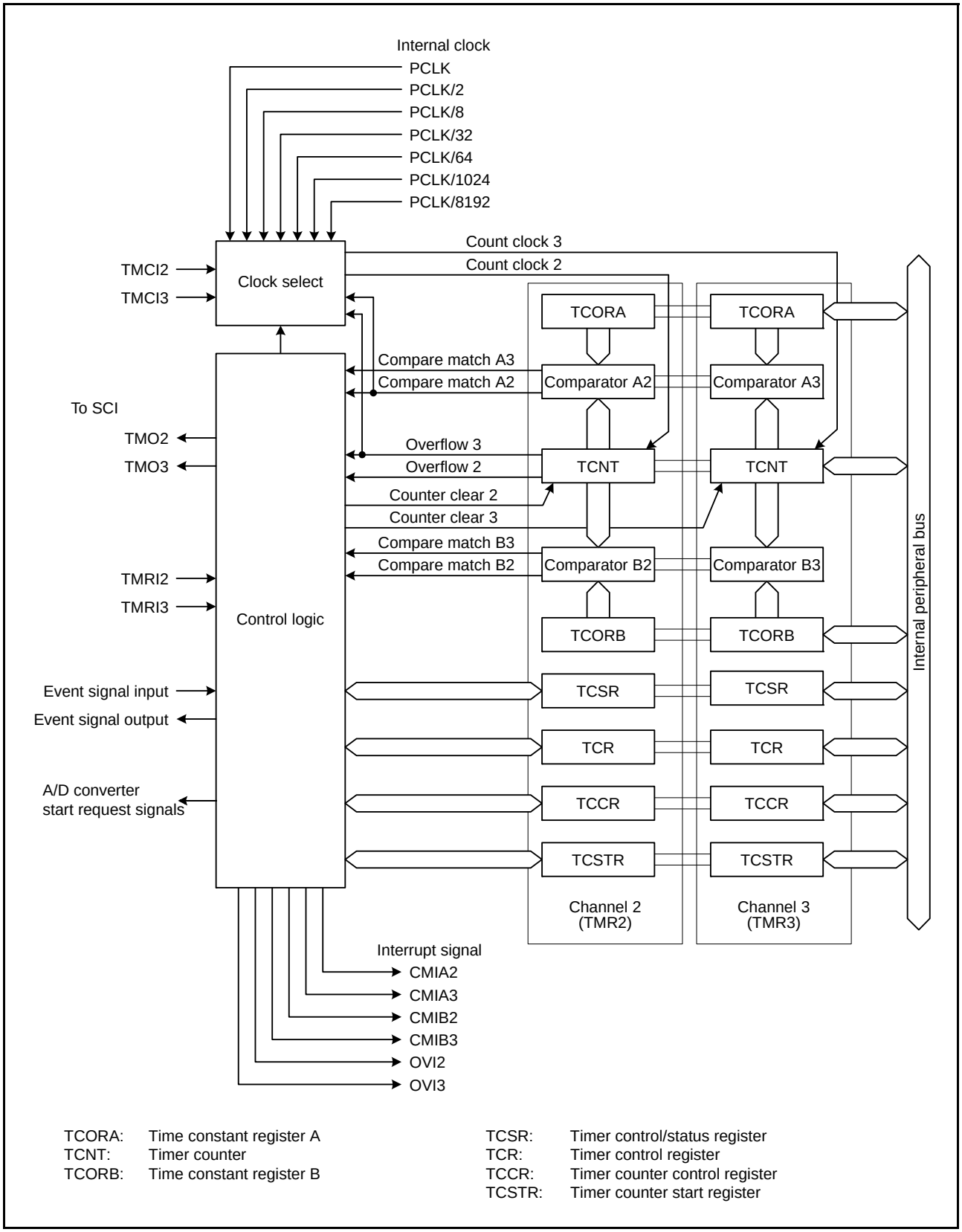


Figure 24.2 Block Diagram of TMR (Unit 1)

Table 24.3 lists the I/O pins of the TMR.

Table 24.3 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

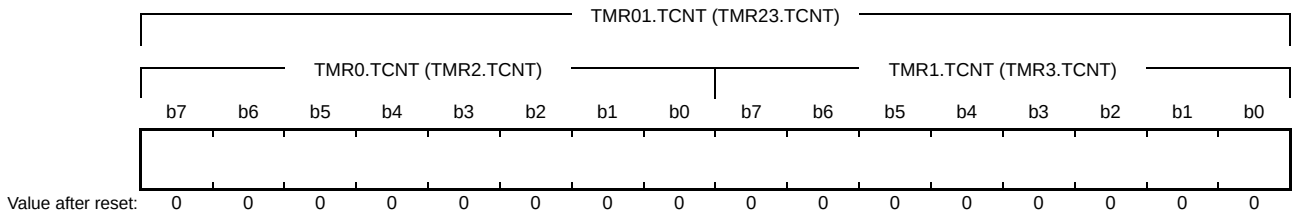
24.2 Register Descriptions

Table 24.4 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

24.2.1 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.

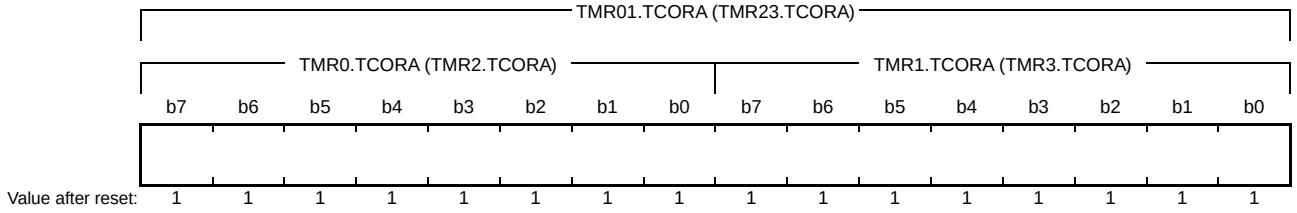
TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

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 14, Interrupt Controller (ICUF), and Table 24.6, TMR Interrupt Sources.

24.2.2 Time Constant Register A (TCORA)

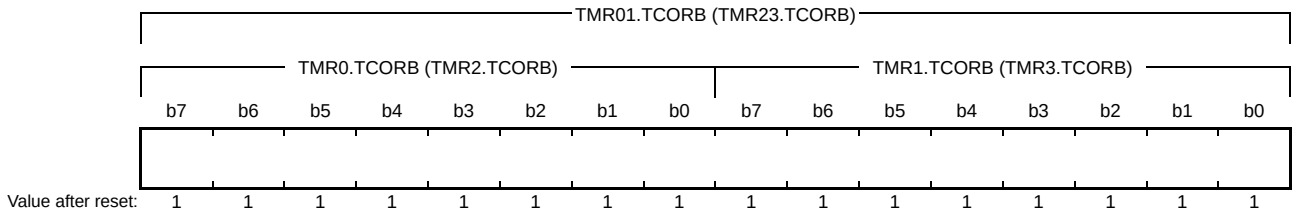
Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h, TMR01.TCORA 0008 8204h, TMR23.TCORA 0008 8214h



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.

24.2.3 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.

24.2.4 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 20, I/O Ports and section 21, 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.

24.2.5 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 24.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 24.5.	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 20, I/O Ports and section 21, 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 24.5.

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 24.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register CSS[1:0] b4	b3	CKS[2:0] b2	b1	b0	Description
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 20, I/O Ports and section 21, 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.

24.2.6 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.

24.2.7 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 24.7, Link Operation by ELC, or section 19, Event Link Controller (ELC).

24.3 Operation

24.3.1 Pulse Output

Figure 24.3 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.

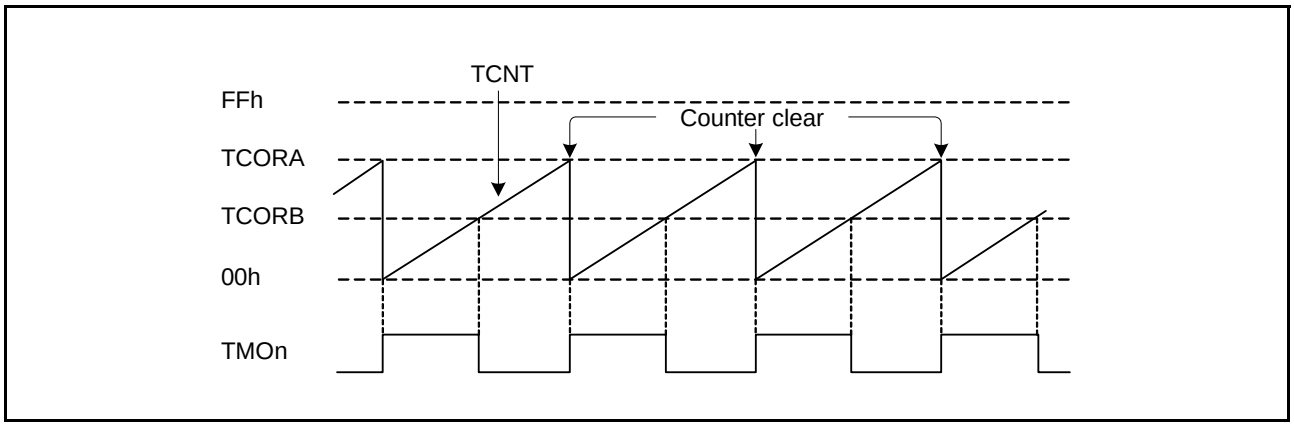


Figure 24.3 Example of Pulse Output (n = 0 to 3)

24.3.2 External Counter Reset Input

Figure 24.4 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.

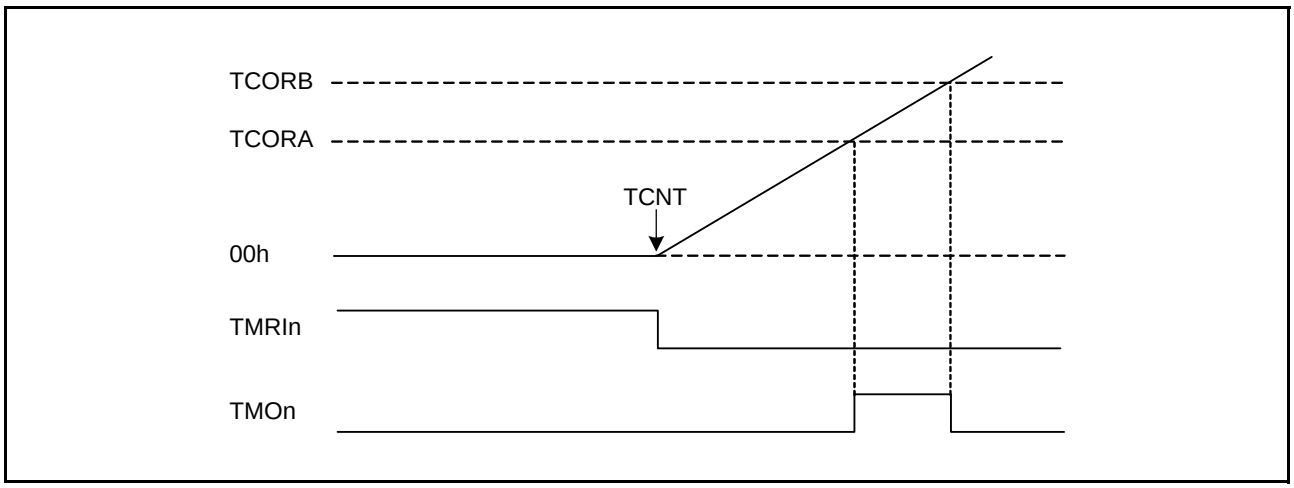


Figure 24.4 Example of External Counter Reset Signal Input (n = 0 to 3)

24.4 Operation Timing

24.4.1 TCNT Count Timing

Figure 24.5 shows the count timing of TCNT for internal clock. Figure 24.6 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.

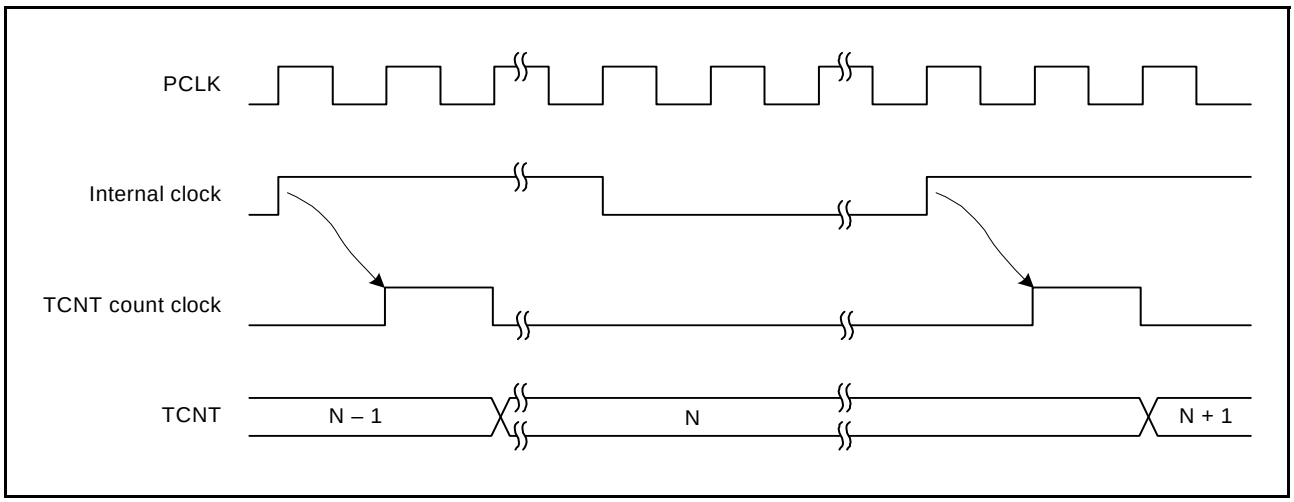


Figure 24.5 Count Timing for Internal Clock

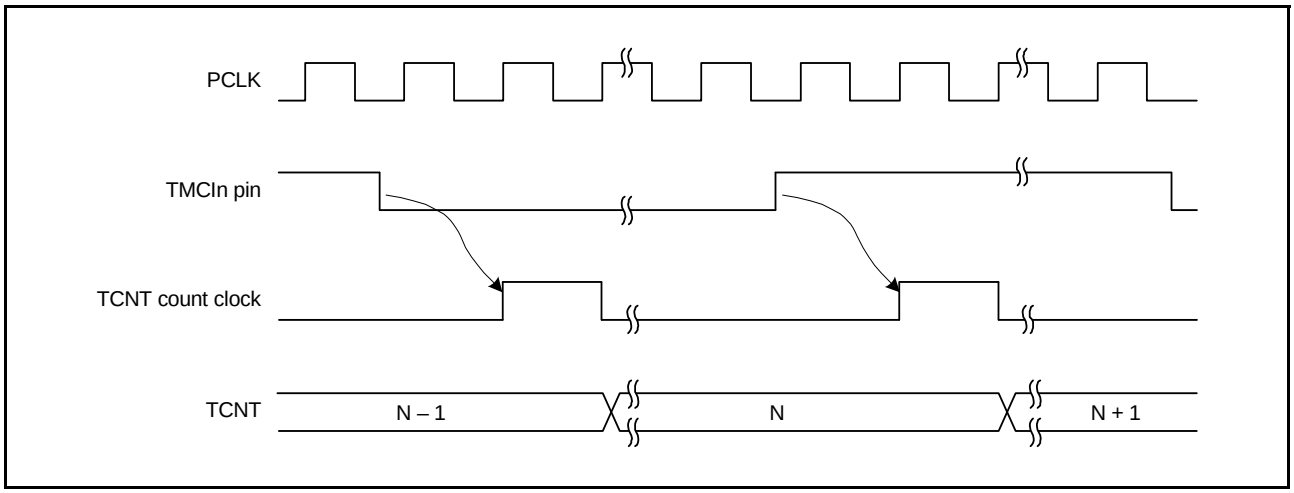


Figure 24.6 Count Timing for External Clock (at Both Edges)

24.4.2 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 24.7 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, refer to section 14, Interrupt Controller (ICUF) and Table 24.6.

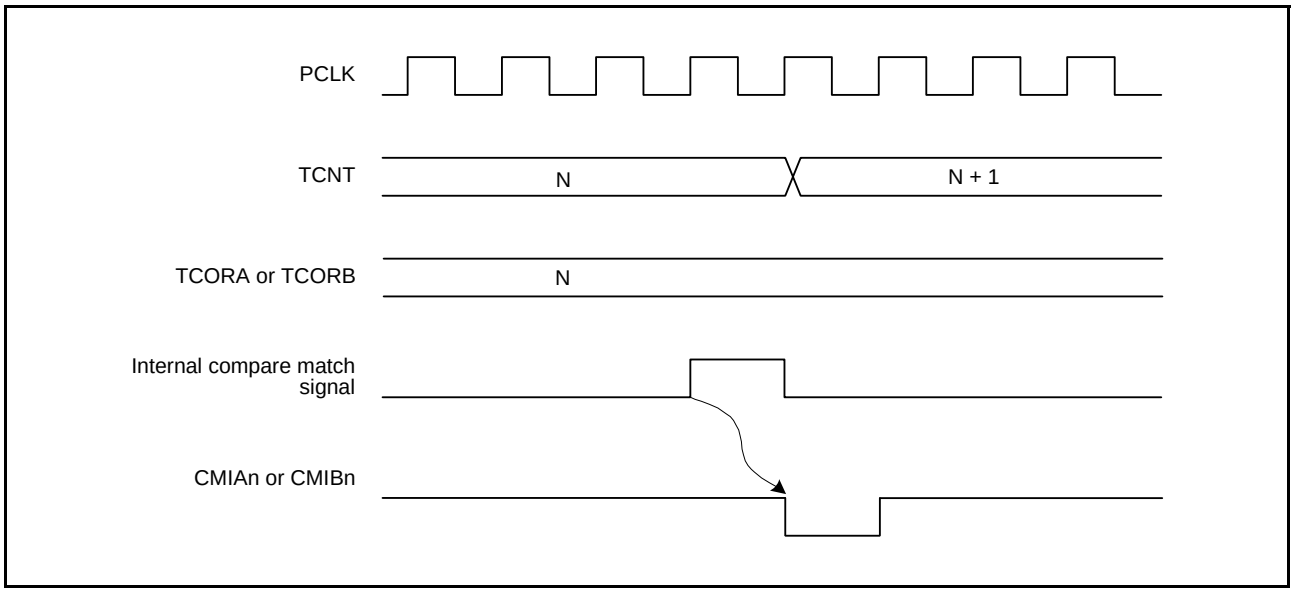


Figure 24.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

24.4.3 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 24.8 shows the timing when the timer output is toggled by the compare match A signal.

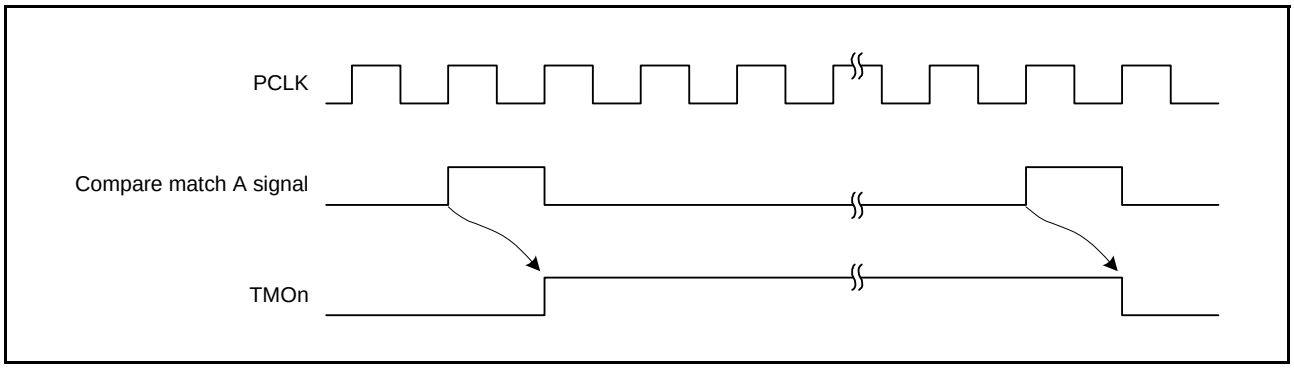


Figure 24.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 3)

24.4.4 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 24.9 shows the timing of this operation.

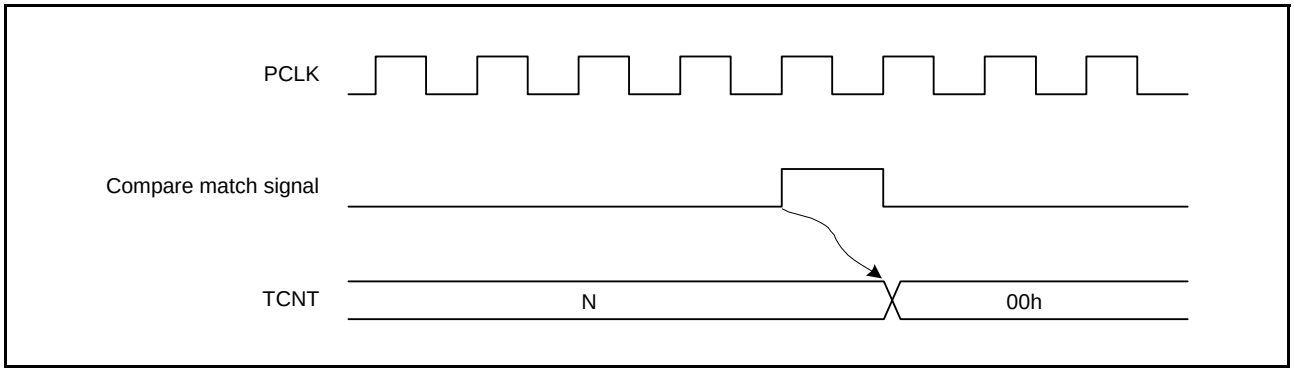


Figure 24.9 Timing of Counter Clear by Compare Match

24.4.5 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 24.10 and Figure 24.11 show the timing of this operation.

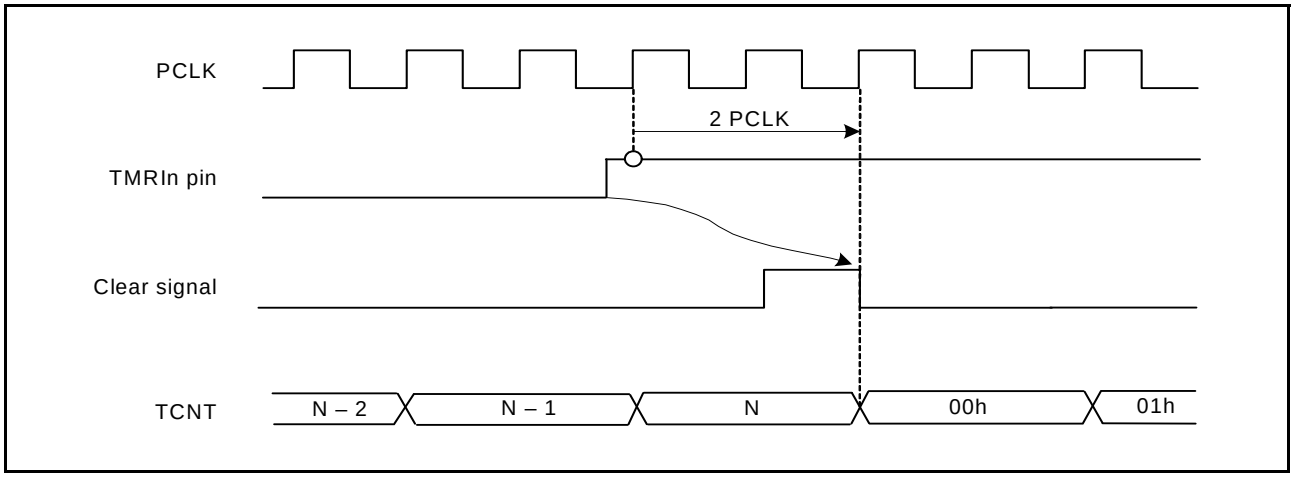


Figure 24.10 Clear Timing by External Counter Reset Signal (Rising Edge)

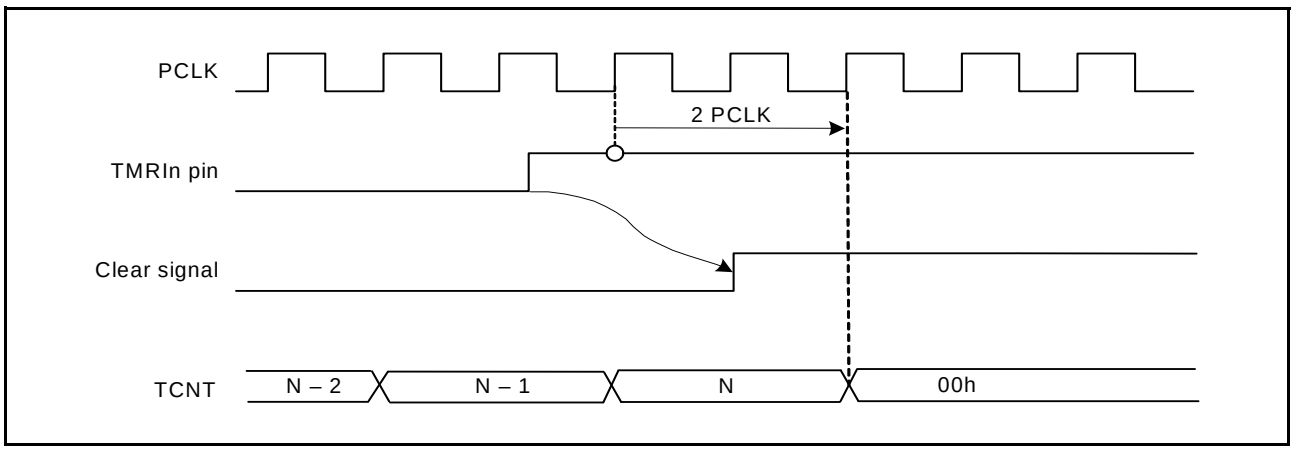


Figure 24.11 Clear Timing by External Counter Reset Signal (High Level)

24.4.6 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 24.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, refer to section 14, Interrupt Controller (ICUF) and Table 24.6.

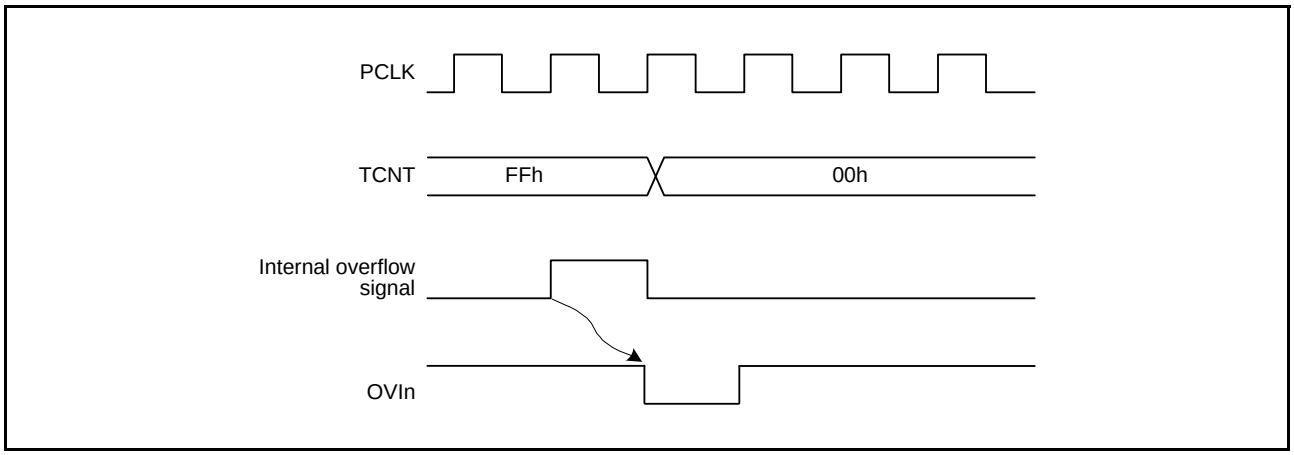


Figure 24.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

24.5 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.

24.5.1 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.

24.5.2 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.

24.6 Interrupt Sources

24.6.1 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 24.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 24.6 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

24.6.2 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 24.7 Startup of A/D Converter

A/D Converter	TMR Unit No.	Target	A/D Conversion Start Request
S12AD (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

24.7 Link Operation by ELC

24.7.1 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 19, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

24.7.2 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.

24.7.3 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.

24.8 Usage Notes

24.8.1 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.

24.8.2 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).

$$f = PCLK / (N + 1)$$

24.8.3 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 24.13.

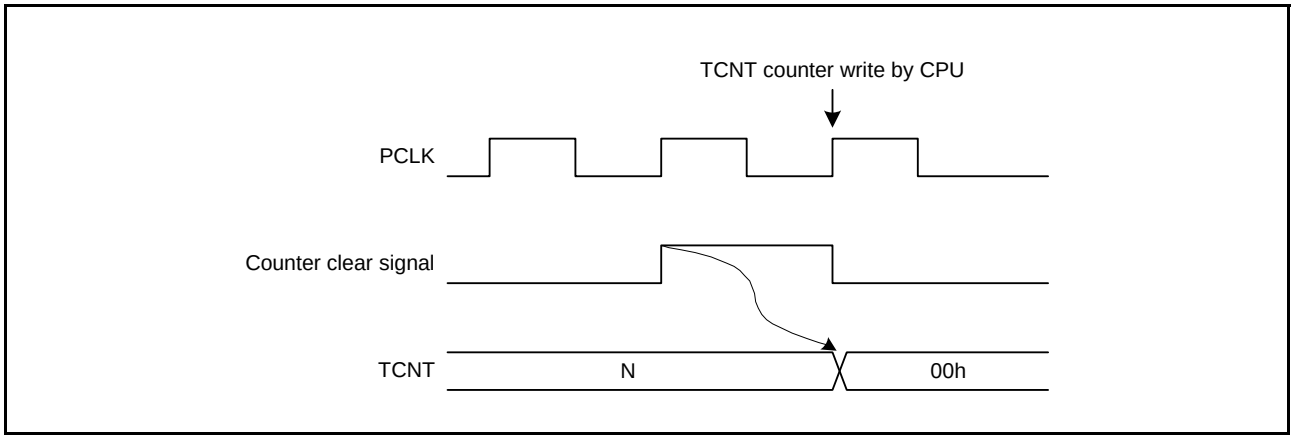


Figure 24.13 Conflict between TCNT Write and Counter Clear

24.8.4 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 24.14.

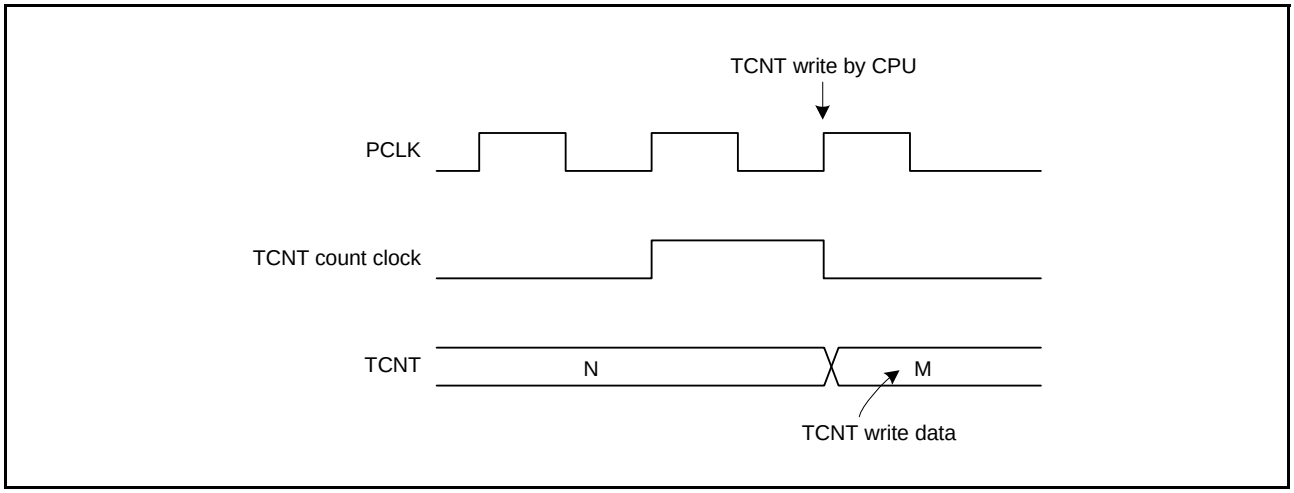


Figure 24.14 Conflict between TCNT Write and Increment

24.8.5 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 24.15, the write takes priority and the compare match signal does not reach High level.

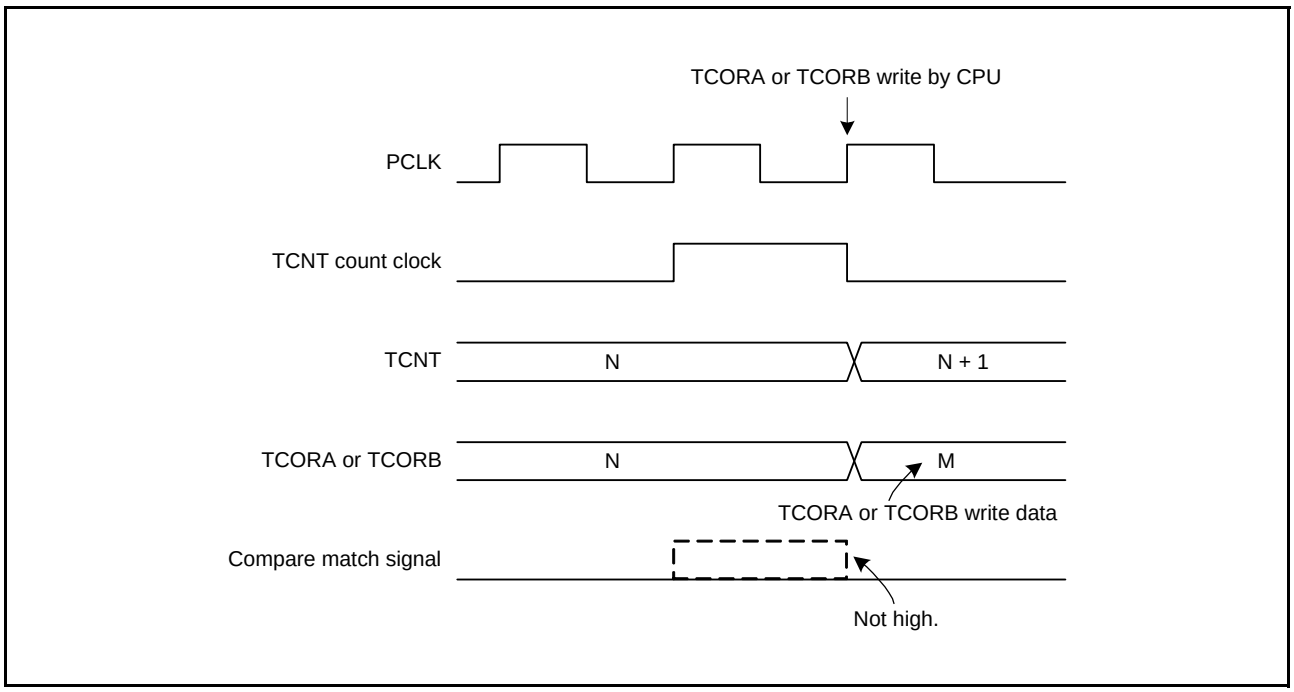


Figure 24.15 Conflict between TCORA or TCORB Write and Compare Match

24.8.6 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 24.8.

Table 24.8 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

24.8.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 24.9 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 24.9, 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 24.9 Switching of Internal Clocks and TCNT Operation (1/2)

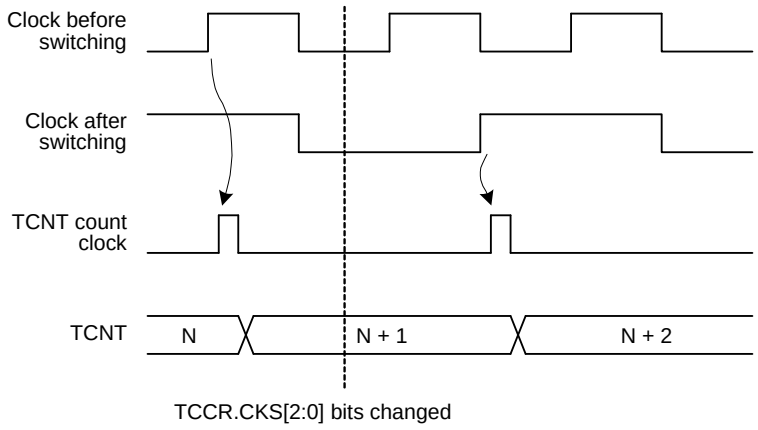
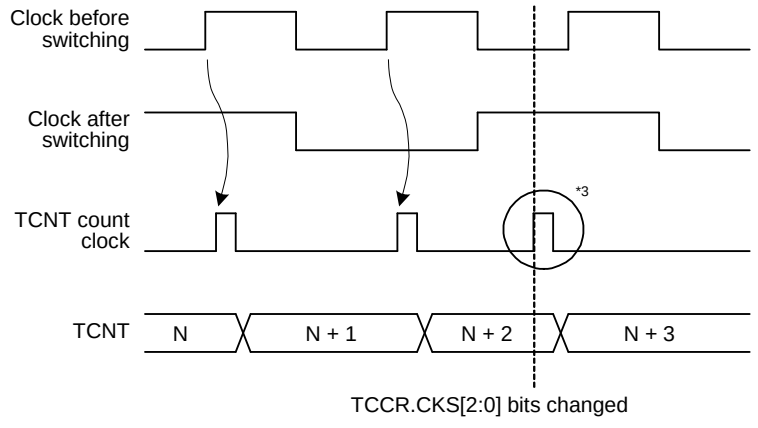
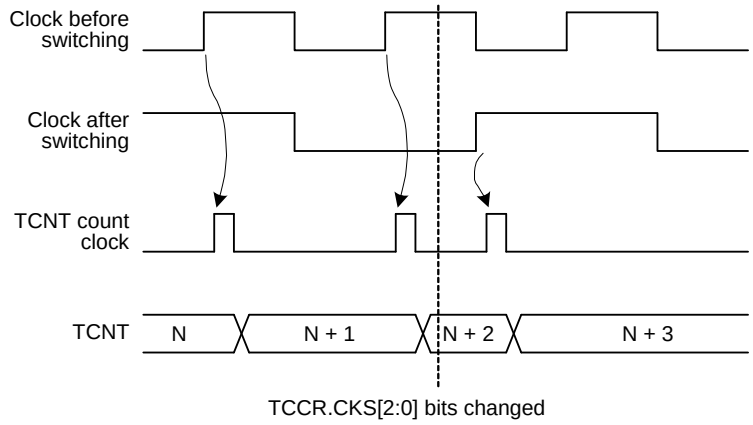
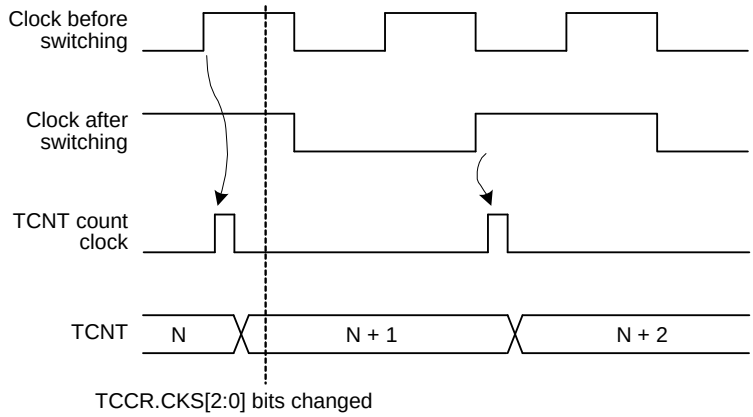
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Table 24.9 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.

24.8.8 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.

24.8.9 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 24.16 shows operation timing when the compare match interrupt signal is continuously output.

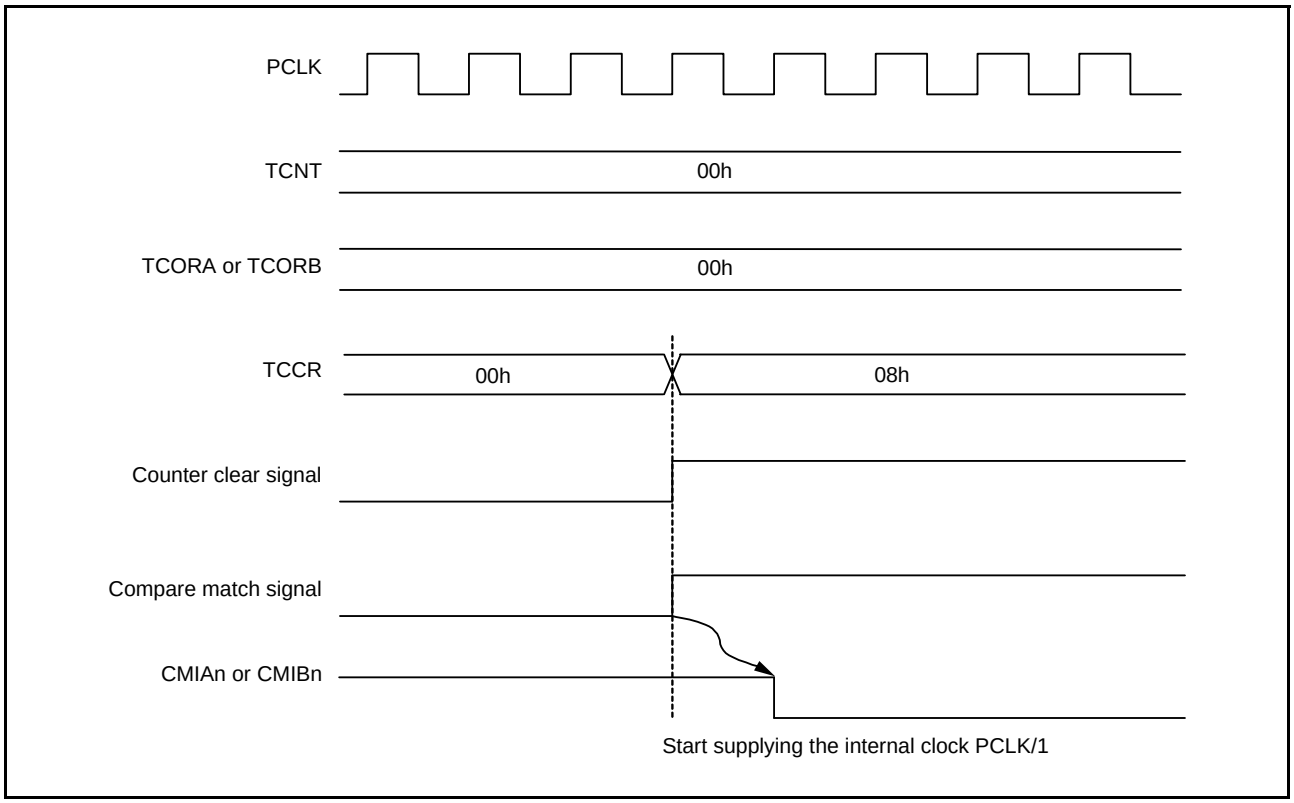


Figure 24.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

25. 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.

25.1 Overview

Table 25.1 lists the specifications for the CMT.
Figure 25.1 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 25.1 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.

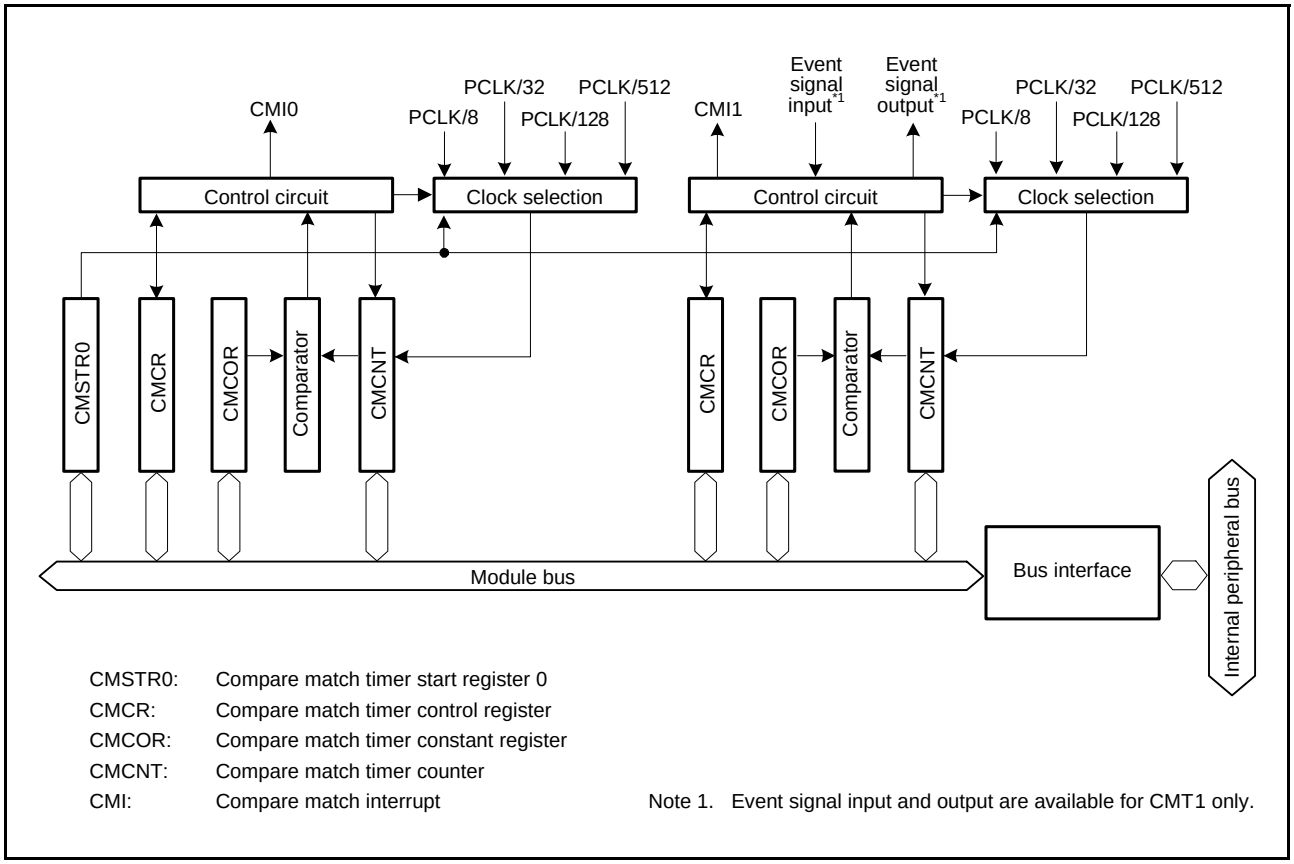


Figure 25.1 CMT (Unit 0) Block Diagram

25.2 Register Descriptions

25.2.1 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

25.2.2 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

25.2.3 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.

25.2.4 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.

25.2.5 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.

25.3 Operation

25.3.1 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 25.2 shows the operation of the CMCNT counter.

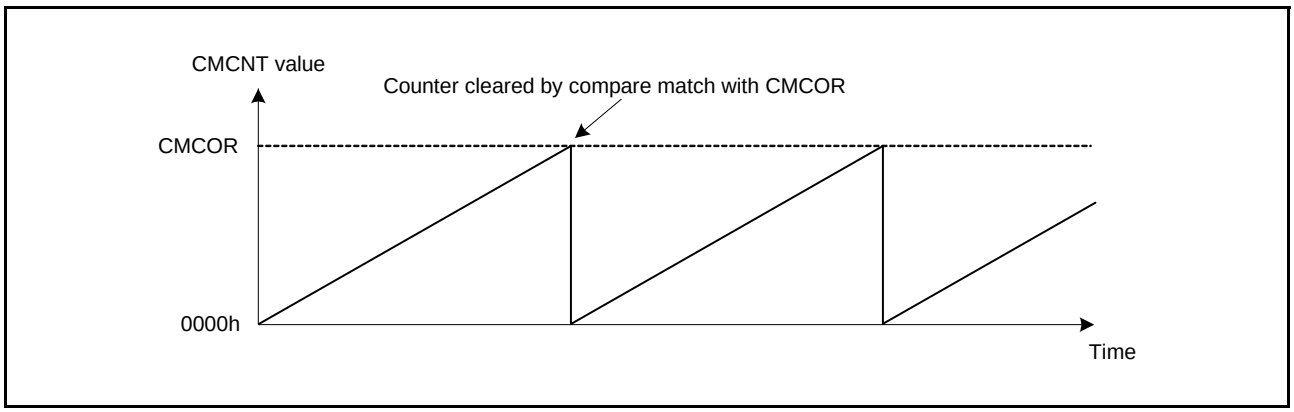


Figure 25.2 CMCNT Counter Operation

25.3.2 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 25.3 shows the timing of the CMCNT counter.

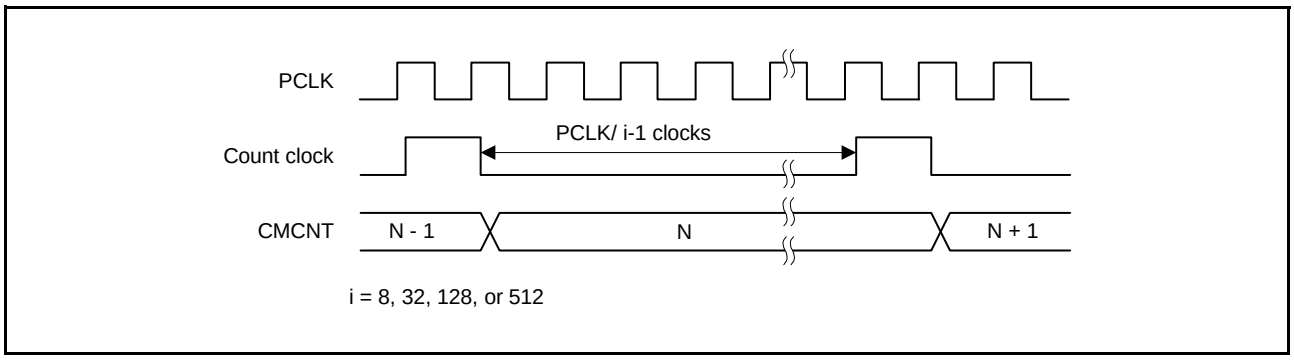


Figure 25.3 CMCNT Count Timing

25.4 Interrupts

25.4.1 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 14, Interrupt Controller (ICUF).

Table 25.2 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

25.4.2 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 25.4 shows the timing of a compare match interrupt.

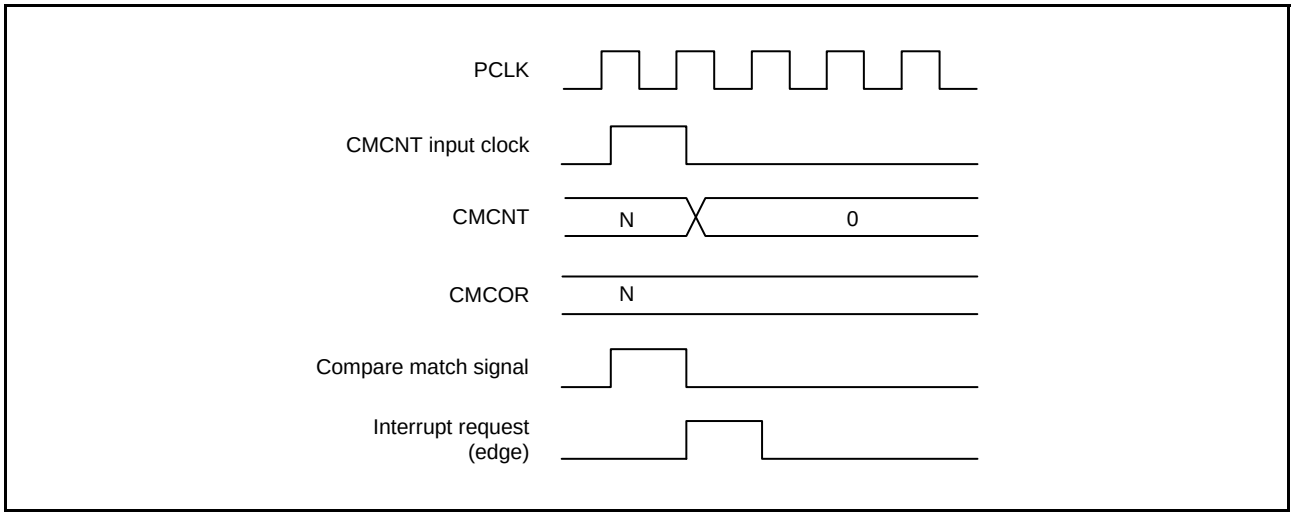


Figure 25.4 Timing of a Compare Match Interrupt

25.5 Link Operations by ELC

25.5.1 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).

25.5.2 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.

25.5.3 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.

25.6 Usage Notes

25.6.1 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.

25.6.2 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 25.5 shows the timing to clear the CMCNT counter.

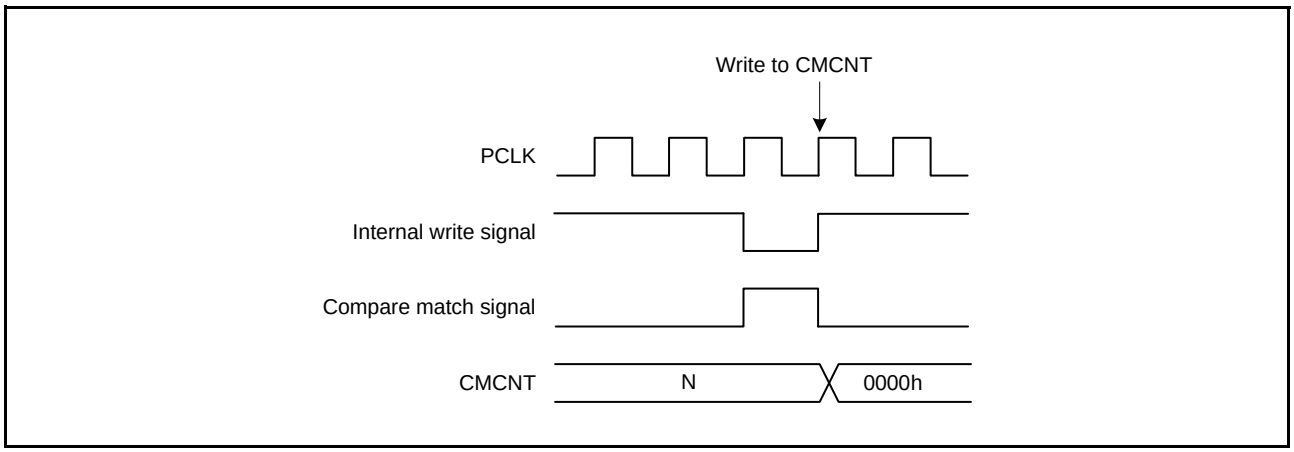


Figure 25.5 Conflict between CMCNT Counter Writing and Compare Match

25.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 25.6 shows the timing to write the CMCNT counter.

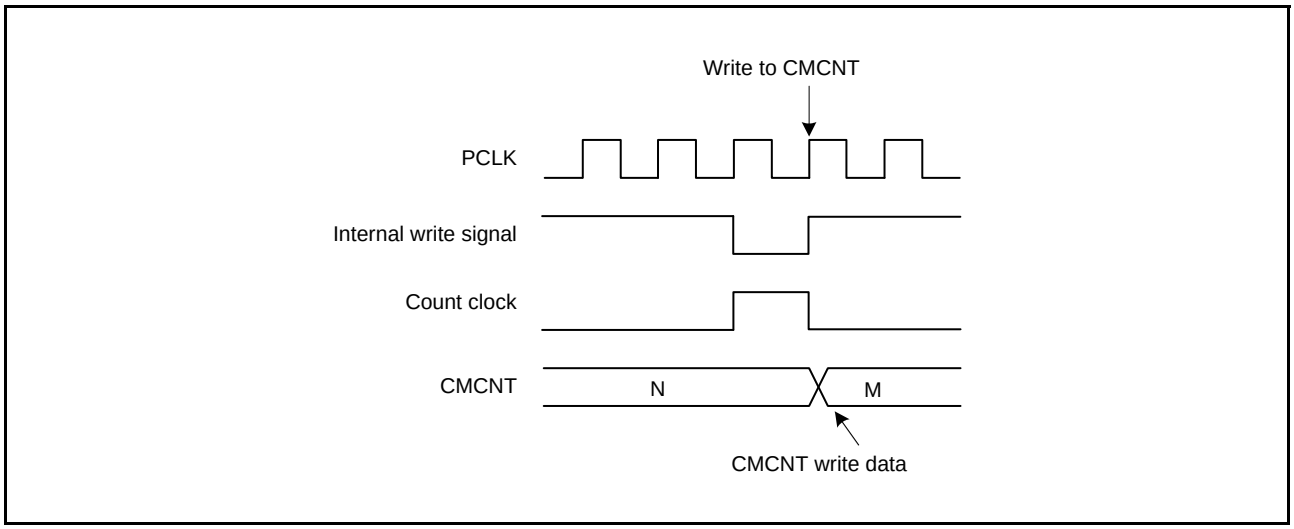


Figure 25.6 Conflict between CMCNT Counter Writing and Incrementing

26. 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.

26.1 Overview

Table 26.1 shows the specifications of the CMTW.
Figure 26.1 shows a block diagram of the CMTW0 and Figure 26.2 shows a block diagram of the CMTW1.

Table 26.1 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
Low power consumption function	Each unit can be placed in the module stop state.

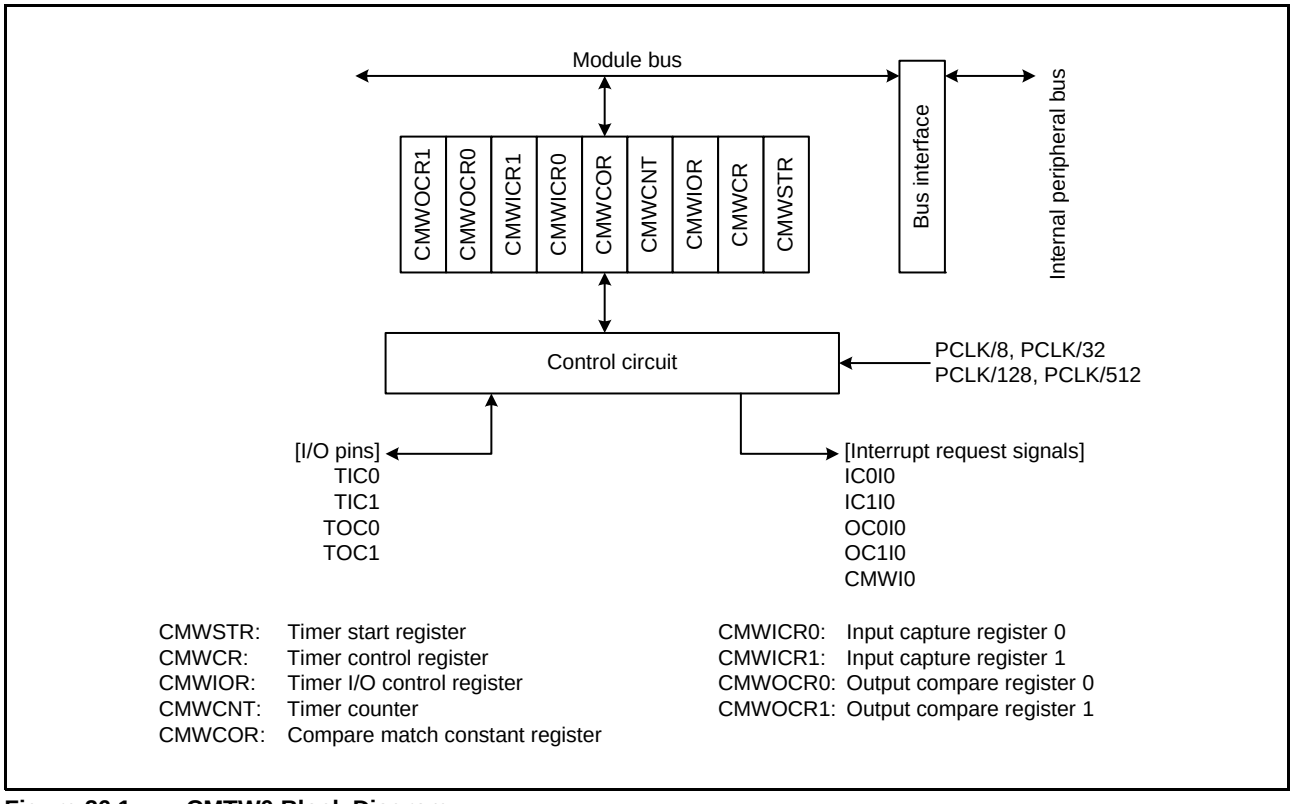


Figure 26.1 CMTW0 Block Diagram

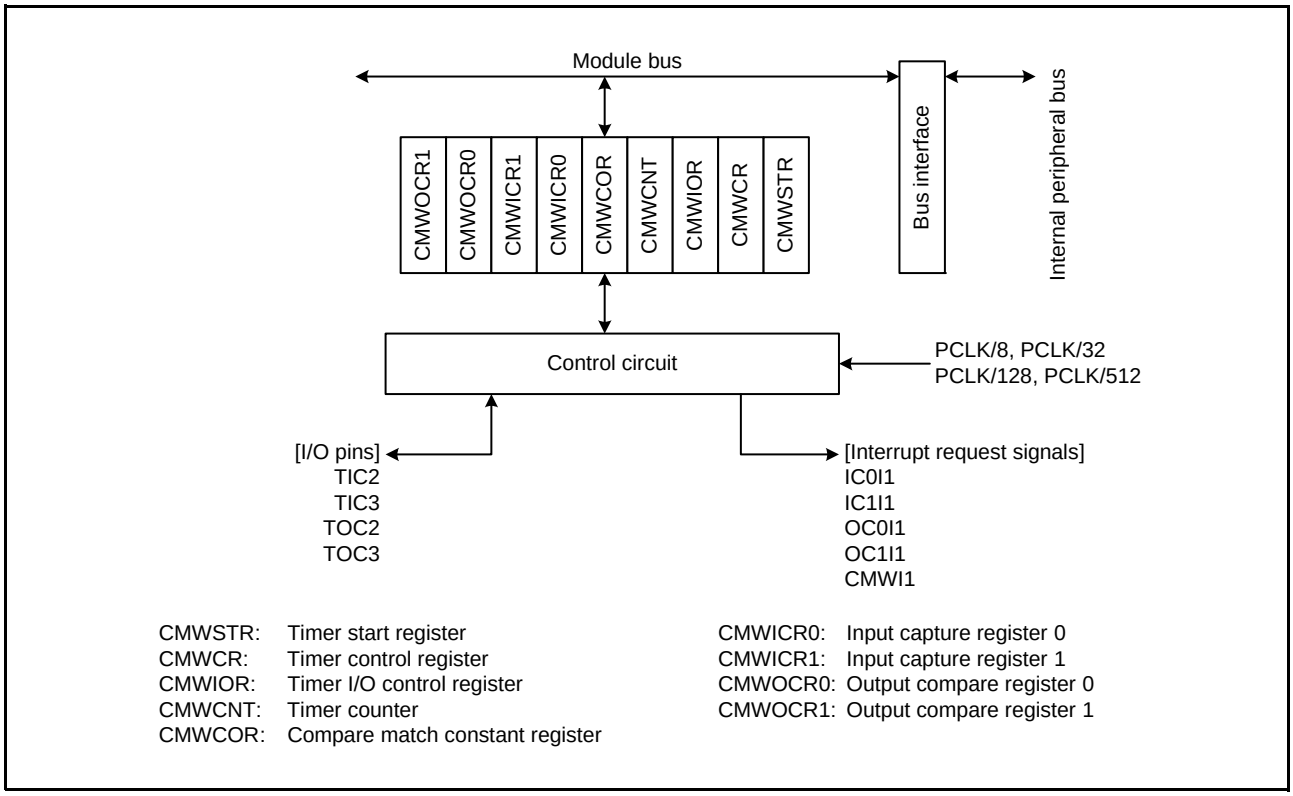


Figure 26.2 CMTW1 Block Diagram

Table 26.2 shows the CMTW pin configuration.

Table 26.2 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

26.2 Register Descriptions

26.2.1 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.

26.2.2 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.

26.2.3 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.

26.2.4 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.

26.2.5 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.

26.2.6 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.

26.2.7 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.

26.3 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.

26.3.1 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.

26.3.2 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 26.3 shows an example of procedure for setting compare match operation.

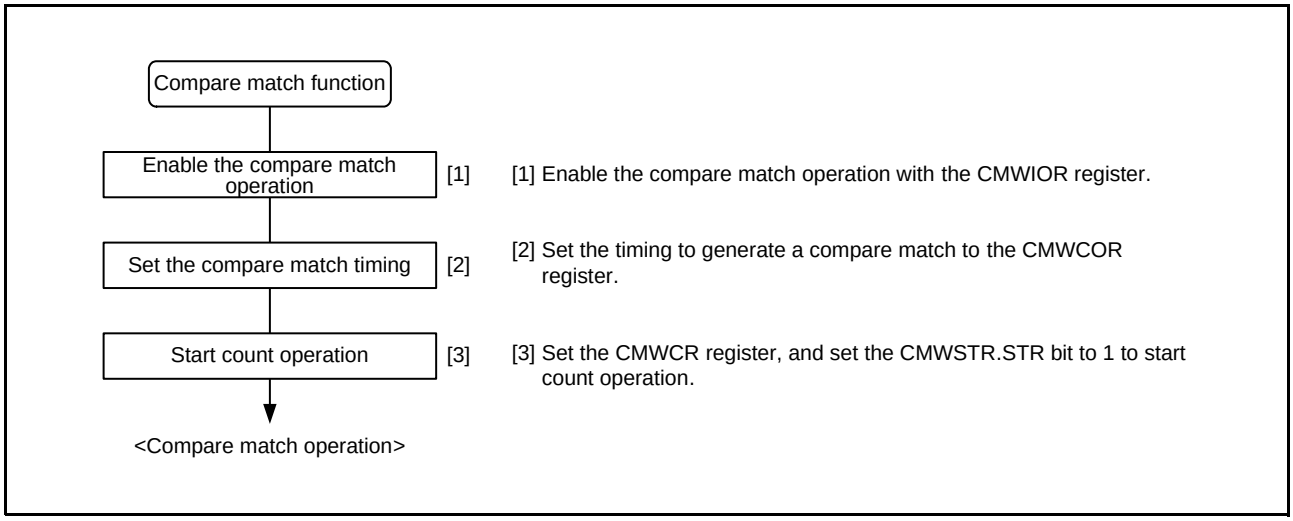


Figure 26.3 Procedure for Setting Compare Match Operation

Figure 26.4 shows an example when compare match with CMWCOR is set as a counter clearing source.

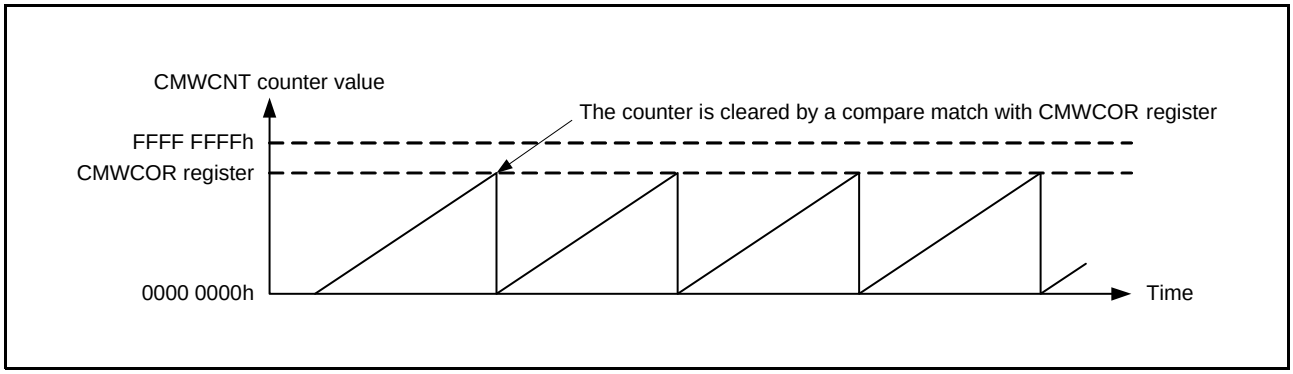


Figure 26.4 Example of Compare Match Operation

Figure 26.5 shows an example when CMWCOR is set to FFFF FFFFh and an overflow is detected.

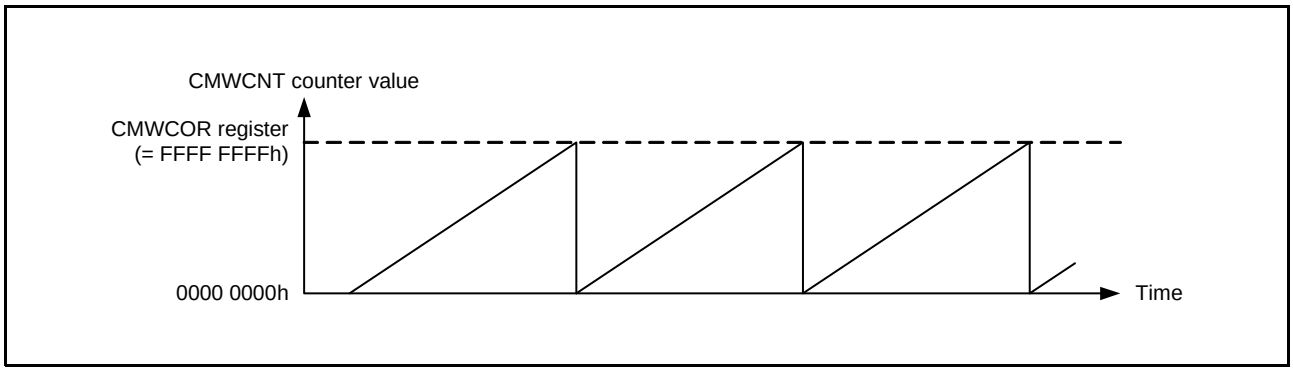


Figure 26.5 Example of Compare Match Operation (Overflow Detected)

26.3.3 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 26.6 shows an example of procedure for setting output compare operation.

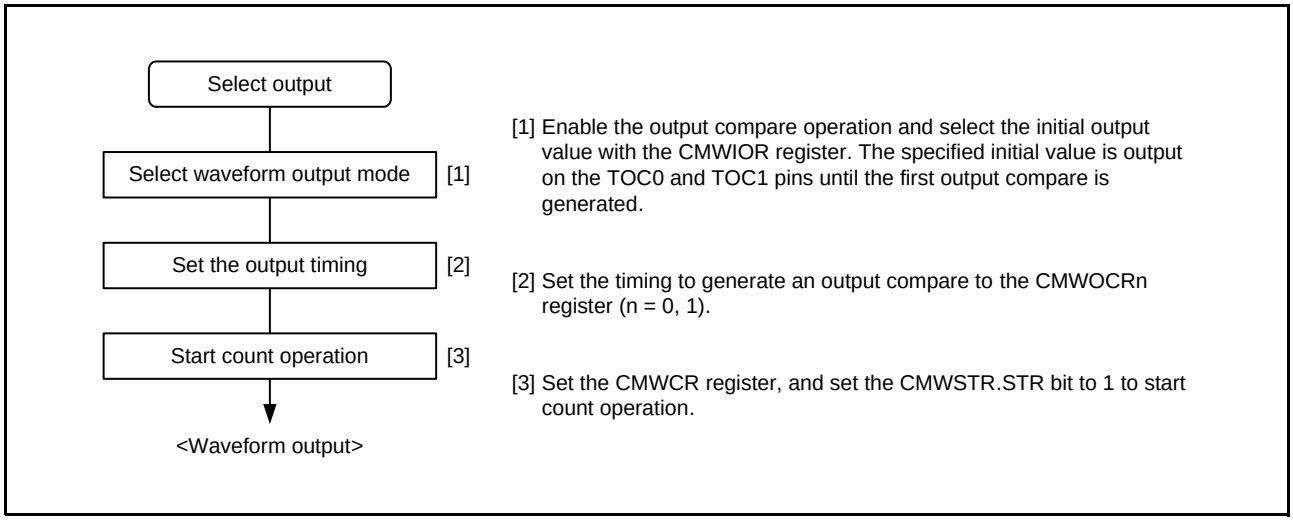


Figure 26.6 Procedure for Setting Output Compare Operation

Figure 26.7 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.

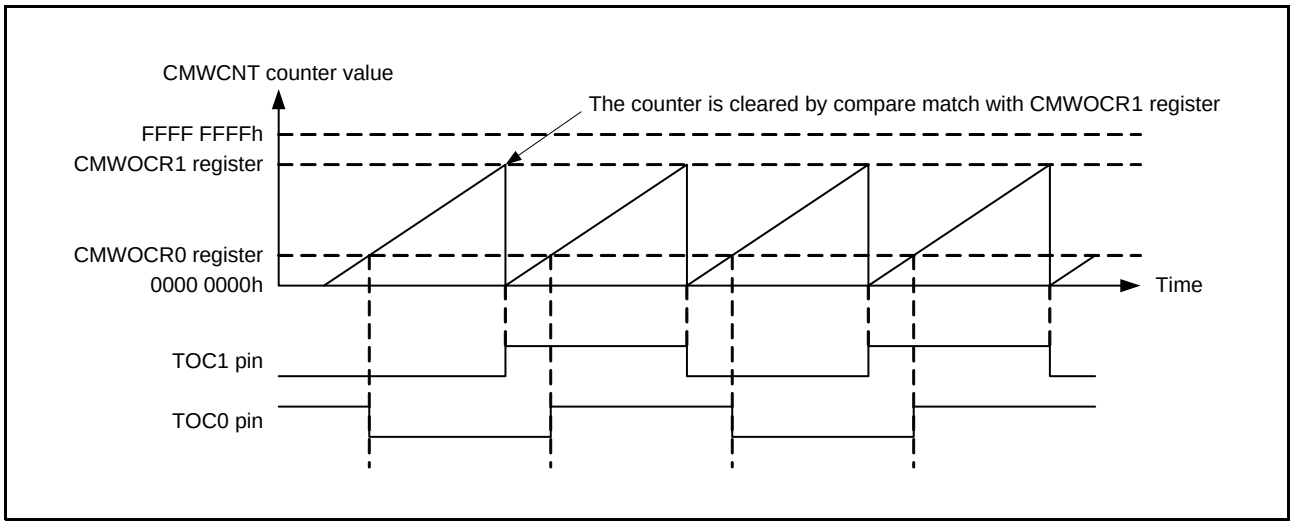


Figure 26.7 Example of Output Compare Operation (Unit 0)

26.3.4 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 26.8 shows an example of procedure for setting input capture operation.

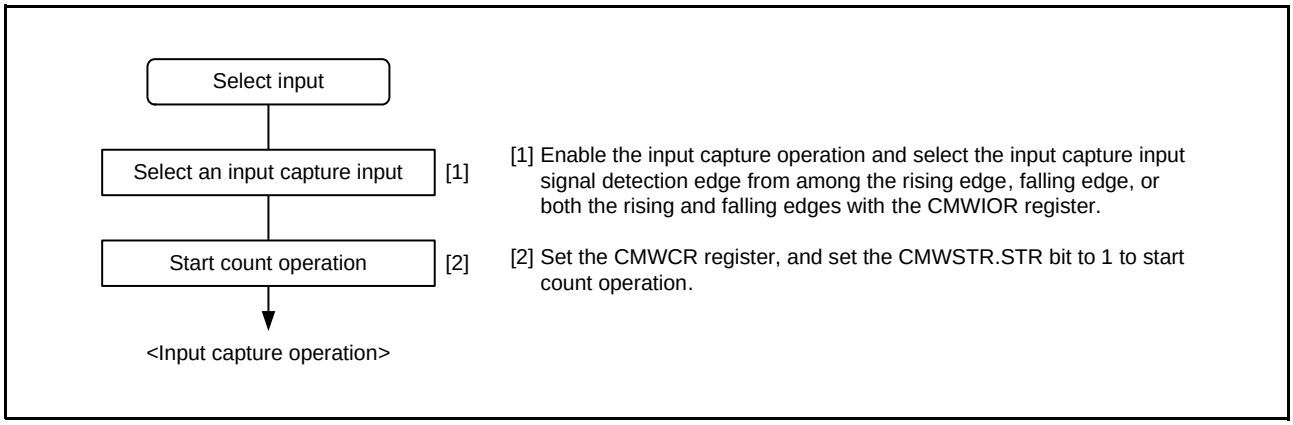


Figure 26.8 Procedure for Setting Input Capture Operation

Figure 26.9 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.

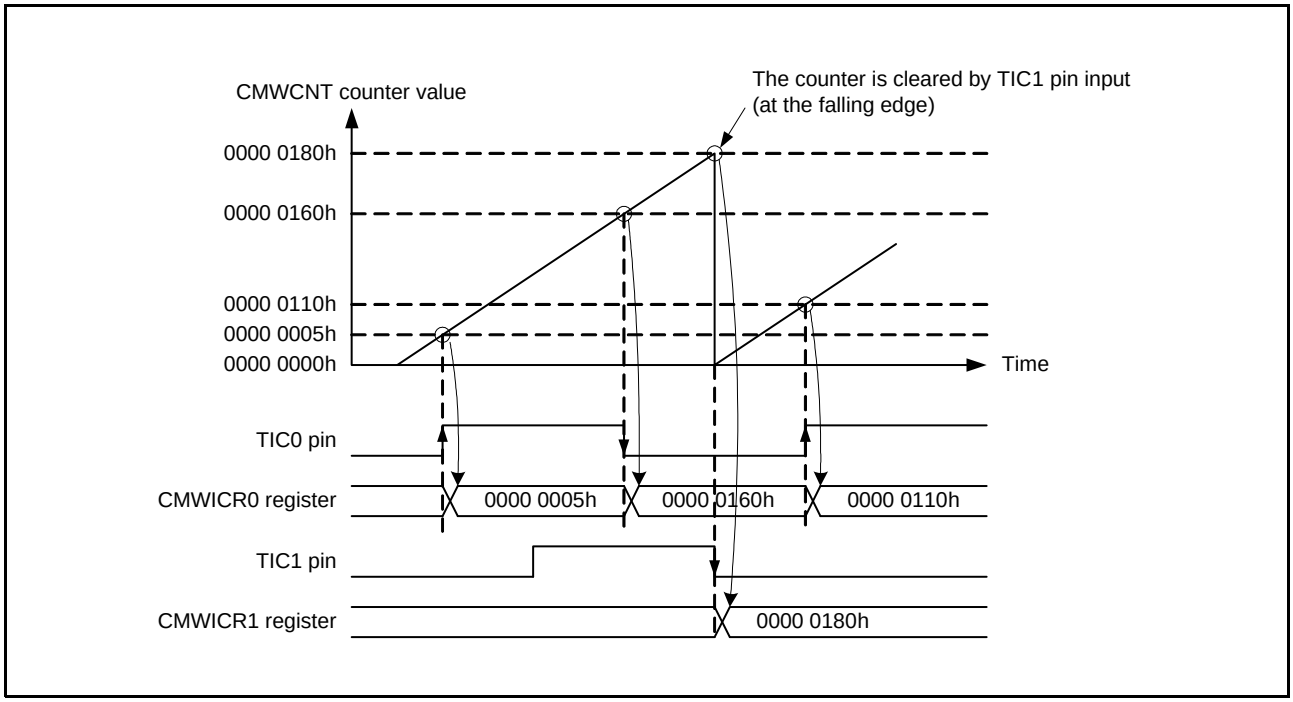


Figure 26.9 Example of Input Capture Operation (Unit 0)

26.3.5 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.

26.3.6 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 26.10 shows the timing.

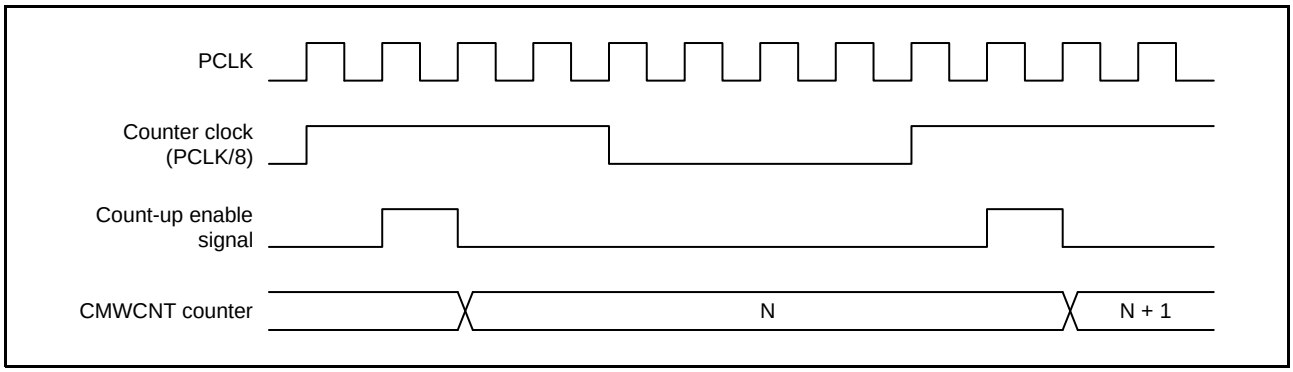


Figure 26.10 Count Timing (PCLK/8)

26.3.7 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 26.11 shows output compare output timing.

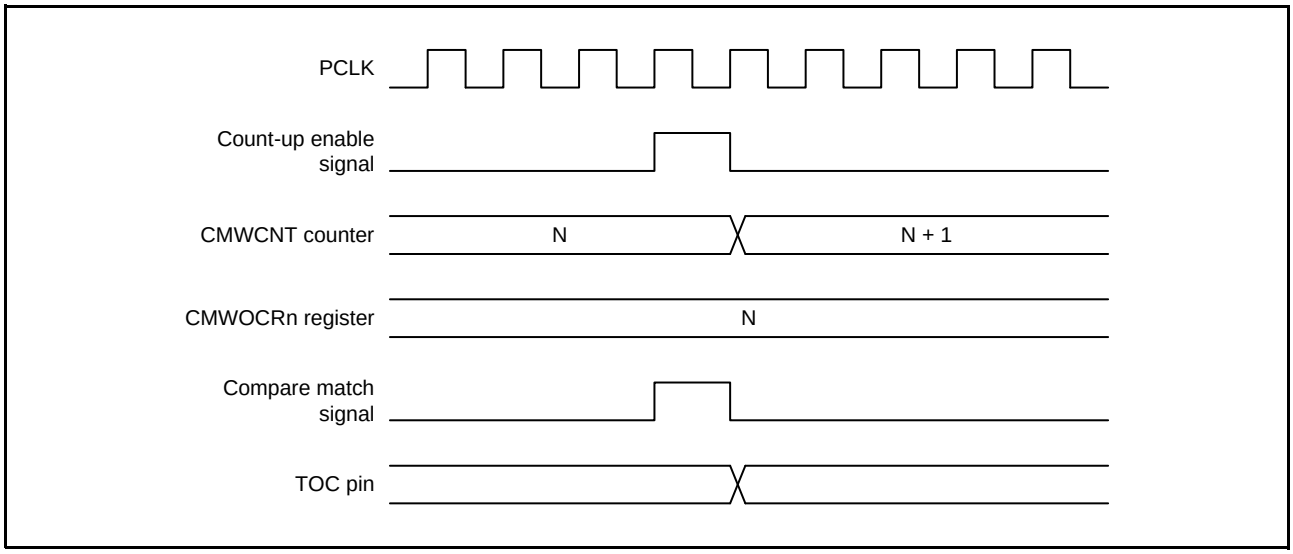


Figure 26.11 Output Compare Output Timing

26.3.8 Input Capture Timing

Figure 26.12 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.

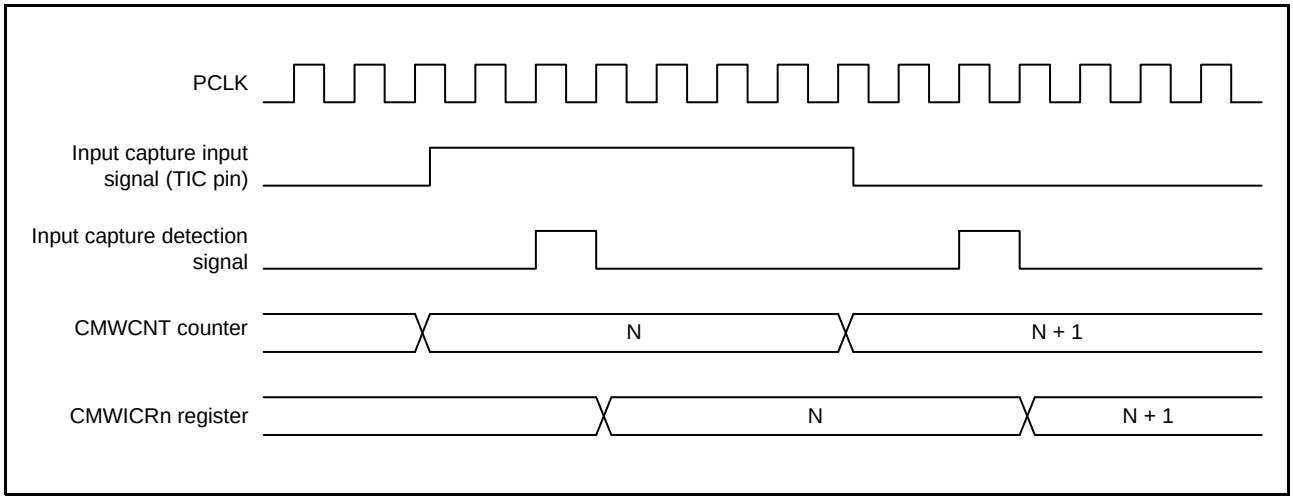


Figure 26.12 Input Capture Timing (Unit 0, Both-Edge Detection) (n = 0, 1)

26.4 Interrupts

26.4.1 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 26.3 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 18, Data Transfer Controller (DTCb).

Table 26.3 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

26.4.2 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 26.13 shows the timing of compare match interrupt generation.

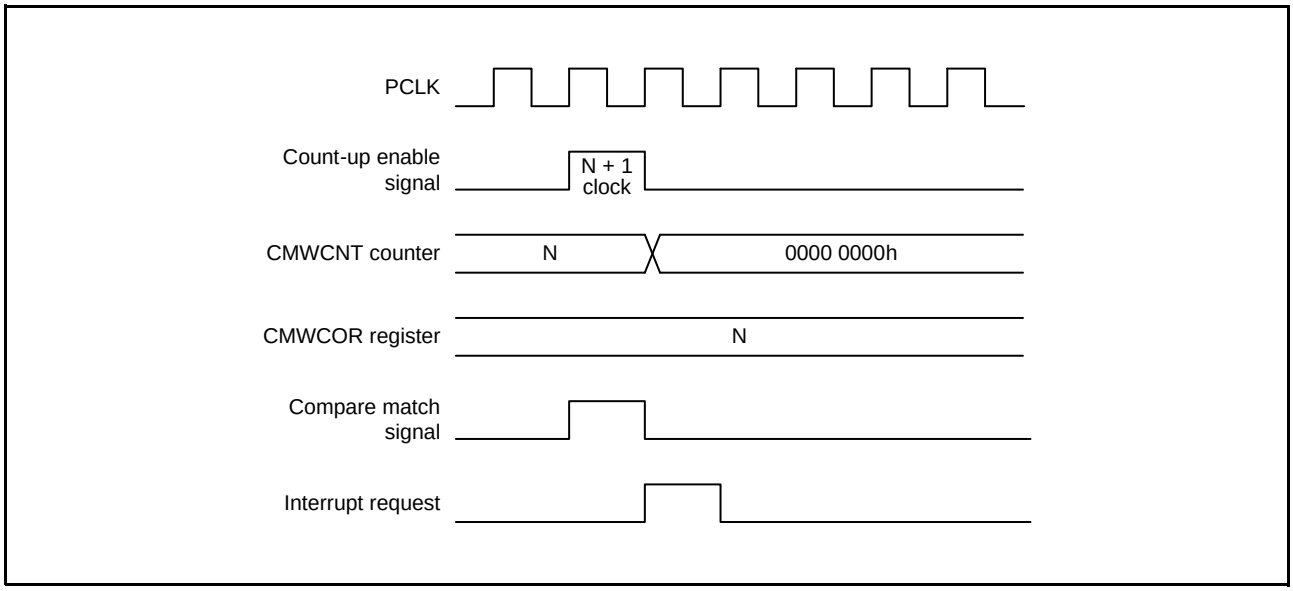


Figure 26.13 Timing of Compare Match Interrupt Generation

(a) Timing of Output Compare Interrupt Generation

Figure 26.14 shows the timing of output compare interrupt generation.

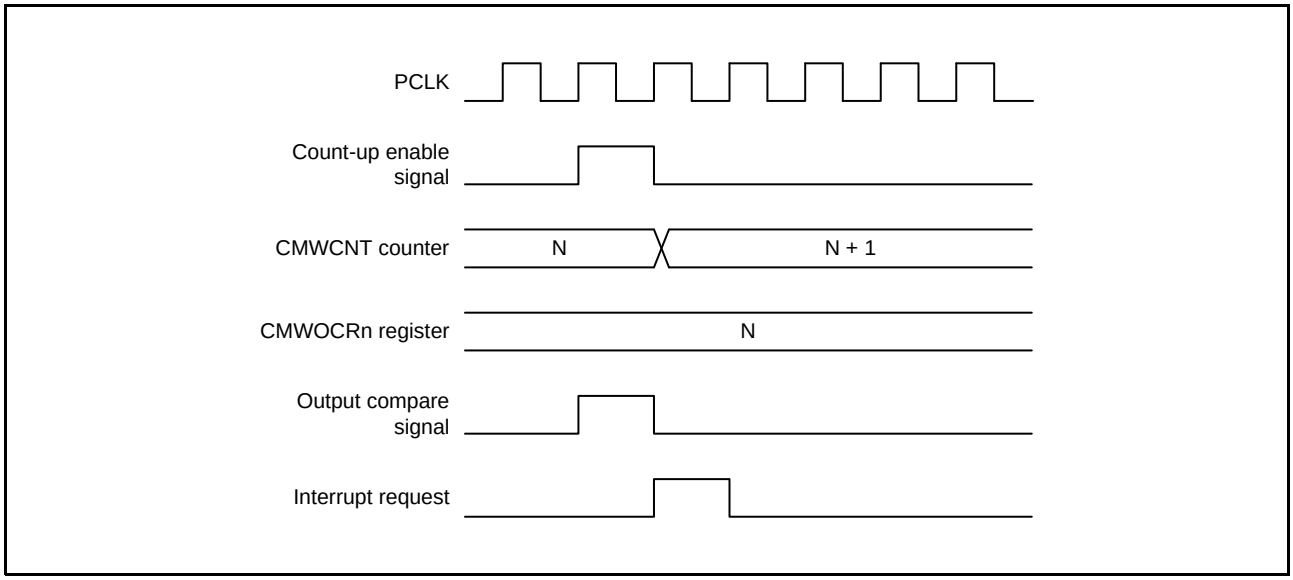


Figure 26.14 Timing of Output Compare Interrupt Generation (n = 0, 1)

(b) Timing of Input Capture Interrupt Generation

Figure 26.15 shows the timing of input capture interrupt generation.

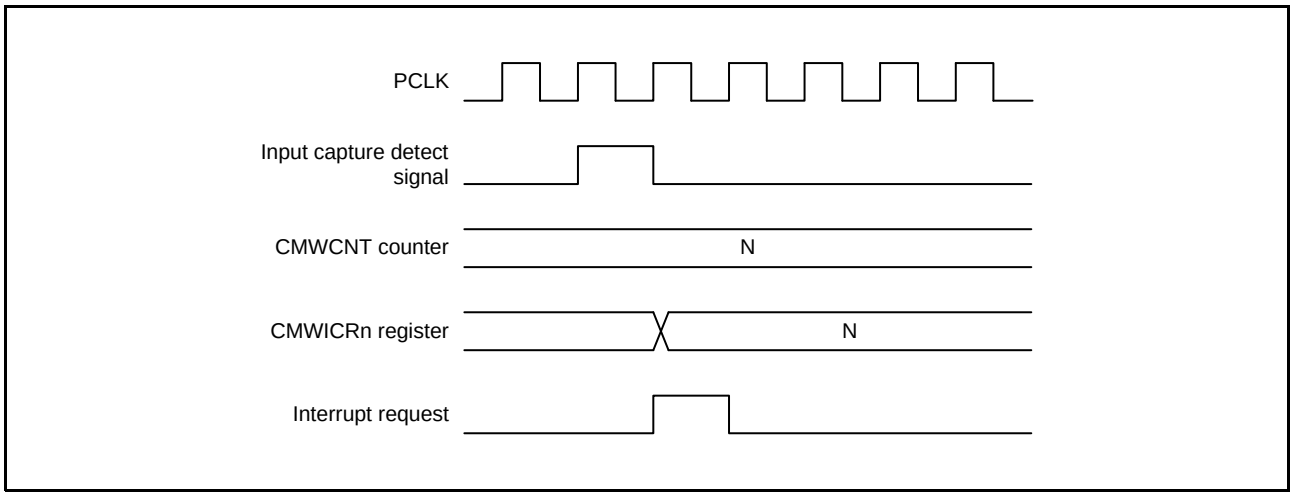


Figure 26.15 Timing of Input Capture Interrupt Generation (n = 0, 1)

26.5 Usage Notes

26.5.1 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.

26.5.2 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 26.16 shows the timing of conflict between CMWCNT counter writing and compare match.

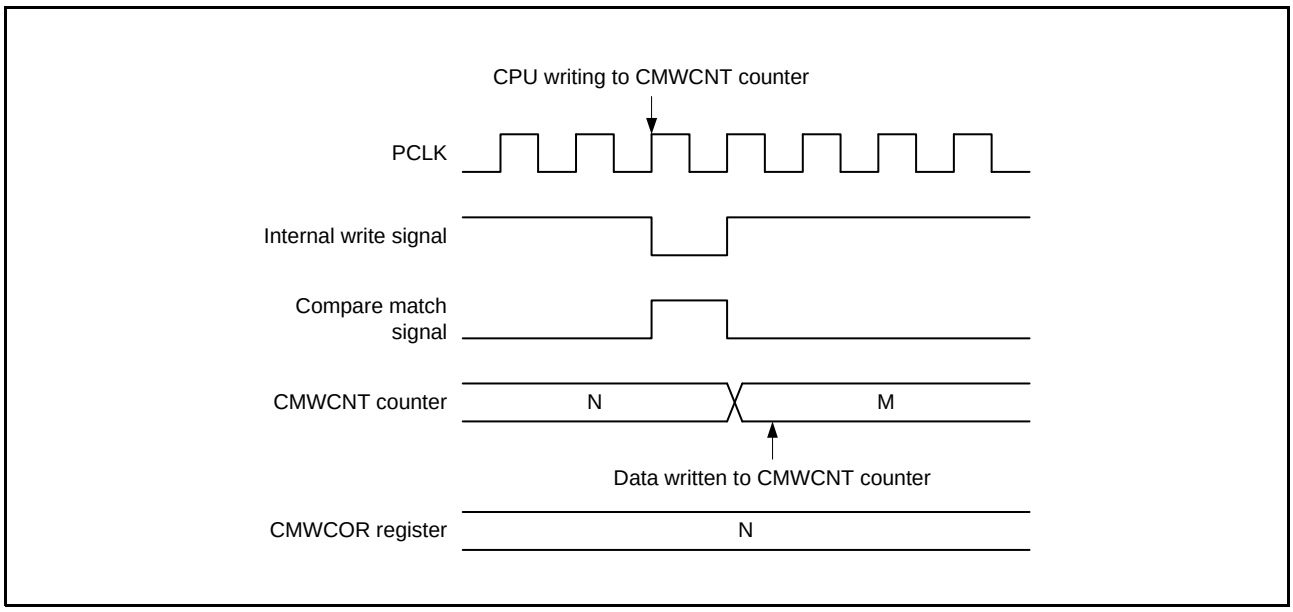


Figure 26.16 Conflict between CMWCNT Counter Writing and Compare Match

26.5.3 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 26.17 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

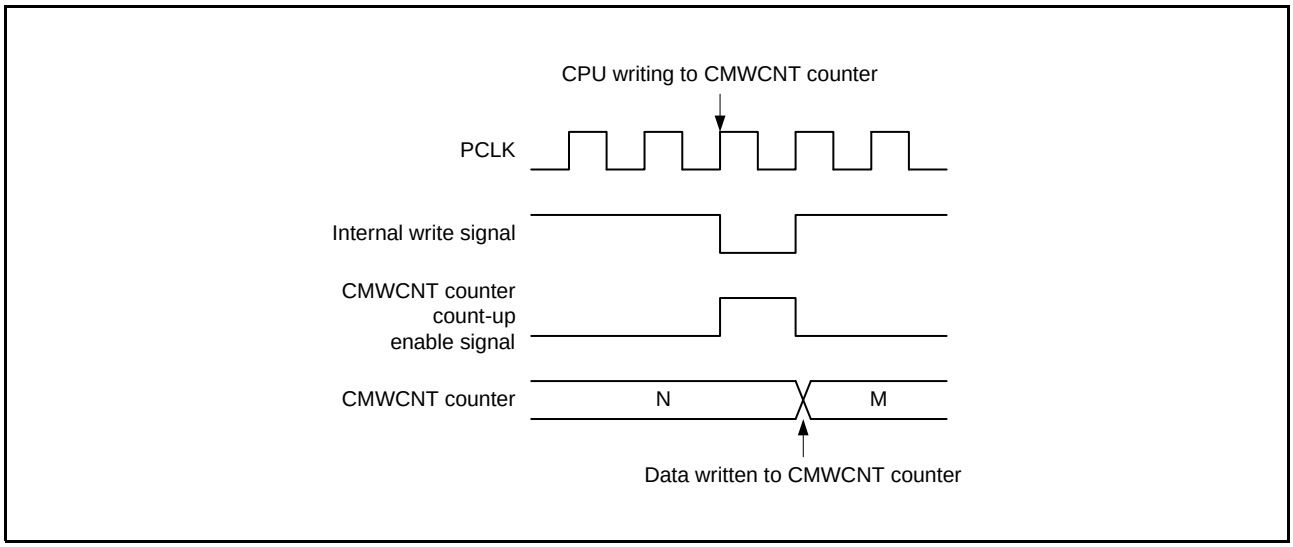


Figure 26.17 Conflict between CMWCNT Counter Writing and Incrementing

26.5.4 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 26.18 shows the timing of conflict between CMWCOR register writing and compare match.

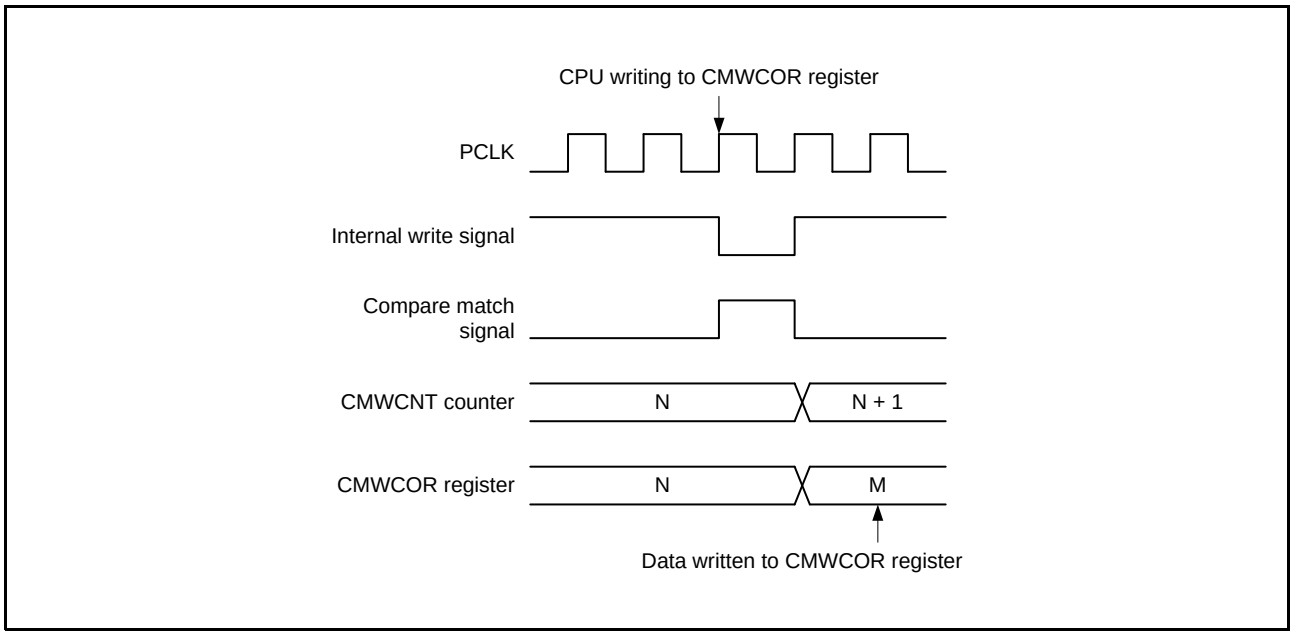


Figure 26.18 Conflict between CMWCOR Register Writing and Compare Match

26.5.5 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 26.19 shows the timing of conflict between CMWOCRn register writing and compare match.

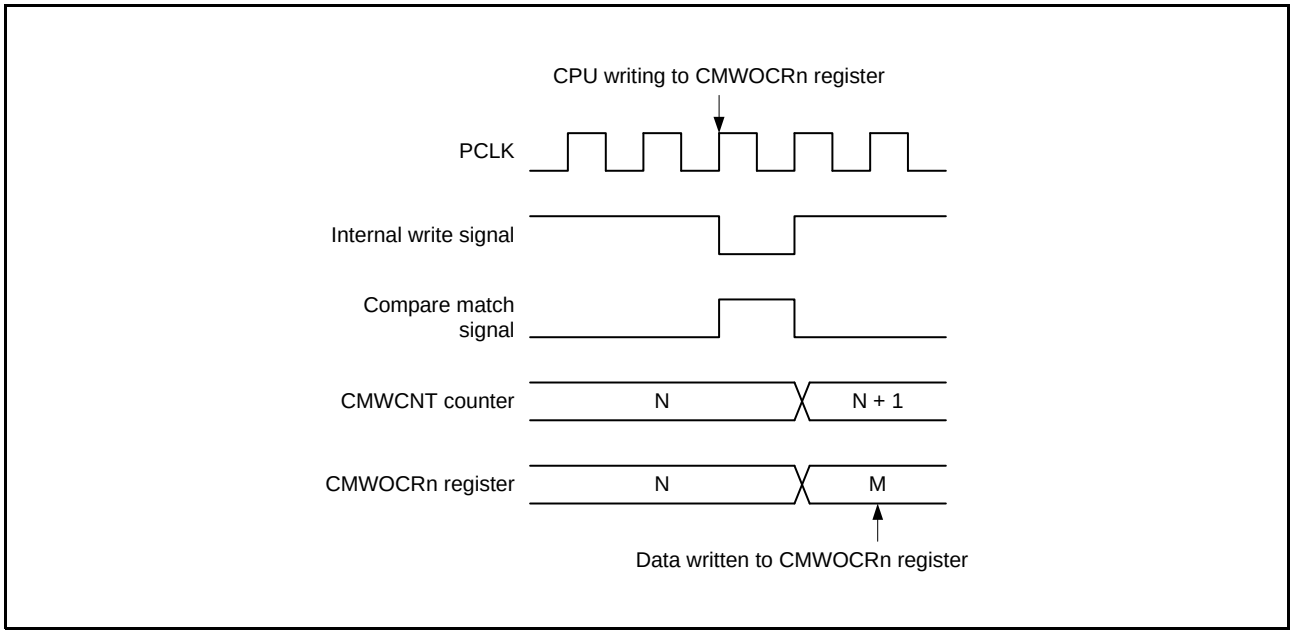


Figure 26.19 Conflict between CMWOCRn Register Writing and Compare Match

26.5.6 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 26.20 shows the timing of conflict between CMWCNT counter reading and incrementing.

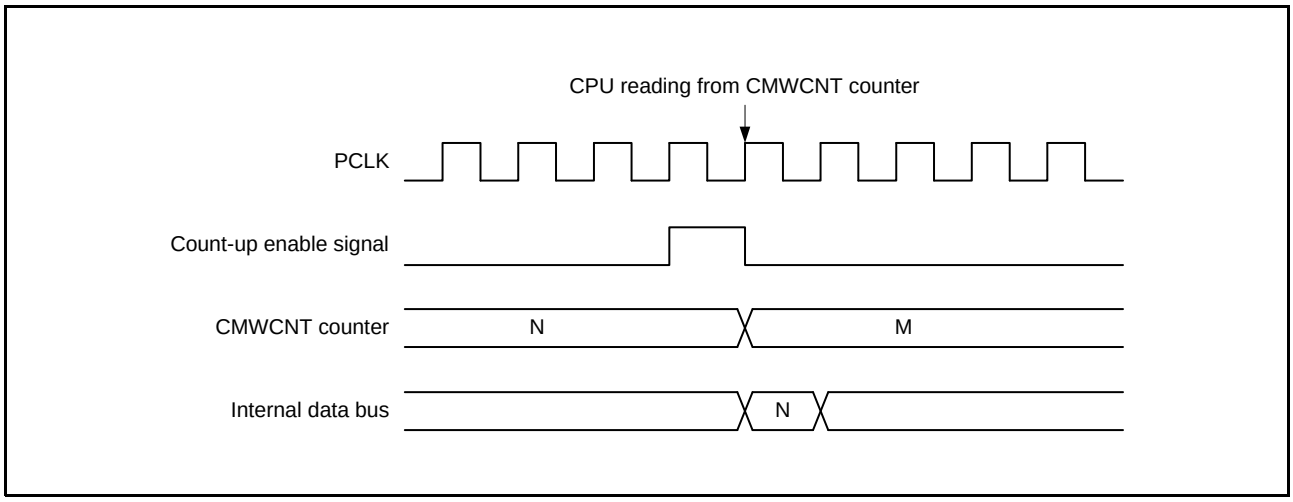


Figure 26.20 Conflict between CMWCNT Counter Reading and Incrementing

26.5.7 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 26.21 shows the timing of conflict between CMWICRn register reading and input capture.

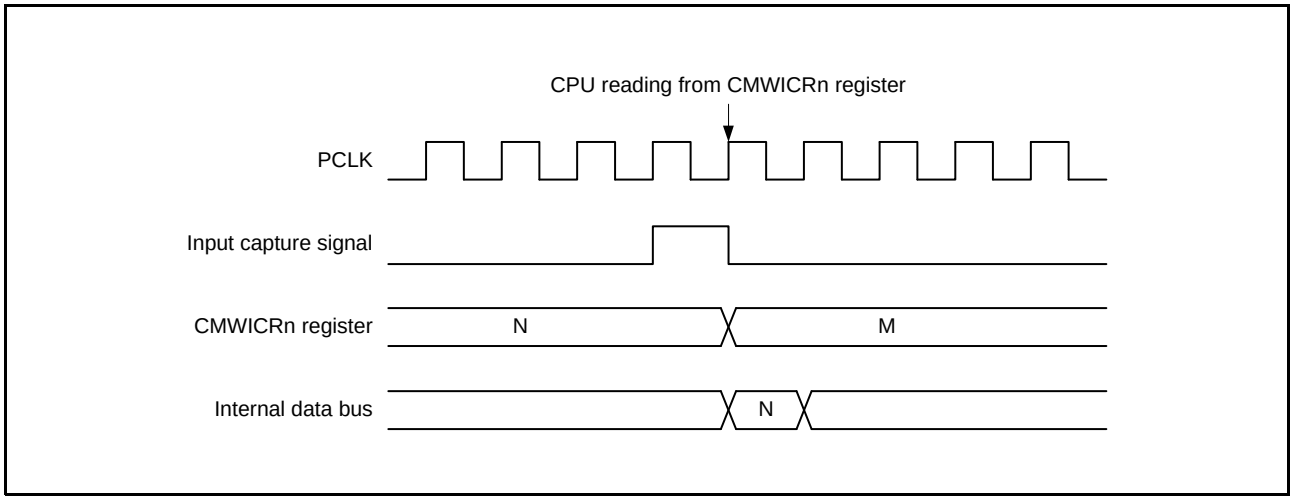


Figure 26.21 Conflict between CMWICRn Register Reading and Input Capture

27. Realtime Clock (RTCC)

In this section, “PCLK” is used to refer to PCLKB.

27.1 Overview

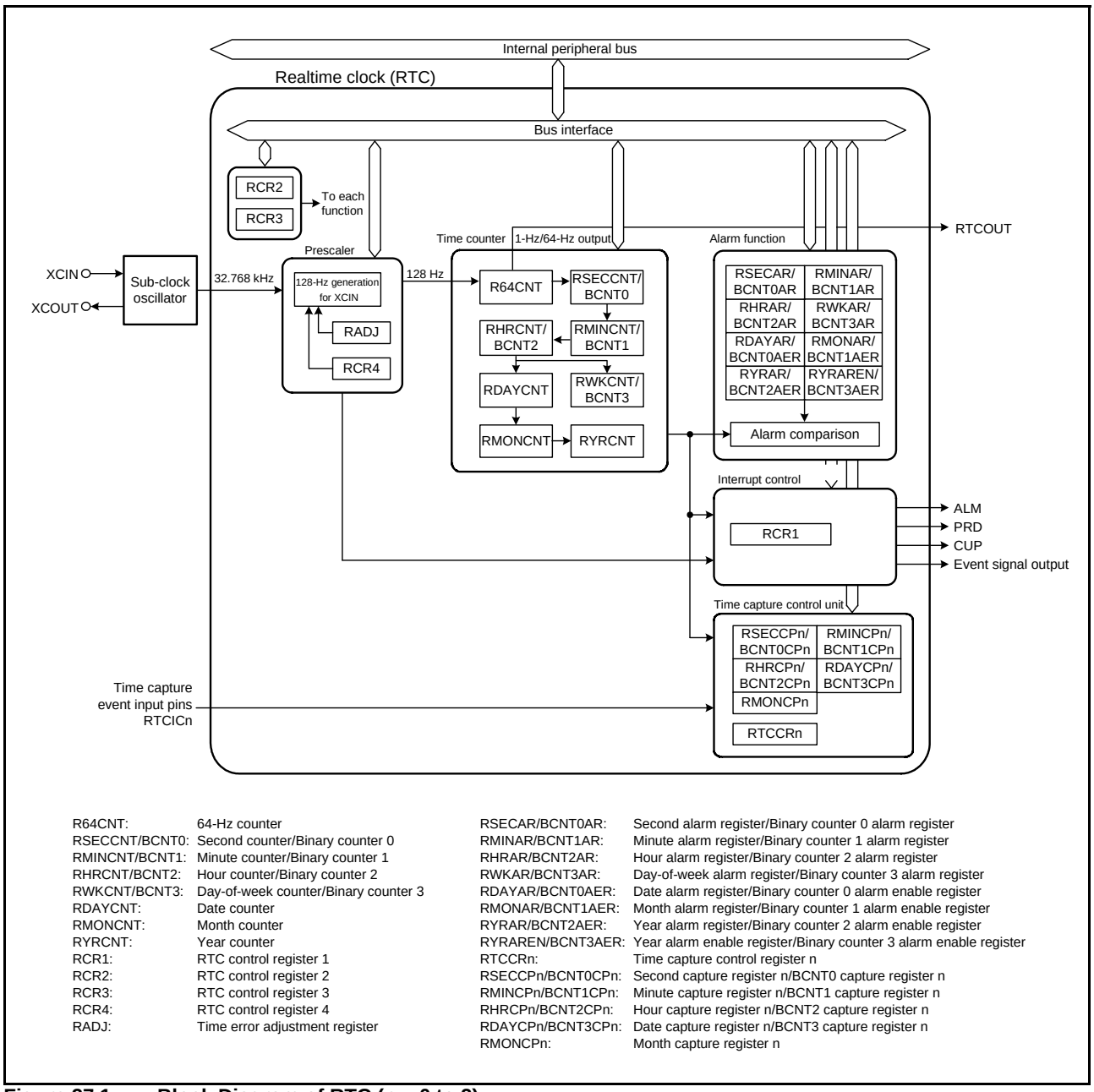
The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.
For binary count mode, the RTC does not count 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 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 27.1 lists the specifications of the RTC, Figure 27.1 shows a block diagram of the RTC, and Table 27.2 shows the pin configuration of the RTC.

Table 27.1 RTC Specifications

Item	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN)
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
RSECCNT/BCNT0:	Second counter/Binary counter 0
RMINCNT/BCNT1:	Minute counter/Binary counter 1
RHRCNT/BCNT2:	Hour counter/Binary counter 2
RWKCNT/BCNT3:	Day-of-week counter/Binary counter 3
RDAYCNT:	Date counter
RMONCNT:	Month counter
RYRCNT:	Year counter
RCR1:	RTC control register 1
RCR2:	RTC control register 2
RCR3:	RTC control register 3
RCR4:	RTC control register 4
RADJ:	Time error adjustment register

RSECAR/BCNT0AR:	Second alarm register/Binary counter 0 alarm register
RMINAR/BCNT1AR:	Minute alarm register/Binary counter 1 alarm register
RHRAR/BCNT2AR:	Hour alarm register/Binary counter 2 alarm register
RWKAR/BCNT3AR:	Day-of-week alarm register/Binary counter 3 alarm register
RDAYAR/BCNT0AER:	Date alarm register/Binary counter 0 alarm enable register
RMONAR/BCNT1AER:	Month alarm register/Binary counter 1 alarm enable register
RYRAR/BCNT2AER:	Year alarm register/Binary counter 2 alarm enable register
RYRAREN/BCNT3AER:	Year alarm enable register/Binary counter 3 alarm enable register
RTCCRn:	Time capture control register n
RSECCPn/BCNT0CPn:	Second capture register n/BCNT0 capture register n
RMINCPn/BCNT1CPn:	Minute capture register n/BCNT1 capture register n
RHRCPn/BCNT2CPn:	Hour capture register n/BCNT2 capture register n
RDAYCPn/BCNT3CPn:	Date capture register n/BCNT3 capture register n
RMONCPn:	Month capture register n

Figure 27.1 Block Diagram of RTC (n = 0 to 2)

Table 27.2 Pin Configuration of RTC

Pin Name	I/O	Function
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	

27.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 27.6.5, 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. 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 27.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

27.2.1 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.

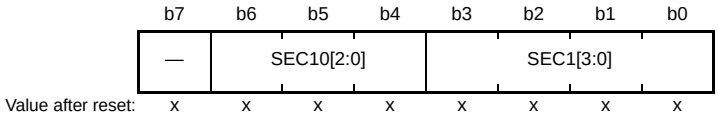
This counter is set to 00h by an RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 27.3.5, Reading 64-Hz Counter and Time.

27.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 0008 C402h



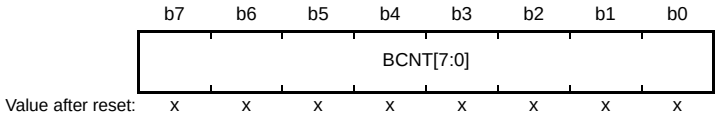
x: Undefined

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 27.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT0 0008 C402h



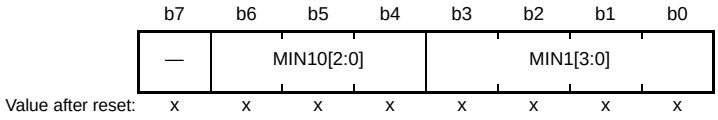
x: Undefined

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 27.3.5, Reading 64-Hz Counter and Time.

27.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 0008 C404h



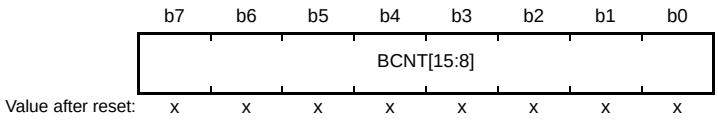
x: Undefined

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 27.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT1 0008 C404h



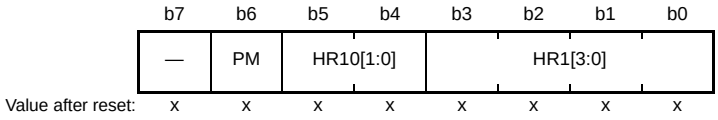
x: Undefined

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 27.3.5, Reading 64-Hz Counter and Time.

27.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 0008 C406h



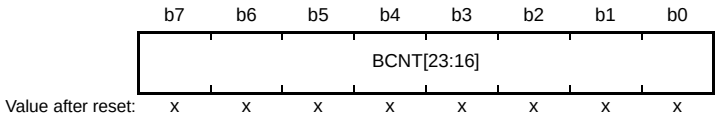
x: Undefined

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 27.6.5, 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 27.3.5, Reading 64-Hz Counter and Time.

27.2.5 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 27.6.5, 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 27.3.5, Reading 64-Hz Counter and Time.

27.2.6 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 27.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

27.2.7 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 27.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

27.2.8 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 27.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

27.2.9 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 27.6.5, 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

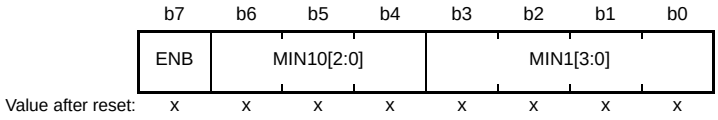
x: Undefined

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.

27.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 0008 C412h



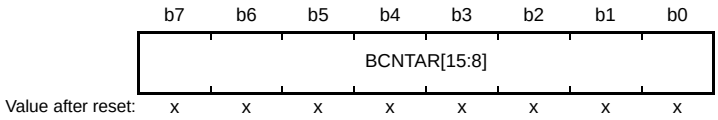
x: Undefined

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 27.6.5, 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.

27.2.11 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 27.6.5, 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.

27.2.12 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 27.6.5, 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.

27.2.13 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 27.6.5, 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.

27.2.14 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 27.6.5, 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.

27.2.15 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 27.6.5, 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.

27.2.16 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.

27.2.17 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. 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

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 21.4.1, 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.

27.2.18 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*3	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	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3	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, *3	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select*3	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. After writing to this bit, confirm that its value has actually changed before proceeding with further processing. Refer to section 27.6.5, 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 27.3.1, Outline of Initial Settings of Registers after Power On.

27.2.19 RTC Control Register 3 (RCR3)

Address(es): RTC.RCR3 0008 C426h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	RTC Enable	0: RTC disabled 1: RTC enabled	R/W
b3 to b1	—	Reserved	Set these bits to 0. They are read as the set value.	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 to enable or disable the RTC operation. Set this register to 00h in products with no RTC.
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.

27.2.20 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: Setting prohibited	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.

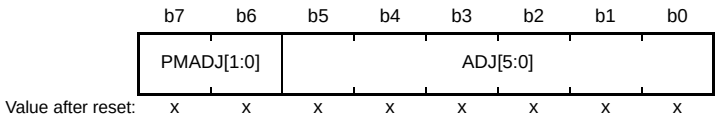
RCKSEL Bit (Count Source Select)

This bit is used to set the count source.

Set this bit to 0 before the initial settings of the RTC registers at power on.

27.2.21 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.

ADJ[5:0] Bits (Adjustment Value)

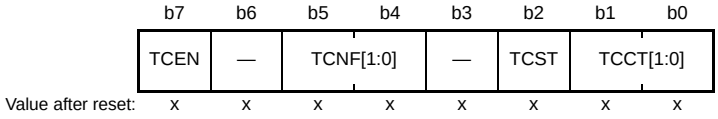
These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

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.

27.2.22 Time Capture Control Register n (RTCCRn) (n = 0 to 2)

Address(es): RTC.RTCCR0 0008 C440h, RTC.RTCCR1 0008 C442h, RTC.RTCCR2 0008 C444h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected. 0 1: Rising edge is detected. 1 0: Falling edge is detected. 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status	0: No event is detected. 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: The noise filter is off. 0 1: Setting prohibited 1 0: The noise filter is on (count source). 1 1: The noise filter is on (count source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input. 1: The RTCICn pin is enabled as the time capture event input.	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 RTC is disabled (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 and then set this bit to 1. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00b.

27.2.23 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

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	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

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPn[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

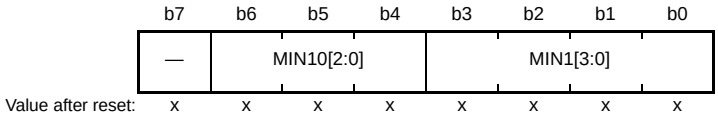
x: Undefined

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.

27.2.24 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



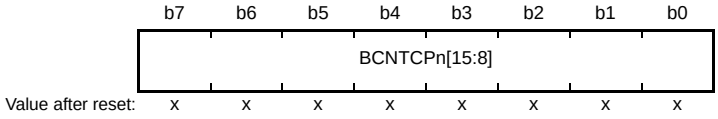
x: Undefined

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.

27.2.25 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

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

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

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPn[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

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.

27.2.26 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.

27.2.27 Month Capture Register n (RMONCPn) (n = 0 to 2)

Address(es): RTC.RMONCP0 0008 C45Ch, RTC.RMONCP1 0008 C46Ch, RTC.RMONCP2 0008 C47Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	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.

27.3 Operation

27.3.1 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.

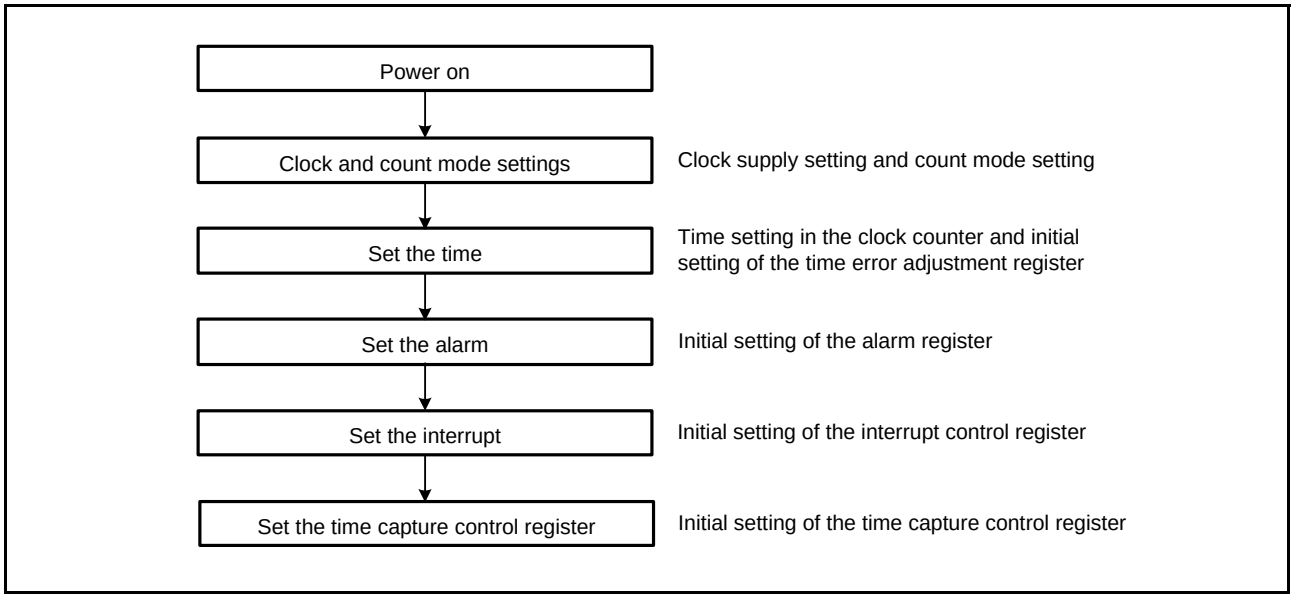
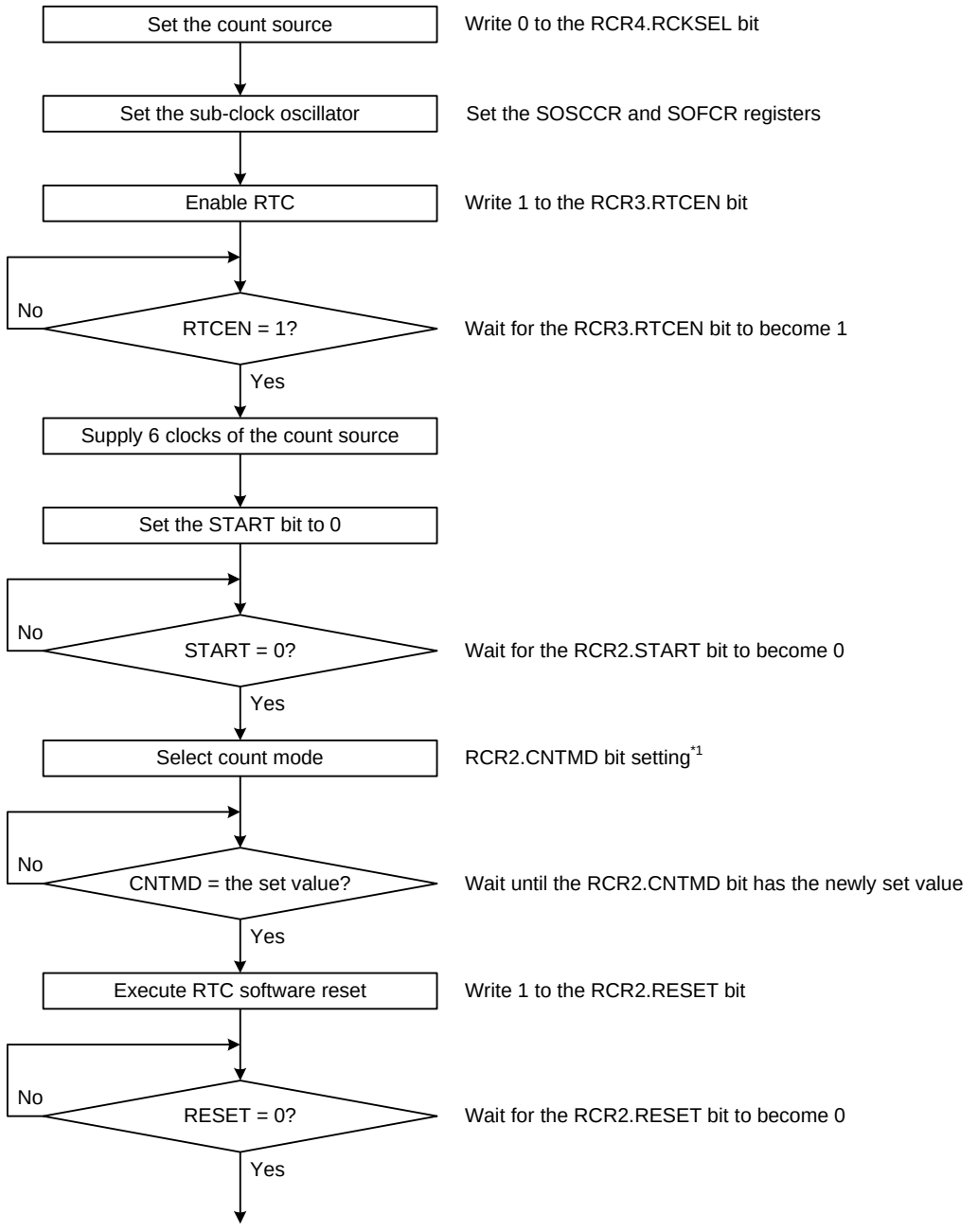


Figure 27.2 Outline of Initial Settings after Power On

27.3.2 Clock and Count Mode Setting Procedure

Figure 27.3 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 27.3 Clock and Count Mode Setting Procedure

27.3.3 Setting the Time

Figure 27.4 shows how to set the time.

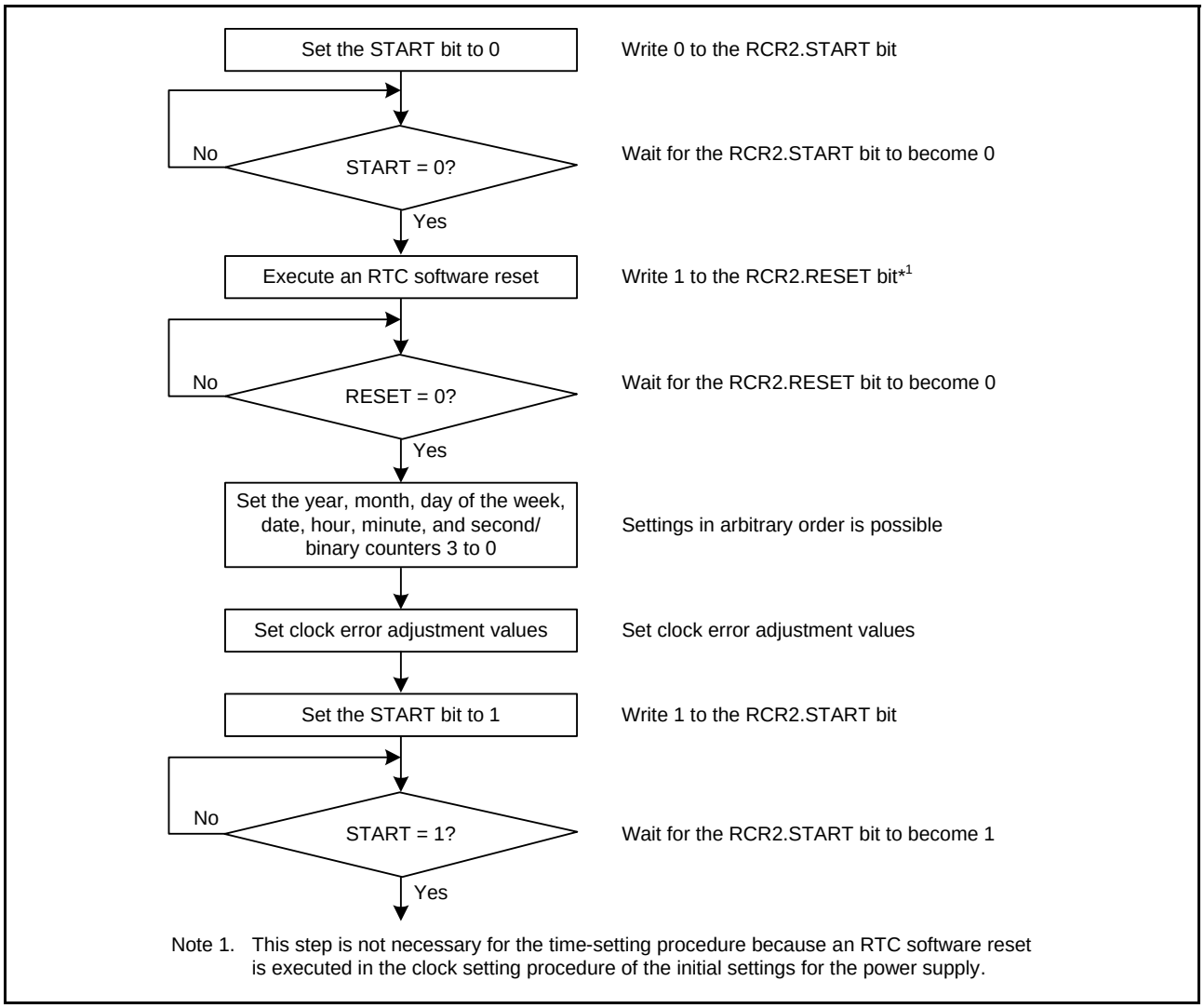


Figure 27.4 Setting the Time

27.3.4 30-Second Adjustment

Figure 27.5 shows how to execute 30-second adjustment.

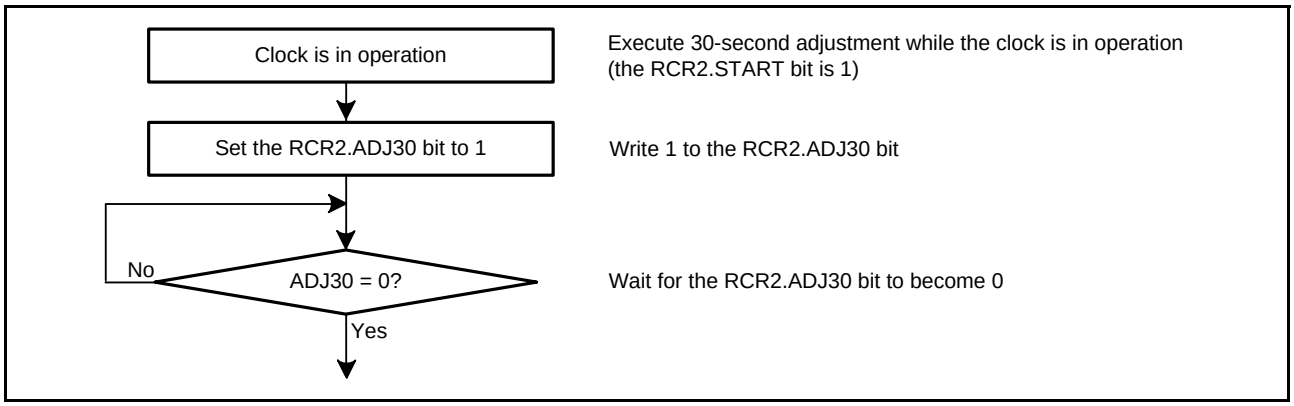


Figure 27.5 30-Second Adjustment

27.3.5 Reading 64-Hz Counter and Time

Figure 27.6 shows how to read the 64-Hz counter and time.

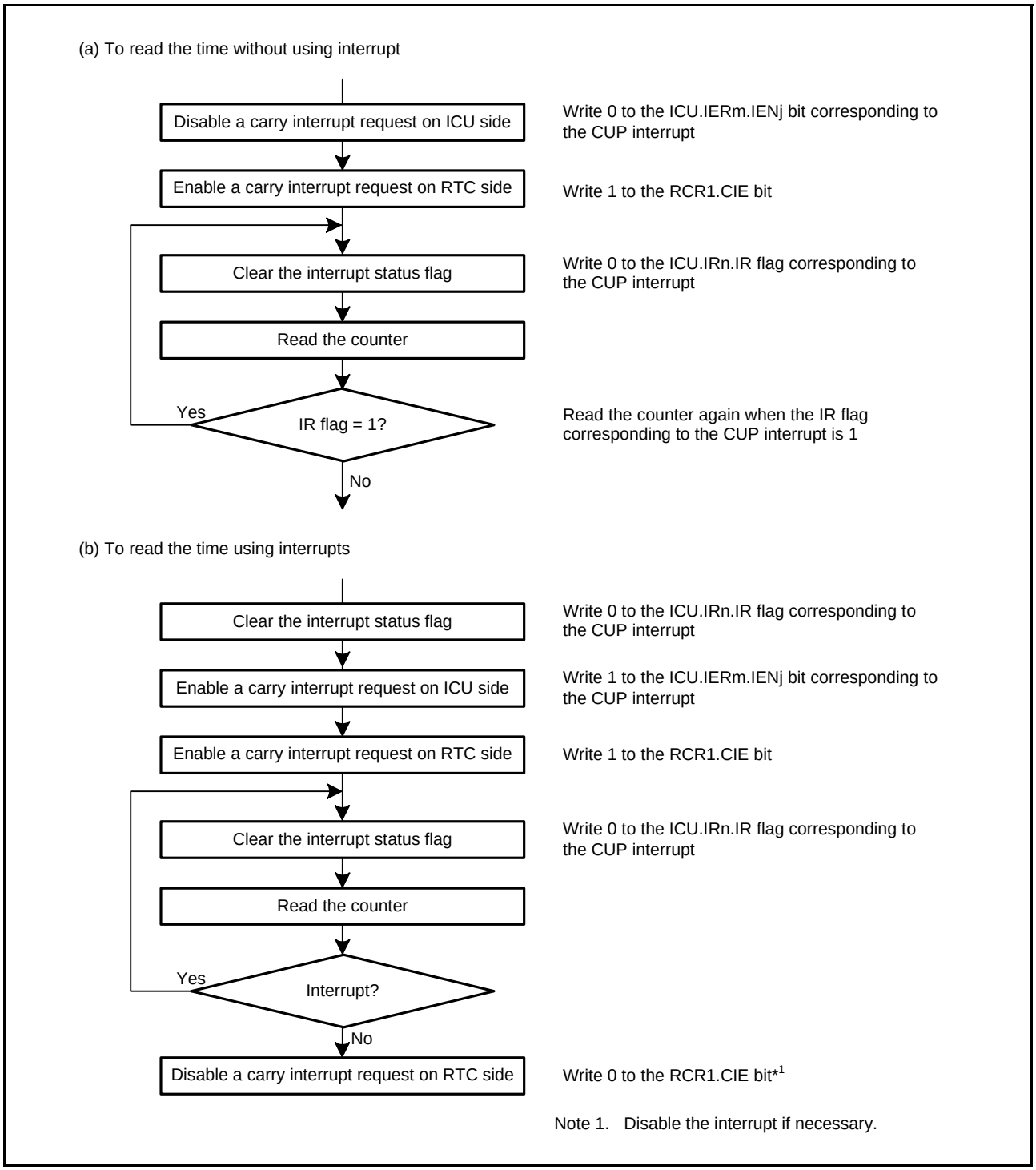


Figure 27.6 Reading Time

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 27.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

27.3.6 Alarm Function

Figure 27.7 shows how to use the alarm function.

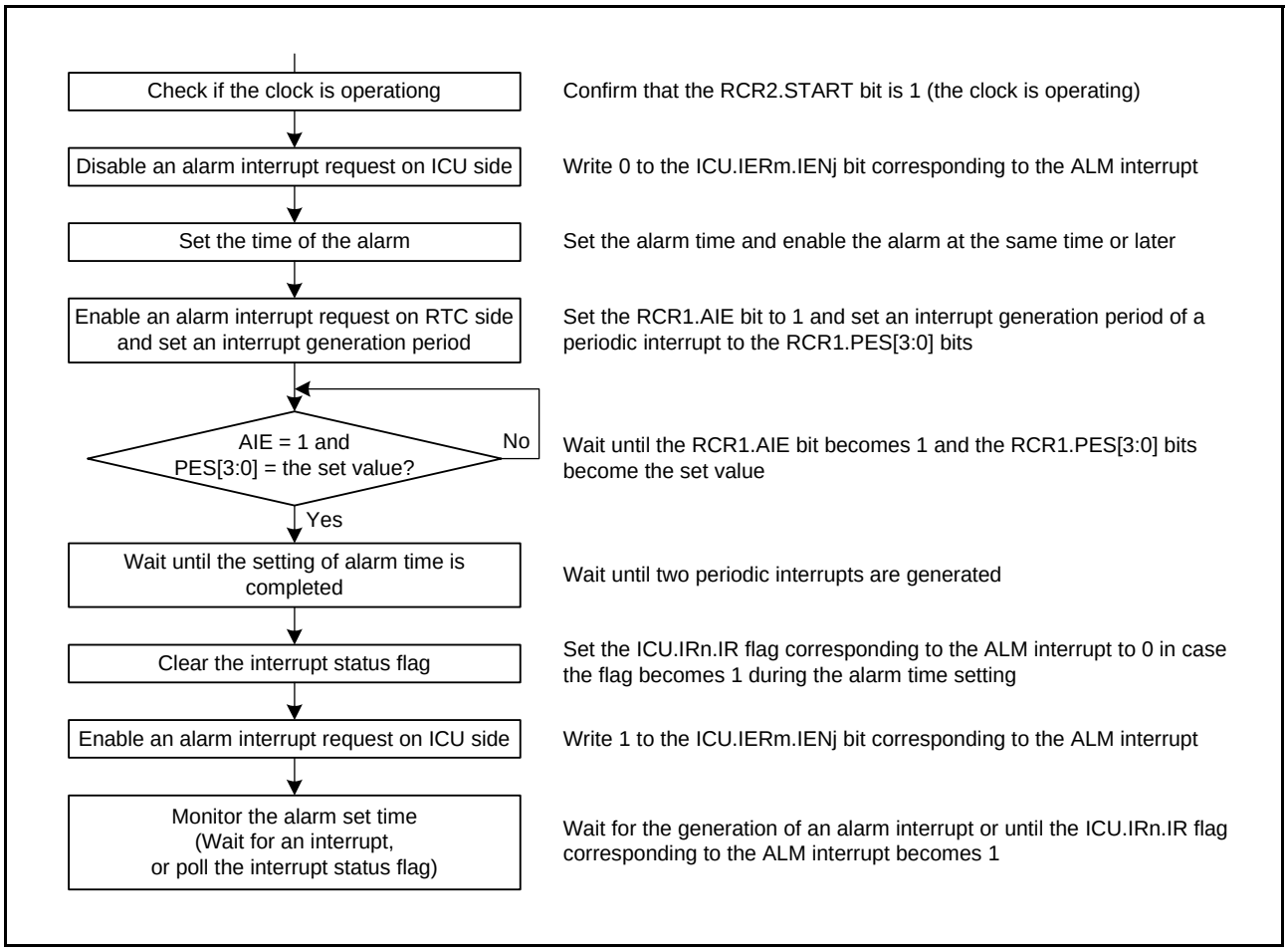


Figure 27.7 Using Alarm Function

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.

27.3.7 Procedure for Disabling Alarm Interrupt

Figure 27.8 shows the procedure for disabling the enabled alarm interrupt request.

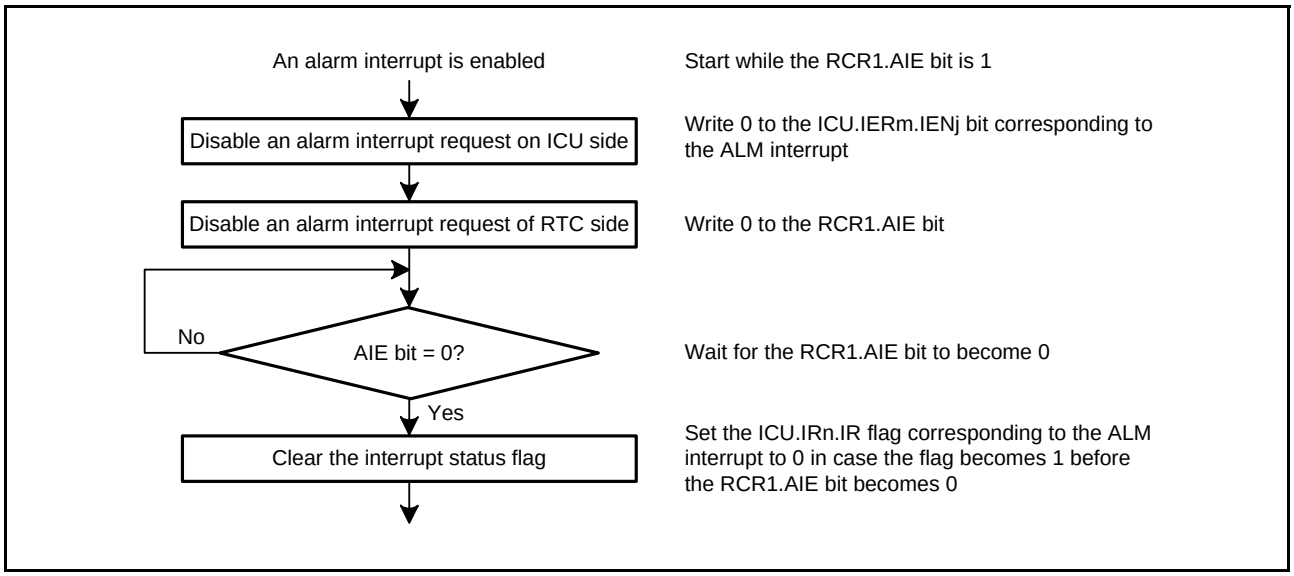


Figure 27.8 Procedure for Disabling Alarm Interrupt Request

27.3.8 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.

27.3.8.1 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)

27.3.8.2 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.

27.3.8.3 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.

27.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

27.3.9 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 27.9 and operation when the noise filter is on is shown in Figure 27.10.

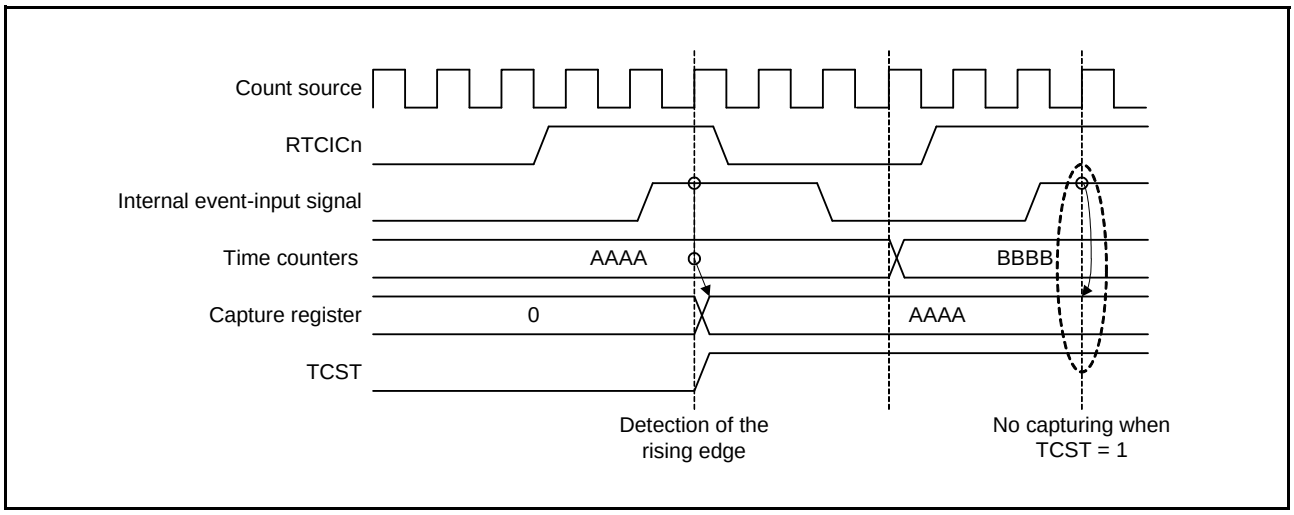


Figure 27.9 Timing of a Time Capture Operation (with the Filter Off) (n = 0 to 2)

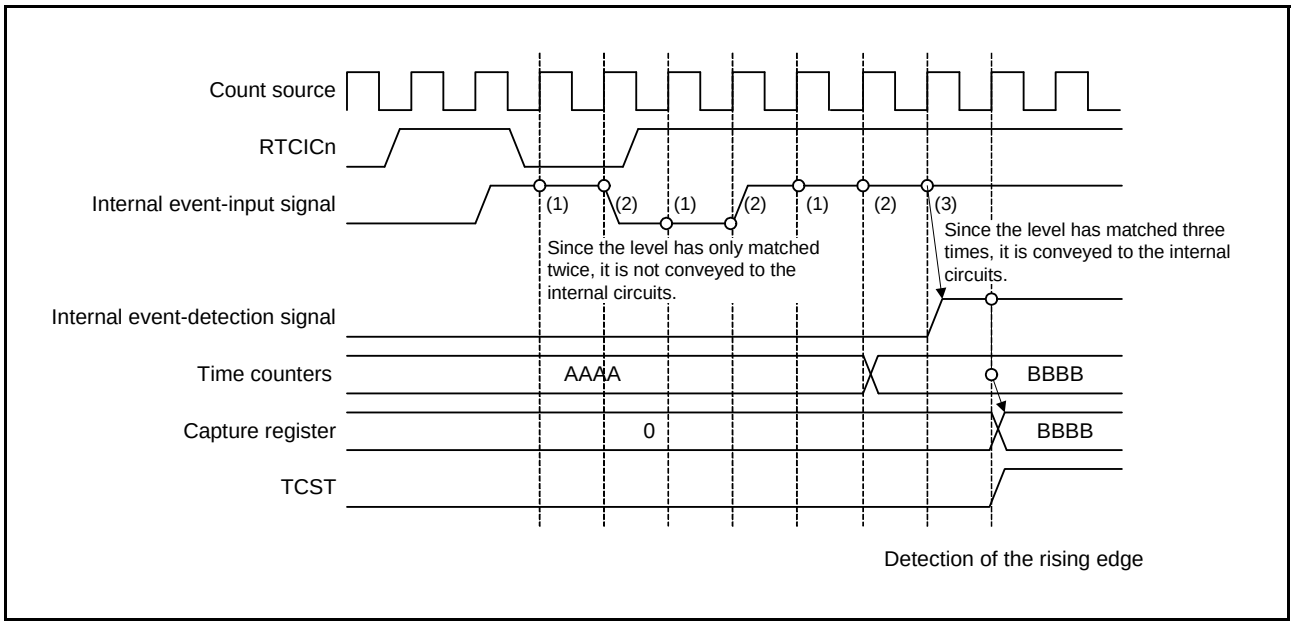


Figure 27.10 Timing of a Time Capture Operation (with the Filter On) (n = 0 to 2)

27.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 27.3 lists interrupt sources for the RTC.

Table 27.3 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 27.3.6, 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.

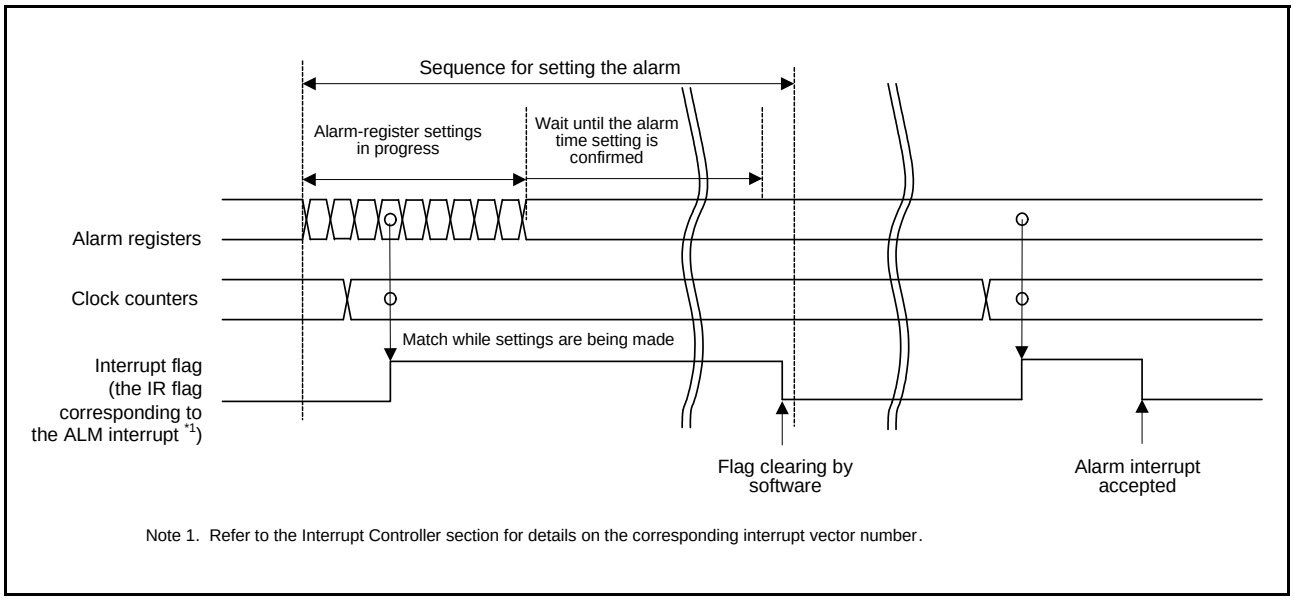


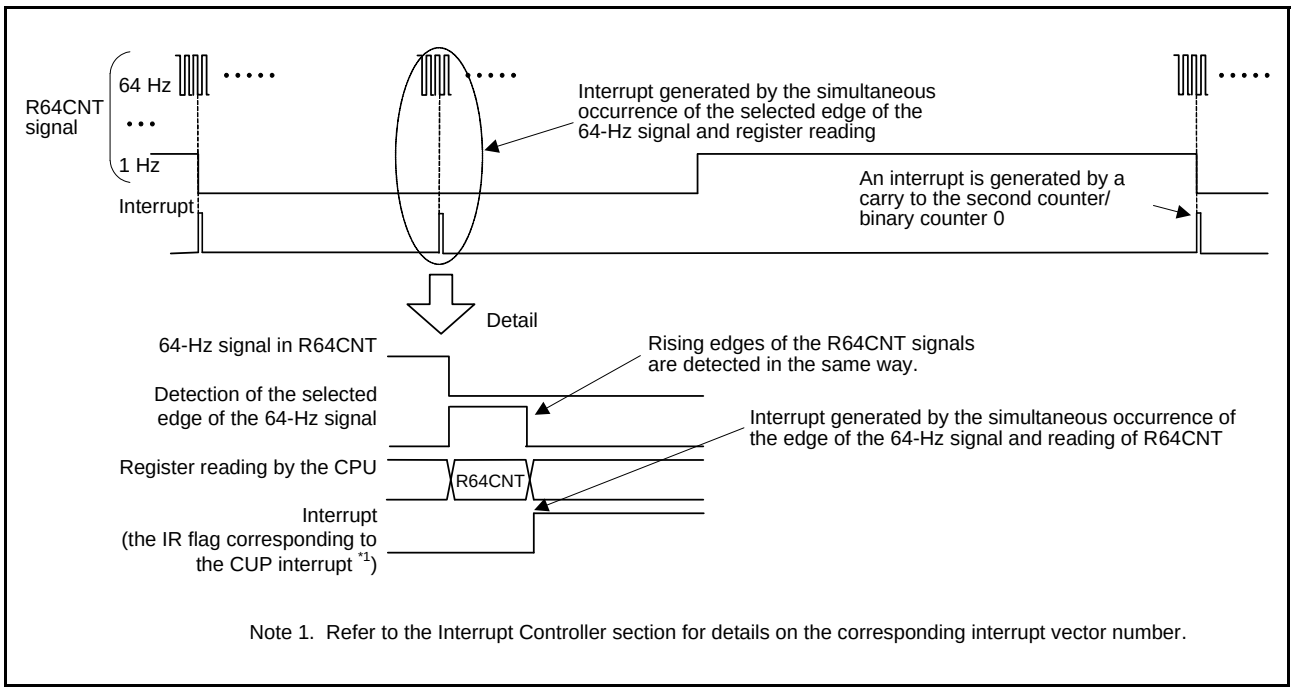
Figure 27.11 Timing Chart for the Alarm Interrupt (ALM)

(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 27.12 Carry Interrupt (CUP) Timing Chart

27.5 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.

27.5.1 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.

27.6 Usage Notes

27.6.1 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

The counter must be stopped before writing to any of the above registers.

27.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 27.13.
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.

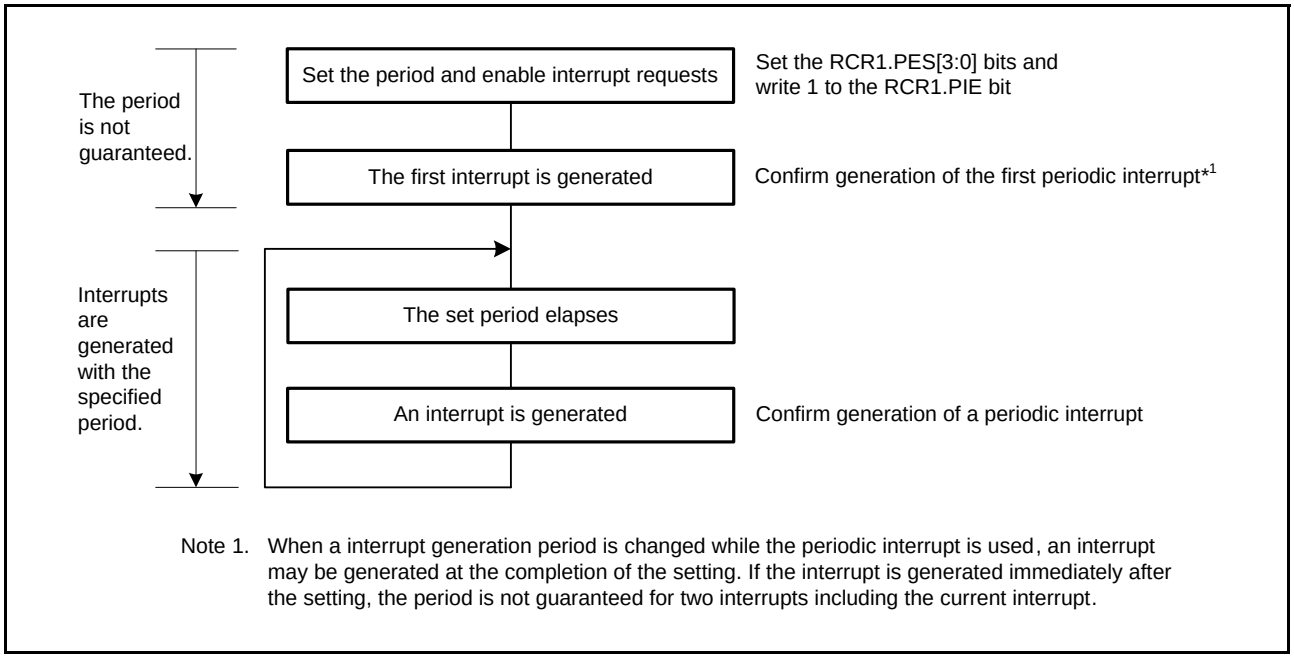


Figure 27.13 Using Periodic Interrupt Function

27.6.3 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.

27.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, deep software standby mode) 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.

27.6.5 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 27.3.5, 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, or RCR4 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, or software standby mode, 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.

27.6.6 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 27.3.1, Outline of Initial Settings of Registers after Power On.

27.6.7 Initialization Procedure When the Realtime Clock is Not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt request or operation of the counter may lead to increased power consumption.
For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 27.14.
When the sub-clock is not used or when using a product with no sub-clock oscillator, set the RCR4.RCKSEL bit to 0 (sub-clock oscillator is selected) and then set the RCR3.RTCEN bit to 0 (RTC disabled).

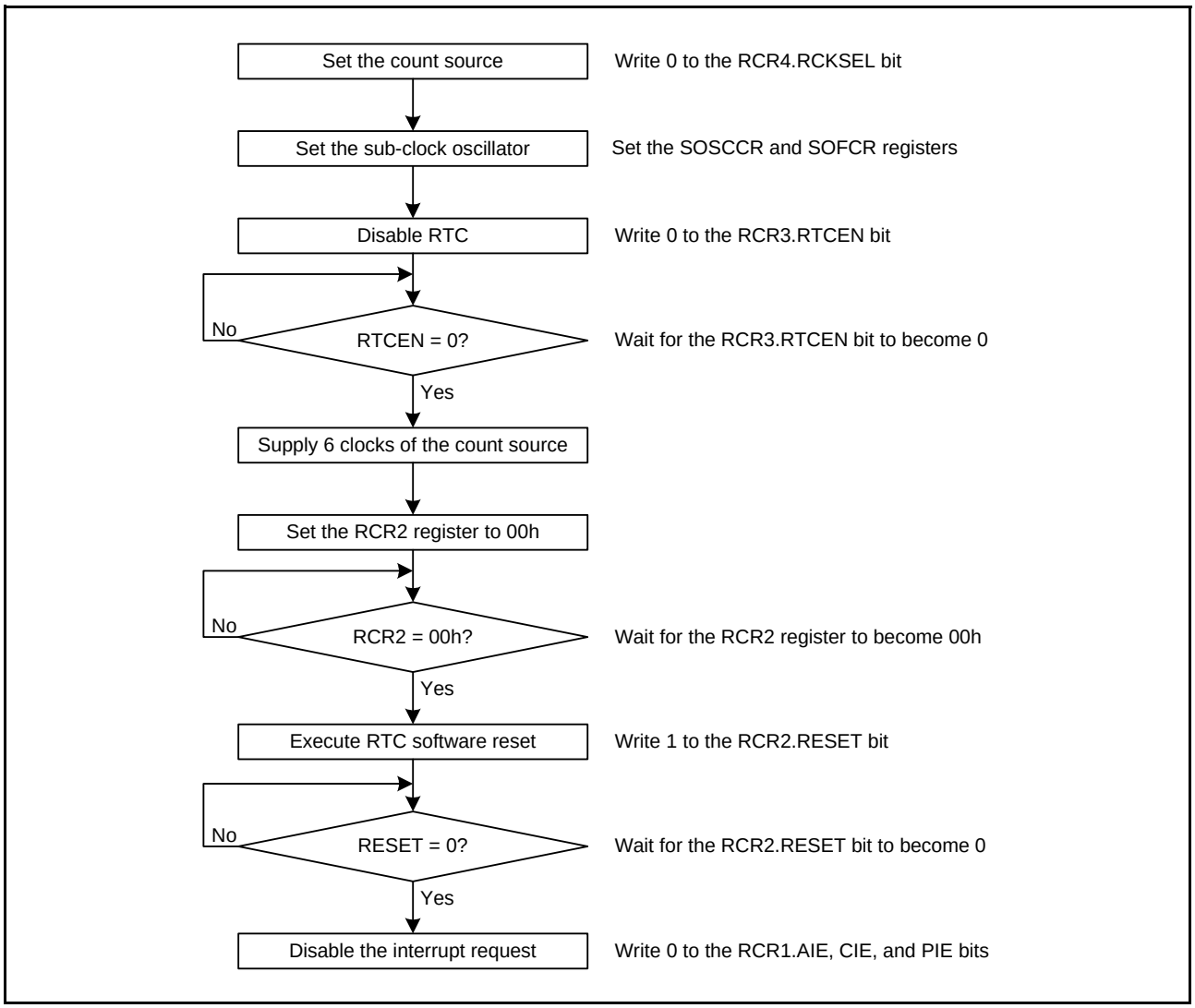


Figure 27.14 Initialization Procedure

28. 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.

28.1 Overview

Table 28.1 lists the specifications of the WDT and Figure 28.1 shows a block diagram of the WDT.

Table 28.1 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.

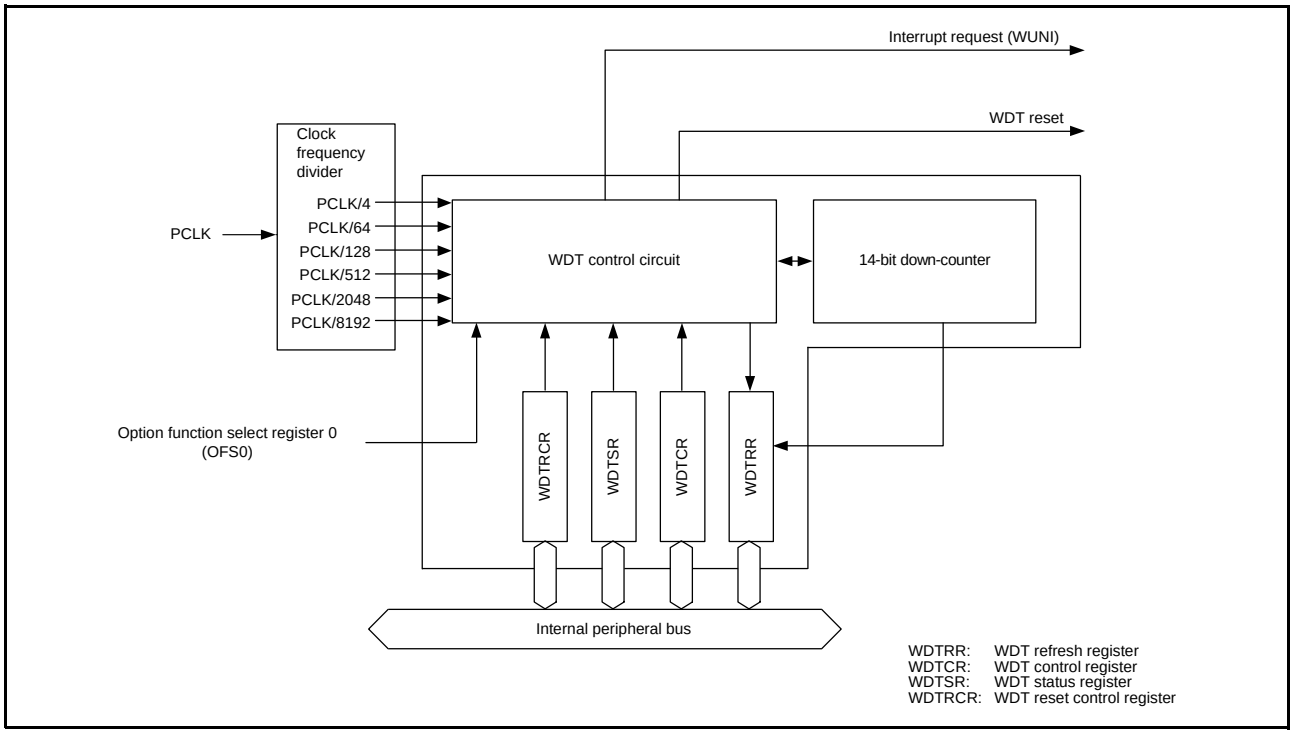
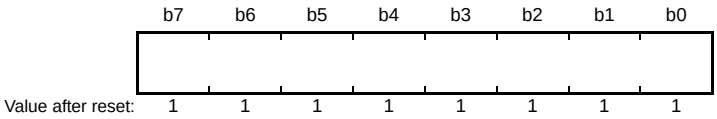


Figure 28.1 WDT Block Diagram

28.2 Register Descriptions

28.2.1 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 28.3.3, Refresh Operation.

28.2.2 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 28.3.2, 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 28.3.7, 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 28.2.

Table 28.2 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 28.3 lists the counter values for the window start and end positions and Figure 28.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 28.3 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%.

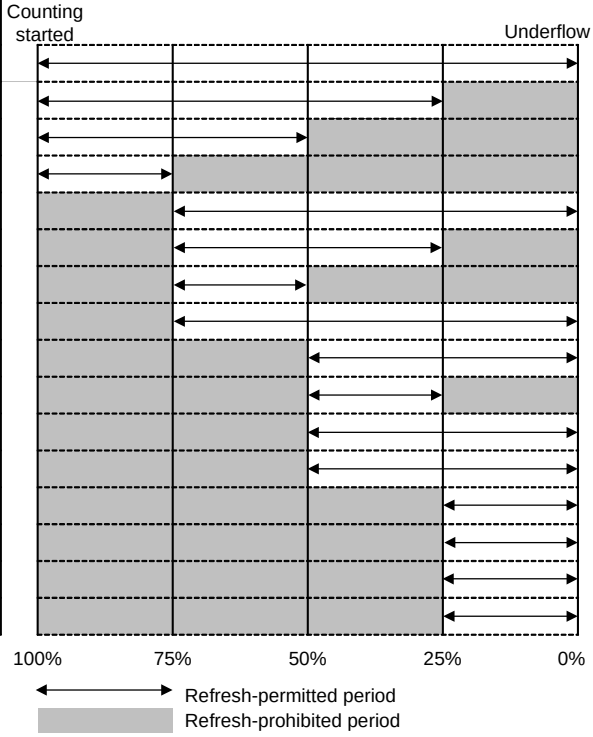


Figure 28.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

28.2.3 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.

28.2.4 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 28.3.2, 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 28.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

28.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 28.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

28.3 Operation

28.3.1 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.

28.3.1.1 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 28.3 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)

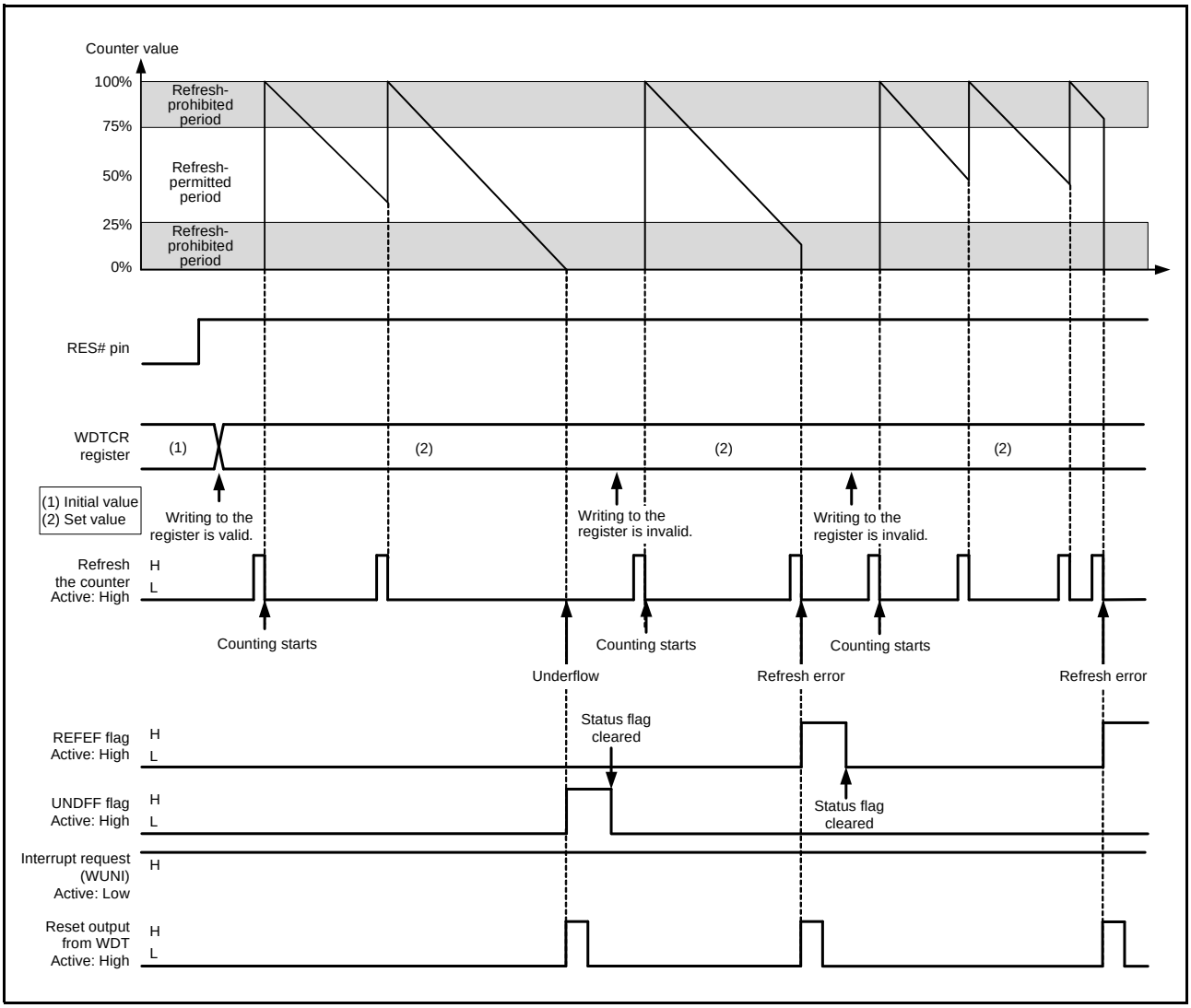


Figure 28.3 Operation Example in Register Start Mode

28.3.1.2 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 28.4 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)

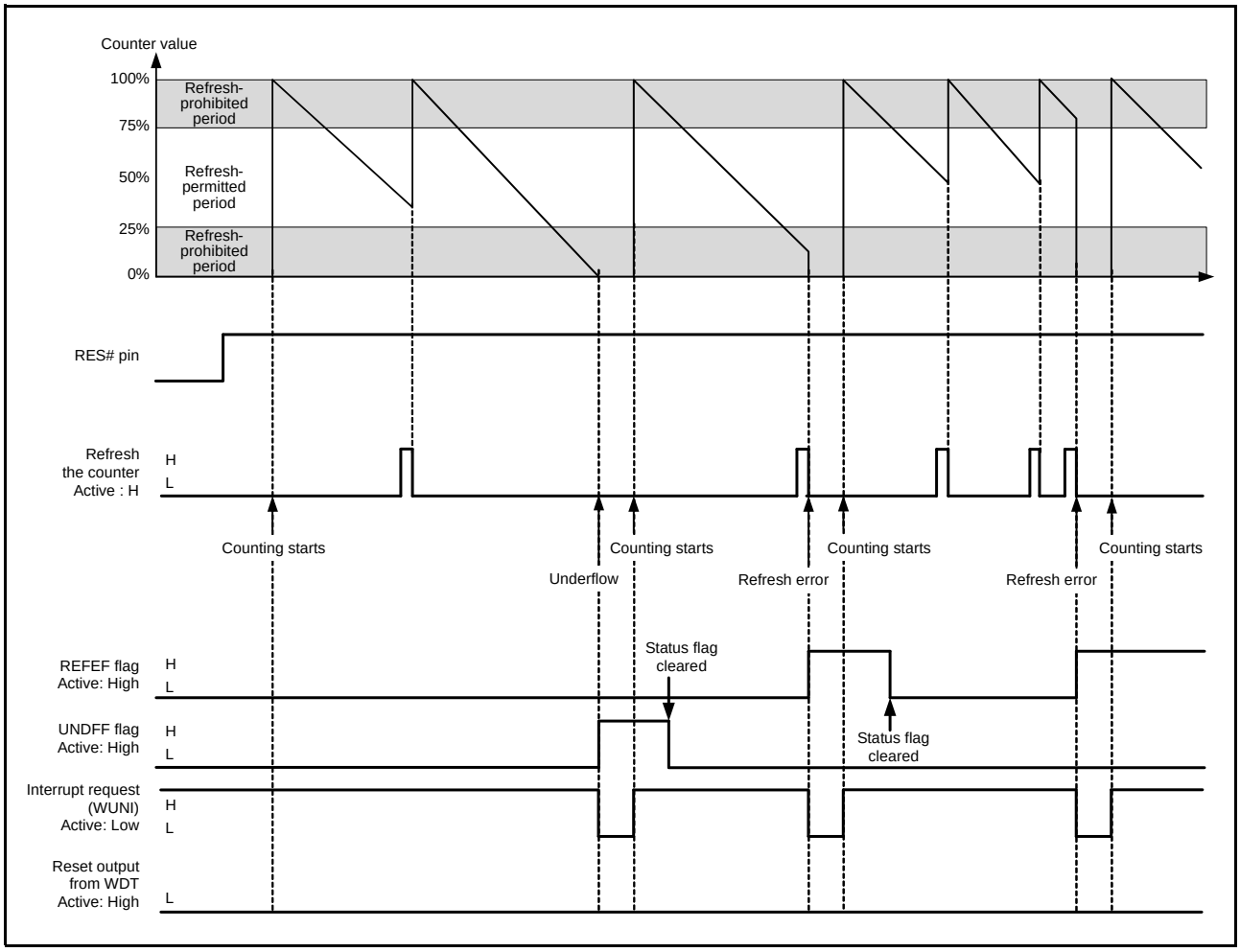


Figure 28.4 Operation Example in Auto-Start Mode

28.3.2 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 28.5 shows control waveforms produced in response to writing to the WDTCR register.

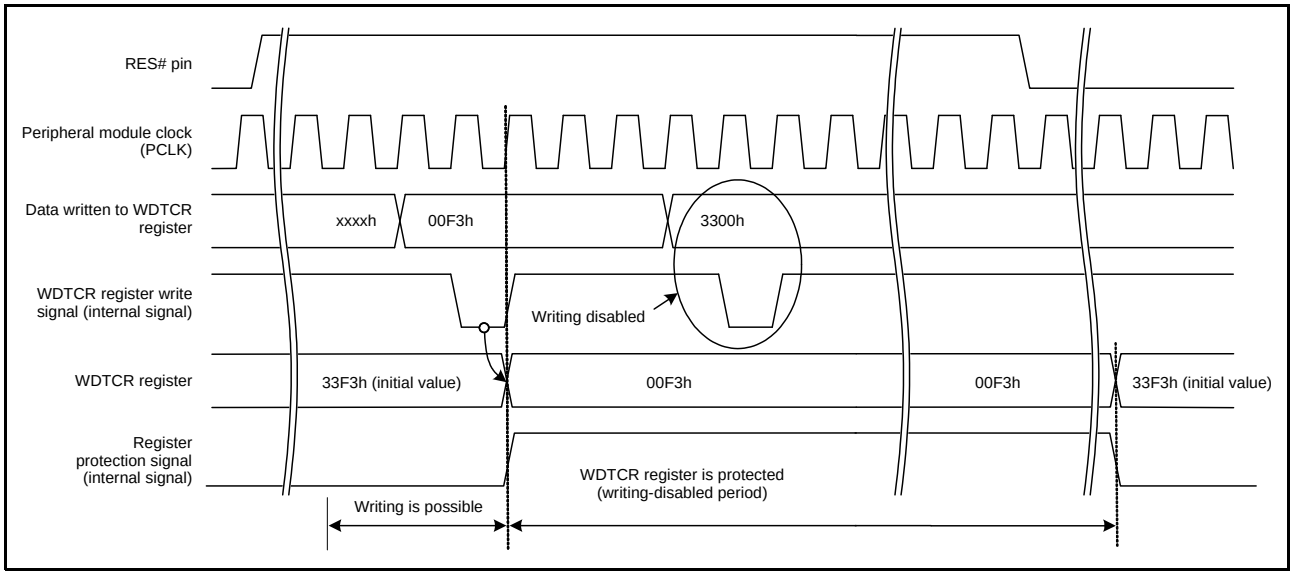


Figure 28.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

28.3.3 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 28.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

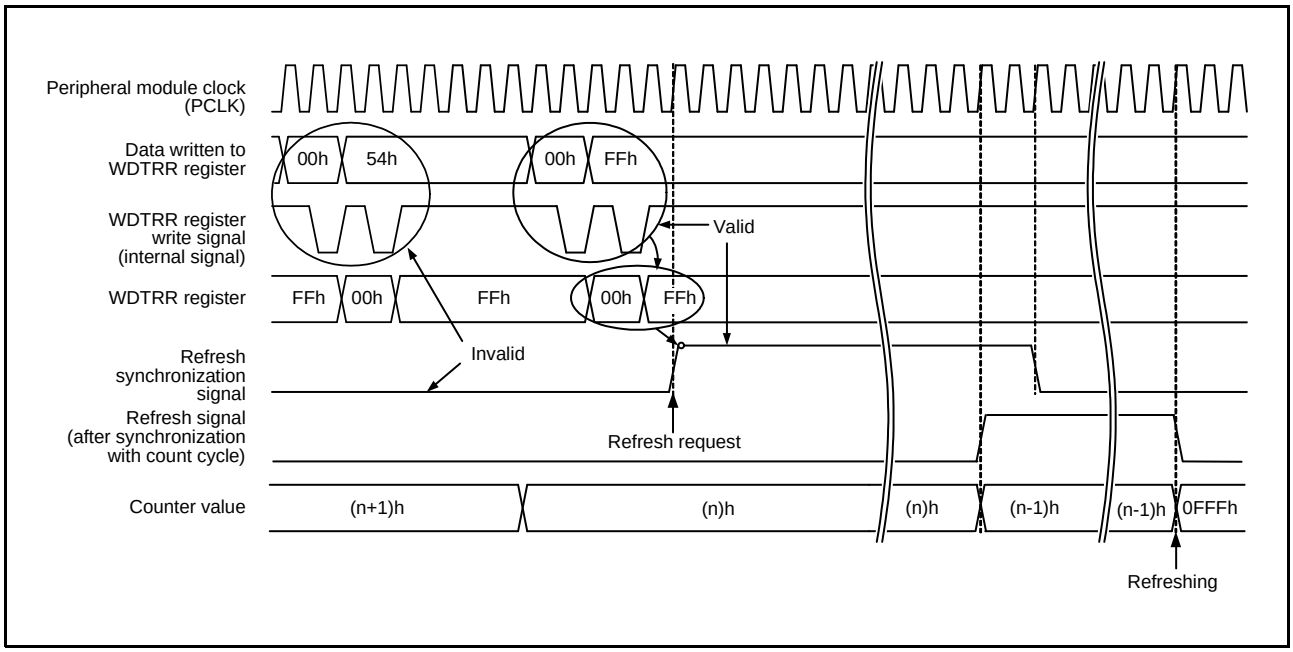


Figure 28.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

28.3.4 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.

28.3.5 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 14, Interrupt Controller (ICUF).

Table 28.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

28.3.6 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 28.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

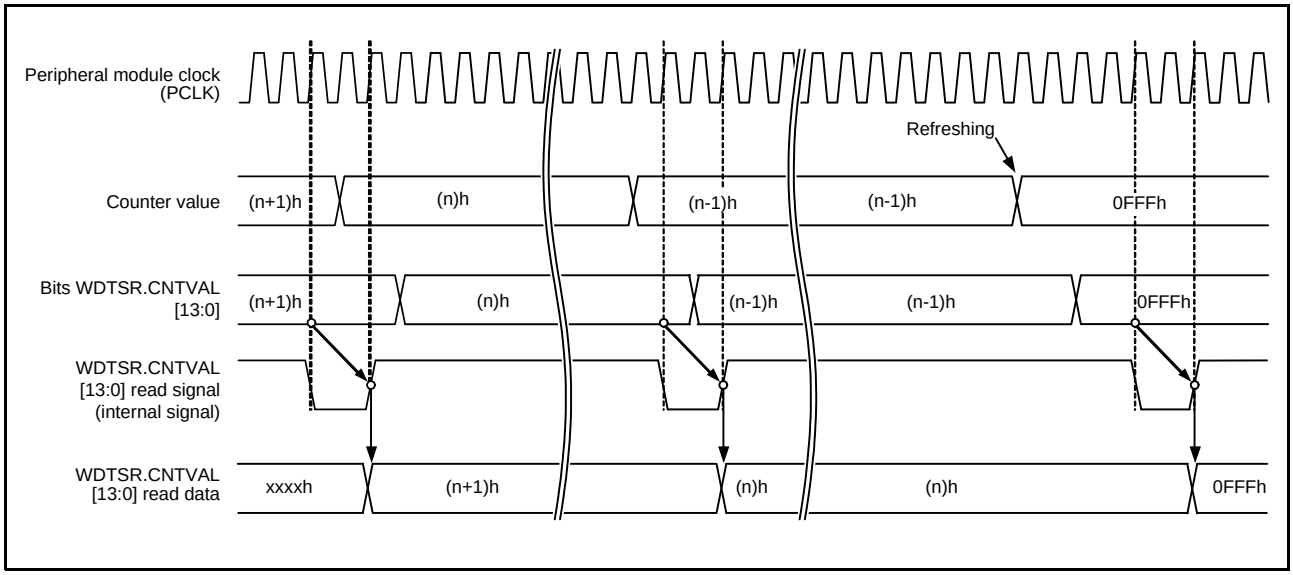


Figure 28.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

28.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 28.5 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 28.5 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

29. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

29.1 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 29.1 lists the specifications of the IWDT and Figure 29.1 shows a block diagram of the IWDT.

Table 29.1 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 29.1 is a block diagram of the IWDT.

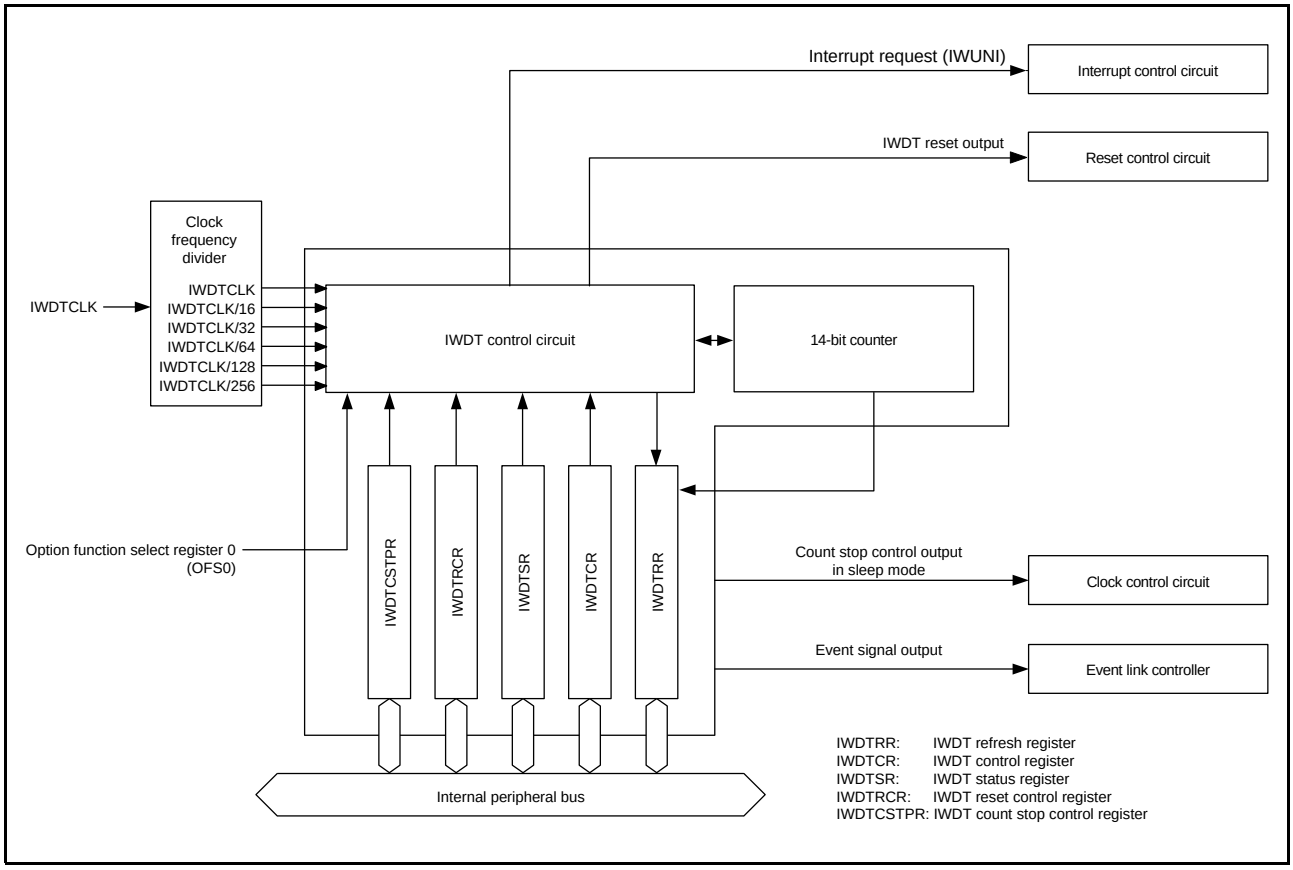
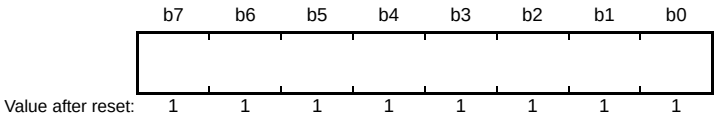


Figure 29.1 IWDT Block Diagram

29.2 Register Descriptions

29.2.1 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 29.3.3, Refresh Operation.

29.2.2 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 29.3.2, 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 29.3.8, 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 29.2.

Table 29.2 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 29.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 29.3 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 29.2 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%.

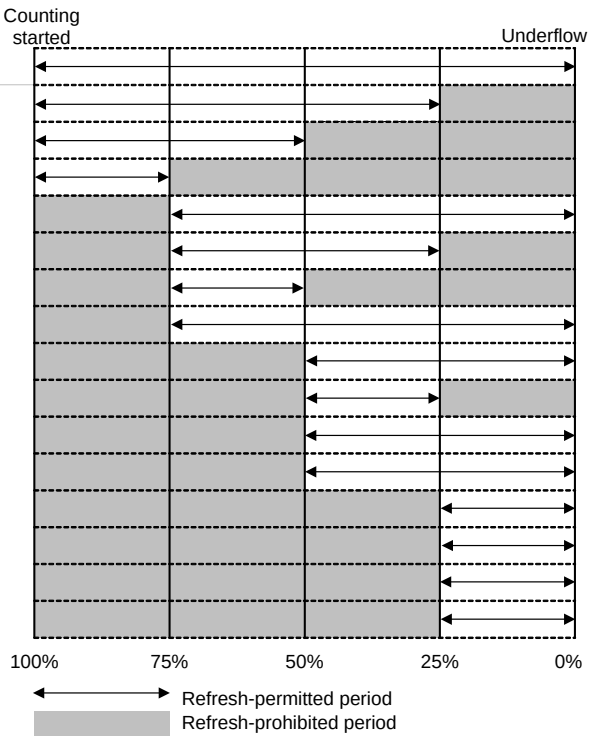


Figure 29.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

29.2.3 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.

29.2.4 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 29.3.2, 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 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

29.2.5 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 29.3.2, 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 29.3.8, 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.

29.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

29.3 Operation

29.3.1 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.

29.3.1.1 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 29.3 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)

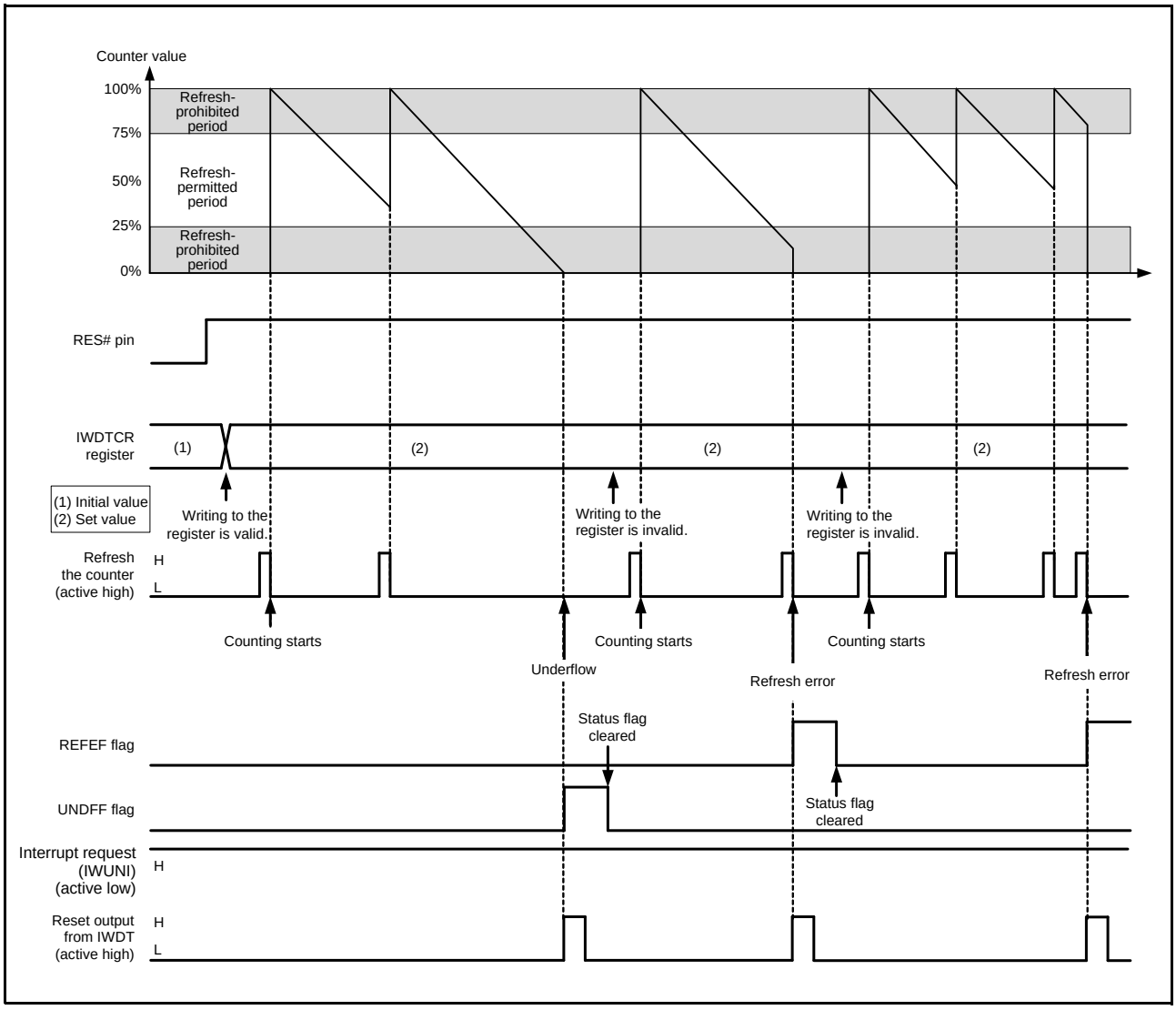


Figure 29.3 Operation Example in Register Start Mode

29.3.1.2 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 29.4 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)

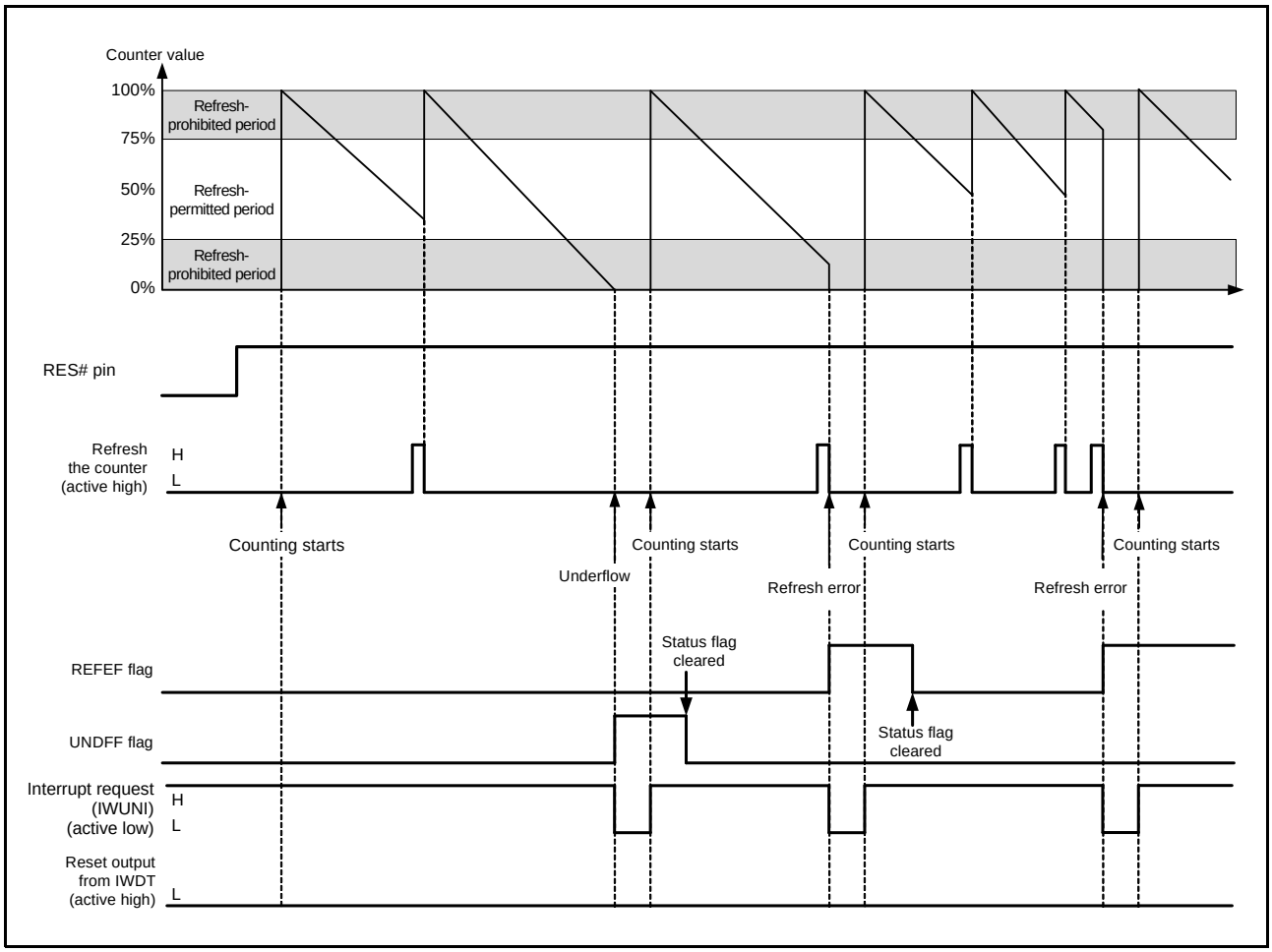


Figure 29.4 Operation Example in Auto-Start Mode

29.3.2 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 29.5 shows control waveforms produced in response to writing to the IWDTCR register.

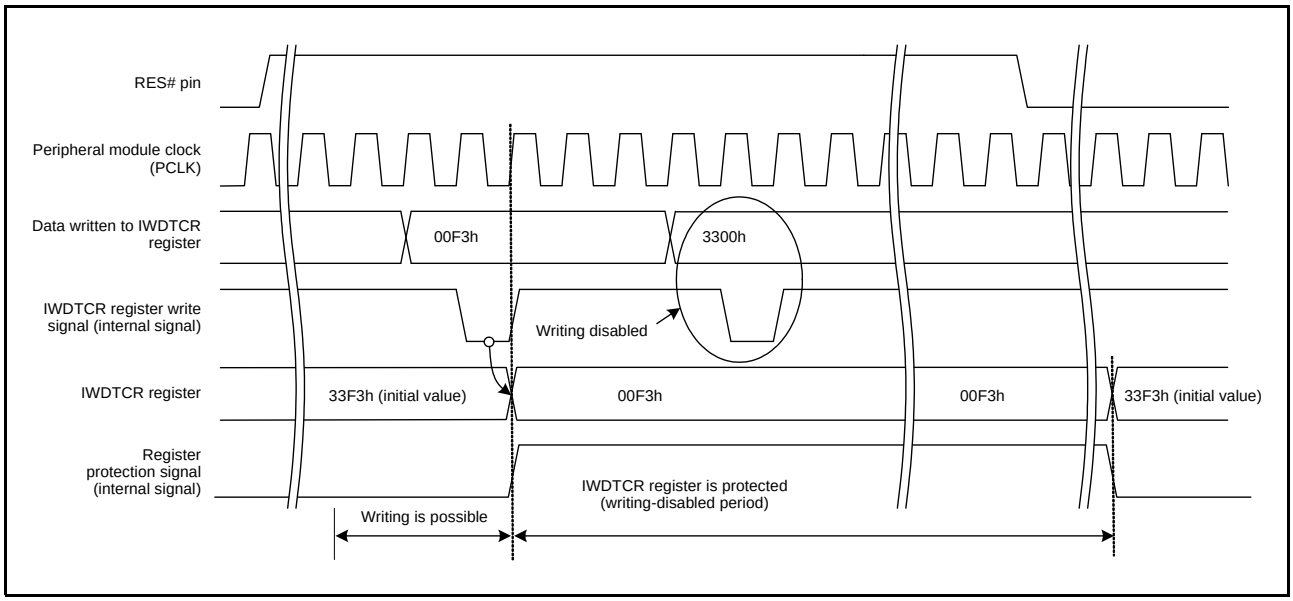


Figure 29.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

29.3.3 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 29.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

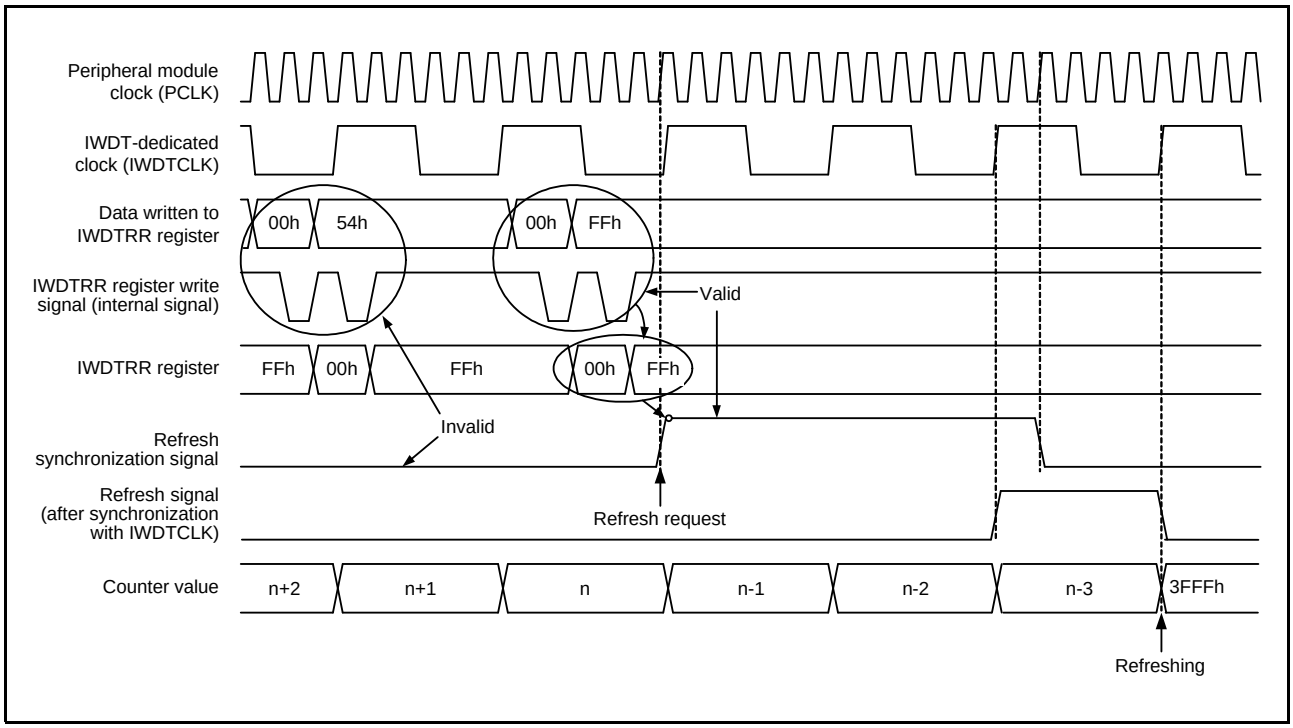


Figure 29.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

29.3.4 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.

29.3.5 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.

29.3.6 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 14, Interrupt Controller (ICUF).

Table 29.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
IWUNI	Counter underflow Refresh error	Not possible	Not possible

29.3.7 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 29.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

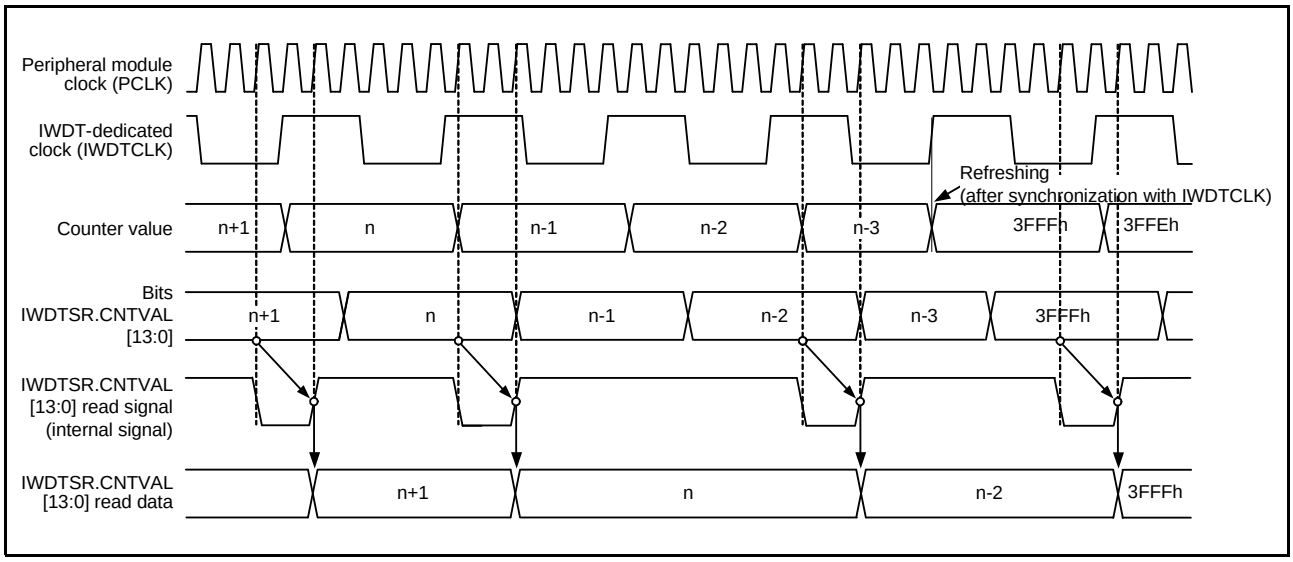


Figure 29.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

29.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 29.5 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 29.5 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

29.4 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 19, Event Link Controller (ELC).

29.5 Usage Notes

29.5.1 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.

29.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

30. 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.

30.1 Overview

Table 30.1 lists the specifications of the SCIk module, Table 30.2 lists the specifications of the SCIm module, Table 30.3 lists the specifications of the SCIh module, and Table 30.4 lists the specifications of the individual SCI channels. Figure 30.1 shows the block diagram of SCI0 to SCI4 and SCI7 to SCI9, Figure 30.2 shows the block diagram of SCI5 and SCI6, Figure 30.3 shows the block diagram of SCI10 and SCI11, and Figure 30.4 shows the block diagram of SCI12 (SCIh).

Table 30.1 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 30.5 to Table 30.7.
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 30.1 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 30.2.13, 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 30.2 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 30.5 to Table 30.7.
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 30.2 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 30.2.13, 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 30.3 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 30.5 to Table 30.8.
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.

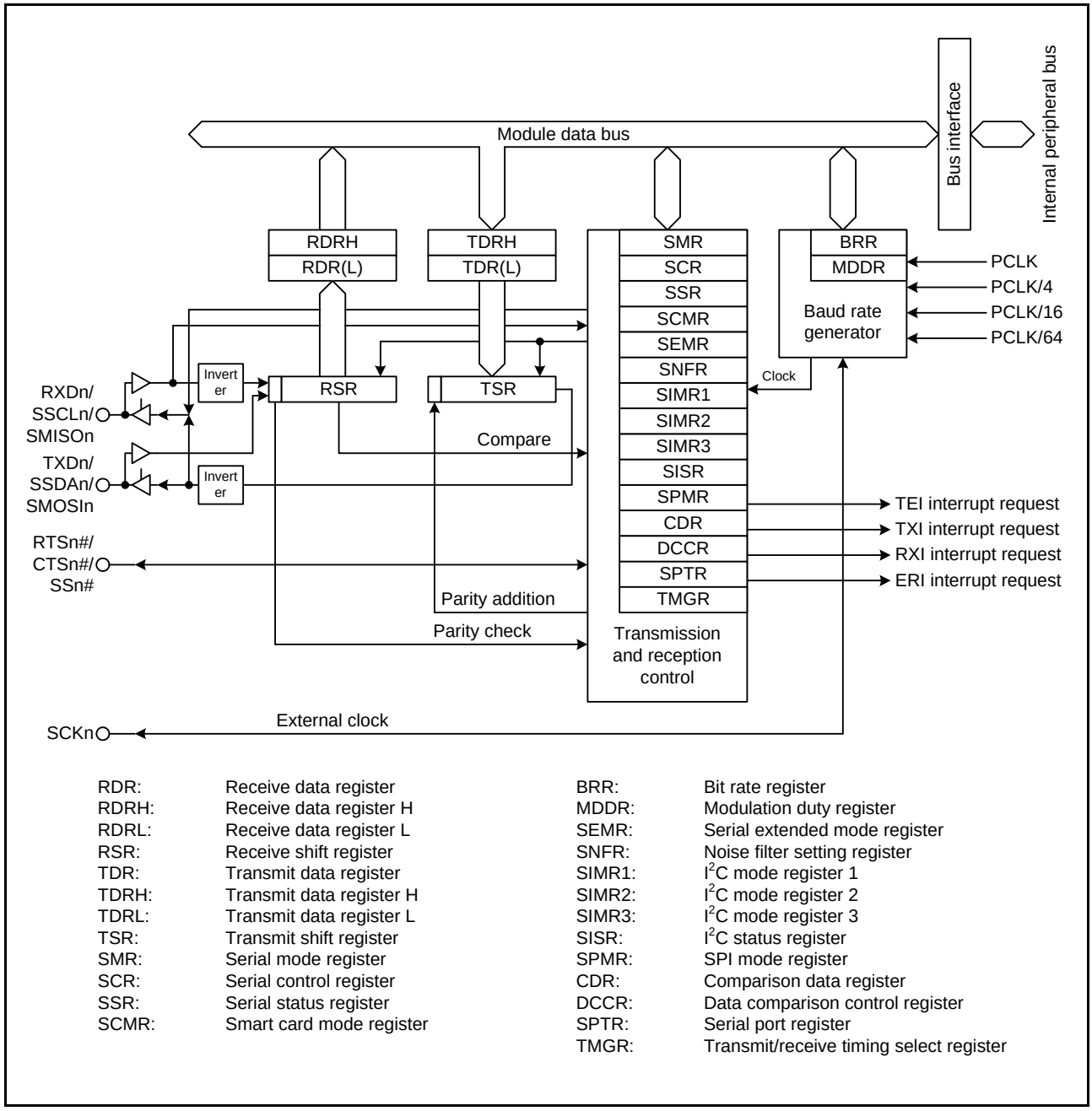
Table 30.3 SCIh Specifications (2/2)

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 I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 30.2.13, 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 I2C mode, only MSB first is available.

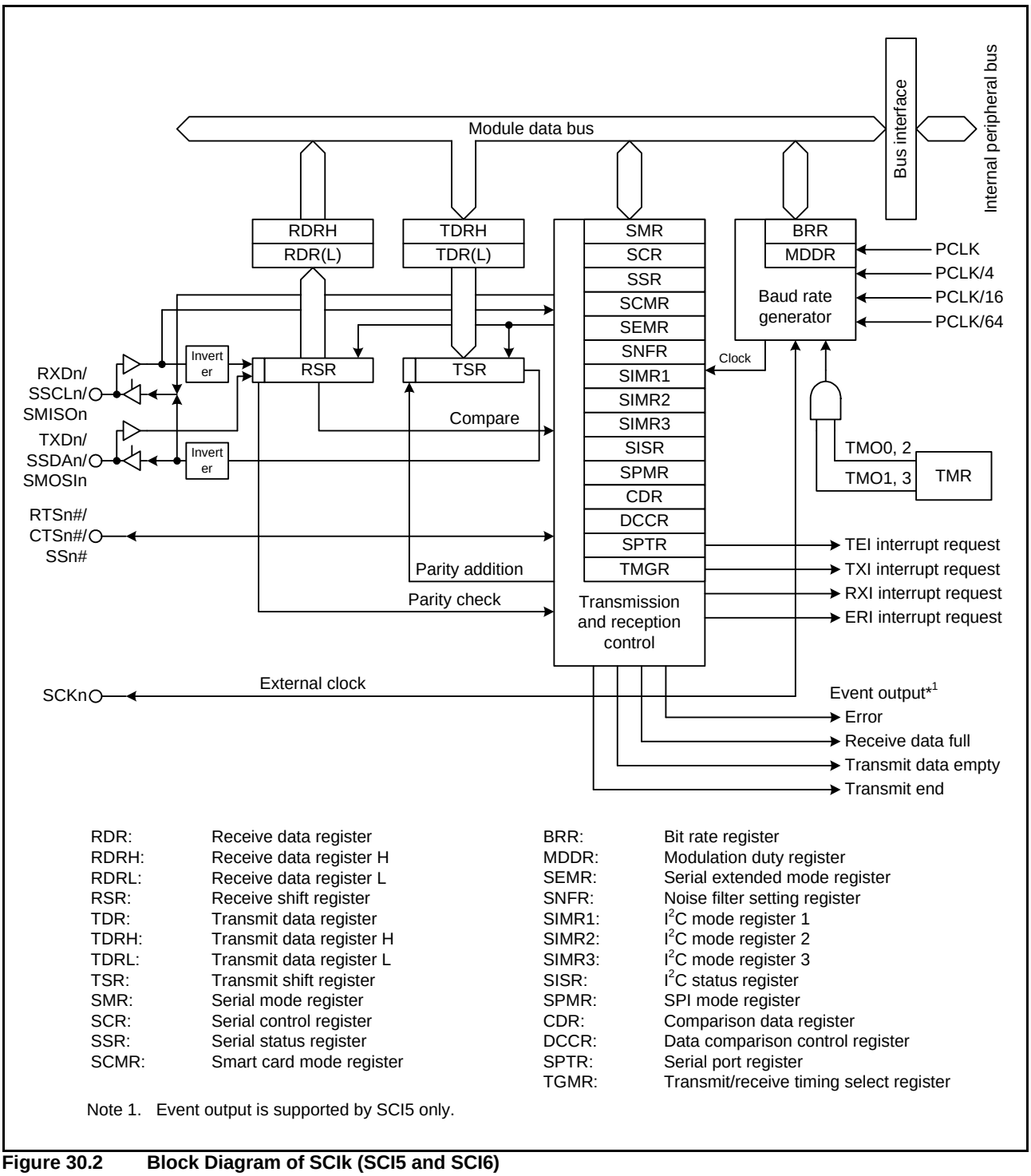
Table 30.4 Functions of SCI Channels

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 I2C 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

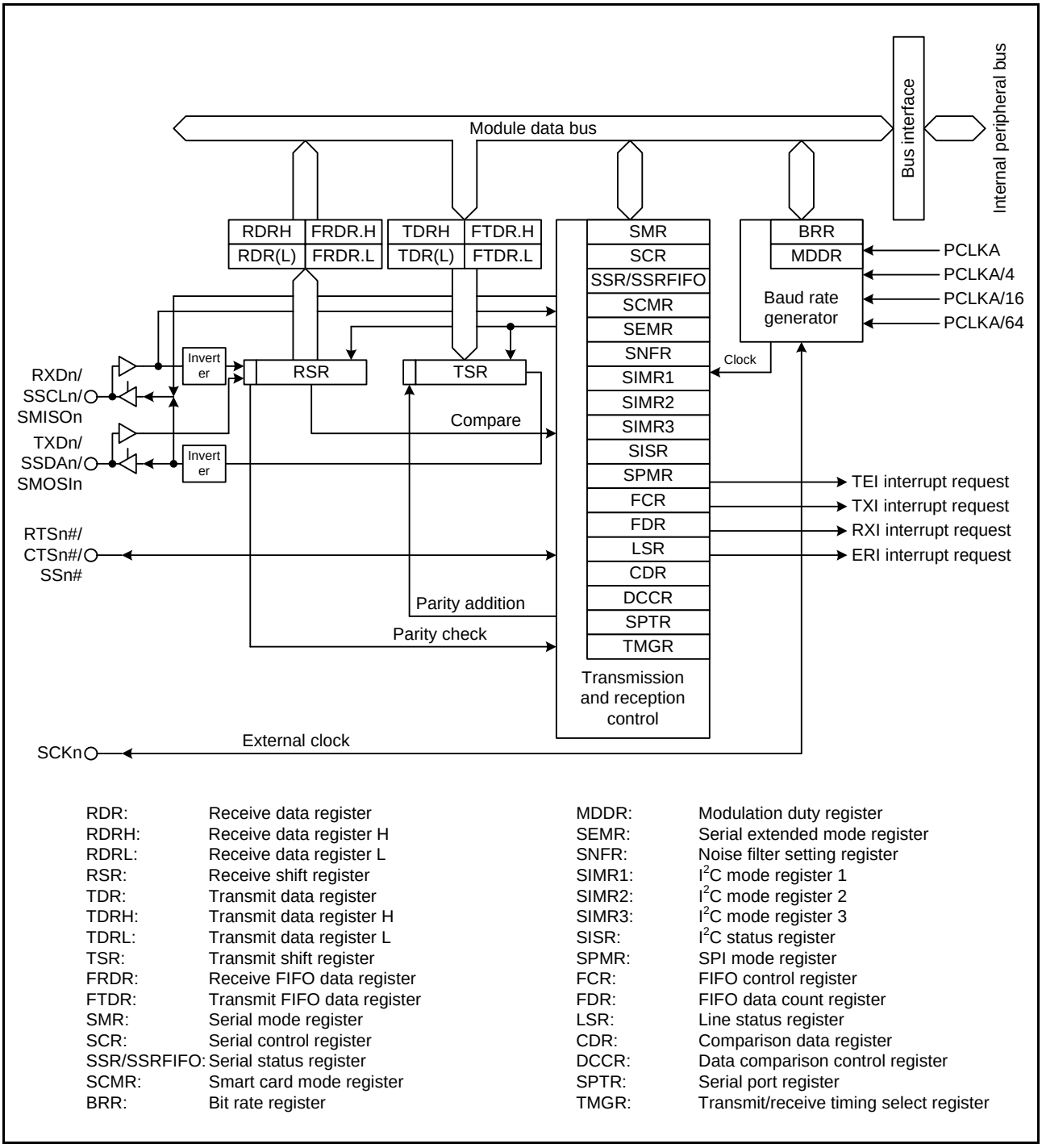
Figure 30.1 Block Diagram of SCIk (SCI0 to SCI4 and SCI7 to SCI9)



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 30.2 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 30.3 Block Diagram of SCIm (SCI10 and SCI11)

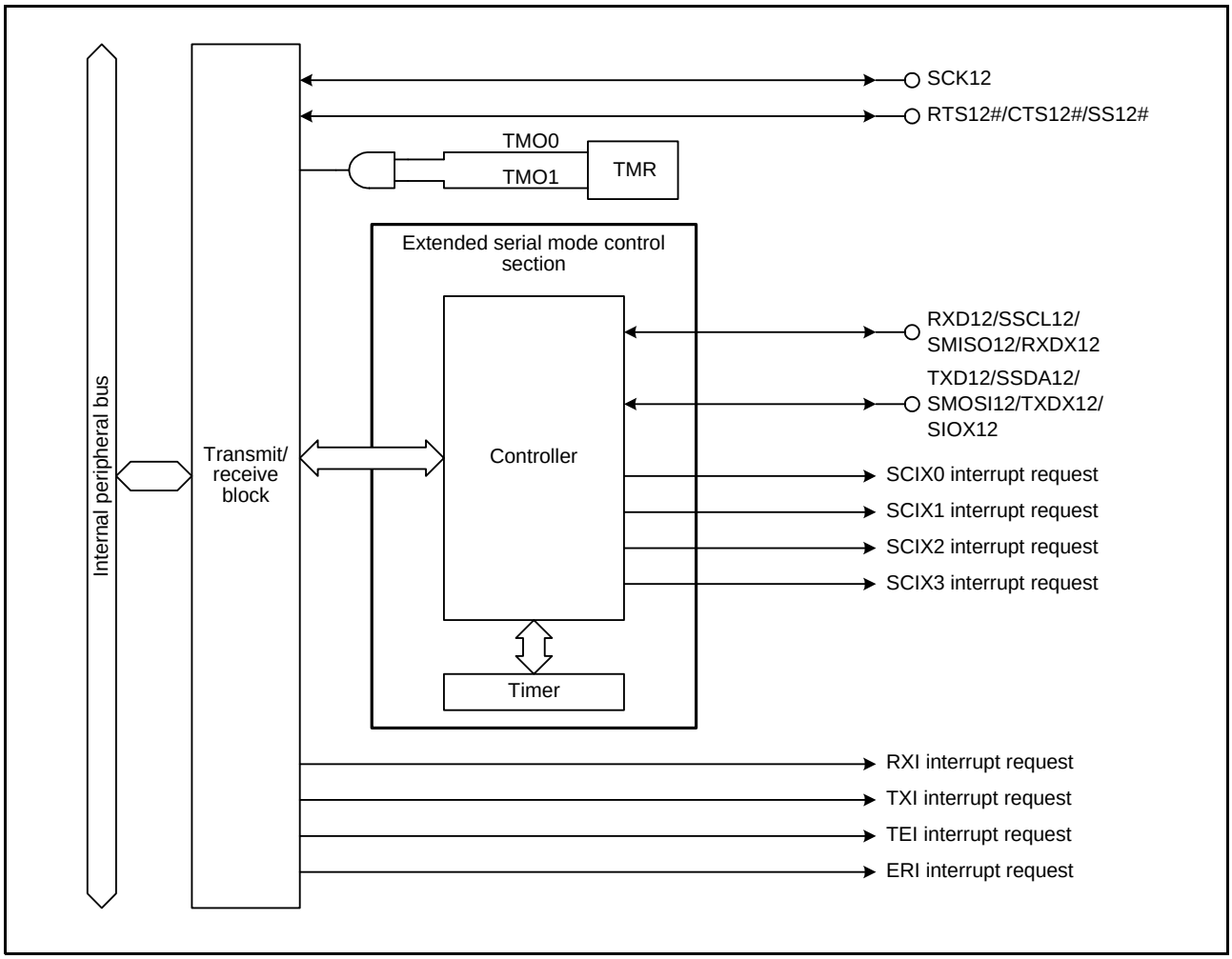


Figure 30.4 Block Diagram of SCIh (SCI12)

Table 30.5 to Table 30.8 list the pin configuration of the SCIs for the individual modes.

Table 30.5 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 30.5 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 30.6 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 30.7 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 30.7 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 30.8 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

30.2 Register Descriptions

30.2.1 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.

30.2.2 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 0005h, SCI11.RDR 000D 0025h, SCI12.RDR 0008 B305h

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

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.

30.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

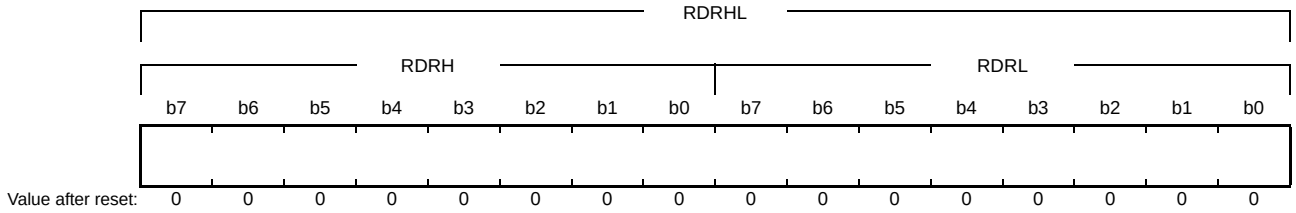
Address(es): SCI0.RDRH 0008 A010h, SCI1.RDRH 0008 A030h, SCI2.RDRH 0008 A050h, SCI3.RDRH 0008 A070h, SCI4.RDRH 0008 A090h, SCI5.RDRH 0008 A0B0h, SCI6.RDRH 0008 A0D0h, SCI7.RDRH 0008 A0F0h, SCI8.RDRH 0008 A110h, SCI9.RDRH 0008 A130h, SCI10.RDRH 000D 0010h, SCI11.RDRH 000D 0030h, SCI12.RDRH 0008 B310h

- 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 0011h, SCI11.RDRL 000D 0031h, 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 0010h, SCI11.RDRHL 000D 0030h, 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.

30.2.4 Receive FIFO Data Register (FRDR)

Address(es): SCI10.FRDR 000D 0010h, SCI11.FRDR 000D 0030h,
SCI10.FRDR.H 000D 0010h, SCI11.FRDR.H 000D 0030h,
SCI10.FRDR.L 000D 0011h, SCI11.FRDR.L 000D 0031h

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.

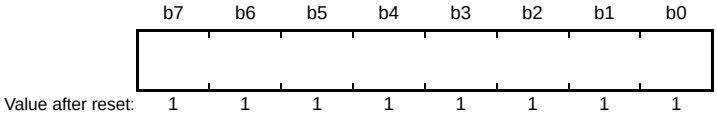
The MPB flag holds the value of the multi-processor bit that was added to the data at the top of the receive FIFO. The SSR.MPB flag is not used when the FCR.FM bit is 1 (FIFO mode).

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.

30.2.5 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 0003h, SCI11.TDR 000D 0023h, 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).

30.2.6 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

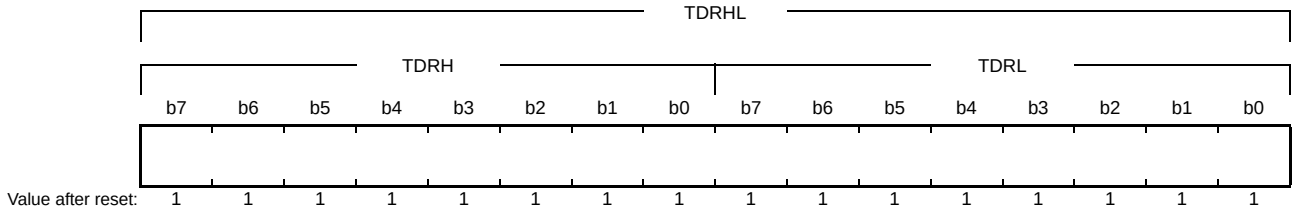
Address(es): SCI0.TDRH 0008 A00Eh, SCI1.TDRH 0008 A02Eh, SCI2.TDRH 0008 A04Eh, SCI3.TDRH 0008 A06Eh, SCI4.TDRH 0008 A08Eh, SCI5.TDRH 0008 A0AEh, SCI6.TDRH 0008 A0CEh, SCI7.TDRH 0008 A0EEh, SCI8.TDRH 0008 A10Eh, SCI9.TDRH 0008 A12Eh, SCI10.TDRH 000D 000Eh, SCI11.TDRH 000D 002Eh, SCI12.TDRH 0008 B30Eh

- 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 000Fh, SCI11.TDRL 000D 002Fh, 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 000Eh, SCI11.TDRHL 000D 002Eh, 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.

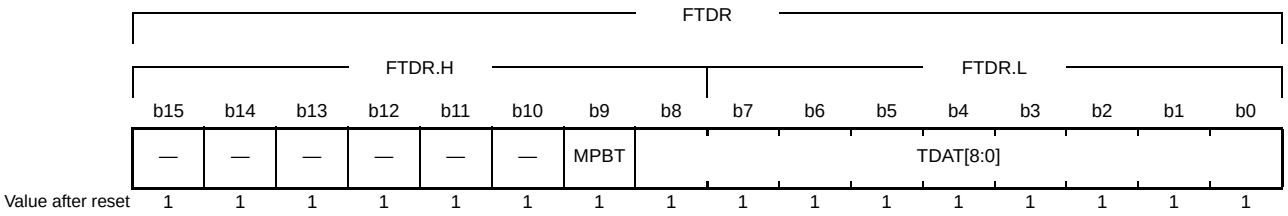
The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in the RDRH register are fixed to 1. These bits are read as 1. The write value should be 1.

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.

30.2.7 Transmit FIFO Data Register (FTDR)

Address(es): SCI10.FTDR 000D 000Eh, SCI11.FTDR 000D 002Eh, SCI10.FTDR.H 000D 000Eh, SCI11.FTDR.H 000D 002Eh, SCI10.FTDR.L 000D 000Fh, SCI11.FTDR.L 000D 002Fh



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).

30.2.8 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.

30.2.9 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 0000h, SCI11.SMR 000D 0020h, 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 30.2.13, 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 30.2.13, 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 0000h, SMCI11.SMR 000D 0020h, 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 30.9 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 30.2.13, 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 30.2.13, 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 30.6.4, Receive Data Sampling Timing and Reception Margin.

Table 30.9 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 30.2.13, 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 30.6.2, 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 30.6.3, 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 30.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 30.6.8, Clock Output Control.

30.2.10 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 0002h, SCI11.SCR 000D 0022h, 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 30.4, 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 0002h, SMCI11.SCR 000D 0022h, 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 30.12, 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 30.6.8, 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.

30.2.11 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 0004h, SCI11.SSR 000D 0024h, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 0004h, SMCI11.SSR 000D 0024h, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 0004h, SCI11.SSRFIFO 000D 0024h

	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 14.5.2, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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.

30.2.12 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 0006h, SCI11.SCMR 000D 0026h, 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 0006h, SMCI11.SCMR 000D 0026h, 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 30.10 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 30.10 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 30.2.13, Bit Rate Register (BRR)).

30.2.13 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 0001h, SCI11.BRR 000D 0021h, 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 30.11 and Table 30.12 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 30.11 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 30.14 and Table 30.15.
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 30.12 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 ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 30.14 and Table 30.15.
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 30.13 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 30.14 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 30.15 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 30.16 lists examples of N settings in the BRR register in normal asynchronous mode. Table 30.18 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 30.21. Examples of BRR (N) settings in smart card interface mode are listed in Table 30.23. Examples of BRR (N) settings in simple I^2C mode are listed in Table 30.25. 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 30.6.4, Receive Data Sampling Timing and Reception Margin. Table 30.19 and Table 30.22 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 30.16. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 30.16 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 30.17 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 30.18 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 30.19 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 30.20 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 30.21 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 30.22 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 30.23 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 30.24 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 30.25 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 30.26 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.

30.2.14 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 0012h, SCI11.MDDR 000D 0032h, 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 30.27 and Table 30.28.
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 30.27 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 30.14 and Table 30.15, section 30.2.13, 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 30.28 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 30.14 and Table 30.15, section 30.2.13, 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.

30.2.15 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 0007h, SCI11.SEMR 000D 0027h, 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 30.29 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 30.29 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 30.5 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

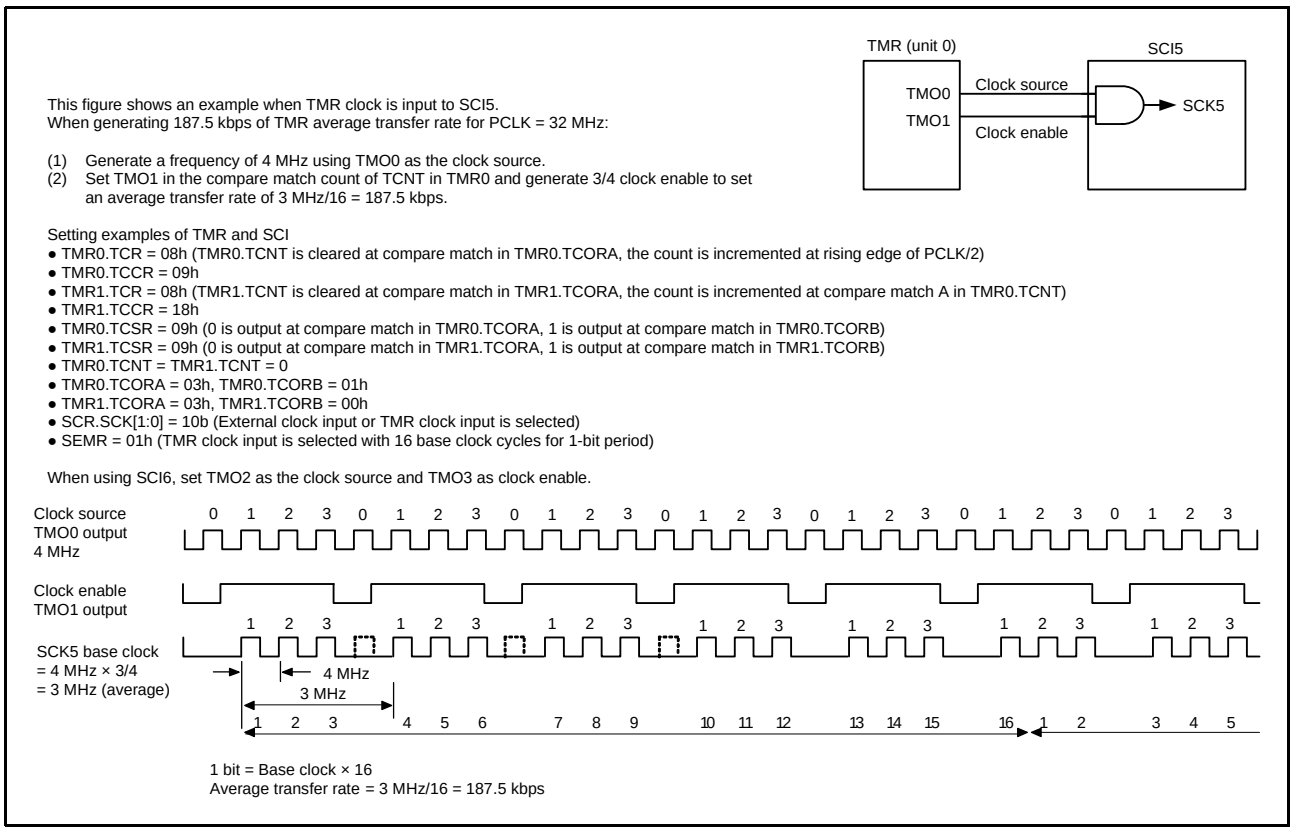


Figure 30.5 Example of Average Transfer Rate Setting When TMR Clock is Input

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)

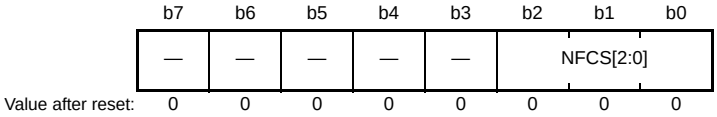
Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. 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.

30.2.16 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 0008h, SCI11.SNFR 000D 0028h, 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.

30.2.17 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 0009h, SCI11.SIMR1 000D 0029h, 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.

30.2.18 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 000Ah, SCI11.SIMR2 000D 002Ah, 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.

30.2.19 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 000Bh, SCI11.SIMR3 000D 002Bh, SCI12.SIMR3 0008 B30Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTP REQ	IICRST AREQ	IICSTA REQ
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.

30.2.20 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 000Ch, SCI11.SISR 000D 002Ch, 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.

30.2.21 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 000Dh, SCI11.SPMR 000D 002Dh, 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)

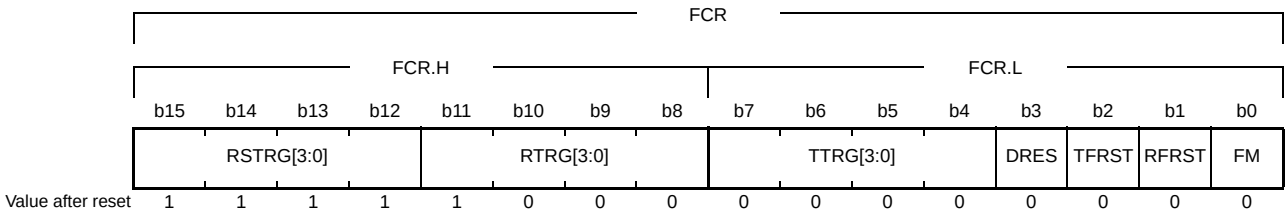
This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 30.64 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 30.64 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

30.2.22 FIFO Control Register (FCR)

Address(es): SCI10.FCR 000D 0014h, SCI11.FCR 000D 0034h,
SCI10.FCR.H 000D 0014h, SCI11.FCR.H 000D 0034h,
SCI10.FCR.L 000D 0015h, SCI11.FCR.L 000D 0035h



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)

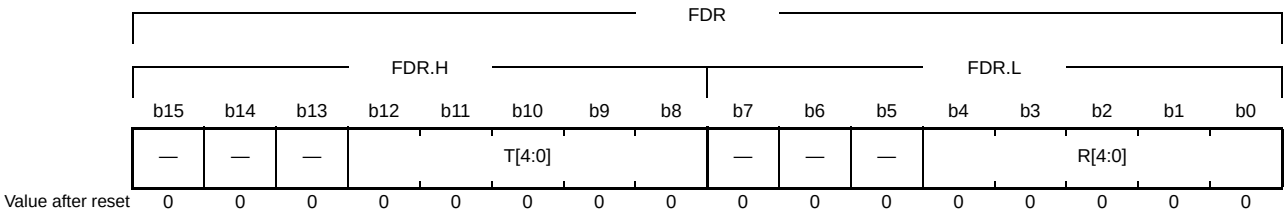
The SSRFIFO.RDF flag becomes 1 when the values of the FDR.R[4:0] bits are equal to or greater than the values set in the RTRG[3:0] bits. At this time, if the SCR.RIE bit is 1, a receive data full interrupt (RXI) request is generated.

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.

30.2.23 FIFO Data Count Register (FDR)

Address(es): SCI10.FDR 000D 0016h, SCI11.FDR 000D 0036h,
SCI10.FDR.H 000D 0016h, SCI11.FDR.H 000D 0036h,
SCI10.FDR.L 000D 0017h, SCI11.FDR.L 000D 0037h



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)

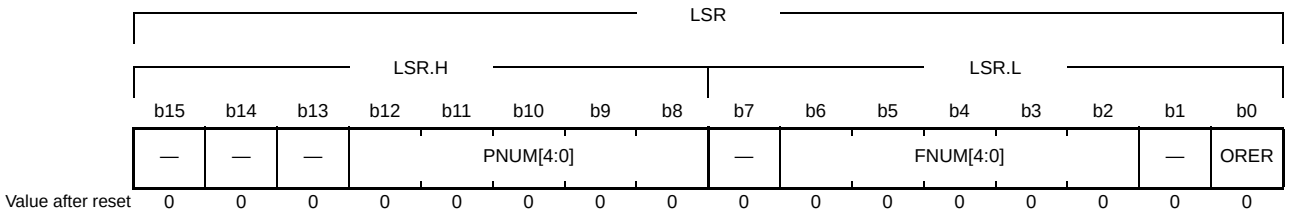
These bits indicate the number of data stored in the receive FIFO. 00h indicates that no received data is in FIFO, and 10h indicates that 16 frame data received have been stored in FIFO.
These bits never have values in the range from 11h to 1Fh.

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.

30.2.24 Line Status Register (LSR)

Address(es): SCI10.LSR 000D 0018h, SCI11.LSR 000D 0038h,
SCI10.LSR.H 000D 0018h, SCI11.LSR.H 000D 0038h,
SCI10.LSR.L 000D 0019h, SCI11.LSR.L 000D 0039h



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)

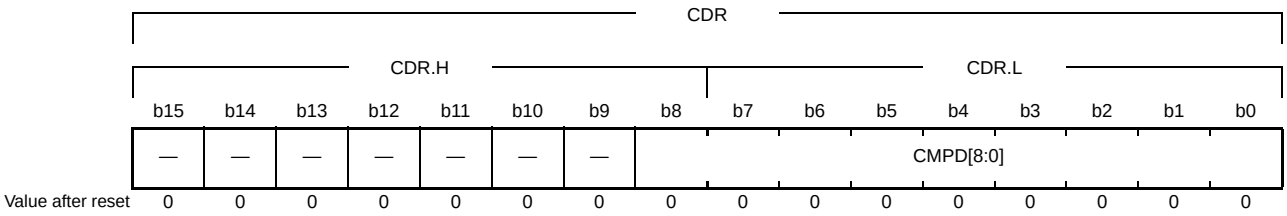
These bits indicate the number of data in which a framing error has occurred among the received data in the receive FIFO.

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.

30.2.25 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 001Ah, SCI11.CDR 000D 003Ah,
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 001Ah, SCI11.CDR.H 000D 003Ah,
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 001Bh, SCI11.CDR.L 000D 003Bh



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.

30.2.26 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 0013h, SCI11.DCCR 000D 0033h

	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 14.5.2, 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 14.5.2, 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.

30.2.27 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 001Ch, SCI11.SPTR 000D 003Ch

	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 30.30 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 30.30 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 30.30 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.

30.2.28 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 001Dh, SCI11.TMGR 000D 003Dh

	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 30.31 for the range of the value that can be set in the RTMG[2:0] bits.

Table 30.31 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 30.32 for the range of the value that can be set in the TTMG[2:0] bits.

Table 30.32 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)

30.2.29 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 30.33 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.

30.2.30 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

30.2.31 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

30.2.32 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.

30.2.33 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.

30.2.34 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.

30.2.35 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

30.2.36 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

30.2.37 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

30.2.38 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.

30.2.39 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

30.2.40 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.

30.2.41 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.

30.2.42 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.

30.2.43 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

30.2.44 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.

30.2.45 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

30.2.46 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.

30.2.47 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.

30.2.48 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.

30.3 Operation in Asynchronous Mode

Figure 30.6 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.

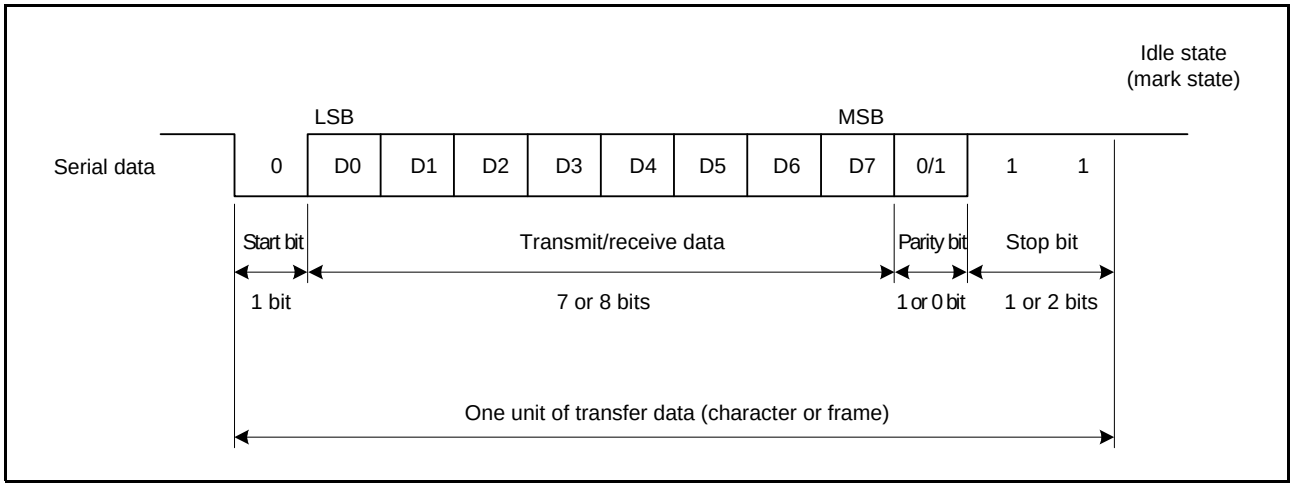


Figure 30.6 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

30.3.1 Serial Data Transfer Format

Table 30.34 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 30.4, Multi-Processor Communications Function.

Table 30.34 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

30.3.2 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 30.7. 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.

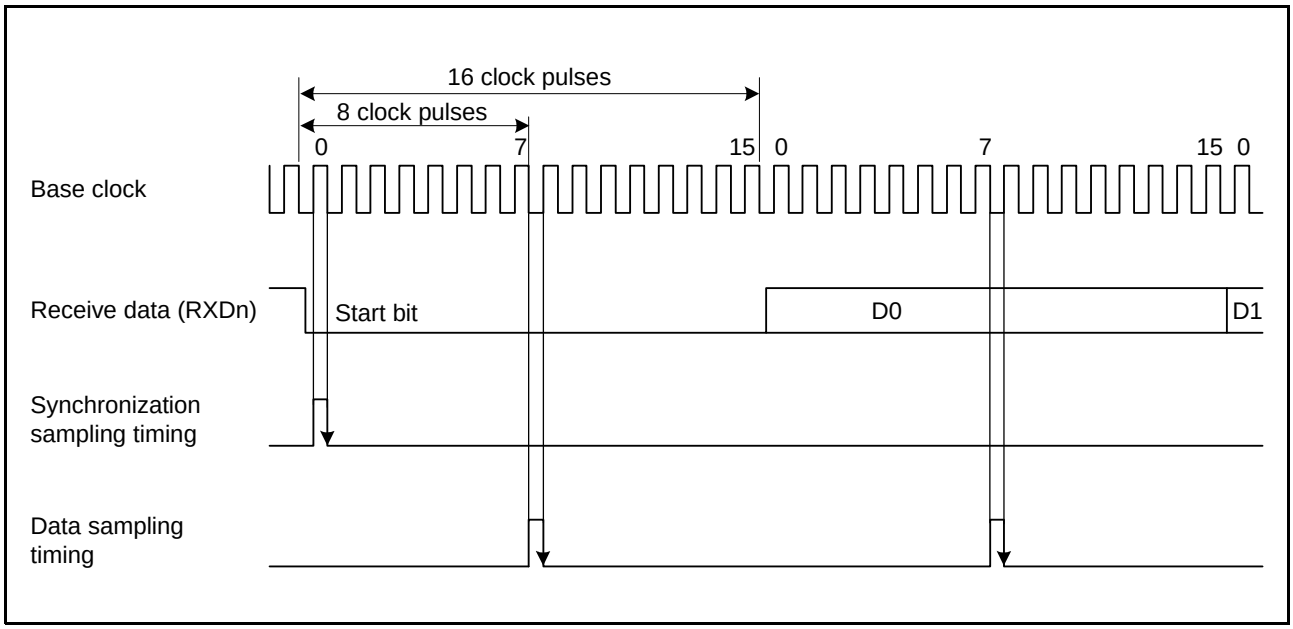


Figure 30.7 Receive Data Sampling Timing in Asynchronous Mode

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.

30.3.2.1 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 30.8 shows an example of the sampling timing adjustment.

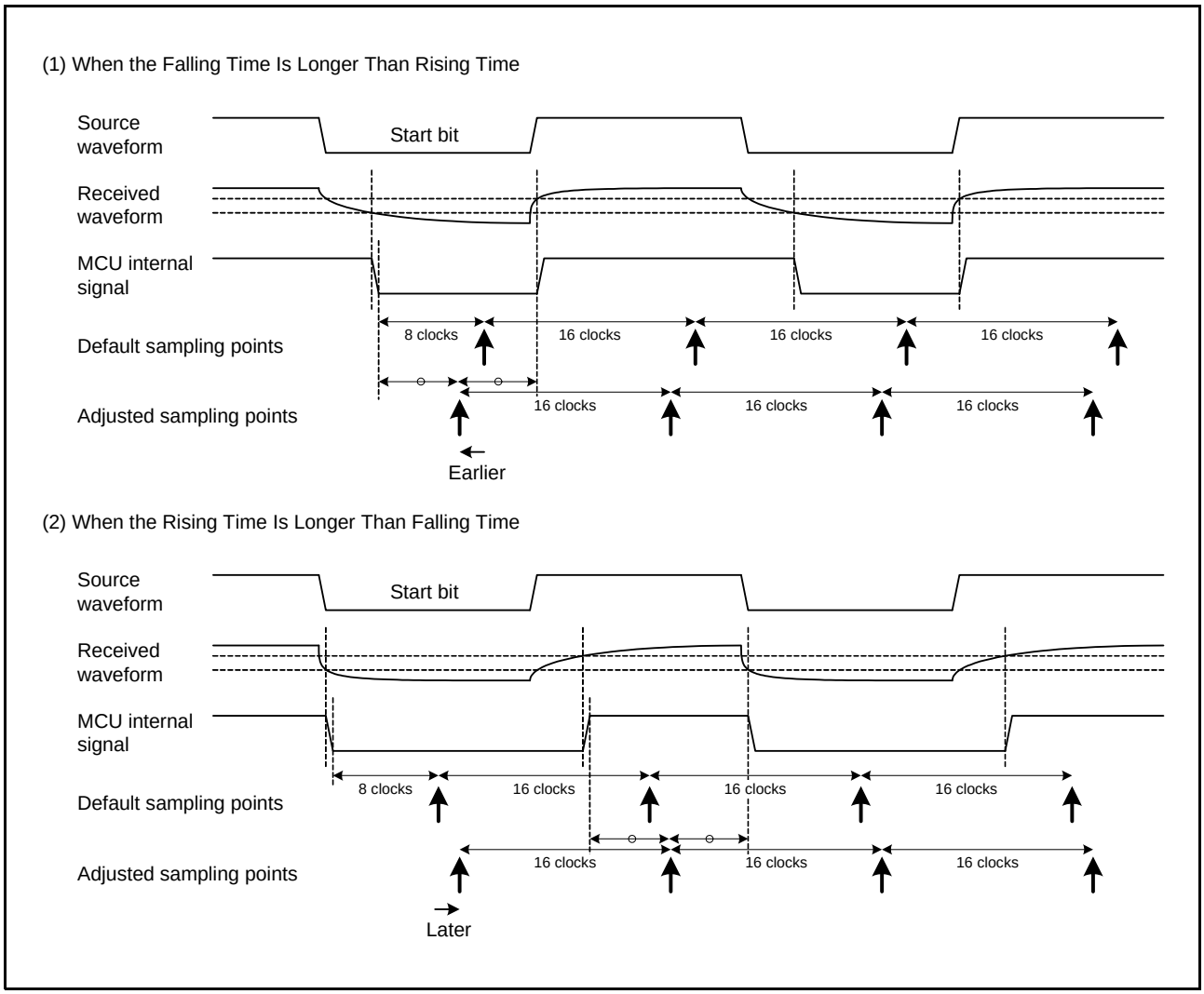


Figure 30.8 Example of Sampling Timing Adjustment (SEMR.ABCSE bit = 0 and SEMR.ABCS bit = 0)

30.3.2.2 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 30.9 shows an example of the transition timing adjustment.

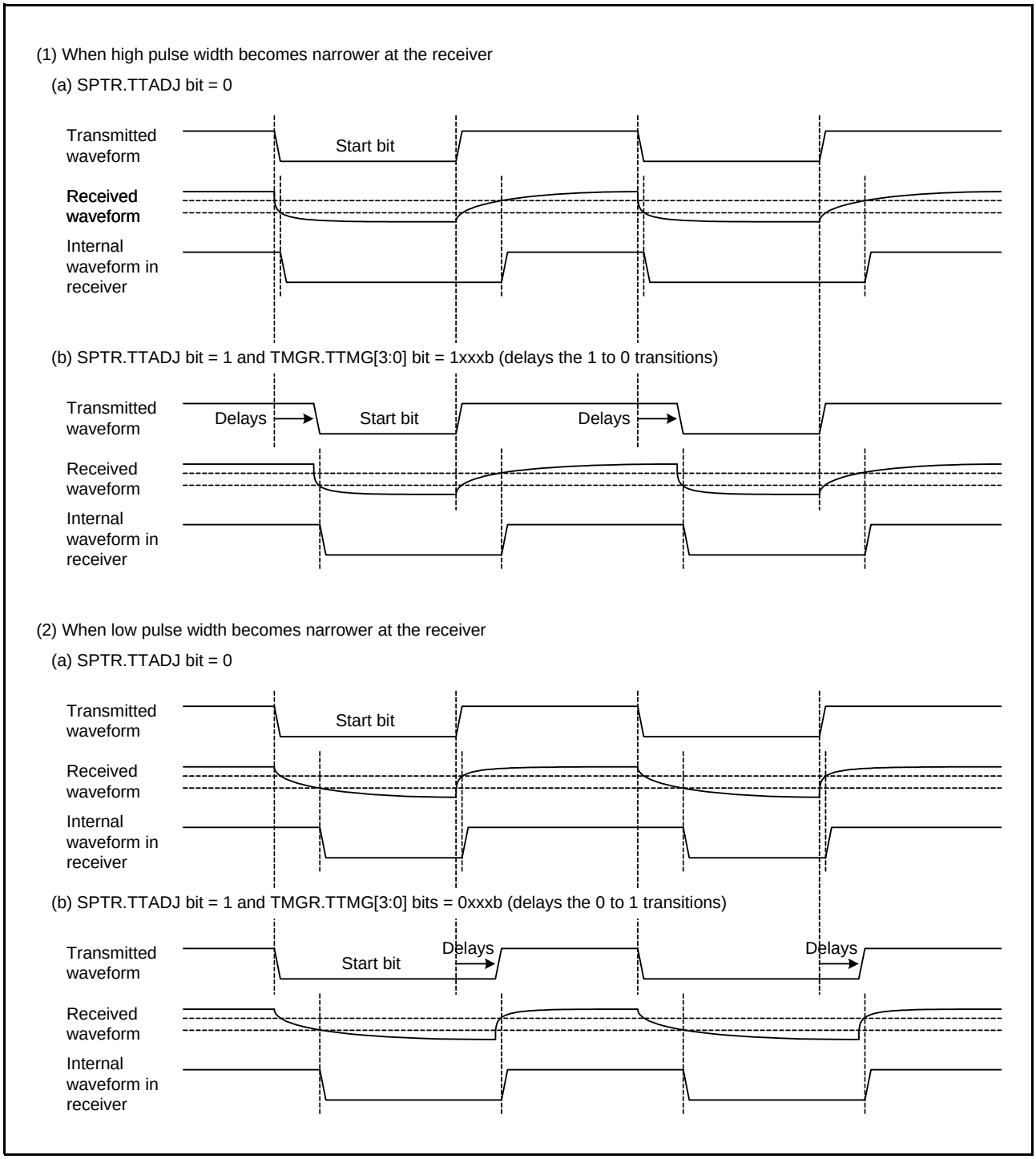


Figure 30.9 Example of Transition Timing Adjustment

30.3.3 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 30.10.

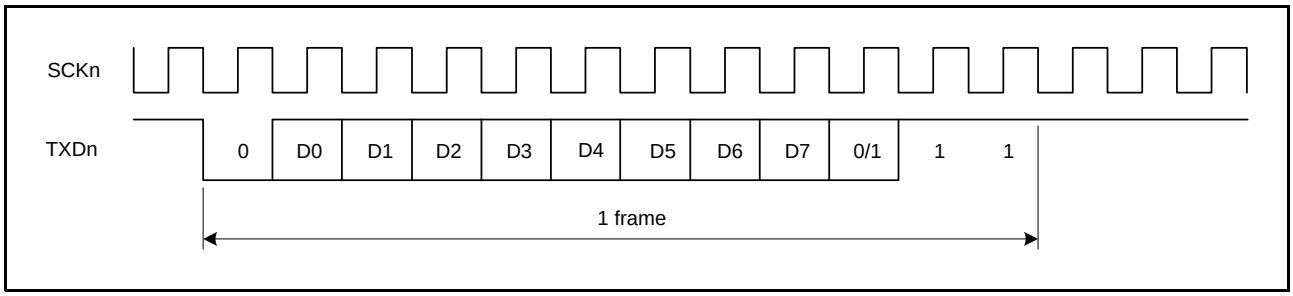


Figure 30.10 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

30.3.4 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 30.3.2, 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.

30.3.5 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.

30.3.6 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 30.11 and Figure 30.12 show examples of data match detection.

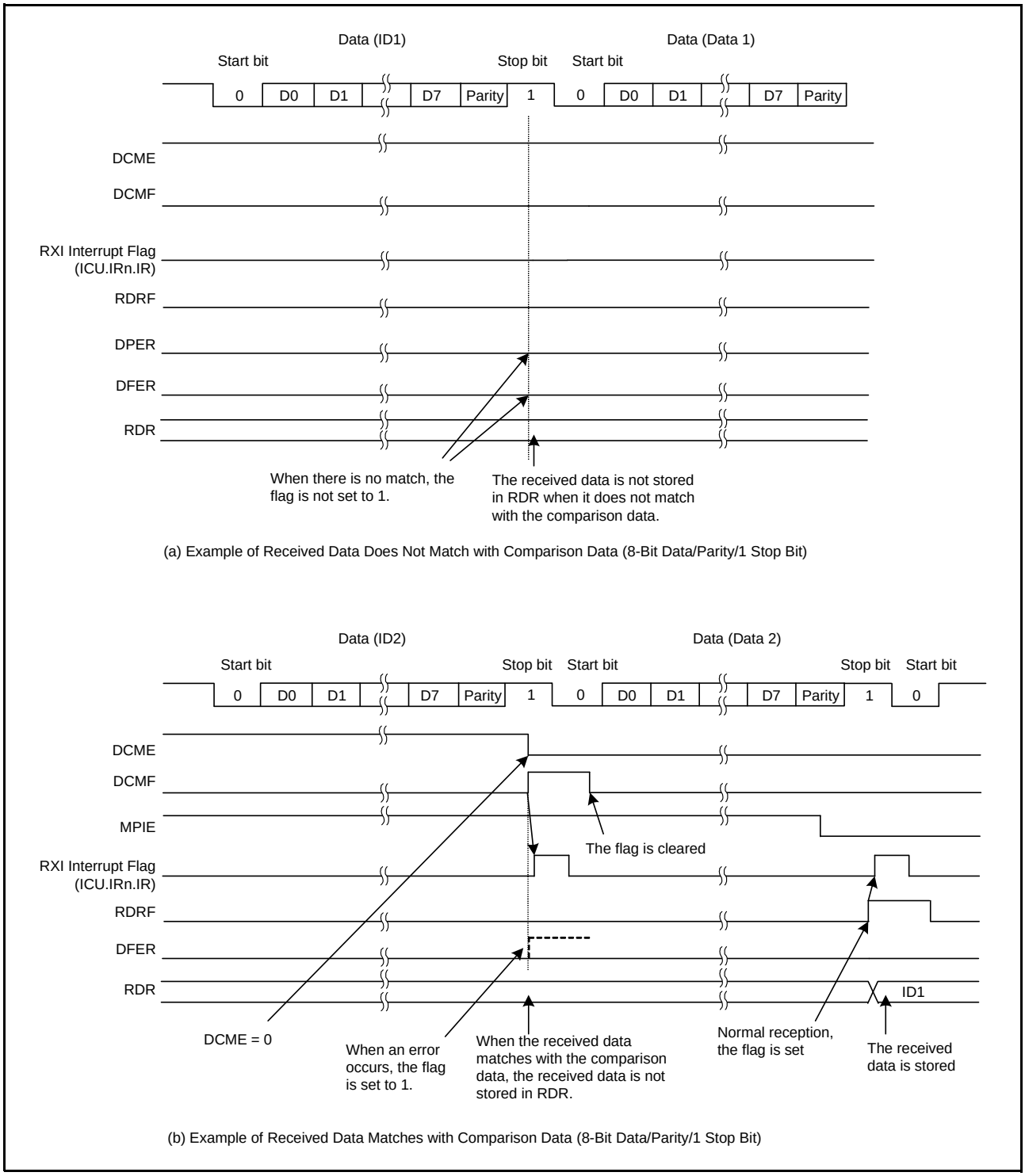


Figure 30.11 Example of Data Match Detection (1) Non Multi-Processor Mode

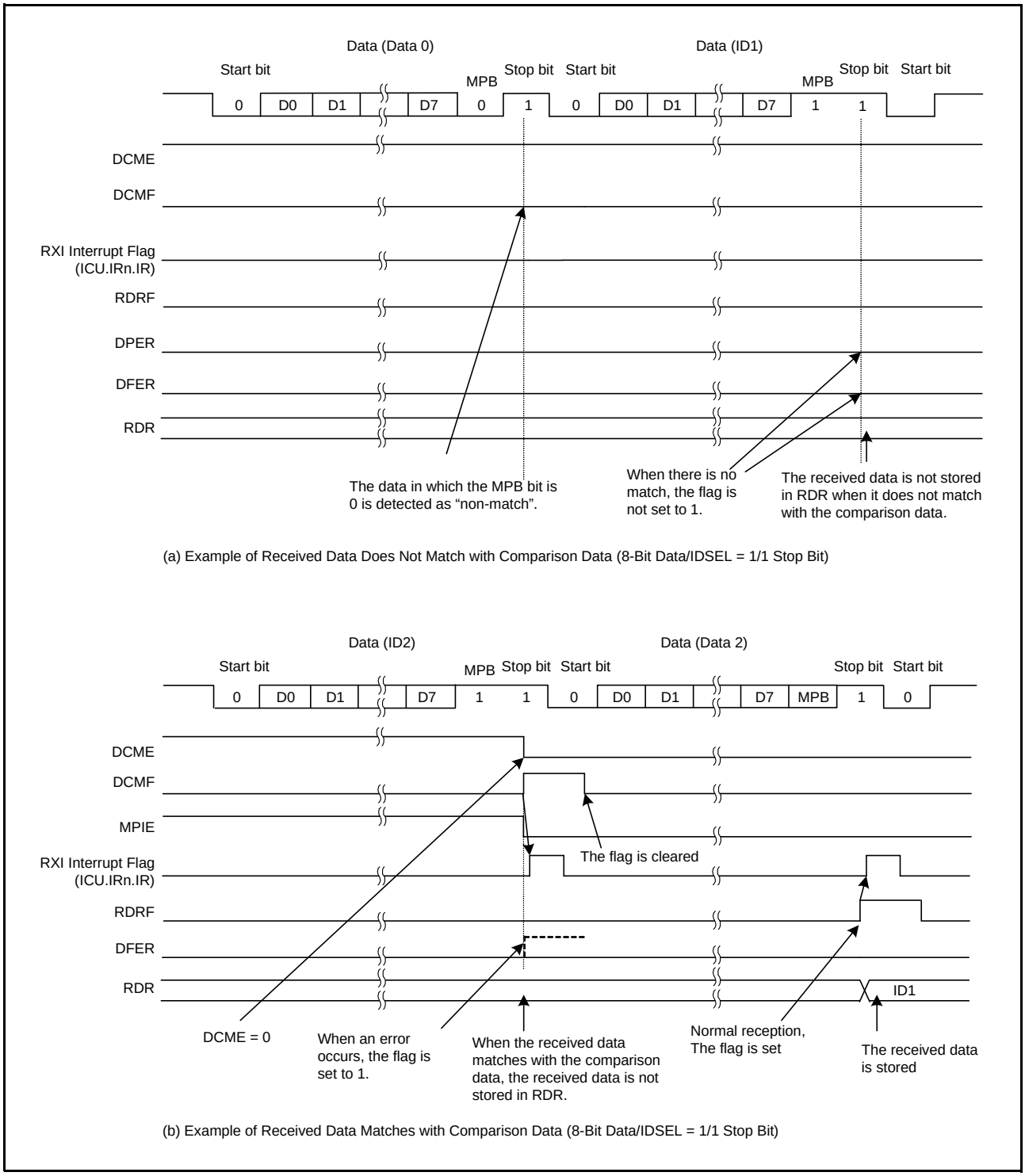


Figure 30.12 Example of Data Match Detection (2) Non Multi-Processor Mode

30.3.7 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 30.13 or Figure 30.14. 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.

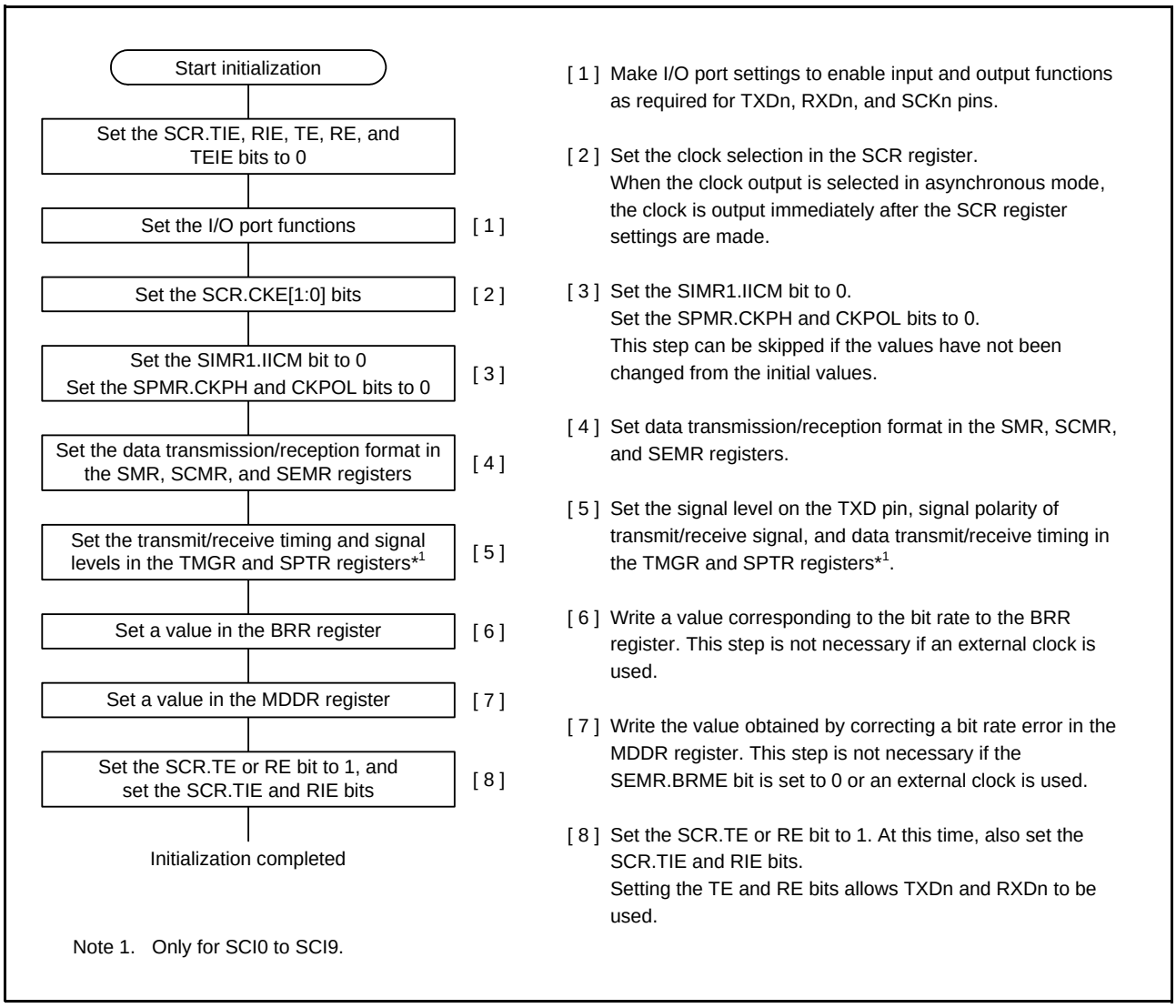


Figure 30.13 Sample SCI Initialization Flowchart (Asynchronous Mode) (for SCI0 to SCI9 and SCI12)

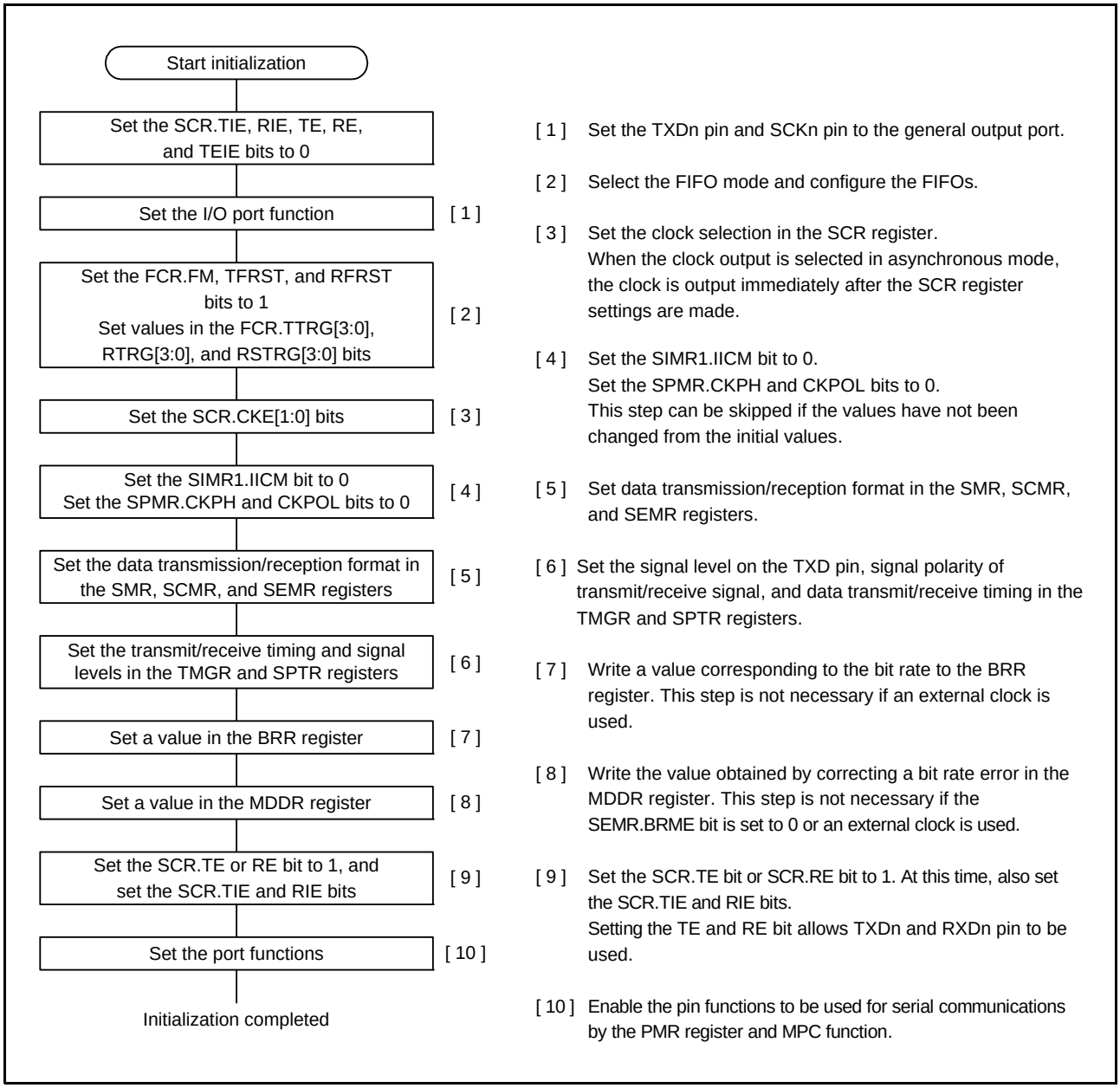


Figure 30.14 Sample SCI Initialization Flowchart (Asynchronous Mode, FIFO enabled) (for SCI10 and SCI11)

Figure 30.15 shows an example of data transmission when the SCI is set to asynchronous mode according to the flow described in Figure 30.13 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.

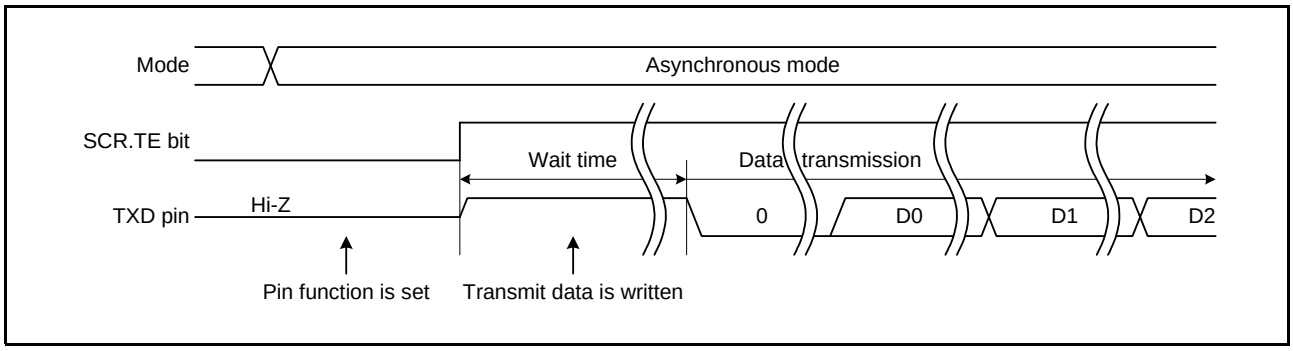


Figure 30.15 Example of Data Transmission Timing in Asynchronous Mode

30.3.8 Serial Data Transmission (Asynchronous Mode)

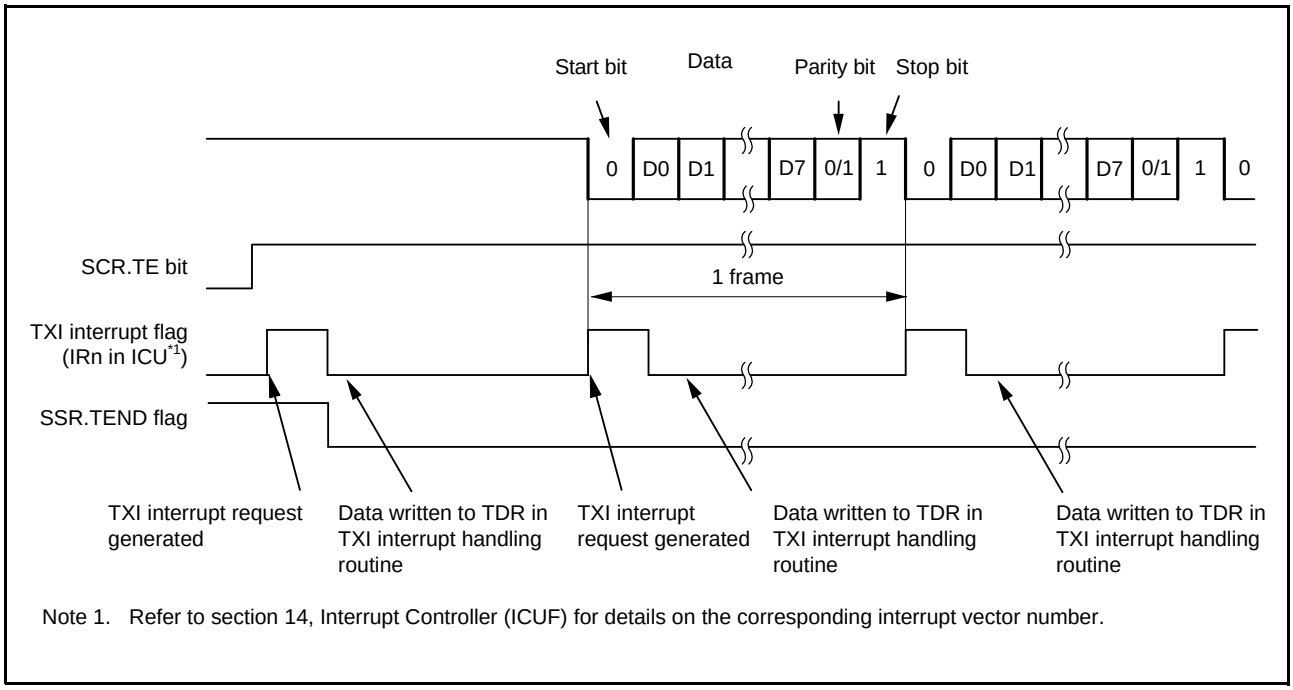
(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Figure 30.16 to Figure 30.18 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.

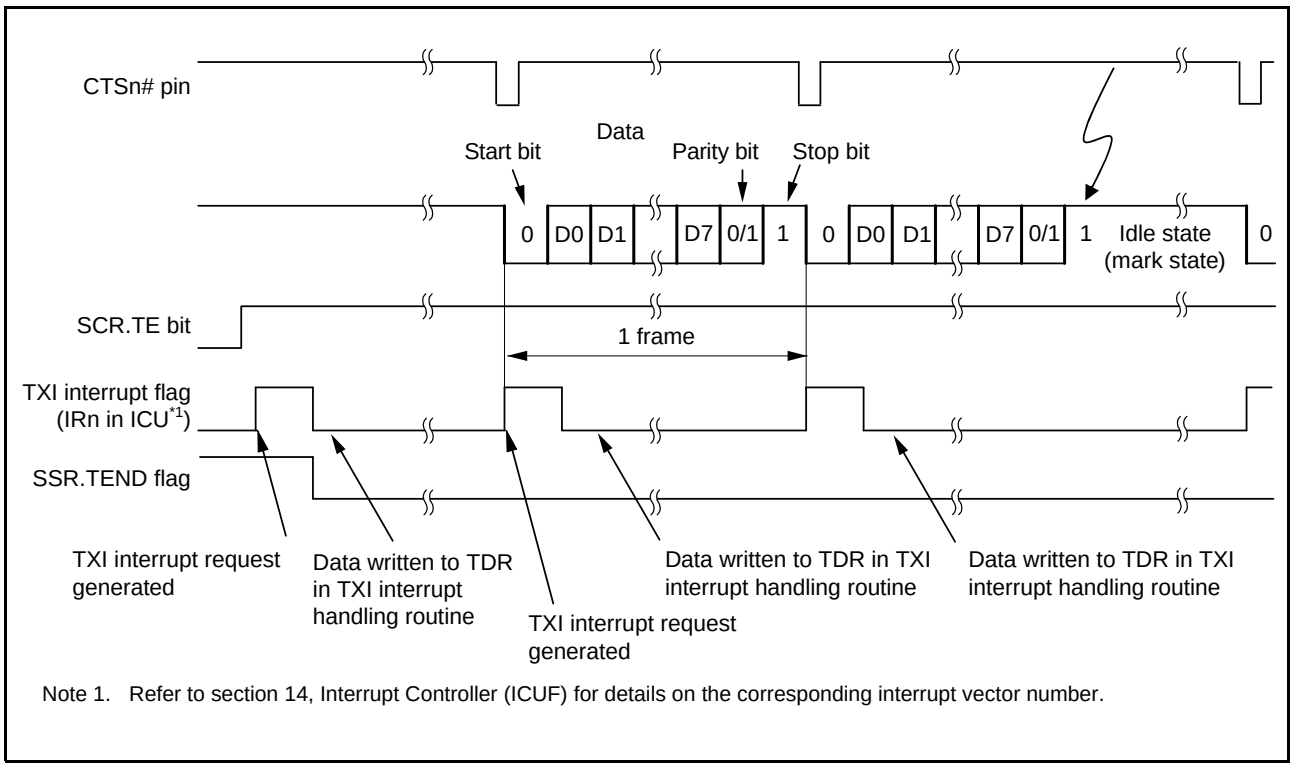
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 30.19 shows a sample flowchart for serial transmission in asynchronous mode.



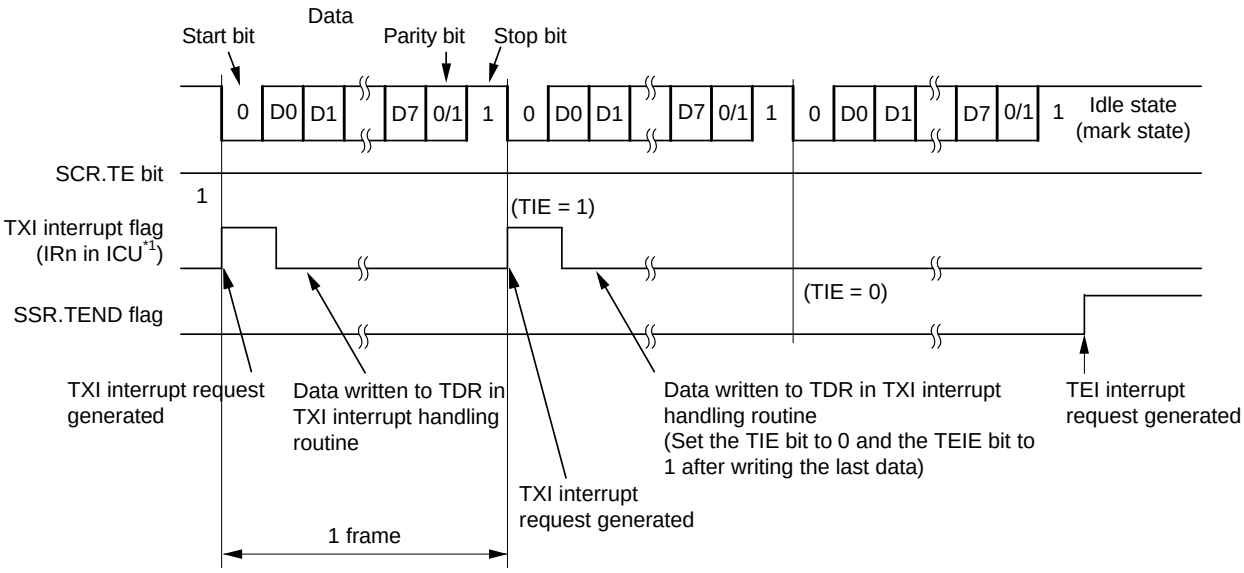
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.16 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)



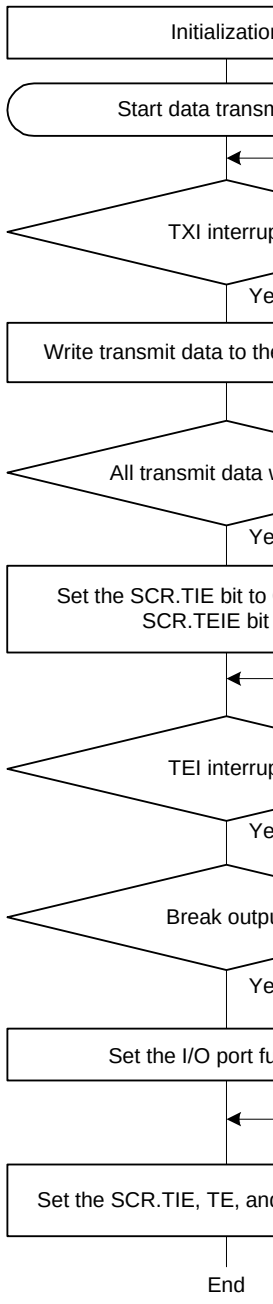
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.17 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.18 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 30.19 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.

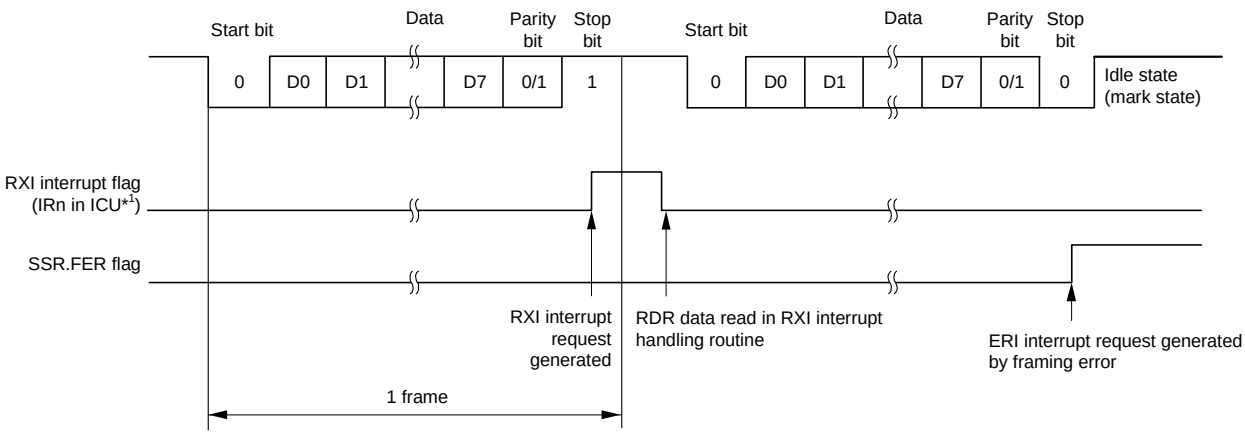
30.3.9 Serial Data Reception (Asynchronous Mode)

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Figure 30.20 and Figure 30.21 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

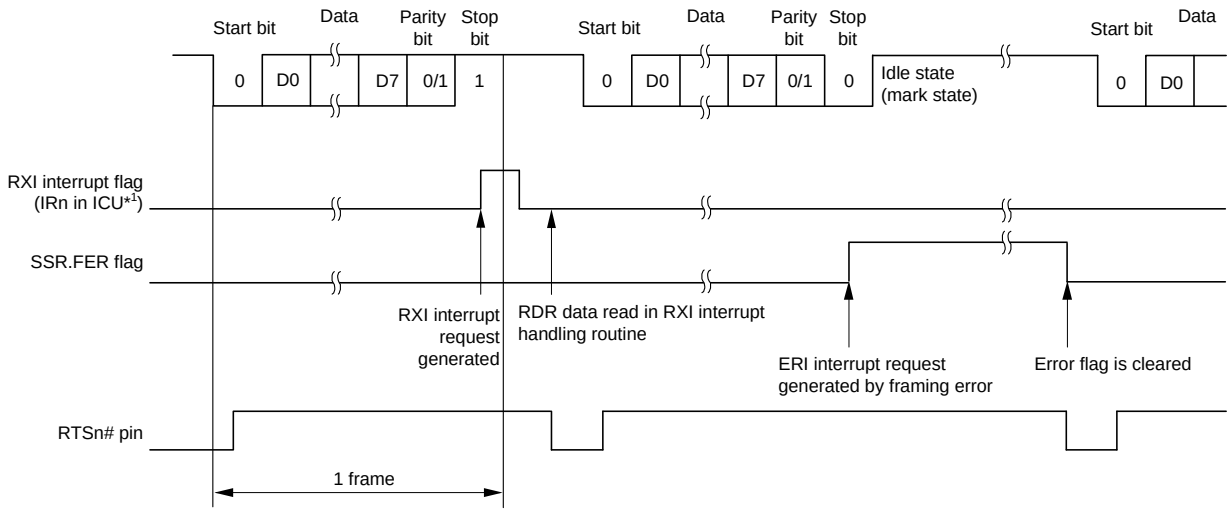
1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.20 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.21 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 30.35 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.

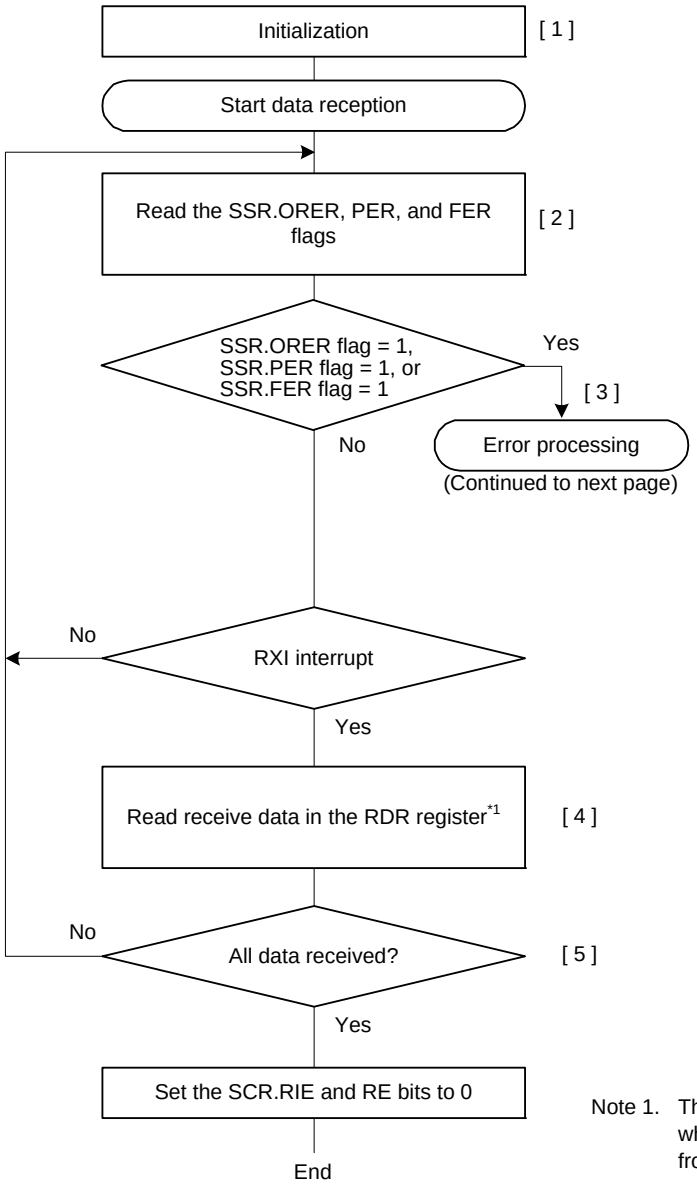
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in the RDR (or the RDRL) register.

Figure 30.22 and Figure 30.23 show samples of flowcharts for serial data reception.

Table 30.35 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 30.22 Example Flowchart of Serial Reception in Asynchronous Mode (1)

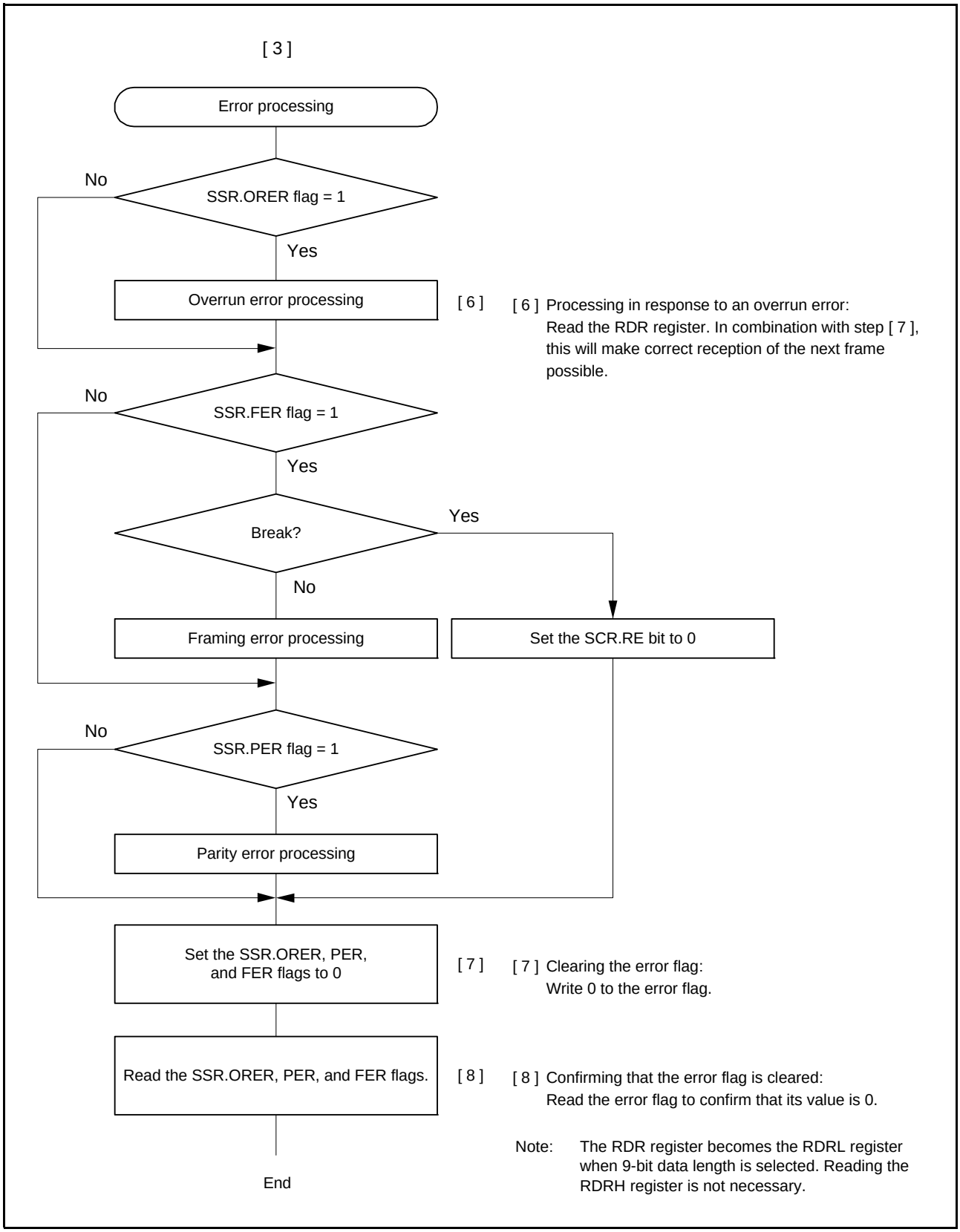


Figure 30.23 Example Flowchart of Serial Reception in Asynchronous Mode (2)

(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.

30.4 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 30.24 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.

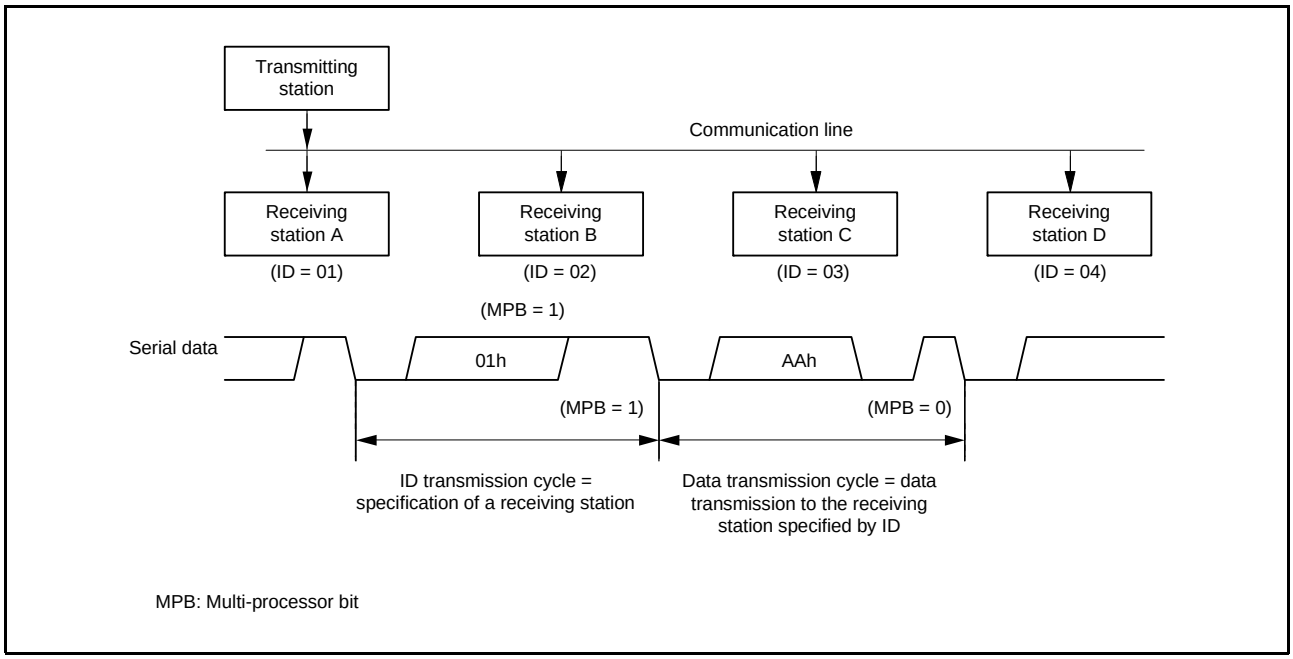


Figure 30.24 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

(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.

30.4.1 Multi-Processor Serial Data Transmission

Figure 30.25 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.

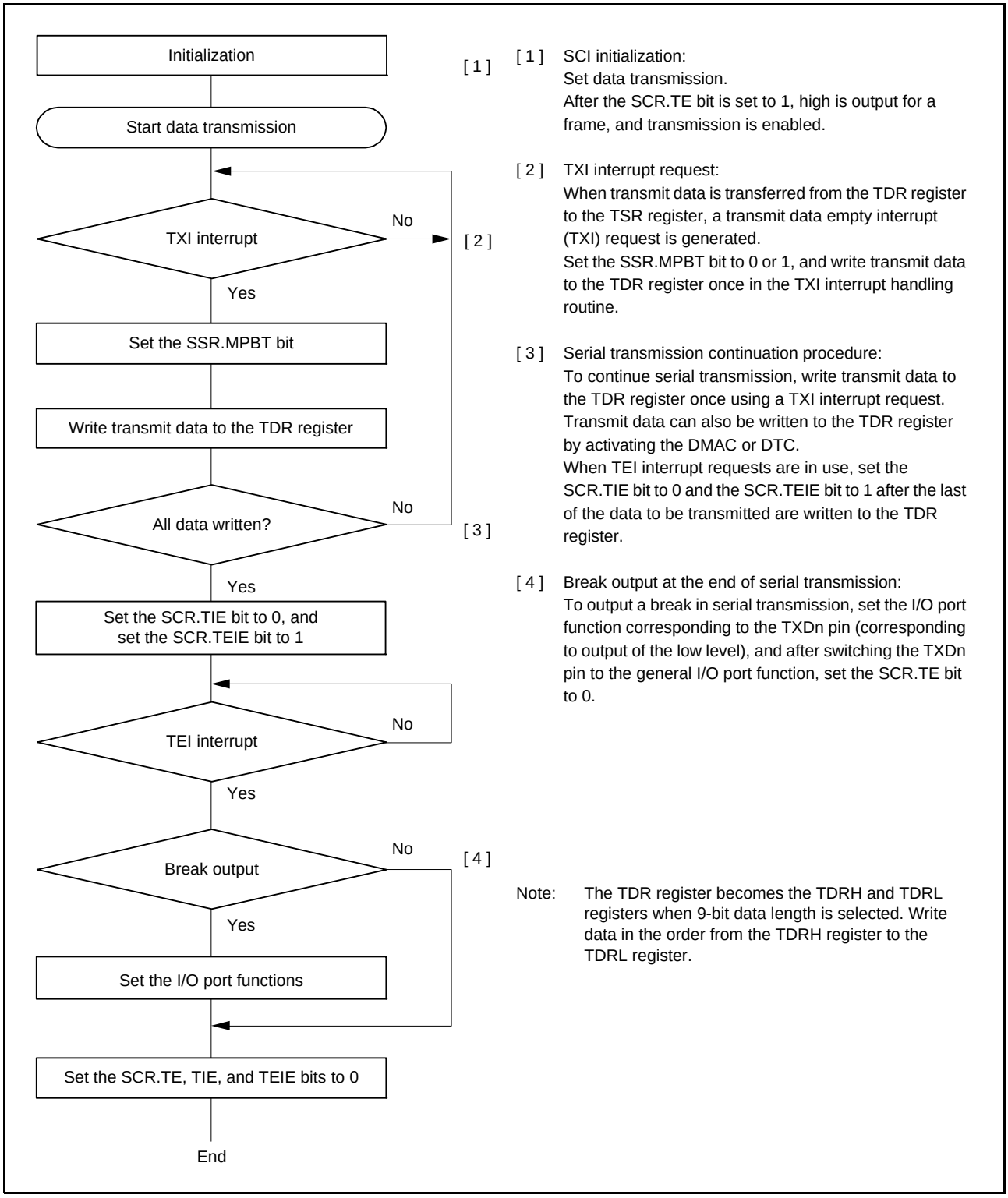
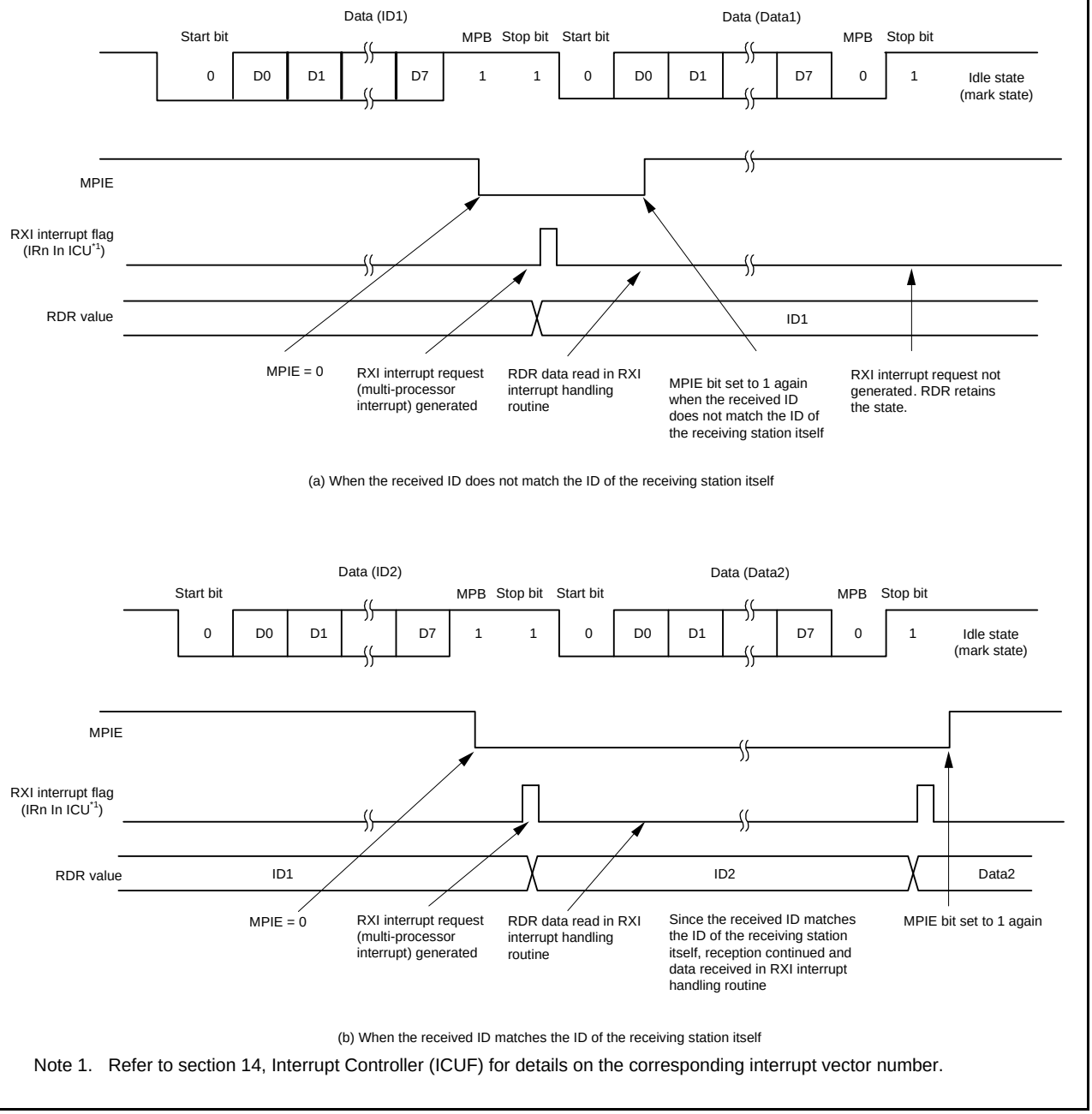


Figure 30.25 Example of Multi-Processor Serial Transmission Flowchart

30.4.2 Multi-Processor Serial Data Reception

Figure 30.27 and Figure 30.28 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 30.26 is the example of operation for reception.



Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.26 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

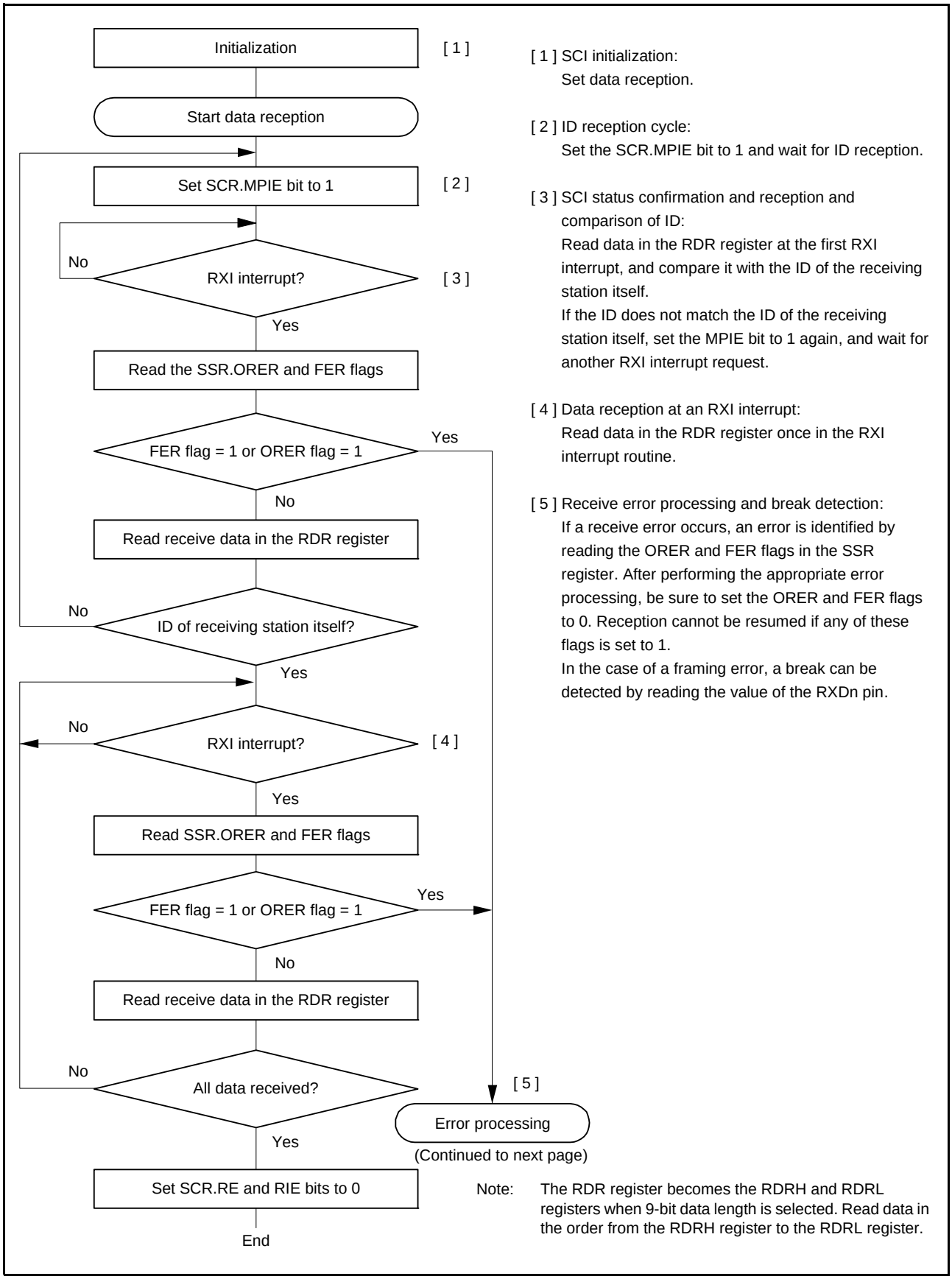


Figure 30.27 Example of Multi-Processor Serial Reception Flowchart (1)

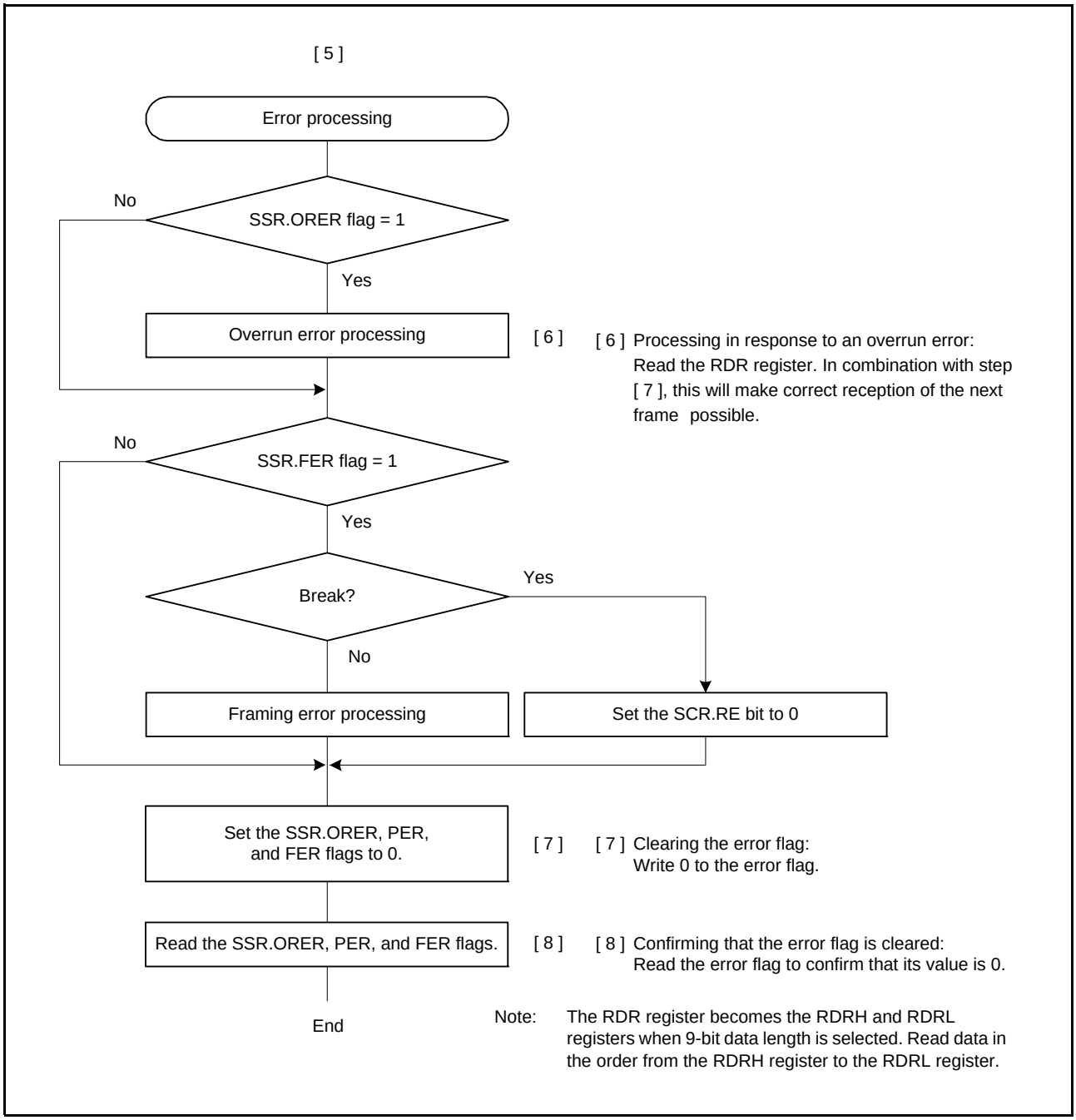


Figure 30.28 Example of Multi-Processor Serial Reception Flowchart (2)

30.5 Operation in Clock Synchronous Mode

Figure 30.29 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.

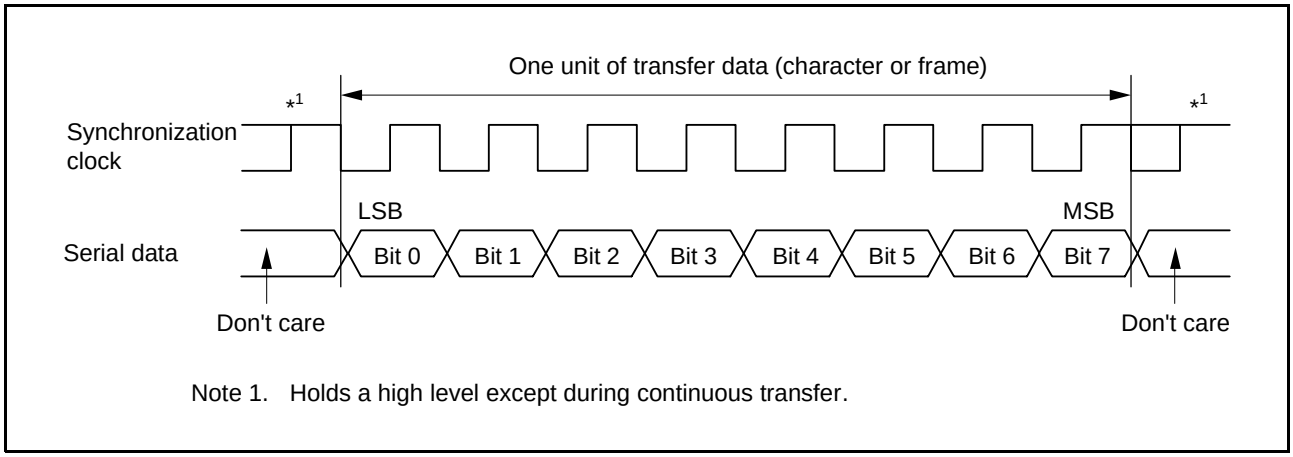


Figure 30.29 Data Format in Clock Synchronous Serial Communications (LSB First)

30.5.1 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.

30.5.2 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

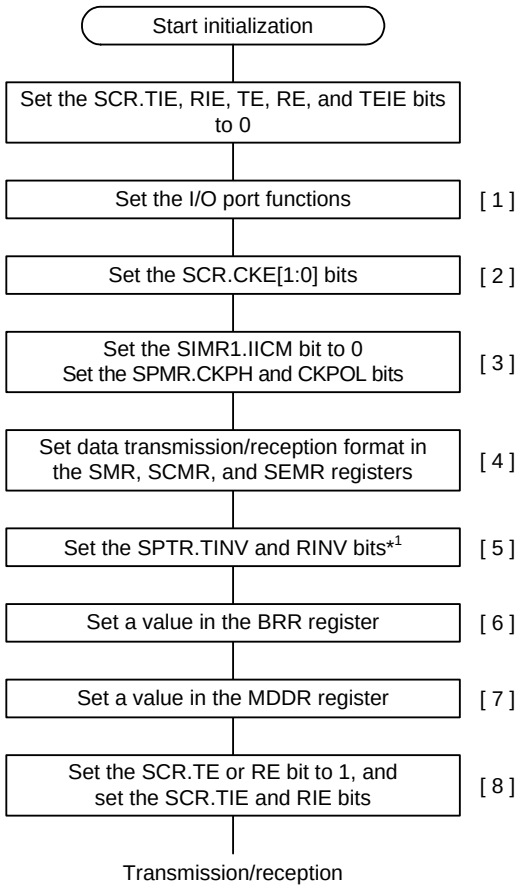
[Conditions for low-level output]
When the following conditions are all satisfied.

- 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.

30.5.3 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 30.30. 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 30.30 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

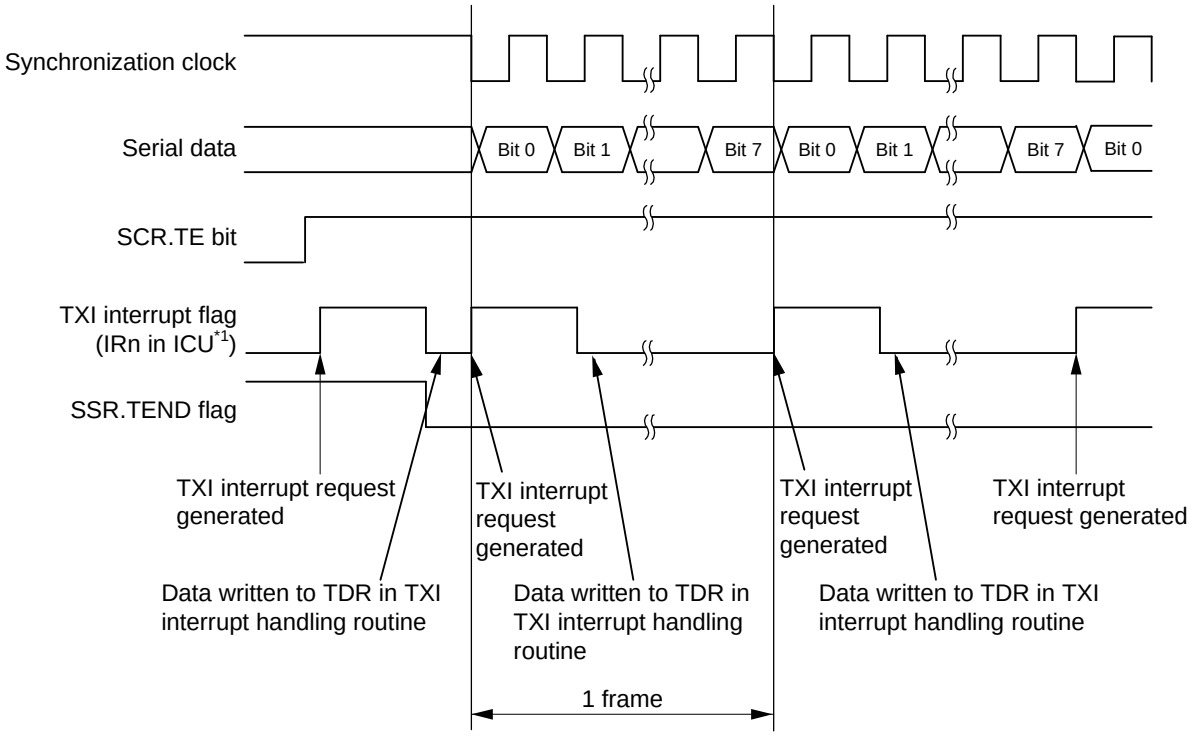
30.5.4 Serial Data Transmission (Clock Synchronous Mode)

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Figure 30.31, Figure 30.32, and Figure 30.33 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

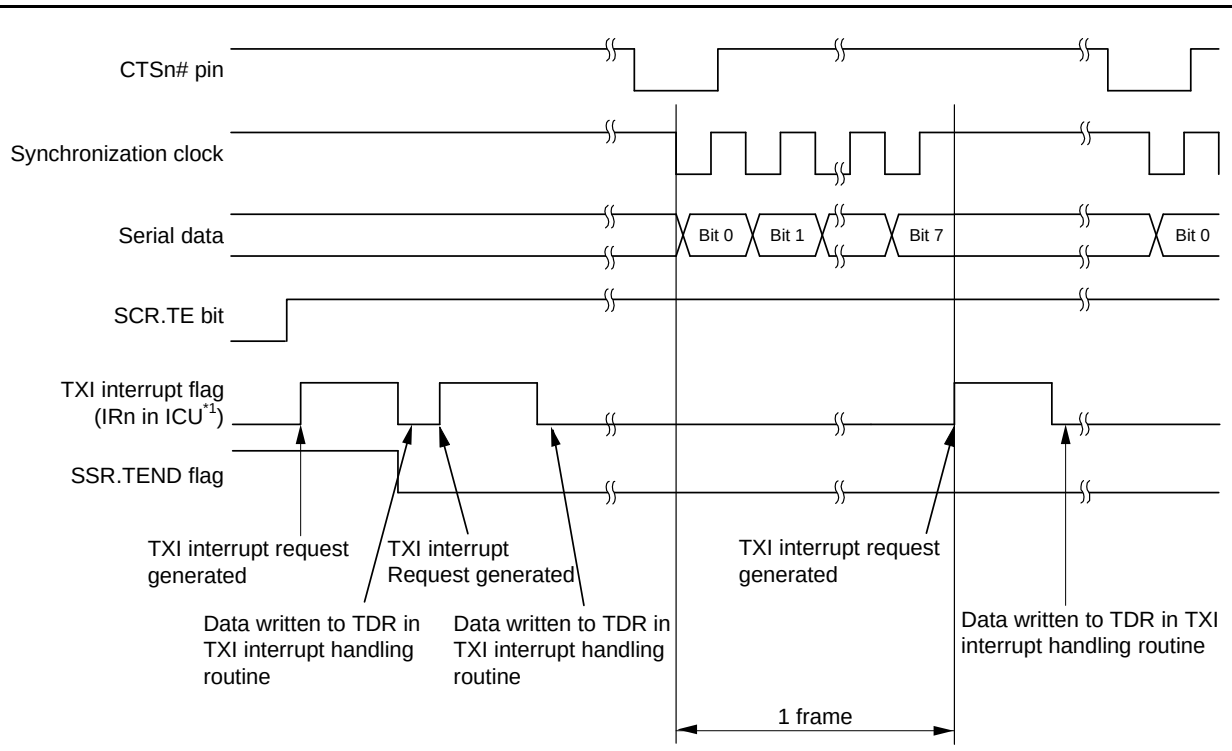
1. The SCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 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 30.34 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.



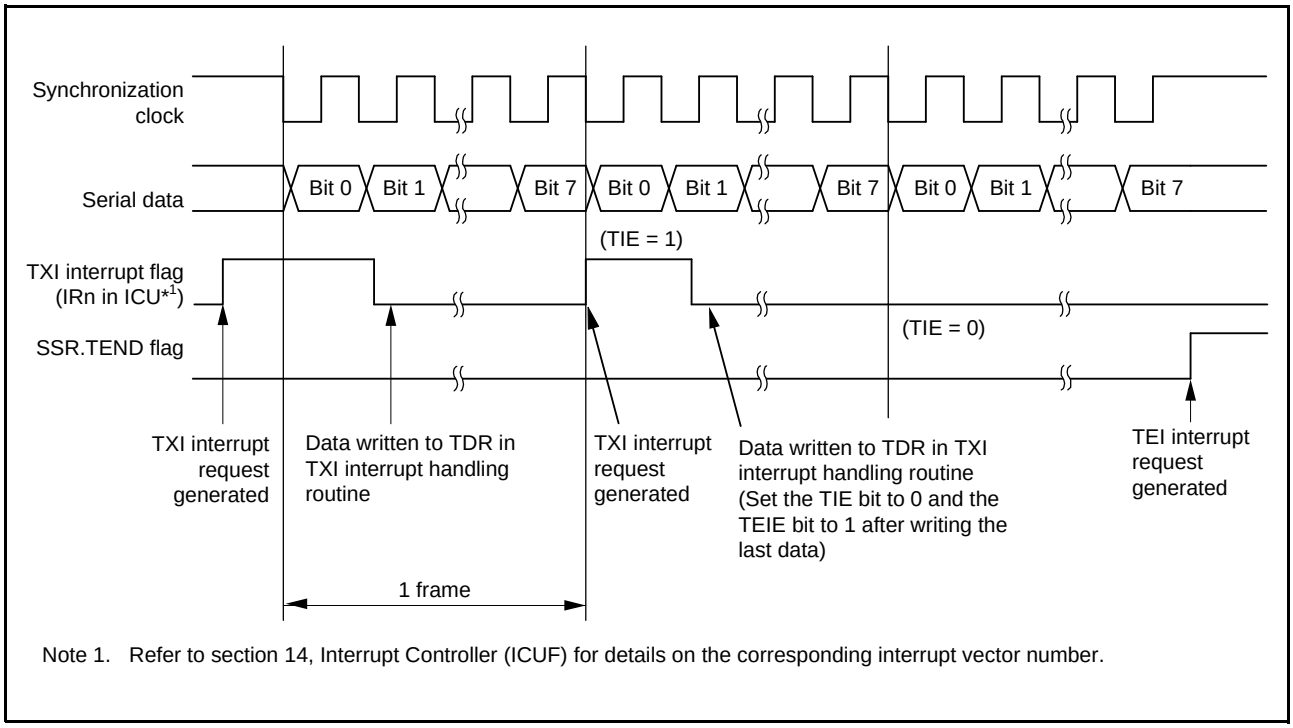
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.31 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



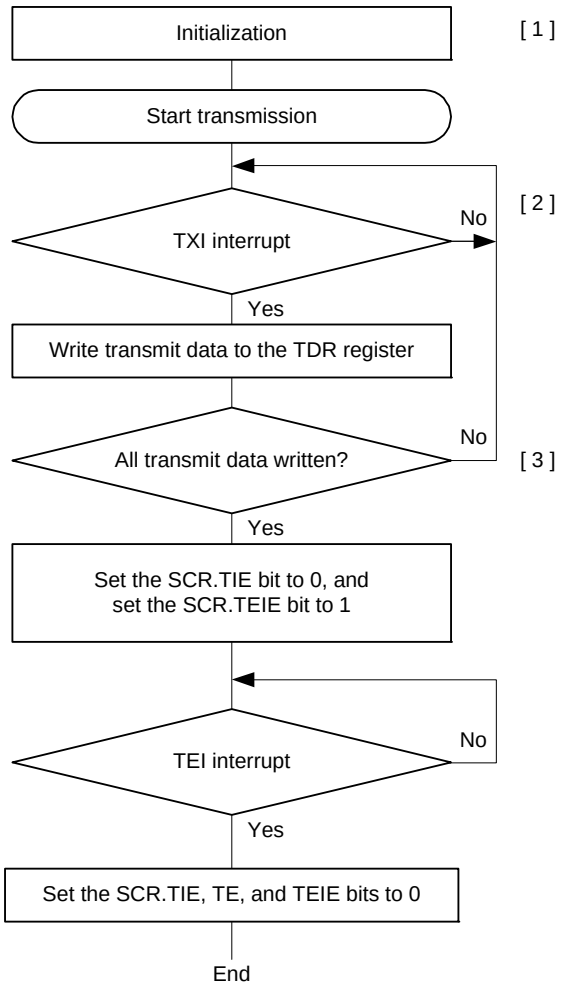
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.32 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.33 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 30.34 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.

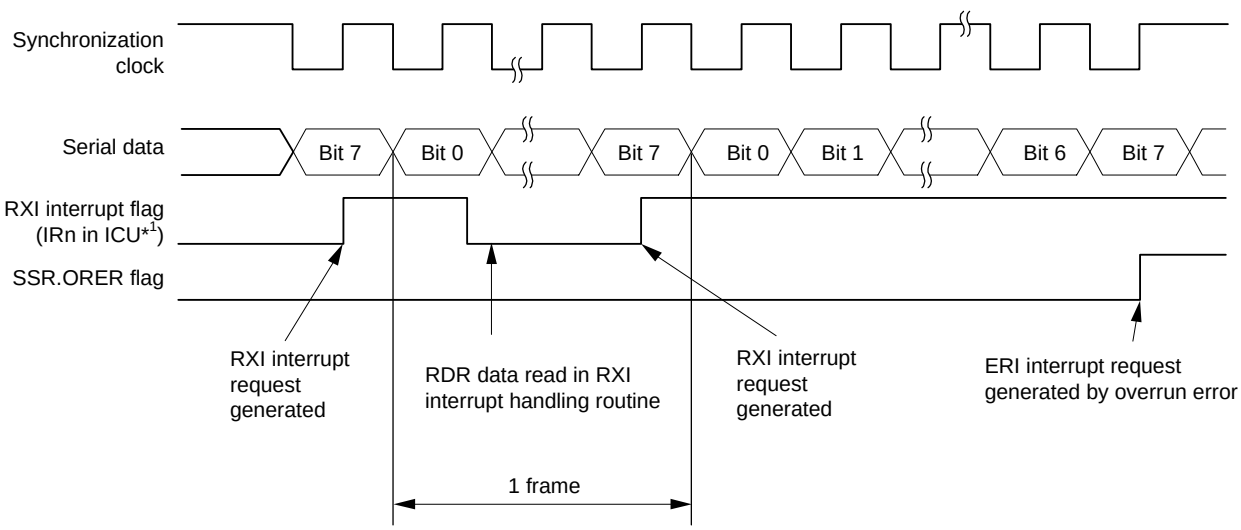
1. When the SCR.TIE and SCR.TE bits are simultaneously set to 1, a TXI interrupt request is generated. Up to 16 minus FDR.T[4:0] bytes of transmit data can be set in the FTDR register (FTDR.L register) within the TXI interrupt processing routine.
2. After the transmit data is written to the FTDR register, 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.

30.5.5 Serial Data Reception (Clock Synchronous Mode)

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

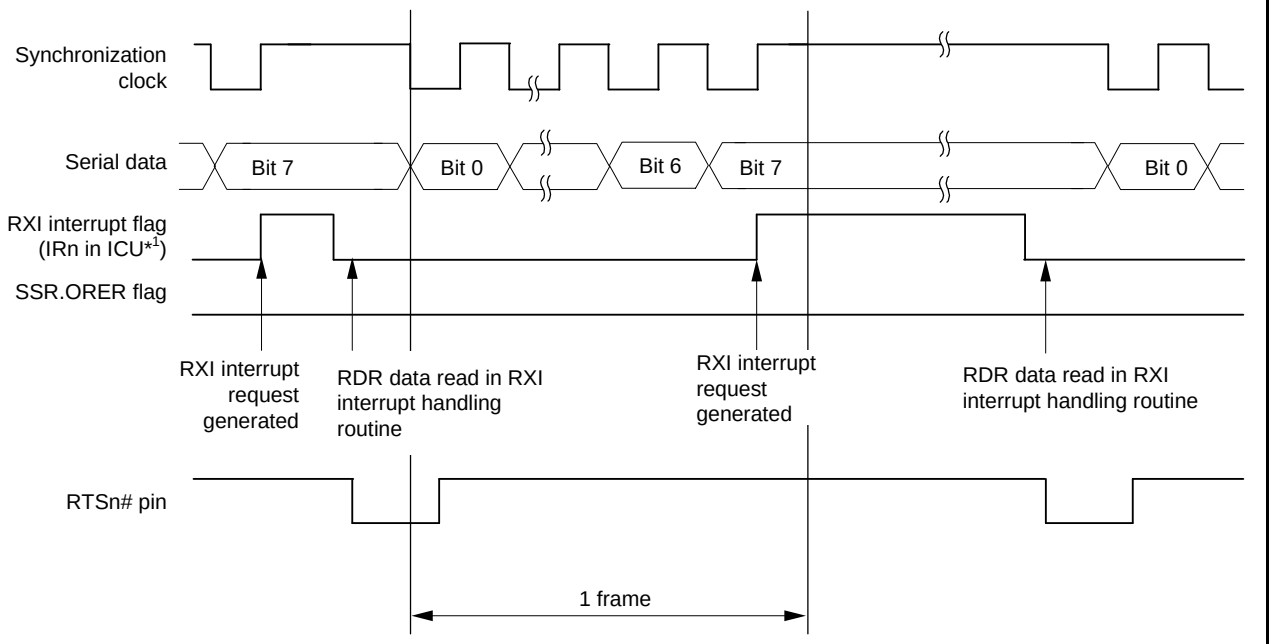
Figure 30.35 and Figure 30.36 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.35 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

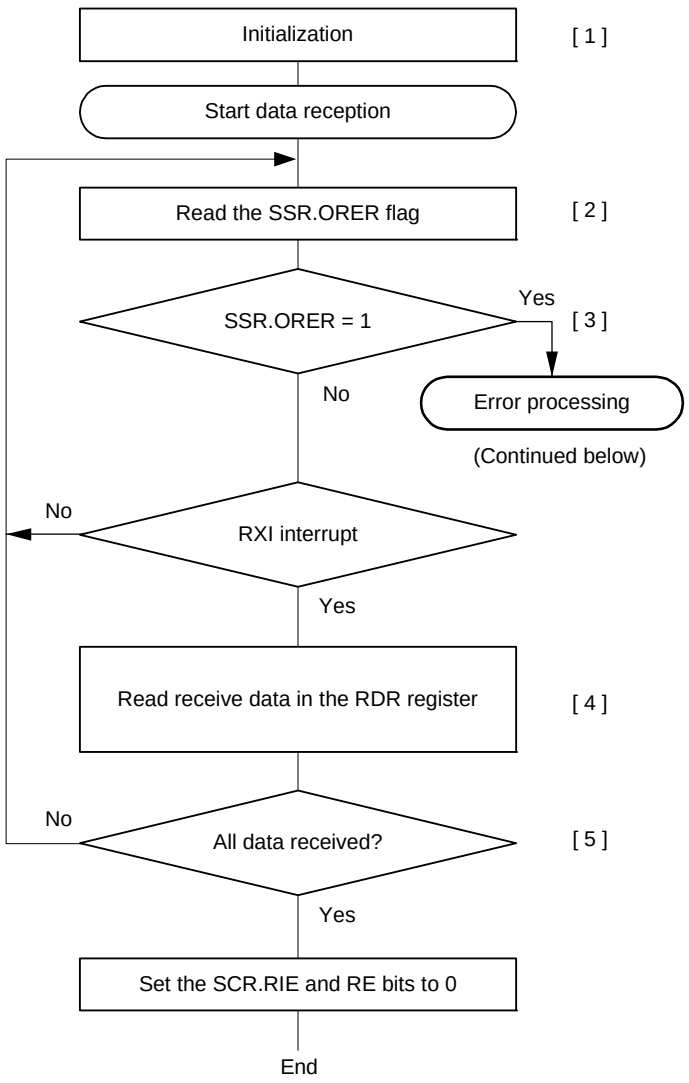


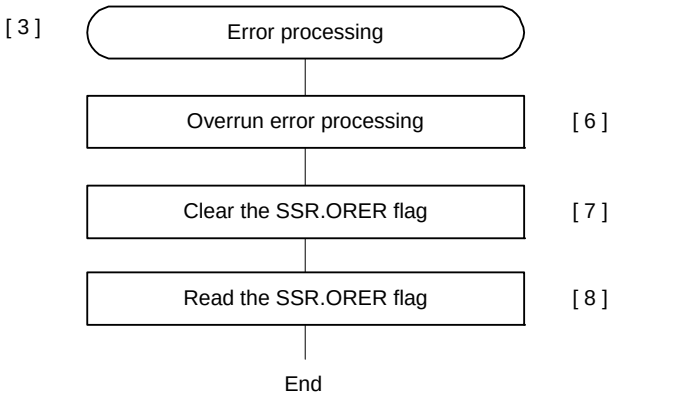
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.36 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 30.37 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 30.37 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.

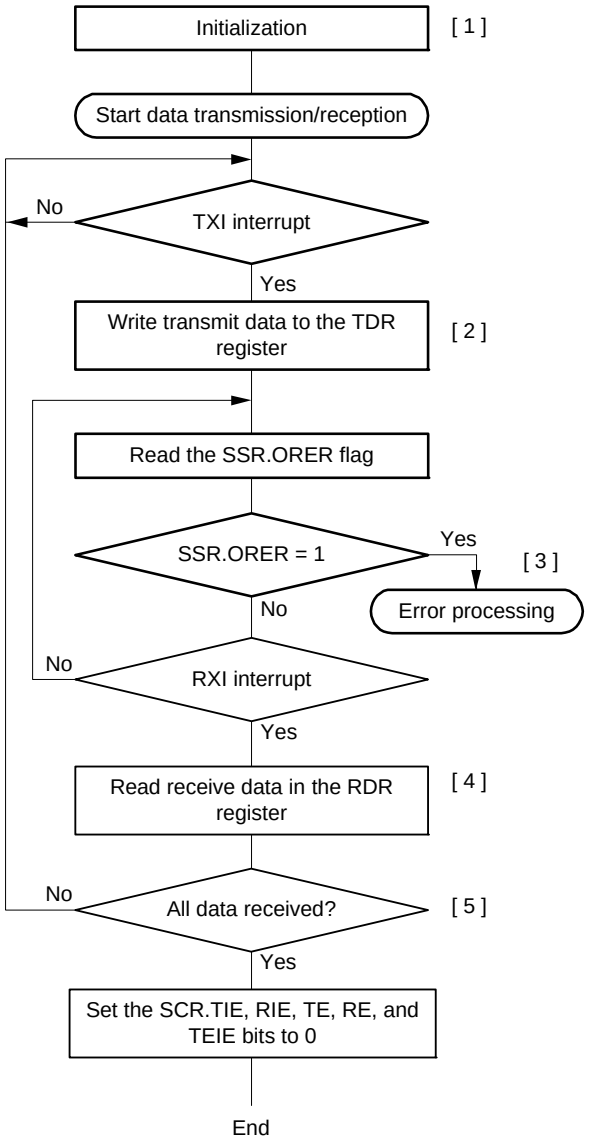
30.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 30.38 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 30.38 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

30.6 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.

30.6.1 Sample Connection

Figure 30.39 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.

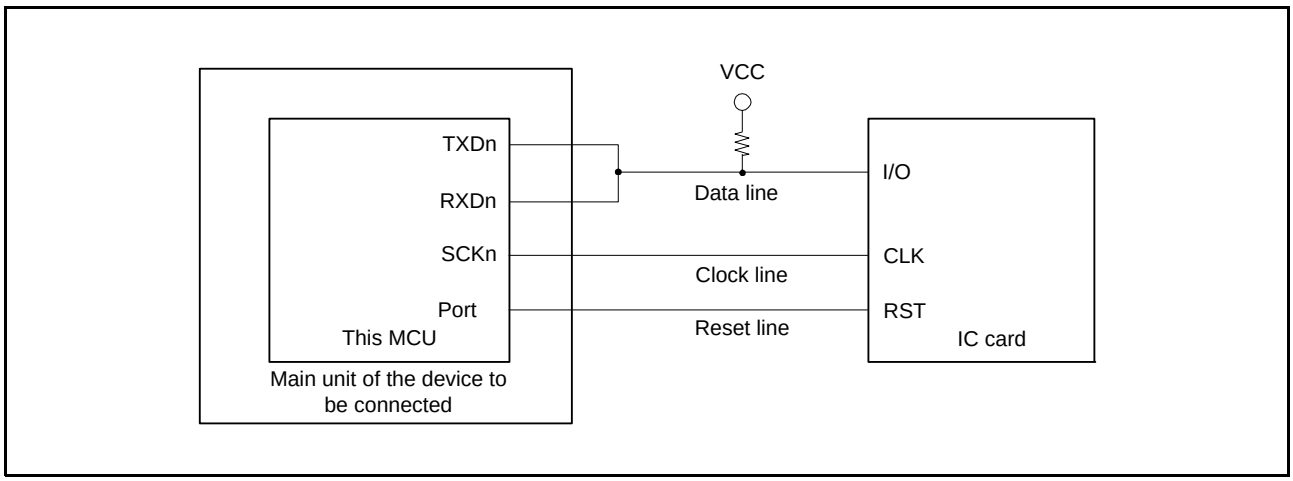


Figure 30.39 Sample Connection with a Smart Card (IC Card)

30.6.2 Data Format (Except in Block Transfer Mode)

Figure 30.40 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.

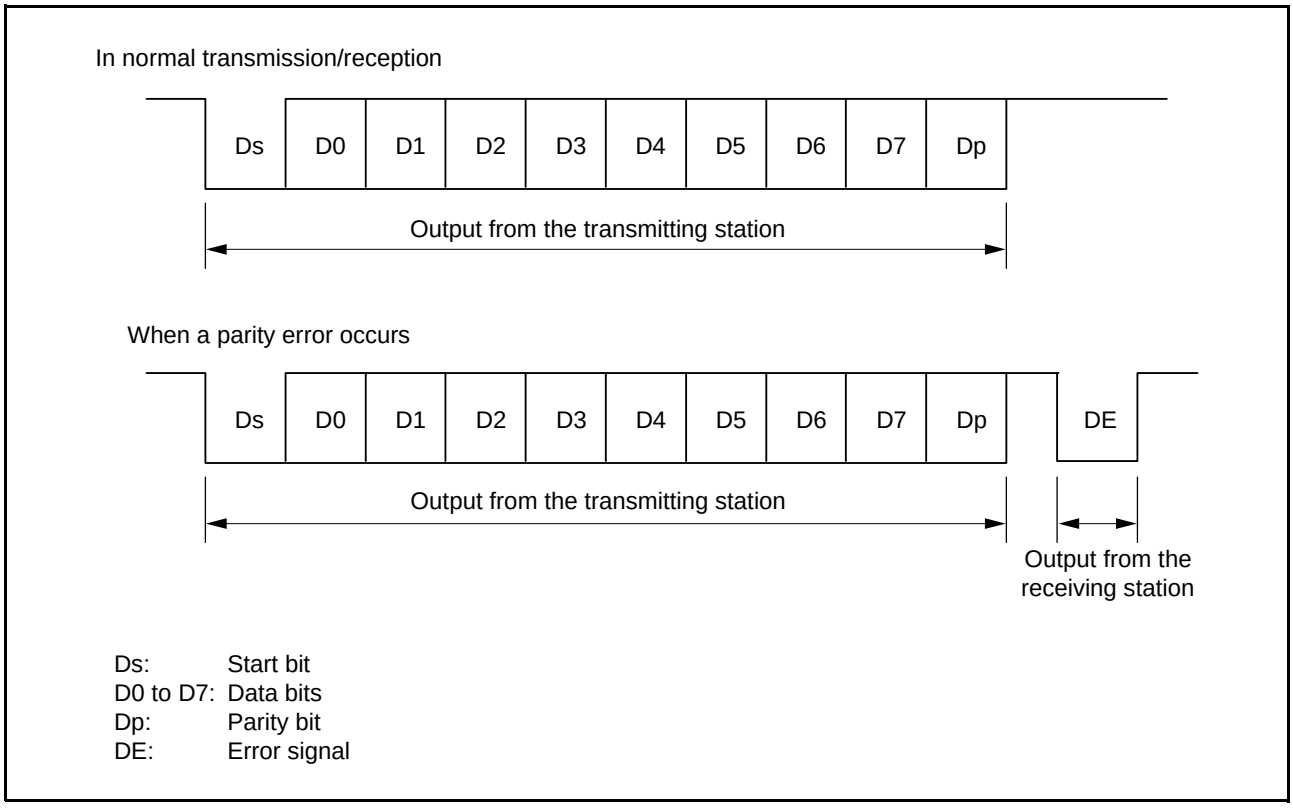


Figure 30.40 Data Formats in Smart Card Interface Mode

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 30.41. 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.

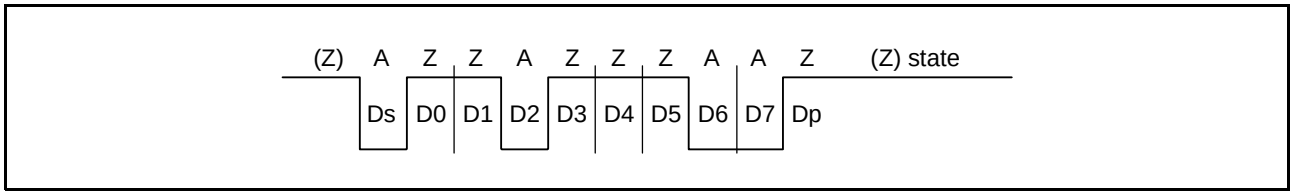


Figure 30.41 Direct Convention (SCMR.SDIR bit = 0, SCMR.SINV bit = 0, SMR.PM bit = 0)

(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 30.42. 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.

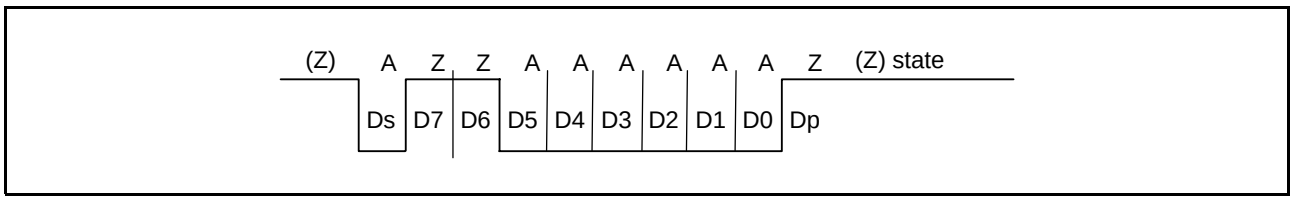


Figure 30.42 Inverse Convention (SCMR.SDIR bit = 1, SCMR.SINV bit = 1, SMR.PM bit = 1)

30.6.3 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.

30.6.4 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 30.43. 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 (%)

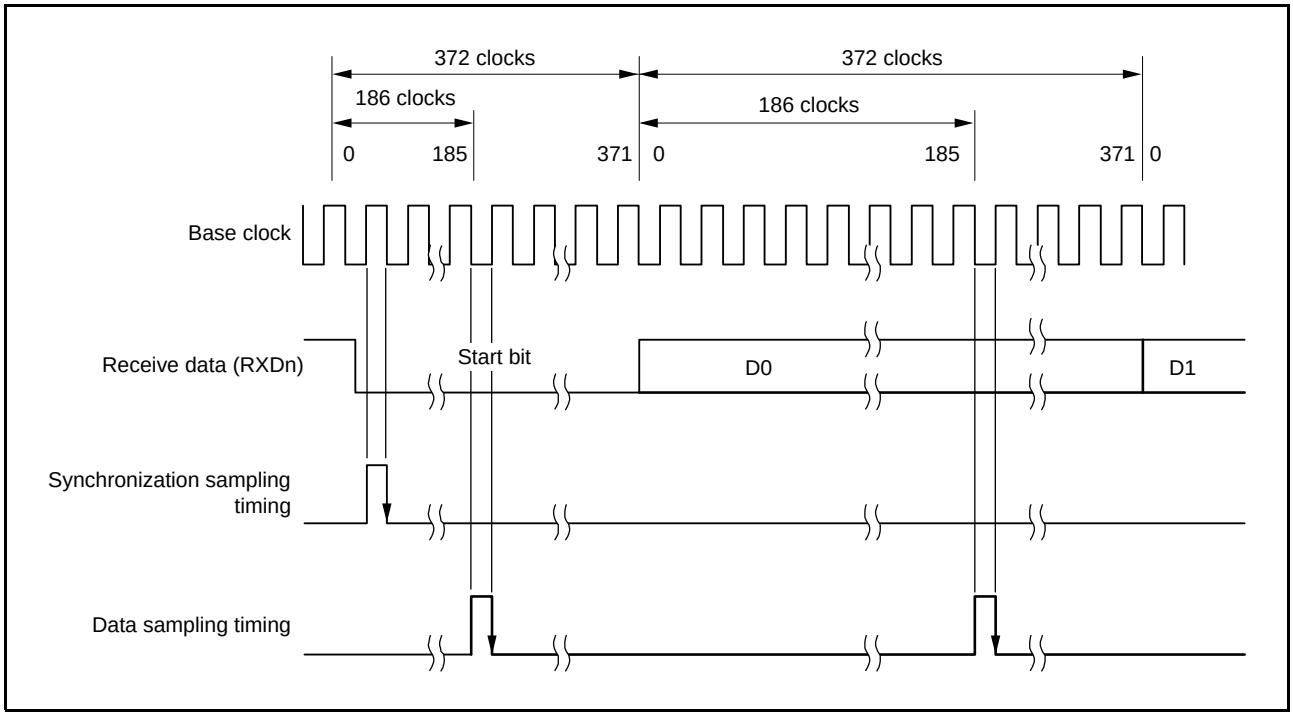
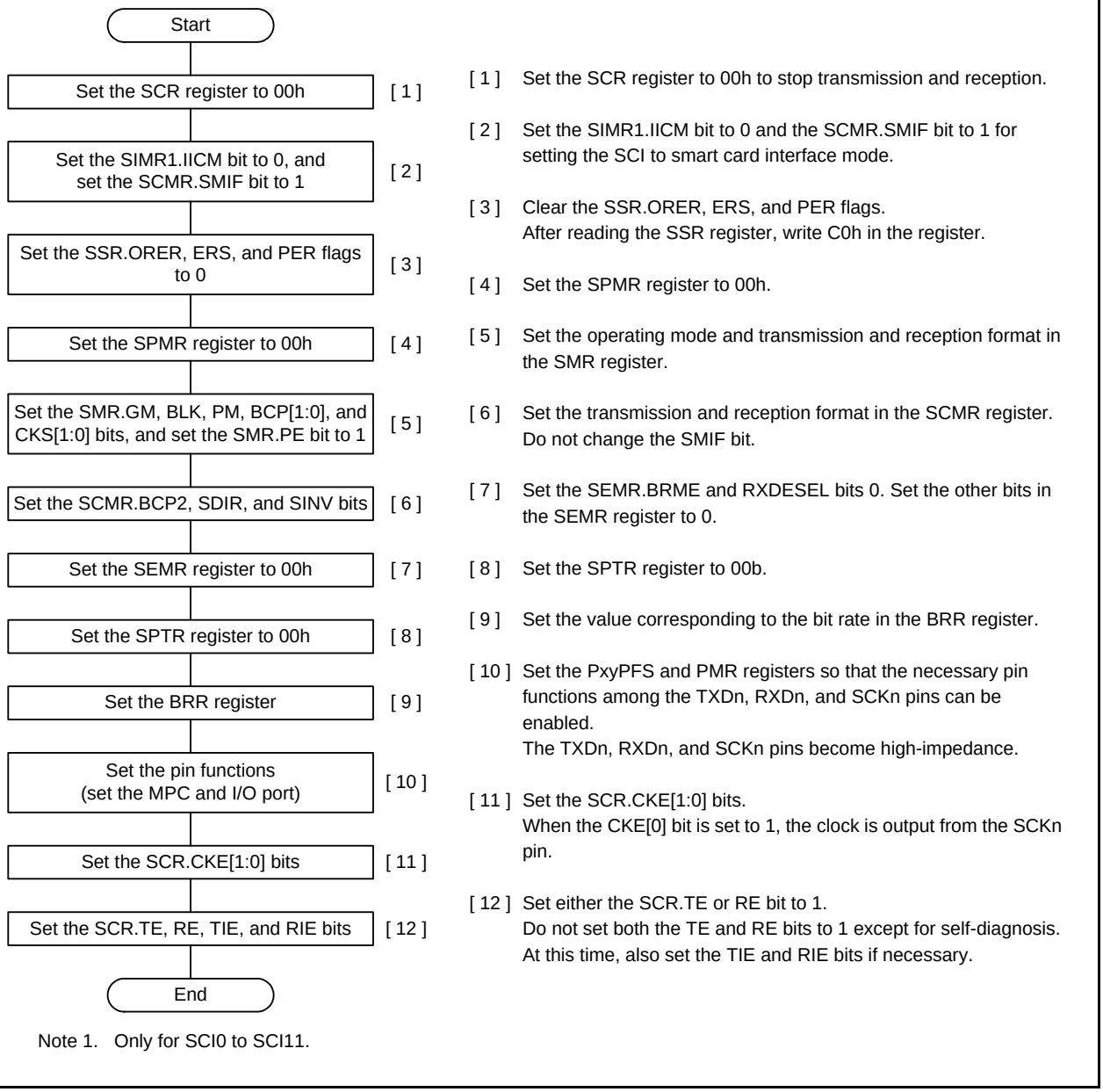


Figure 30.43 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

30.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 30.44.
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 30.44. 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 30.44. Set TE = 0 and RE = 1 at step [11]. Transmission completion can be verified by reading the SSR.TEND flag.



Note 1. Only for SCI0 to SCI11.

Figure 30.44 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

Figure 30.45 shows an example of data transmission when the SCI is set to smart card interface mode according to the flow described in Figure 30.44 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.

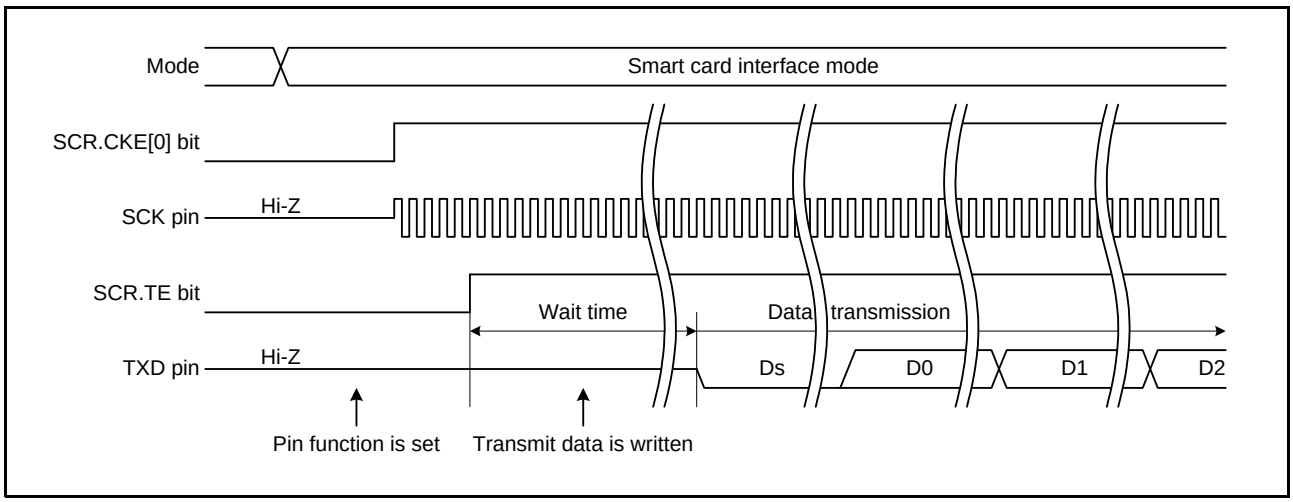


Figure 30.45 Example of Data Transmission Timing in Smart Card Interface Mode

30.6.6 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 30.46 shows the data retransmit operation during transmission.

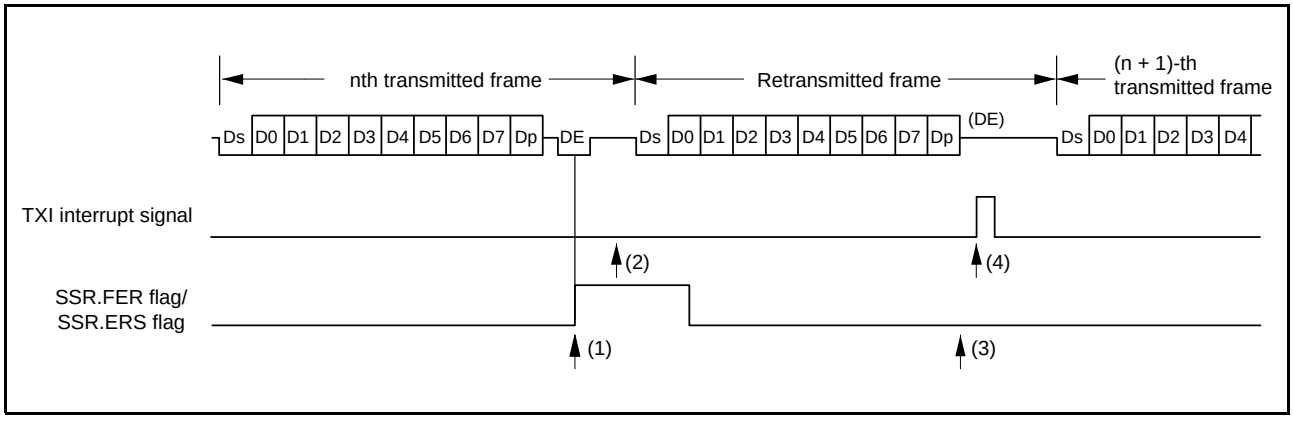


Figure 30.46 Data Retransmit Operation in SCI Transmit Mode

(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 30.47 shows a sample flowchart of serial transmission.

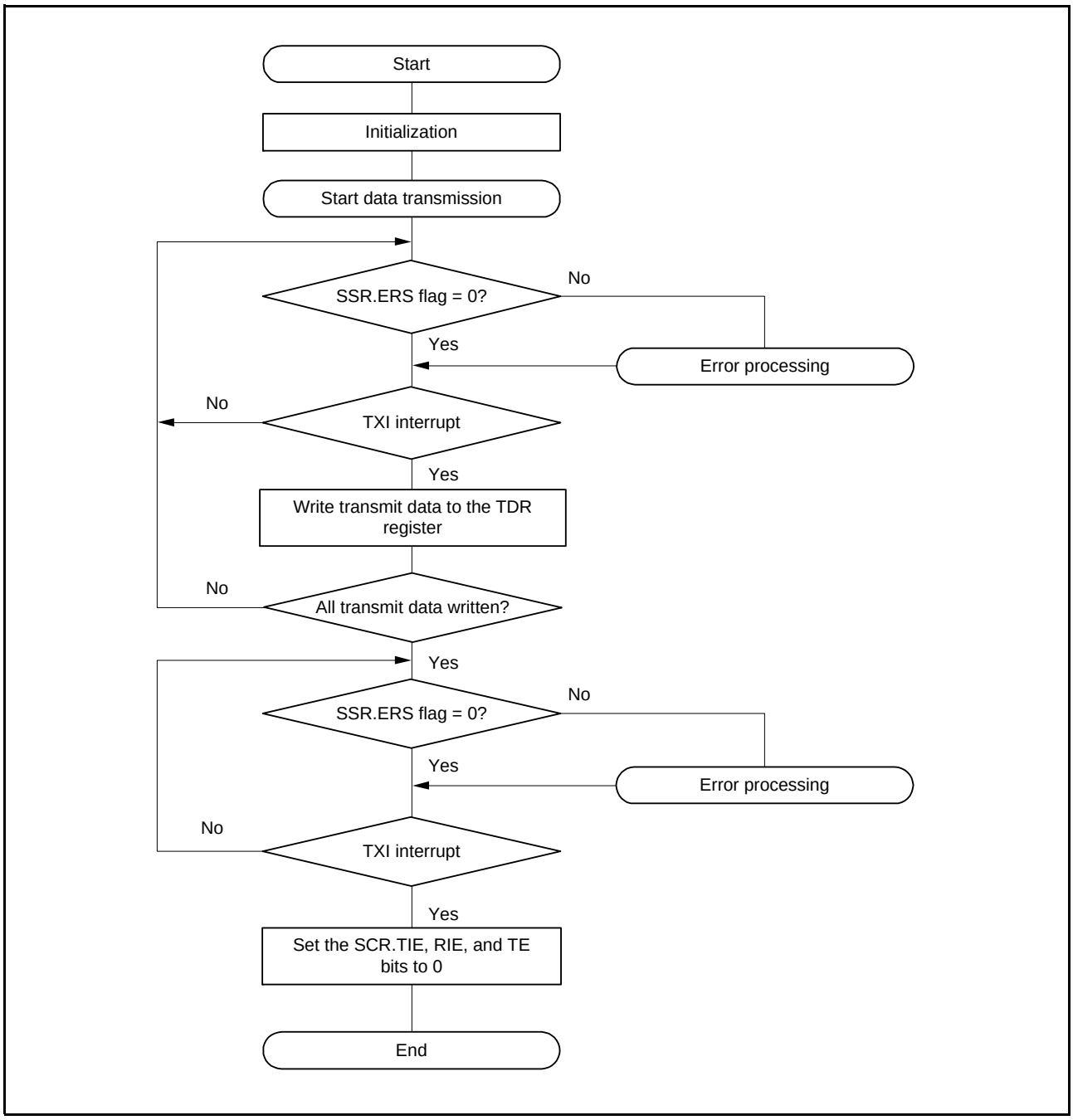


Figure 30.47 Sample Smart Card Interface Transmission Flowchart

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 18, Data Transfer Controller (DTCb), section 17, DMA Controller (DMACAa).

Note that the SSR.TEND flag is set in different timings depending on the SMR.GM bit setting. Figure 30.48 shows the TEND flag generation timing.

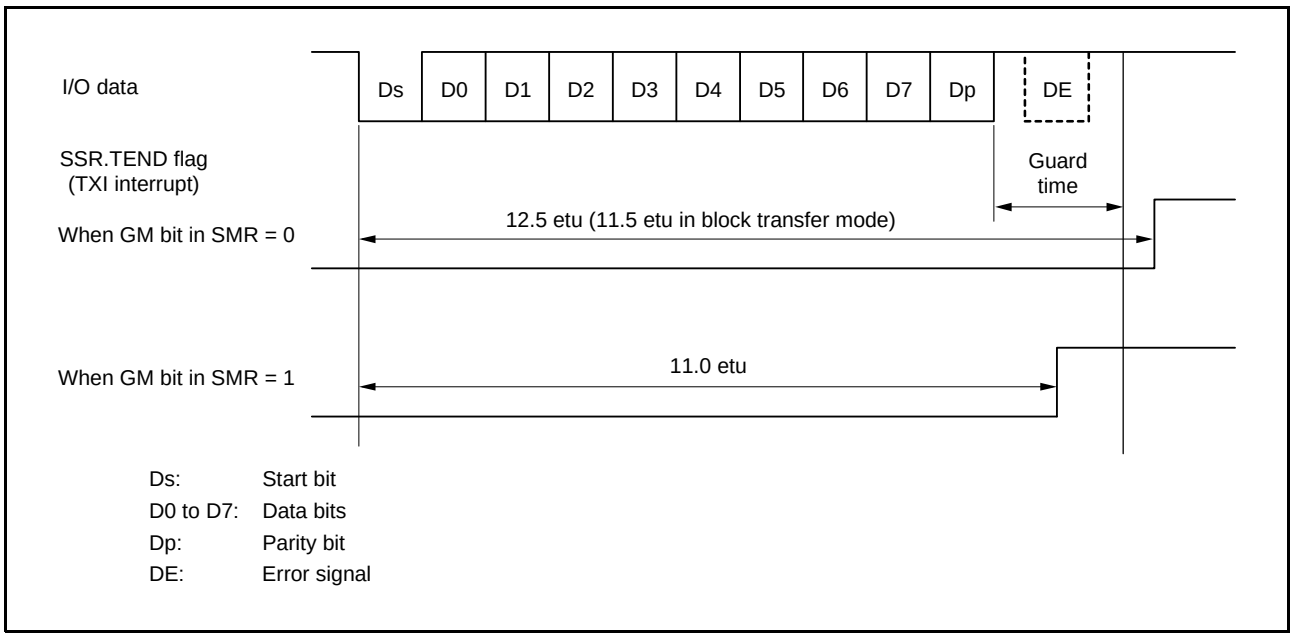


Figure 30.48 SSR.TEND Flag Generation Timing during Transmission

30.6.7 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 30.49 shows the data retransmit operation in receive mode.

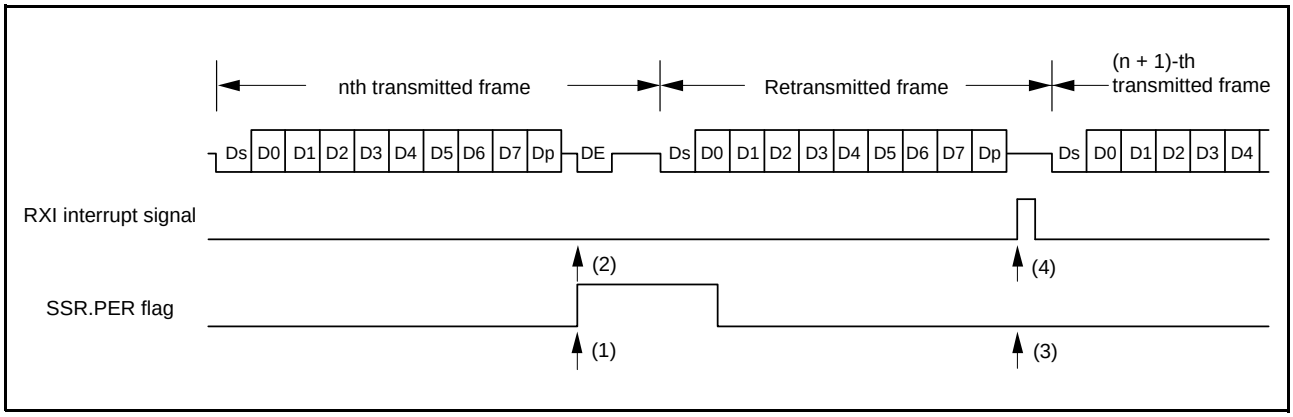


Figure 30.49 Data Retransmit Operation in SCI Receive Mode (Data Retransmit Operation during Reception)

(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 30.50 shows a sample flowchart for serial data reception.

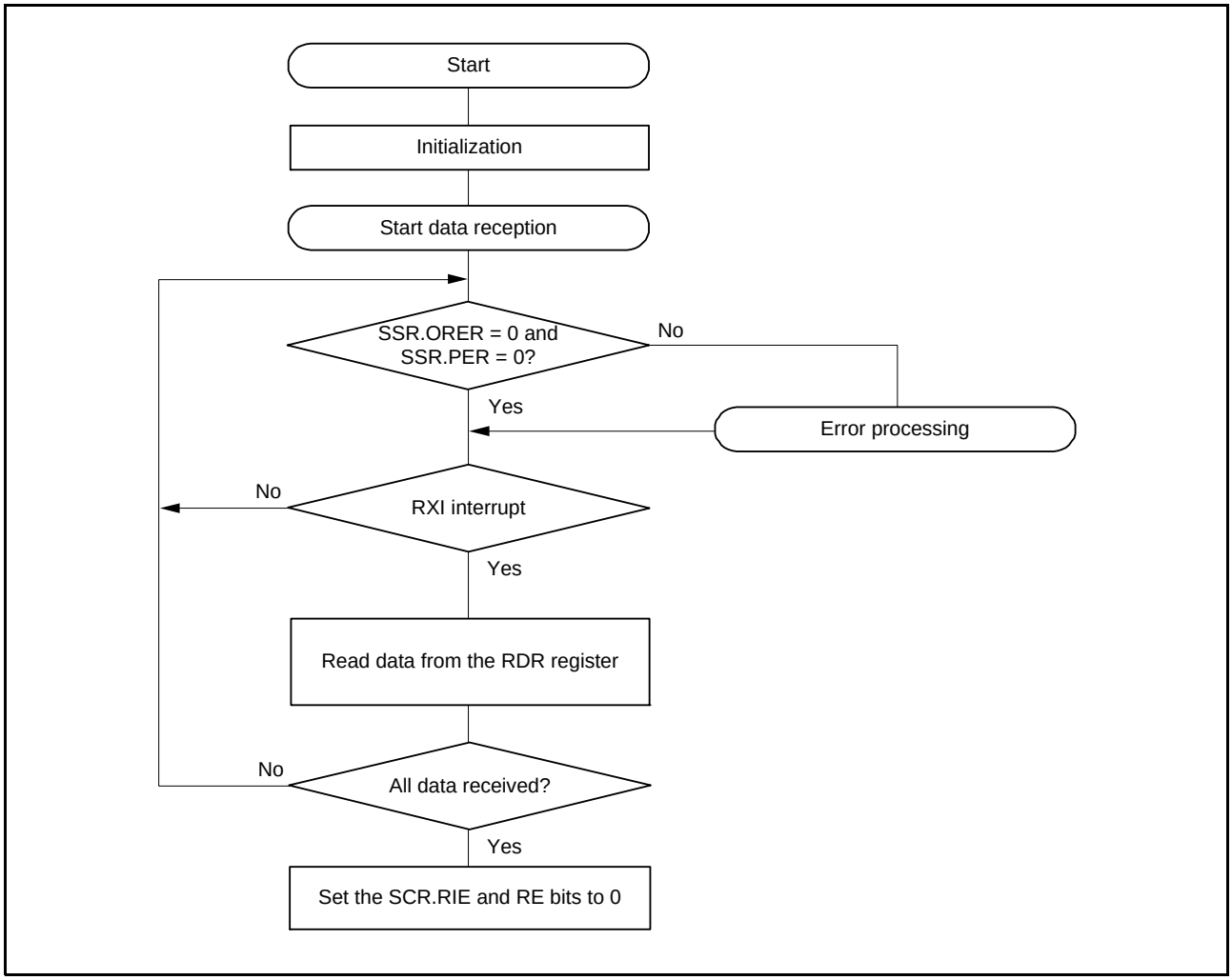


Figure 30.50 Sample Smart Card Interface Reception Flowchart

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 30.3, Operation in Asynchronous Mode.

30.6.8 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 30.2.13, 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 30.51 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.

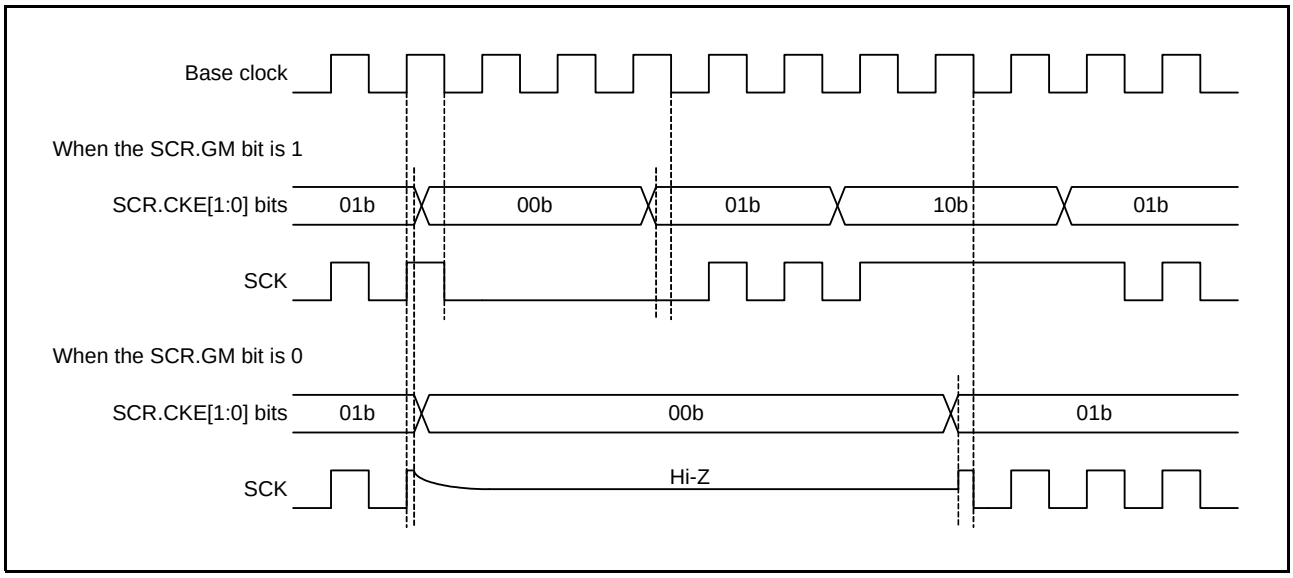


Figure 30.51 Clock Output Control

30.7 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 30.52 and Figure 30.53.

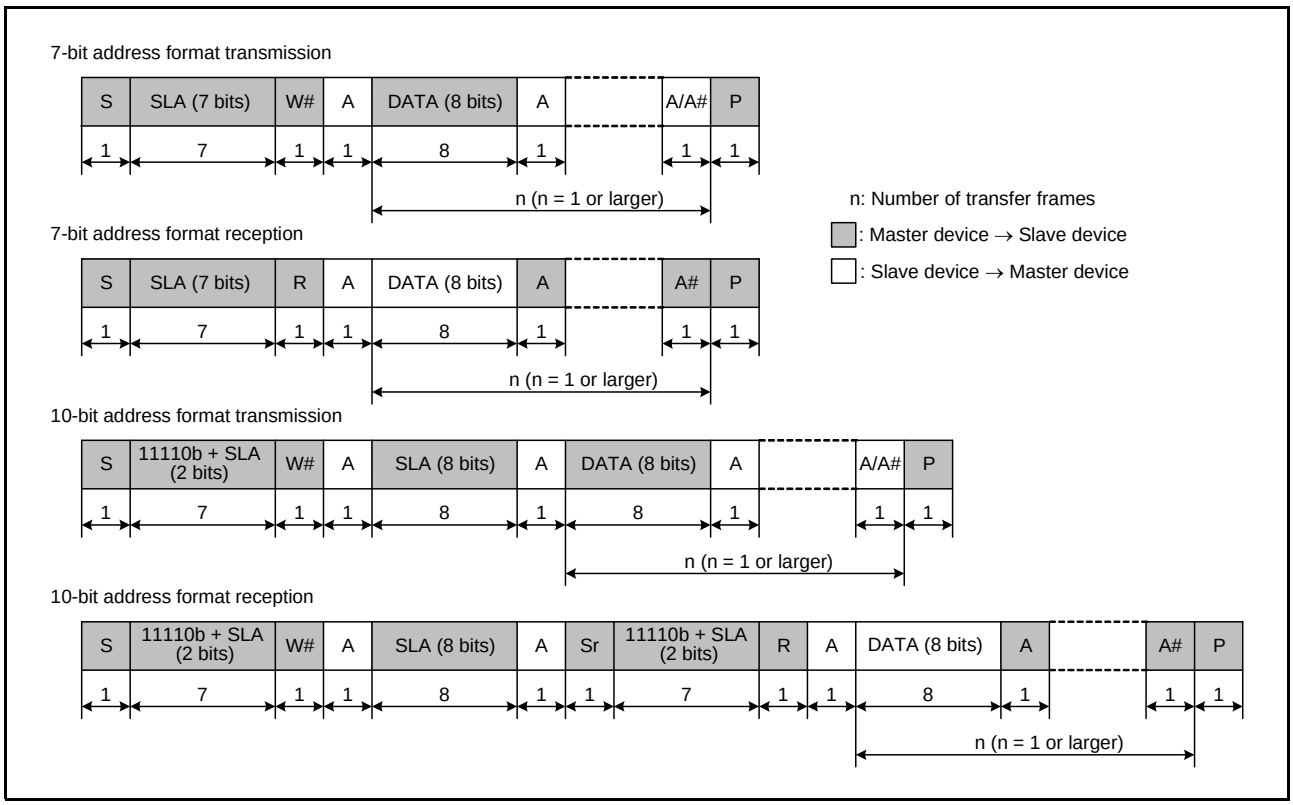


Figure 30.52 I²C-bus Format

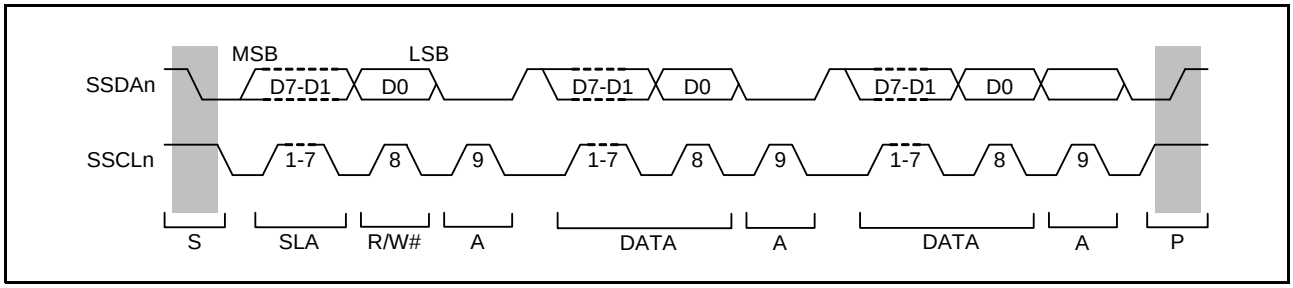


Figure 30.53 I²C-bus Timing (When SLA is 7 Bits)

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.

30.7.1 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 30.54 shows the timing of operations in the generation of start, restart, and stop conditions.

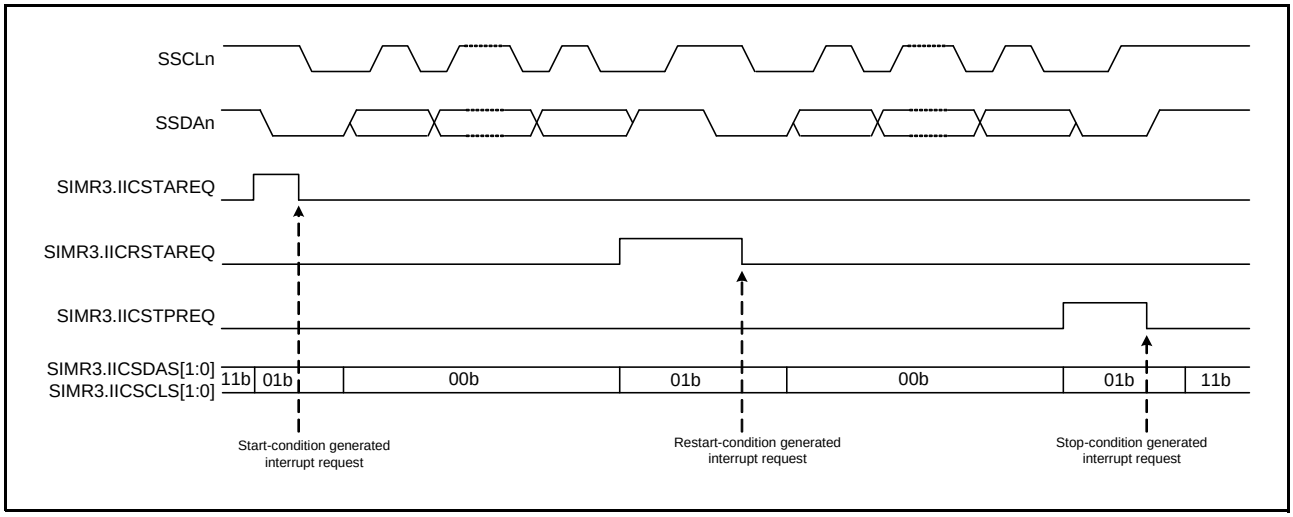


Figure 30.54 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

30.7.2 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 30.55 shows an example of operations to synchronize the clocks.

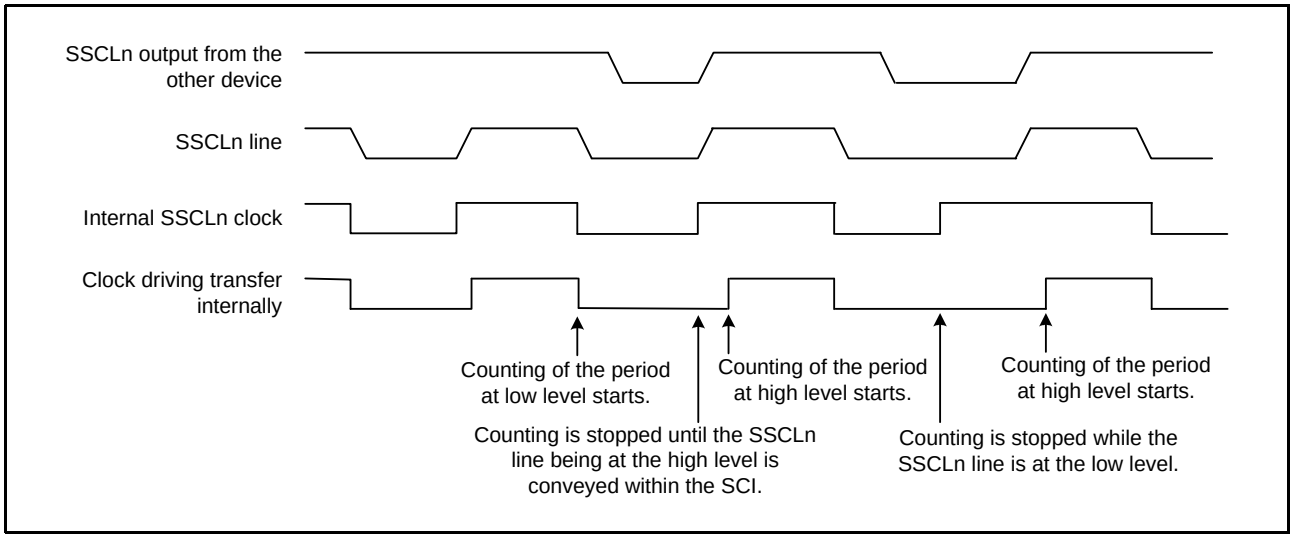


Figure 30.55 Example of Operations for Clock Synchronization

30.7.3 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 30.56 shows the timing of delays in SSDA output.

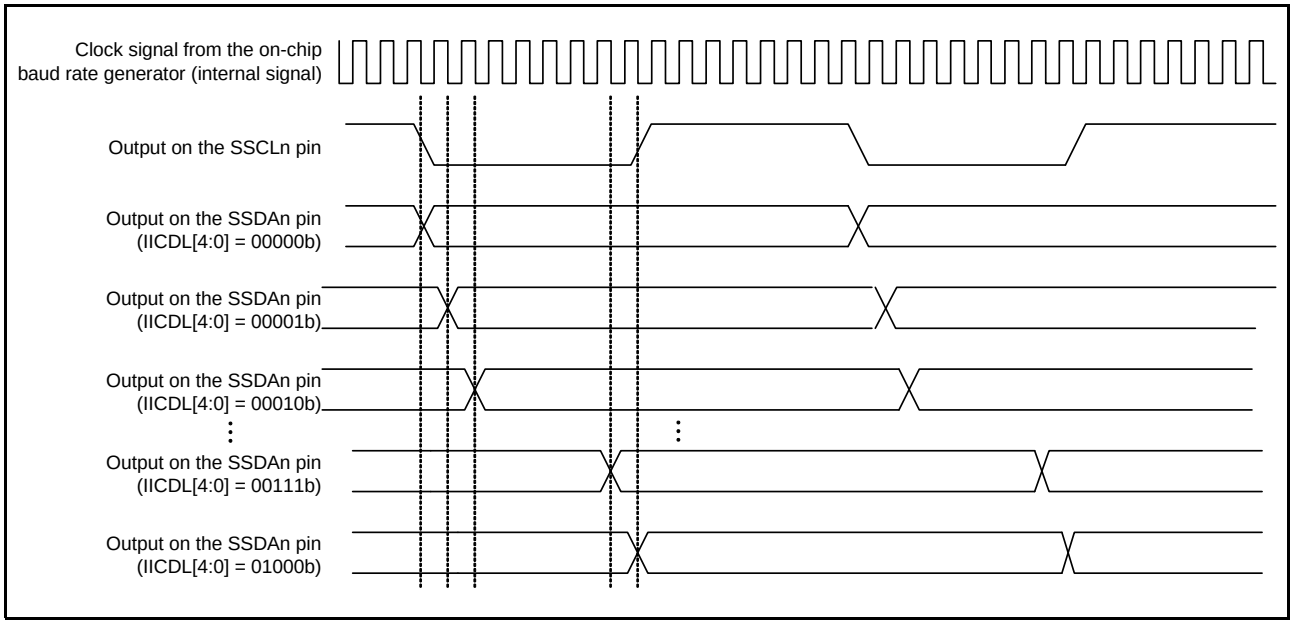
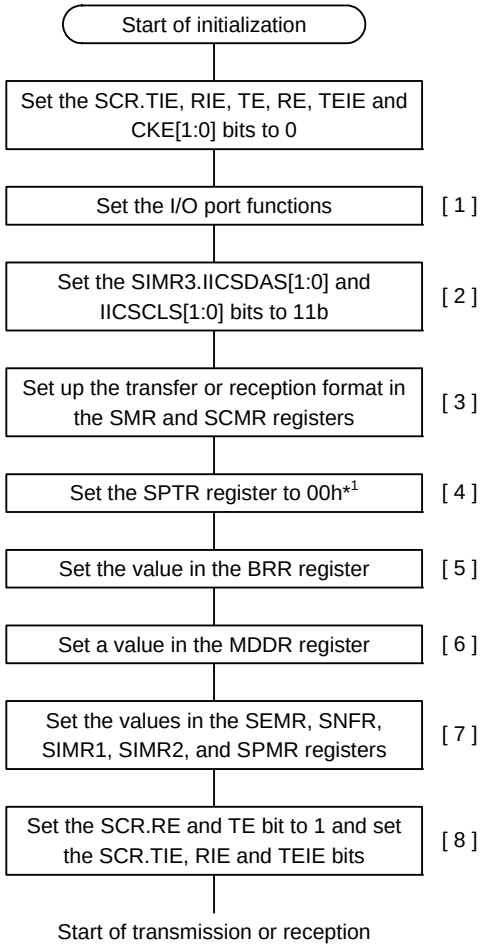


Figure 30.56 Timing of Delays in SSDA Output

30.7.4 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 30.57.
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.

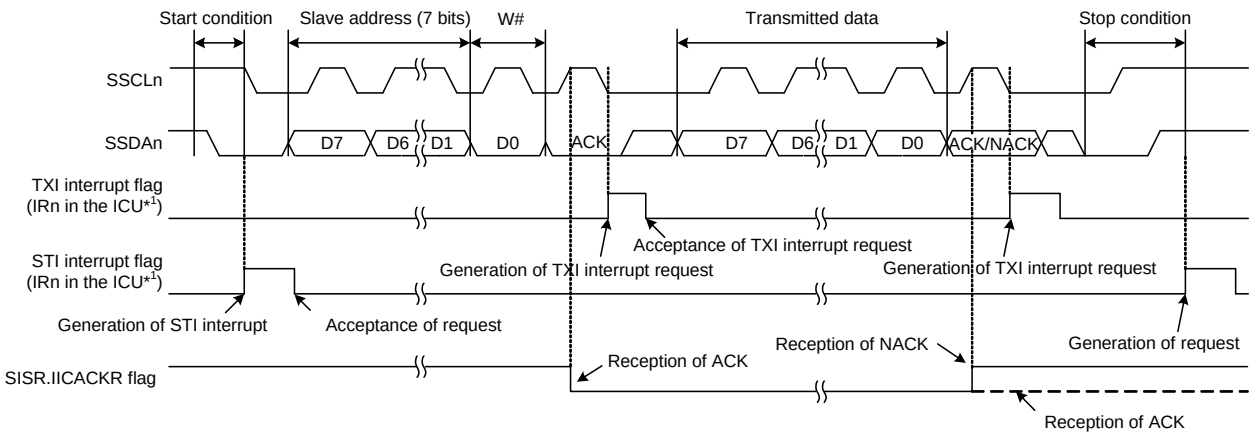
[8] Set the SCR.RE and TE bits to 1. At this time, also set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0).
Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.
Note 1. Only for SCI0 to SCI11.

Figure 30.57 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

30.7.5 Operation in Master Transmission (Simple I2C Mode)

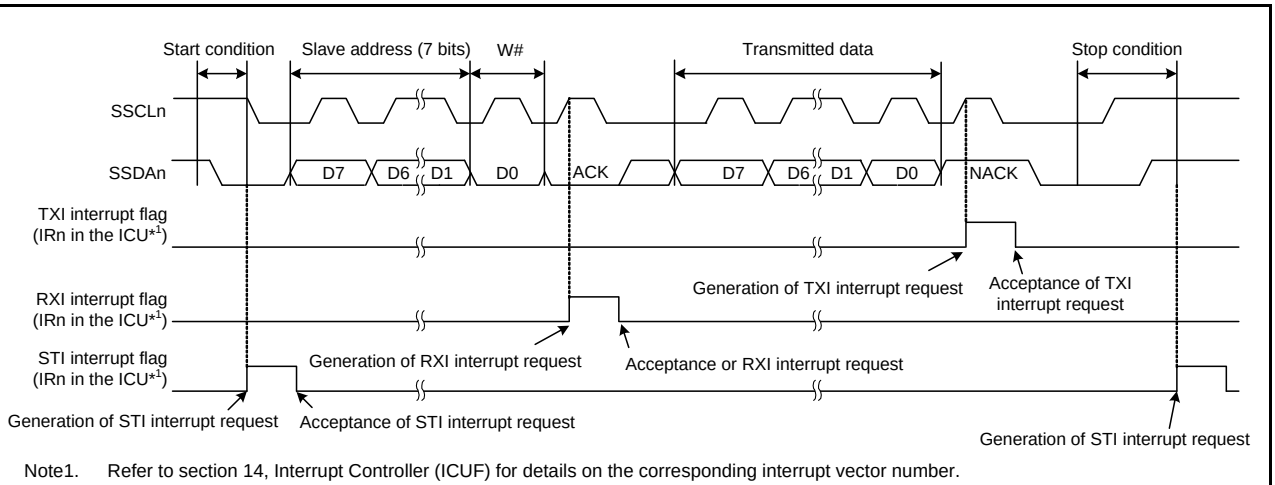
Figure 30.58 and Figure 30.59 show examples of operations in master transmission and Figure 30.60 is a flowchart showing the procedure for data transmission. Refer to Table 30.41 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 30.60 are repeated 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

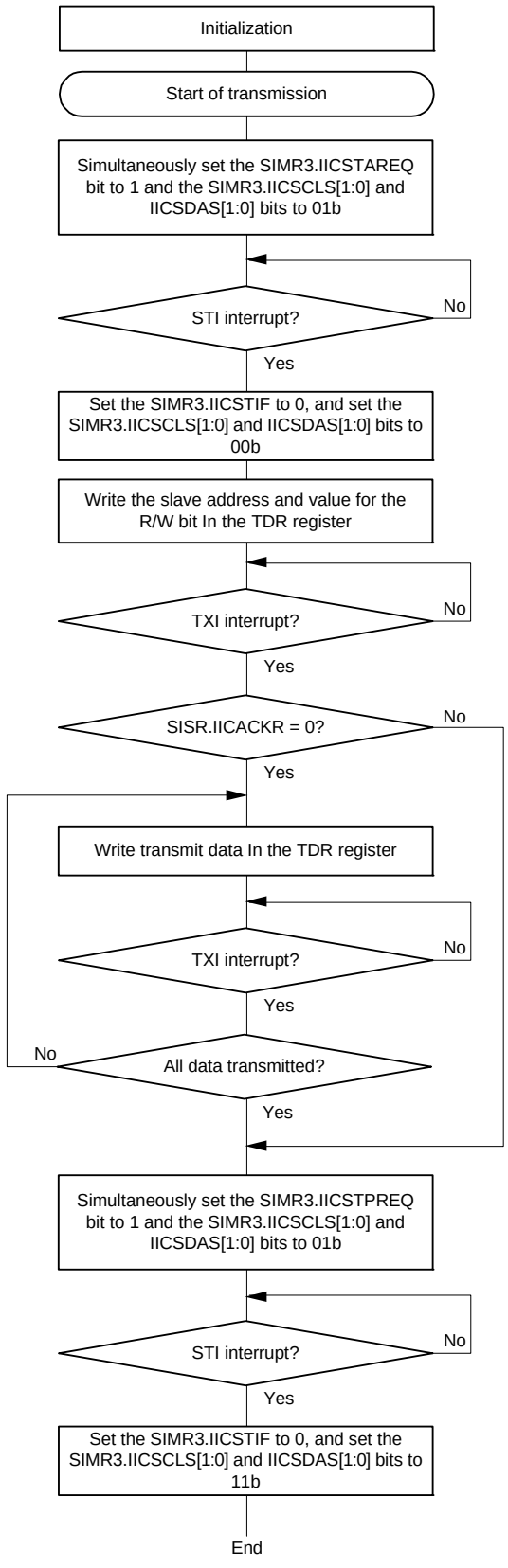
Figure 30.58 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 30.59 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.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

[5]

[6]

Figure 30.60 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

30.7.6 Master Reception (Simple I2C Mode)

Figure 30.61 shows an example of operations in simple I2C mode master reception and Figure 30.62 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.

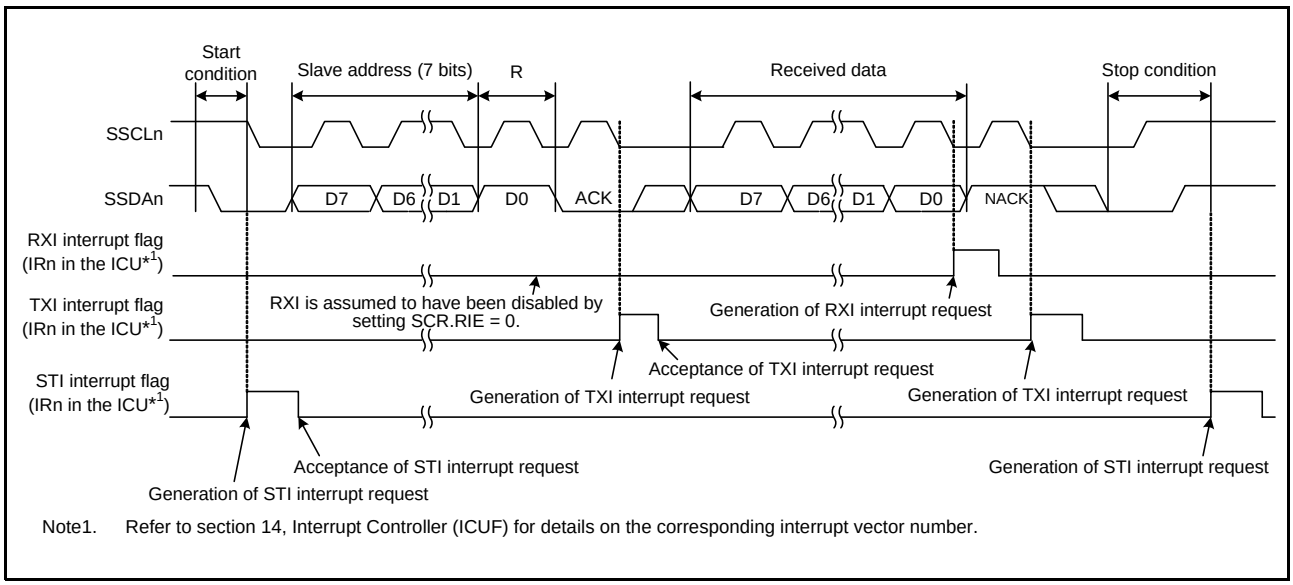
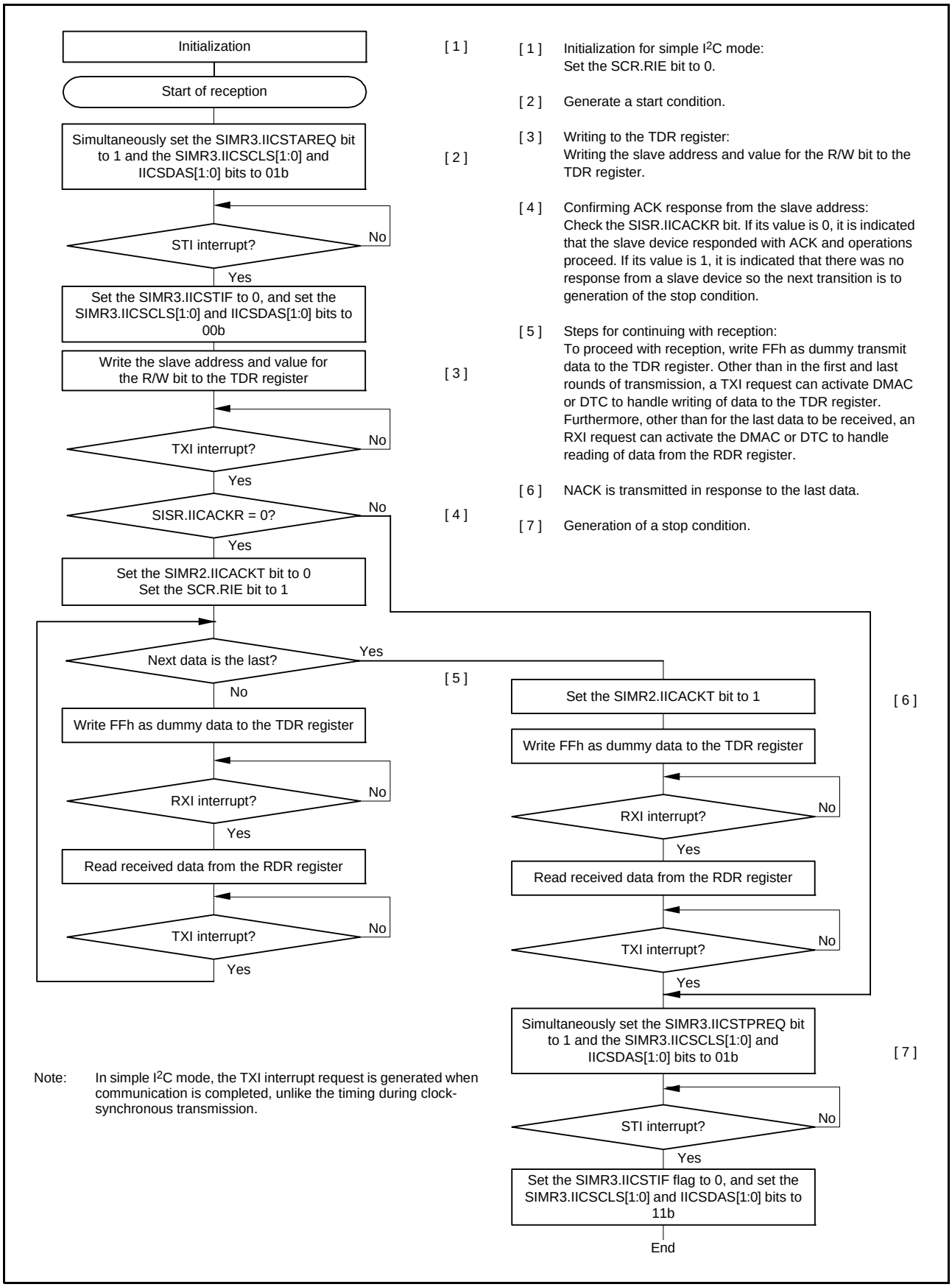


Figure 30.61 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Set the SCR.RIE bit to 0.

[2] Generate a start condition.

[3] Writing to the TDR register:
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 reception:
To proceed with reception, write FFh as dummy transmit data to the TDR register. Other than in the first and last rounds of transmission, a TXI request can activate DMAC or DTC to handle writing of data to the TDR register. Furthermore, other than for the last data to be received, an RXI request can activate the DMAC or DTC to handle reading of data from the RDR register.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

Figure 30.62 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

30.7.7 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.

30.8 Operation in Simple SPI Mode

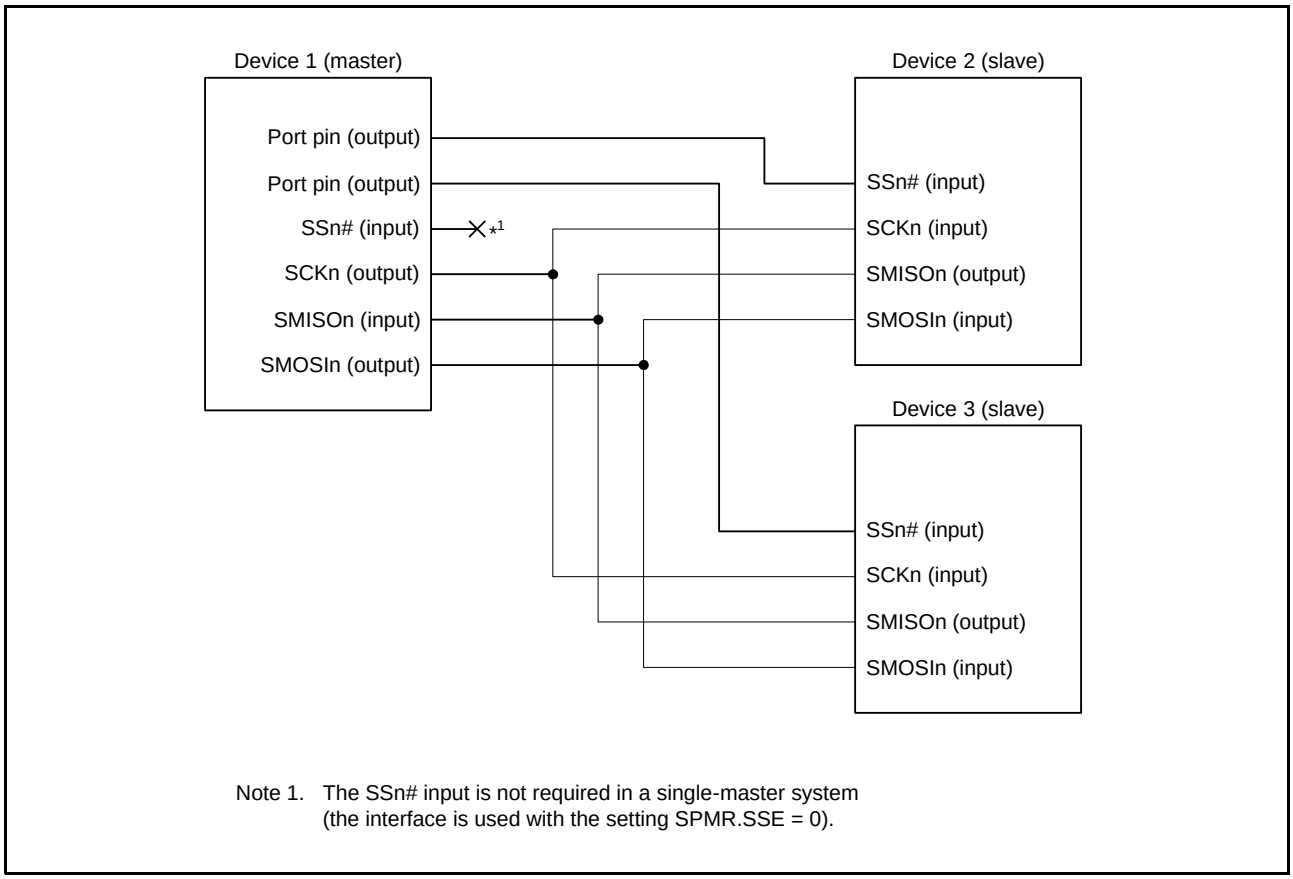
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

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 30.63 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 30.63 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

30.8.1 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 30.36 lists the states of pins according to the mode and the level on the SSn# pin.

Table 30.36 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).

30.8.2 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.

30.8.3 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.

30.8.4 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 30.64. The relation is the same for both master and slave operation.

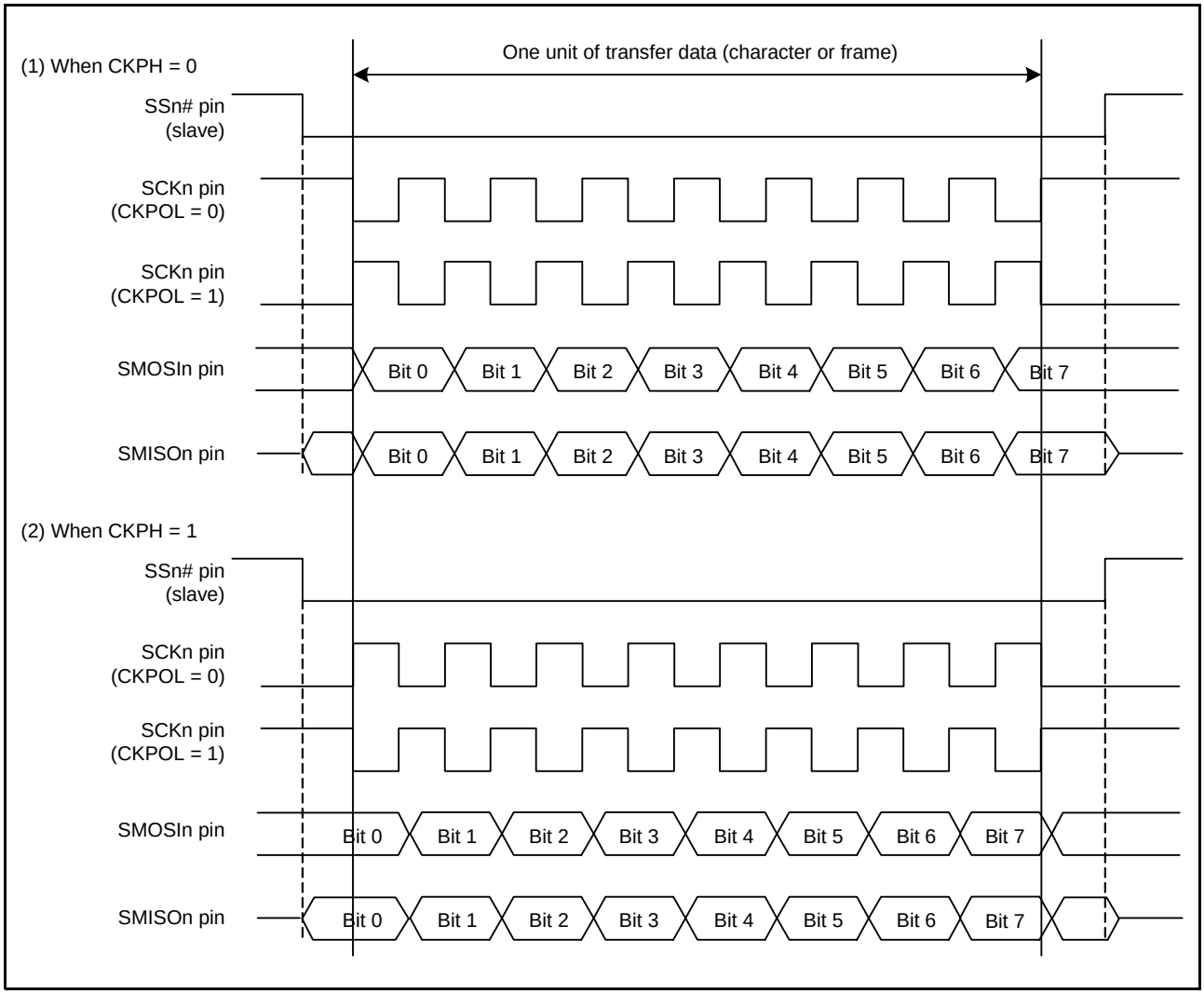


Figure 30.64 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

30.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 30.30, 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.

30.8.6 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.

30.9 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 30.65 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.

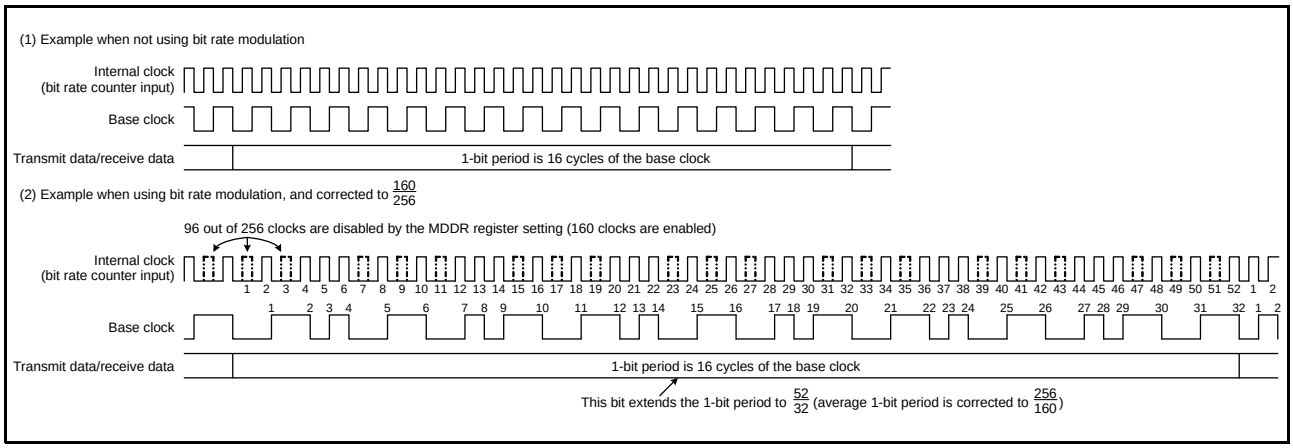


Figure 30.65 Example of the Base Clock When the Bit Rate Modulation Function is Used

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.

30.10 Extended Serial Mode Control Section: Description of Operation

30.10.1 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 30.66.
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.

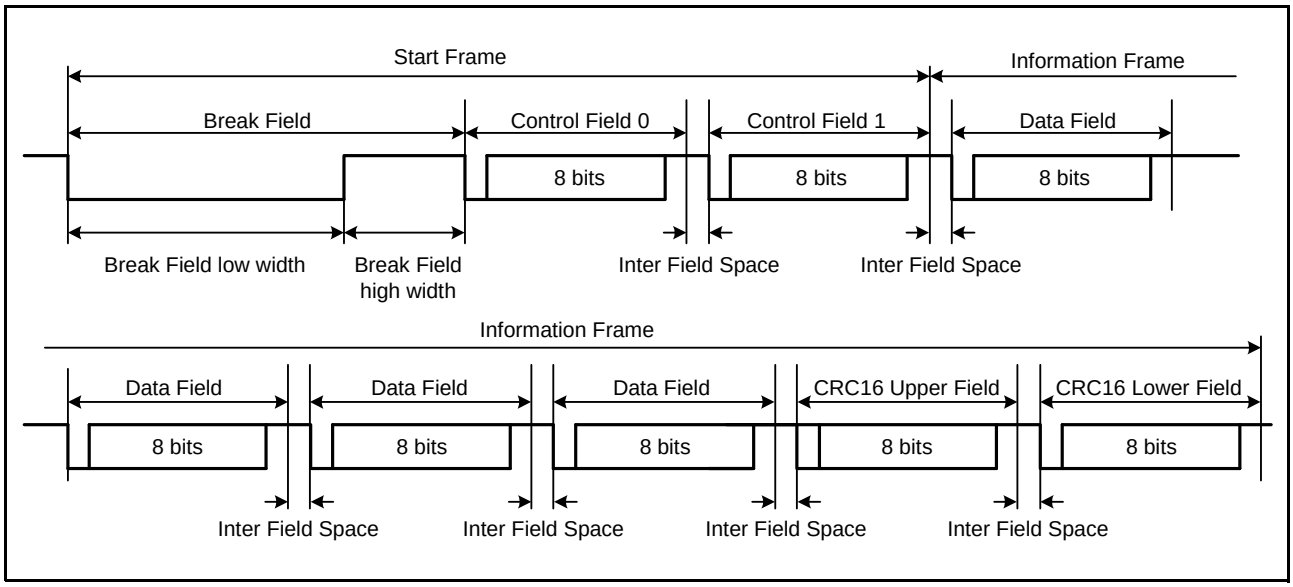


Figure 30.66 Protocol for Serial Transfer by the Extended Serial Mode Control Section

30.10.2 Transmitting a Start Frame

Figure 30.67 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 30.68 and Figure 30.69 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.

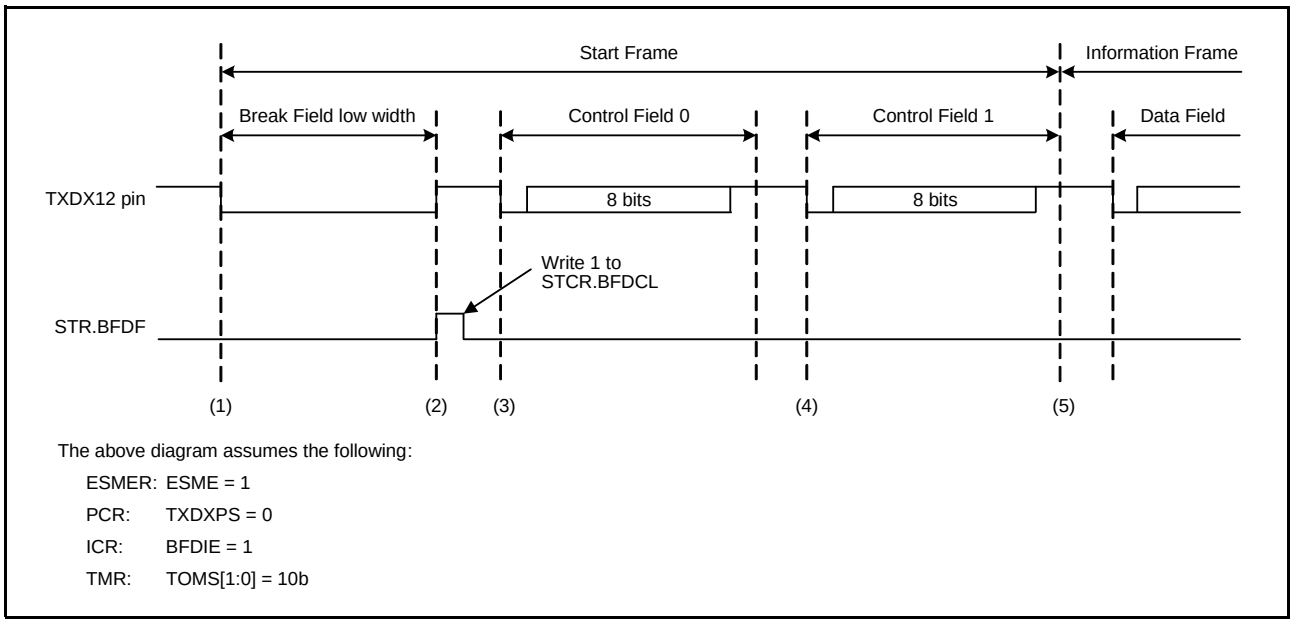


Figure 30.67 Example of Operations When Transmitting a Start Frame

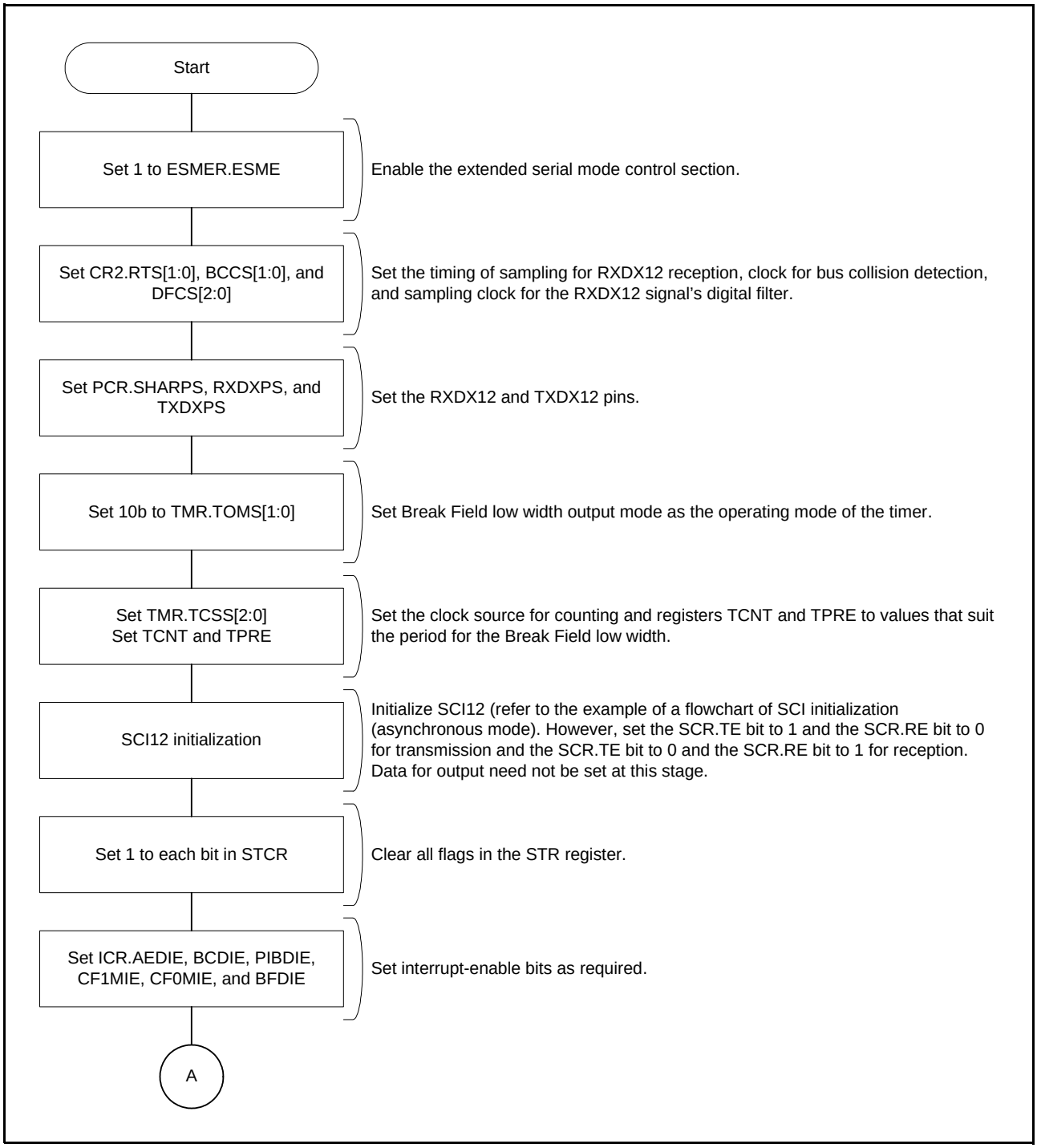


Figure 30.68 Example of Start Frame Transmission (1/2)

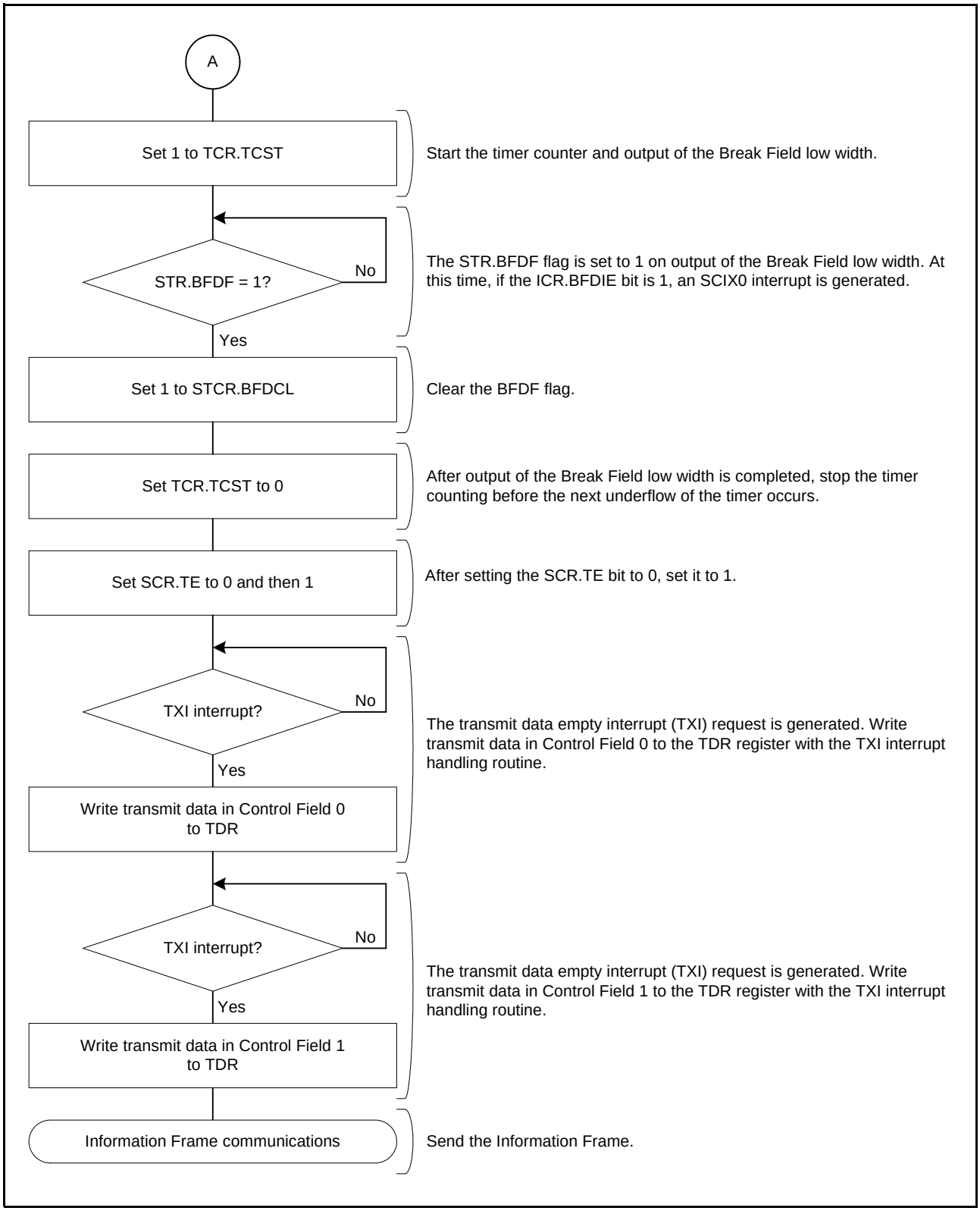


Figure 30.69 Example of Start Frame Transmission (2/2)

30.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 30.37.

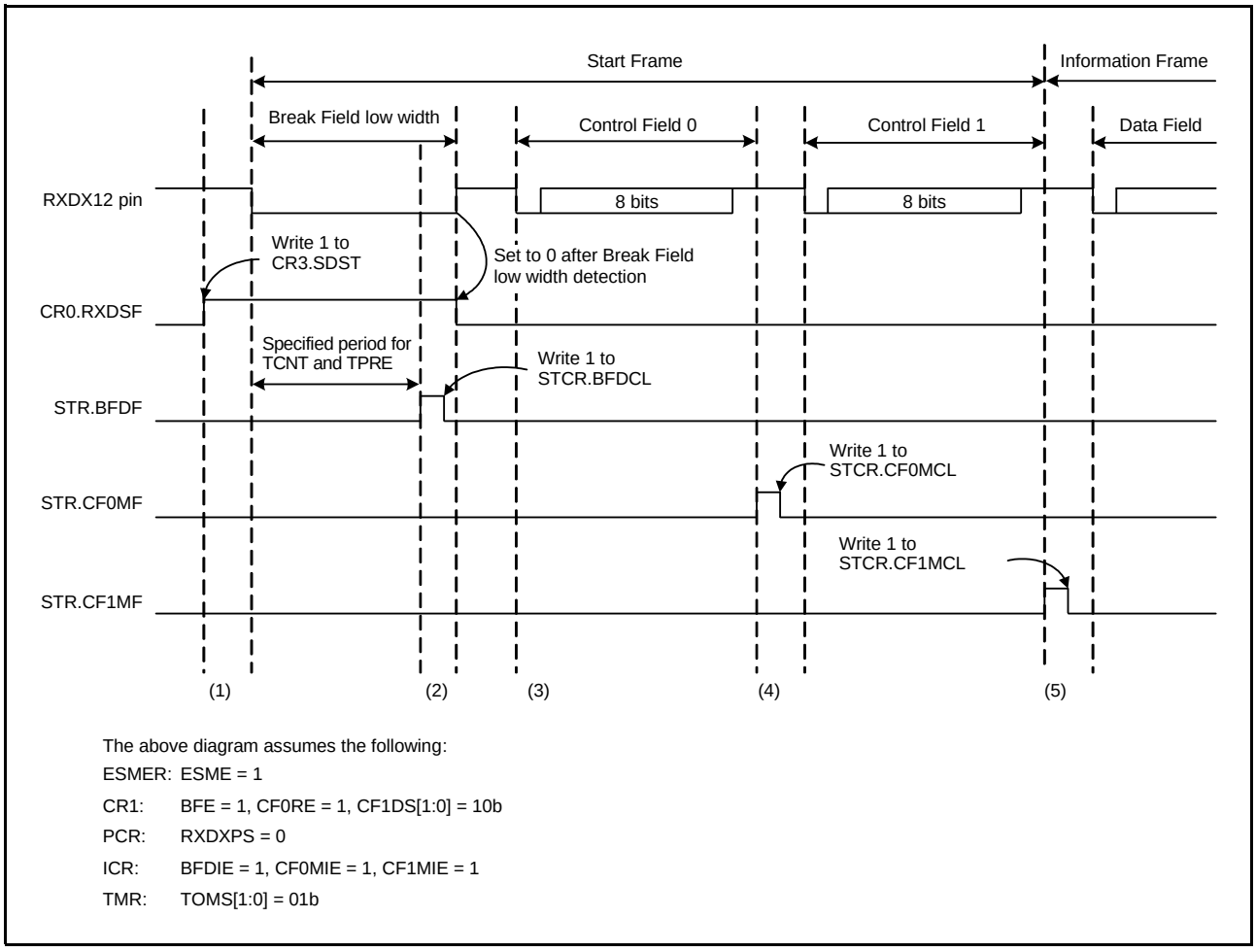
Table 30.37 Structures of Start Frames

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 30.70 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 30.71 and Figure 30.72 are flowcharts for the reception of a Start Frame, and Figure 30.73 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 30.70 Example of Operations at the Time of Start Frame Reception

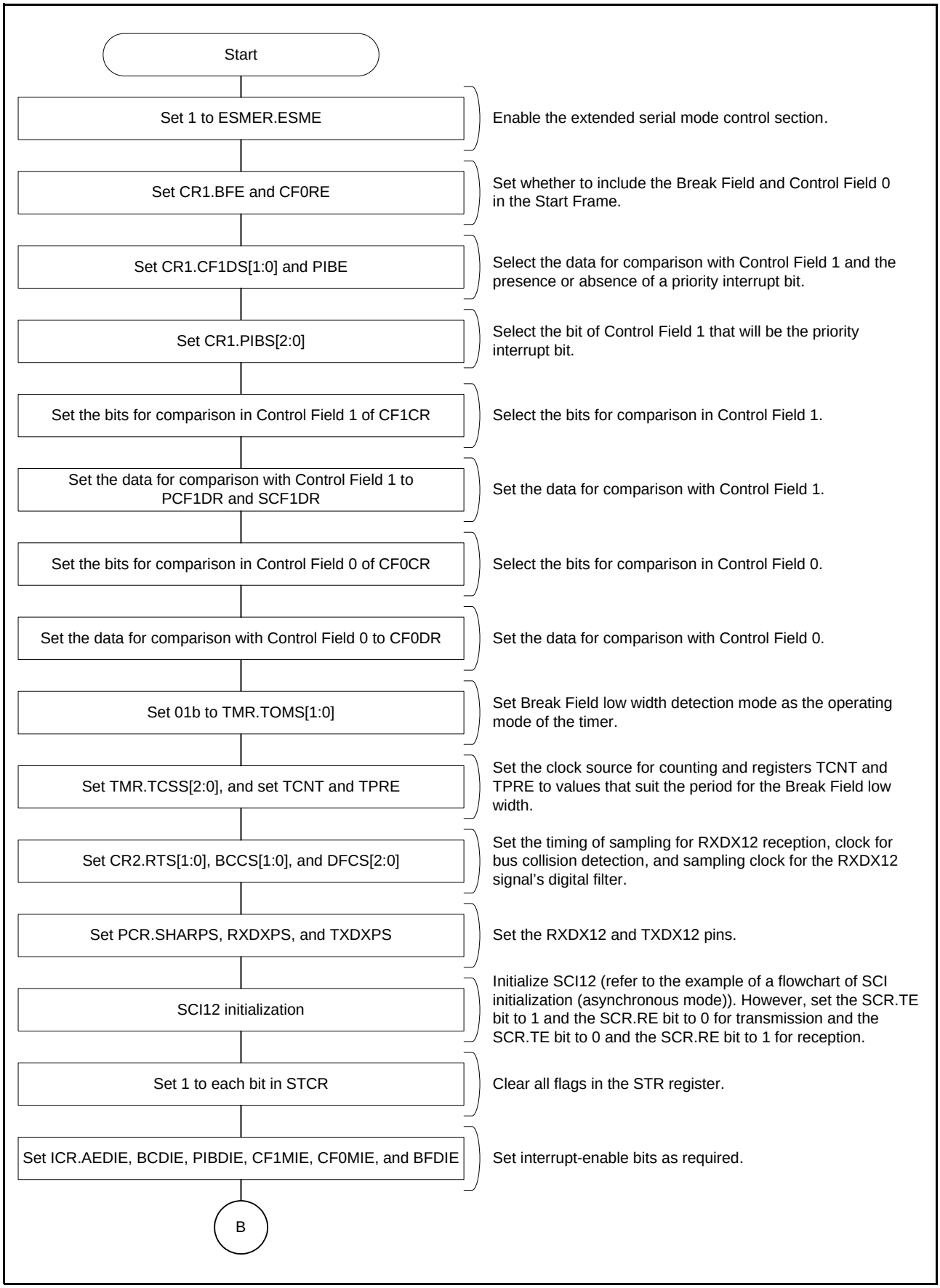


Figure 30.71 Sample Flowchart for Reception of a Start Frame (1)

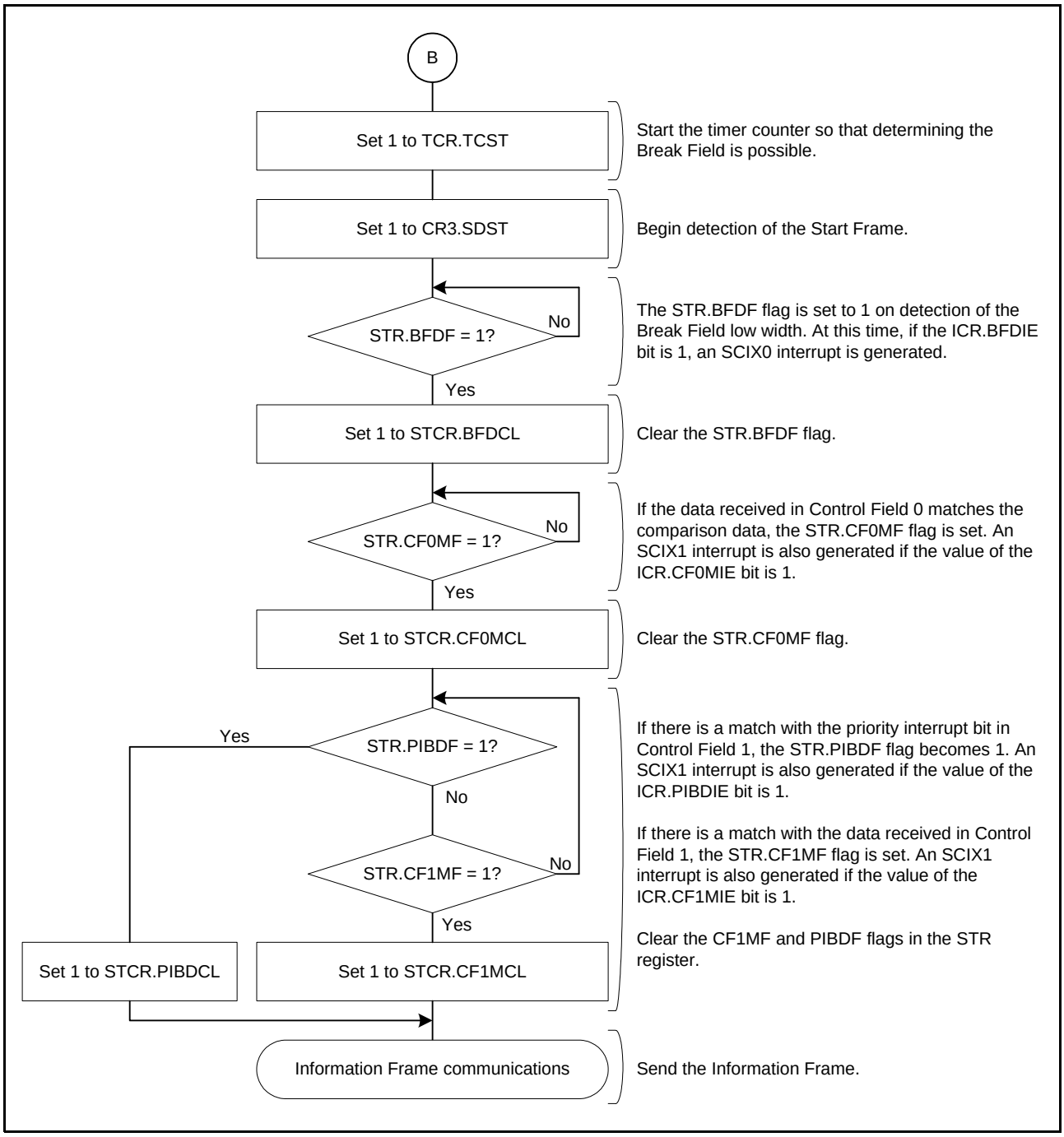


Figure 30.72 Sample Flowchart for Reception of a Start Frame (2)

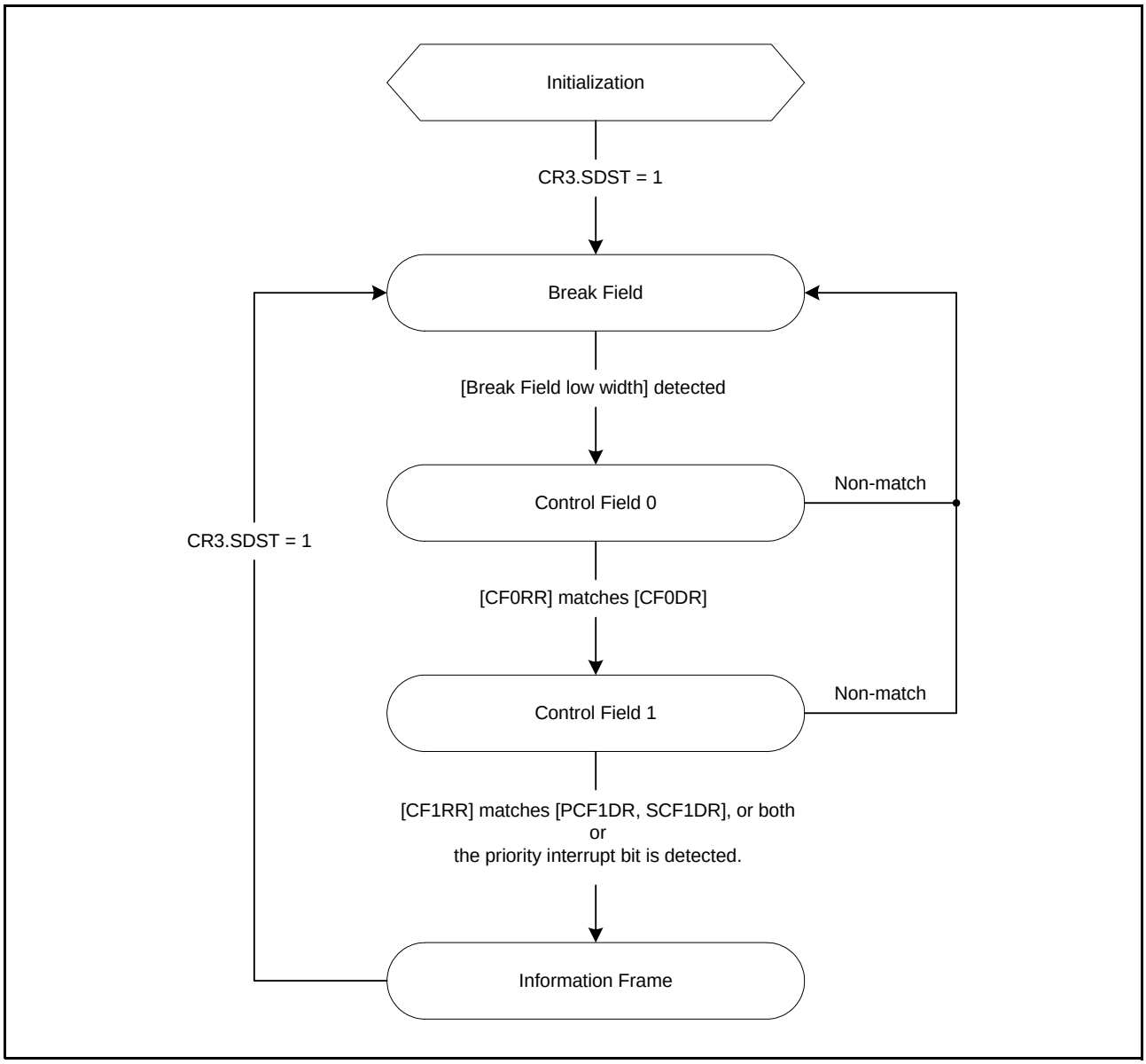


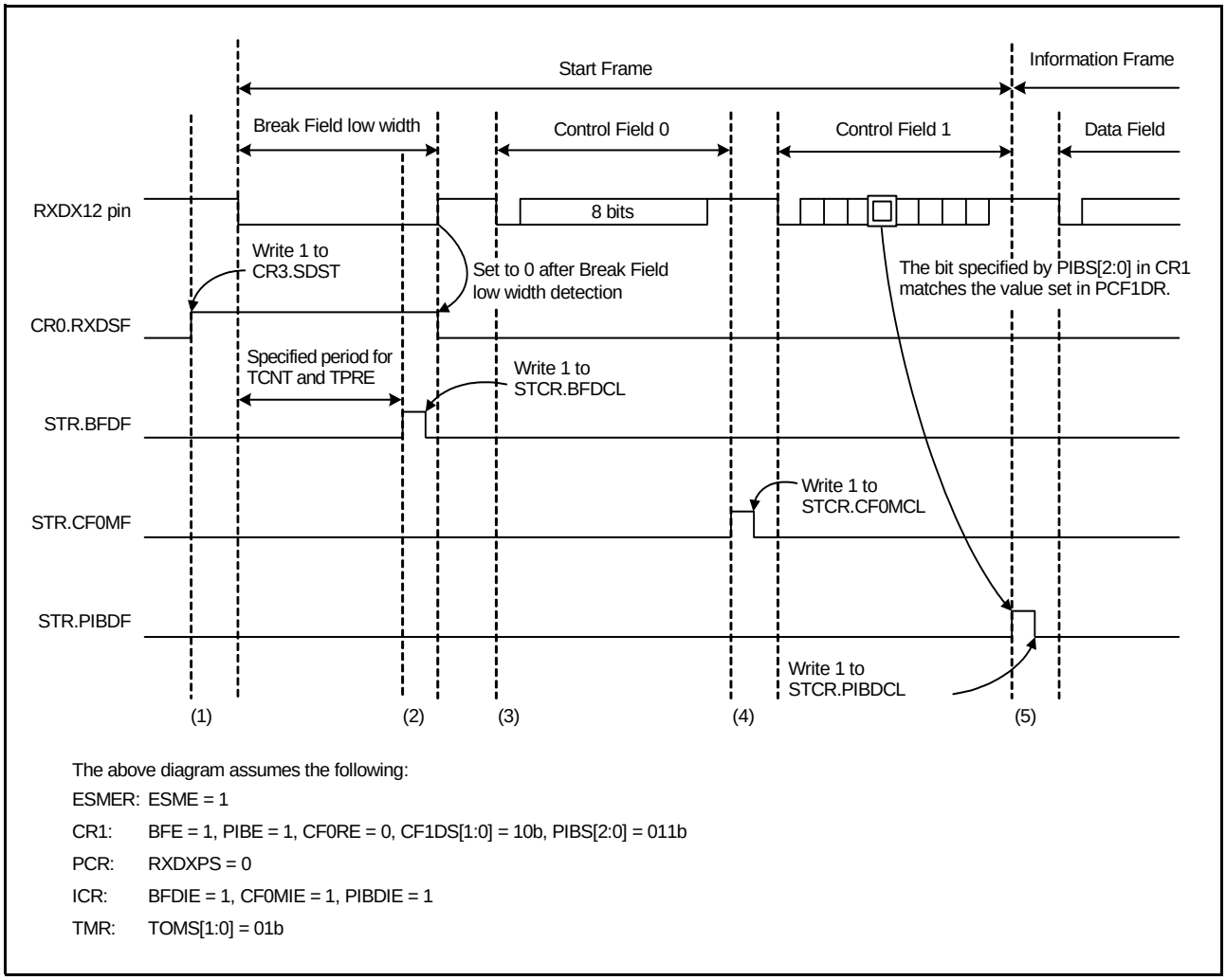
Figure 30.73 State Transitions When Receiving a Start Frame

30.10.3.1 Priority Interrupt Bit

Figure 30.74 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 30.70, 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 30.74 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

30.10.4 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 30.75 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.

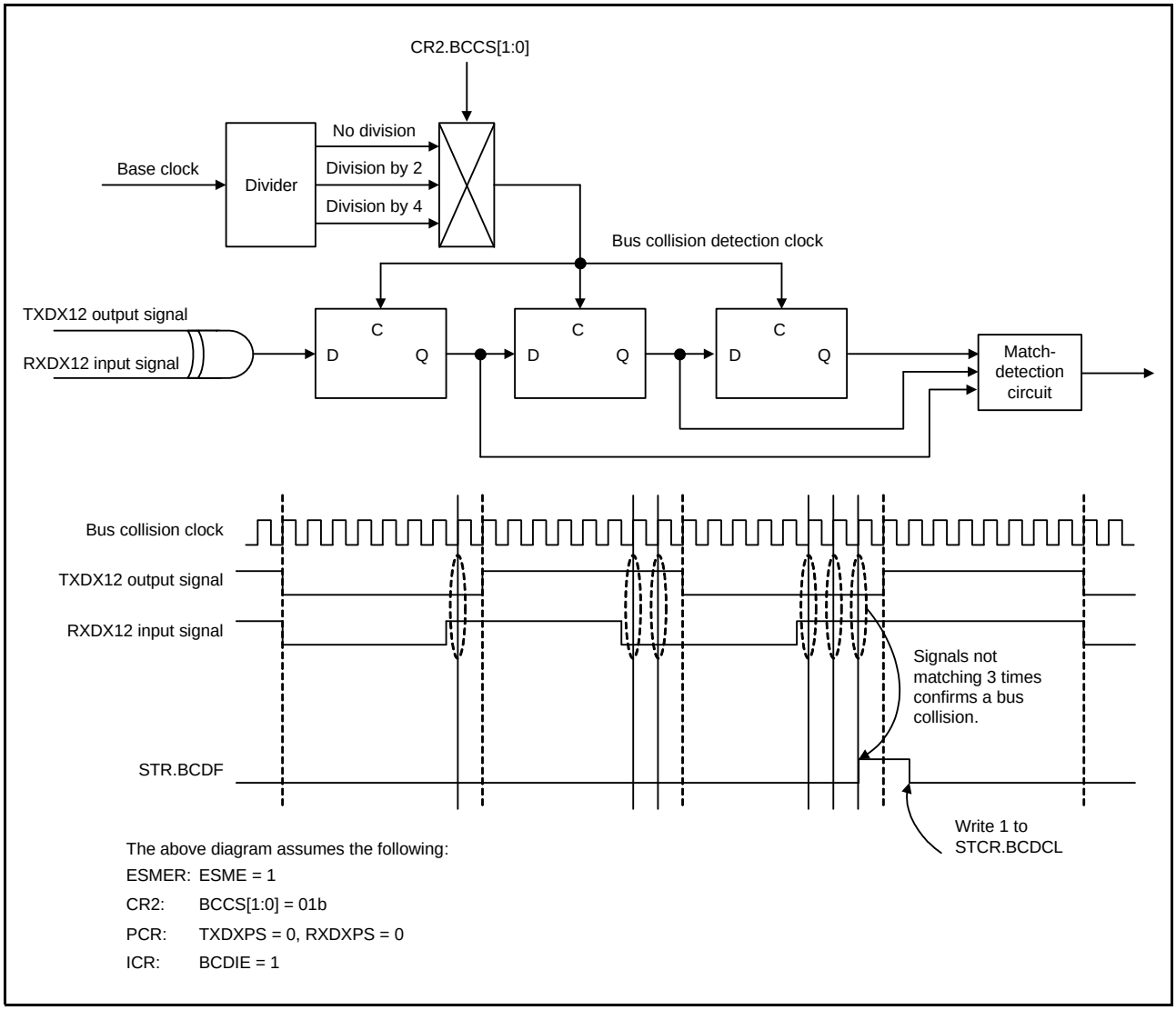


Figure 30.75 Example of Operations with Bus Collision Detection

30.10.5 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 30.76 shows an example of operations with the digital filter.

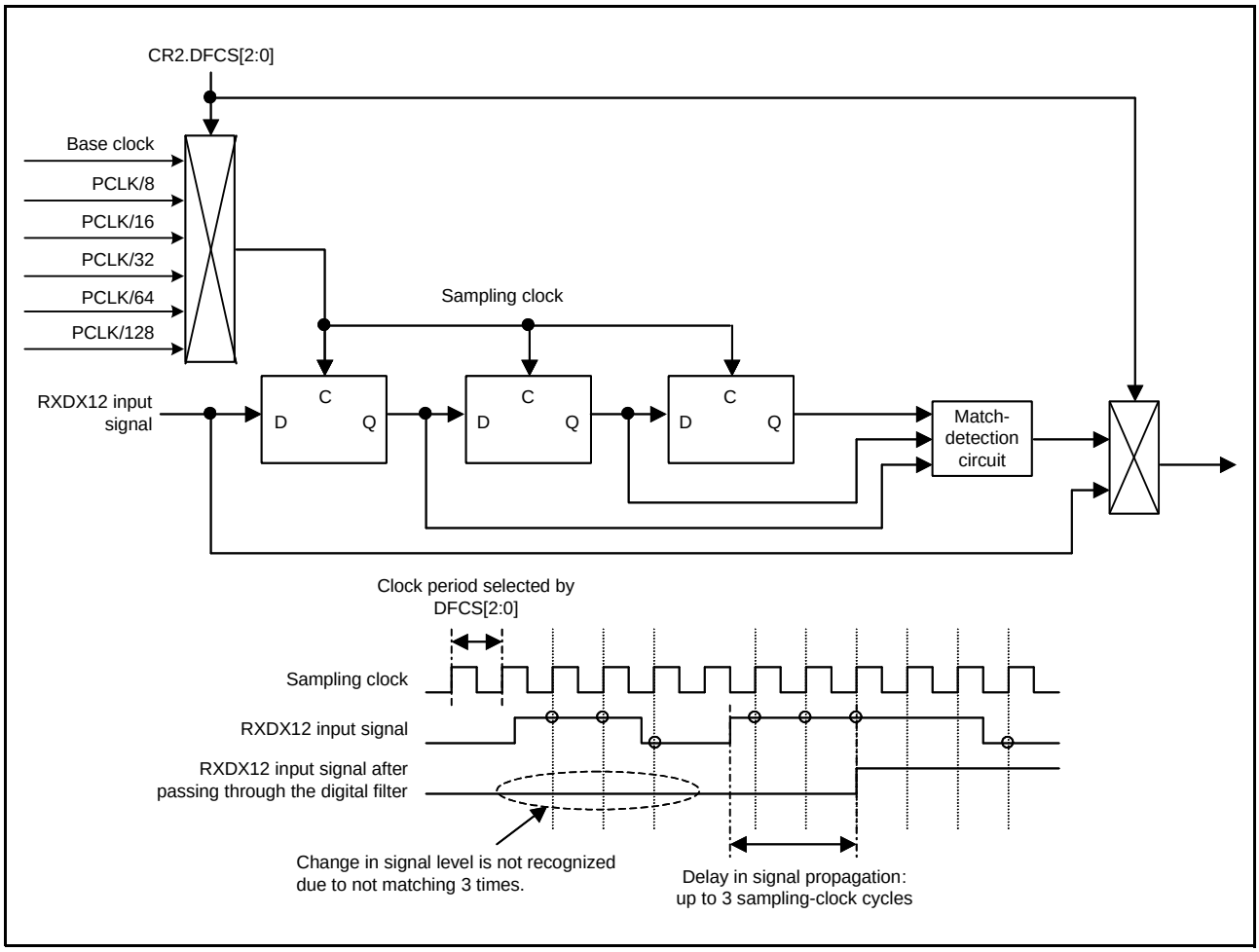
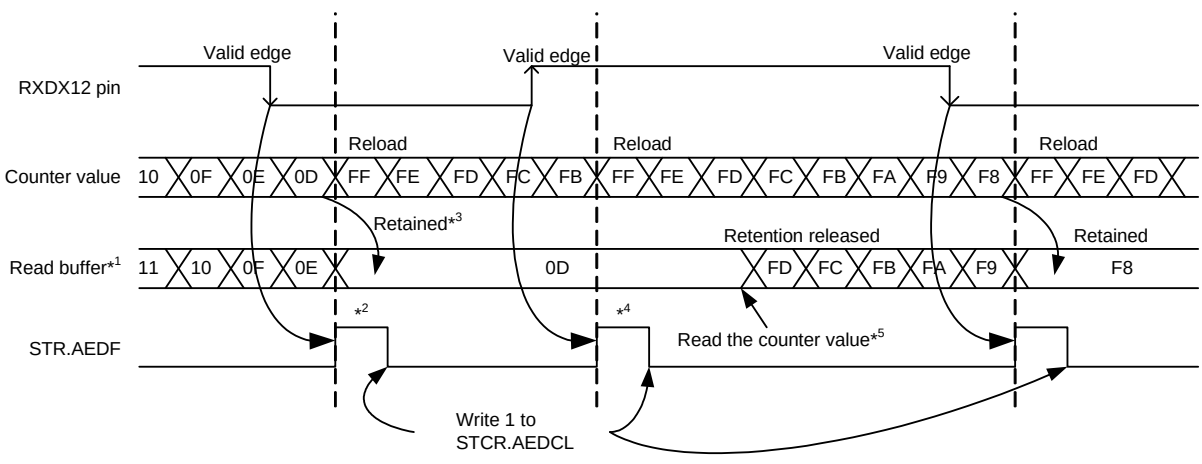


Figure 30.76 Example of Operations with the Digital Filter

30.10.6 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 30.77 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 30.77 Example of Operations for Bit Rate Measurement

30.10.7 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 30.78 shows timing for the sampling of data received through RXDX12.

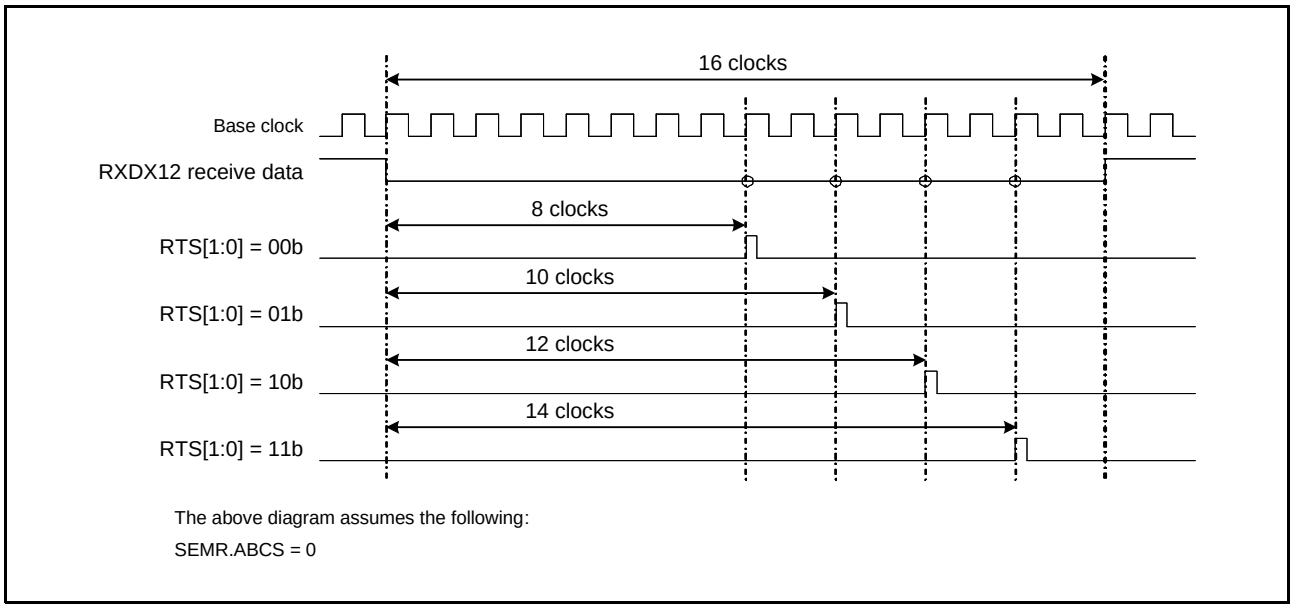


Figure 30.78 Timing for Sampling of Data Received through RXDX12

30.10.8 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 30.79 shows an example of operations in Break Field low width output mode.

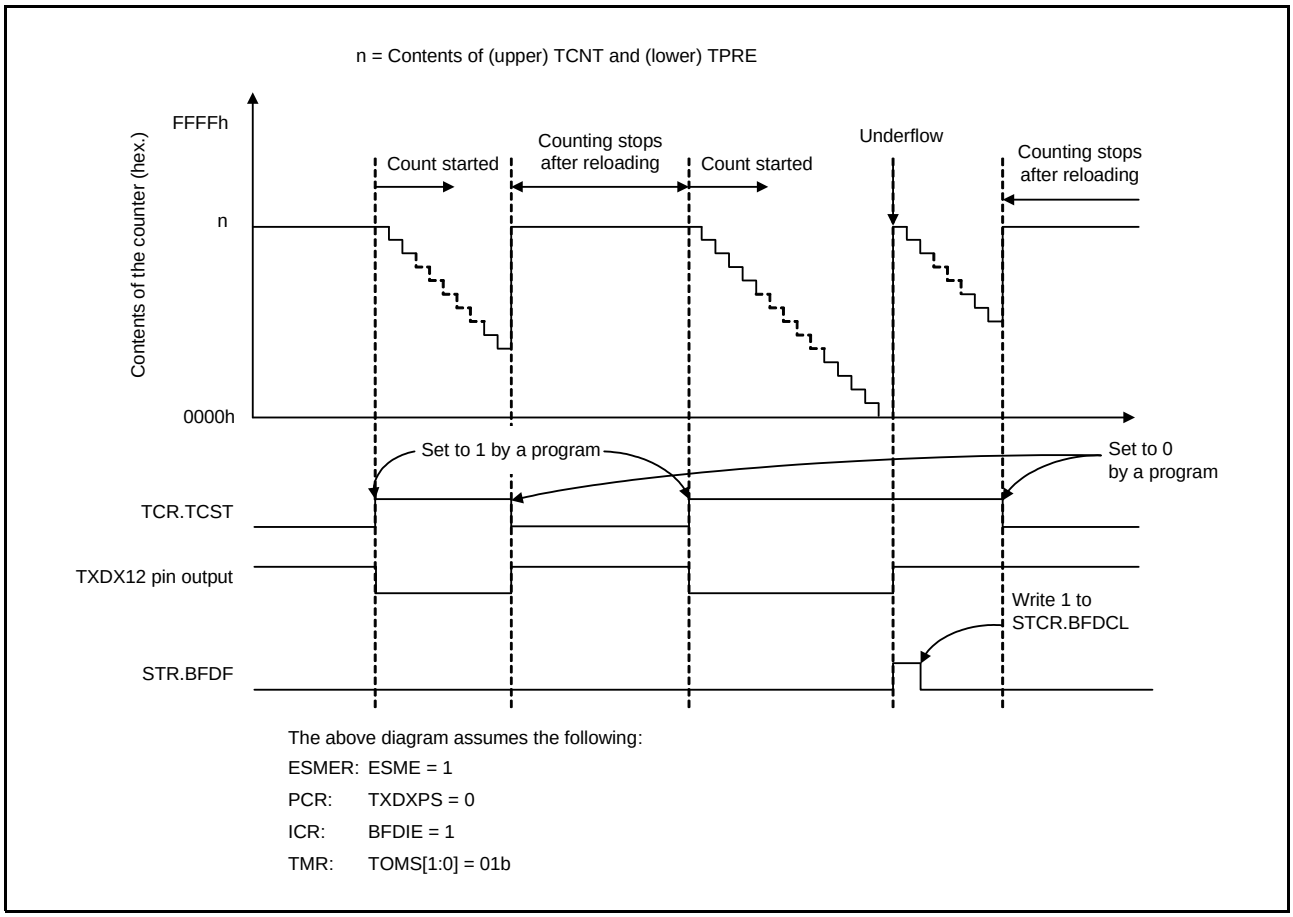


Figure 30.79 Example of Operations in Break Field Low Width Output Mode

(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 30.80 shows an example of operations in Break Field low width output mode.

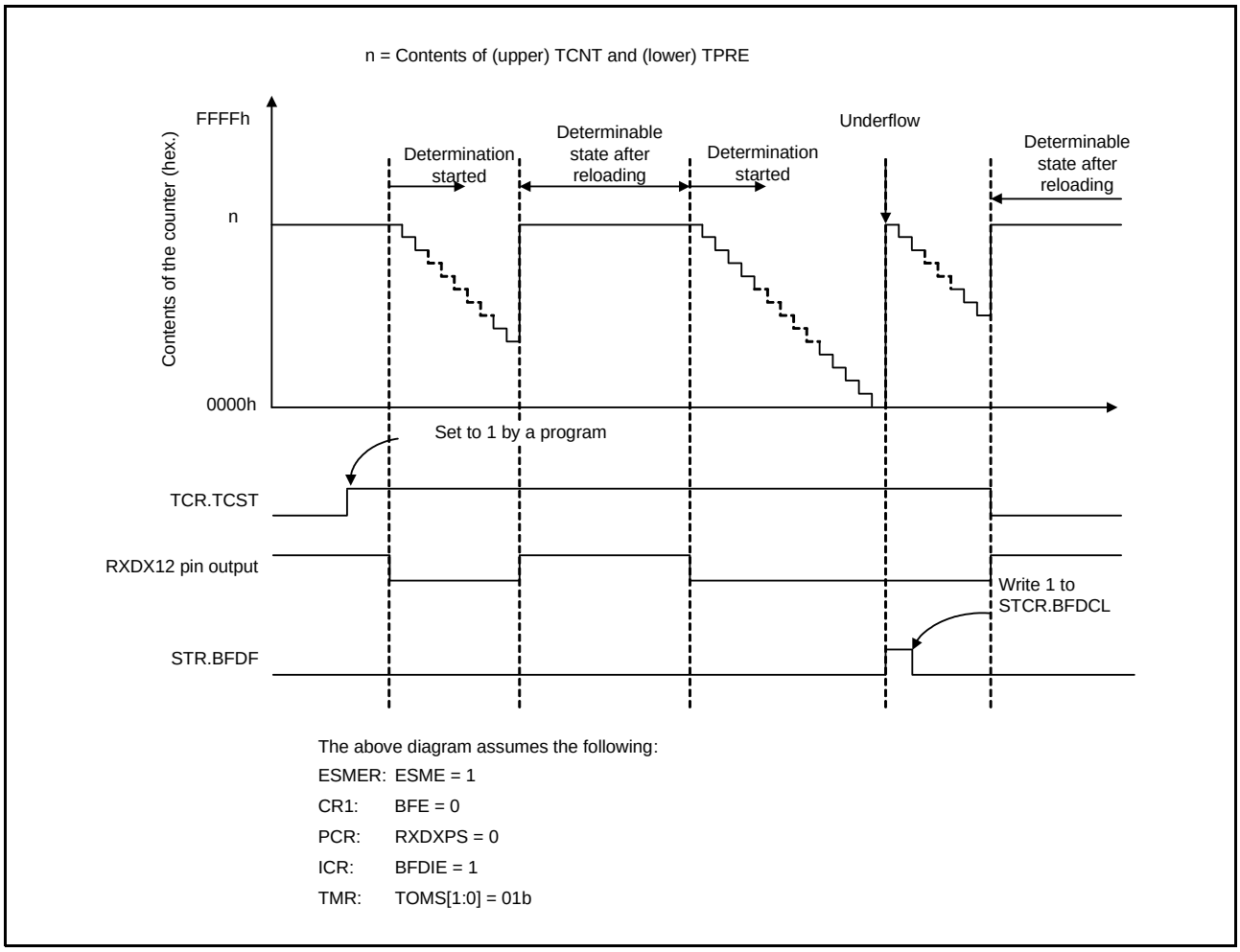


Figure 30.80 Example of Operations in Break Field Low Width Determination Mode

(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.

30.11 Noise Cancellation Function

Figure 30.81 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.

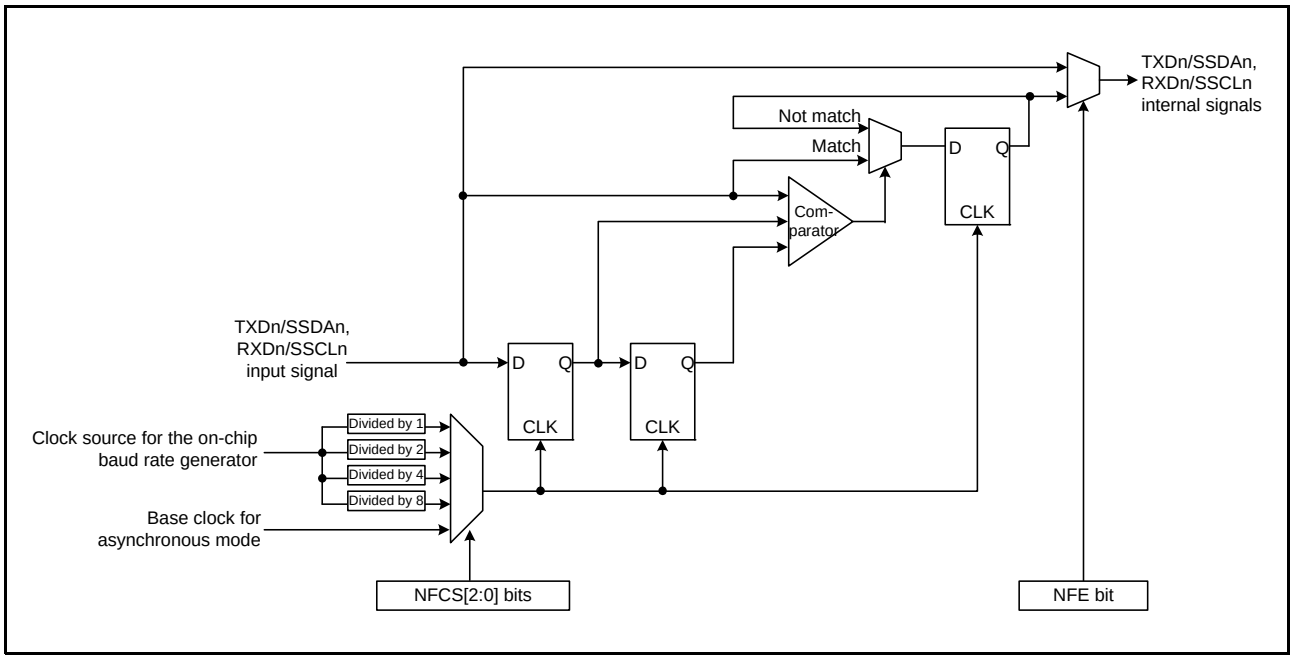


Figure 30.81 Block Diagram of Digital Noise Filter

30.12 Interrupt Sources

30.12.1 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.

30.12.2 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 30.38 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 30.38 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 30.39 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 30.39 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

30.12.3 Interrupts in Smart Card Interface Mode

Table 30.40 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 30.40 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 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa).

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.

30.12.4 Interrupts in Simple I²C Mode

The interrupt sources in simple I²C mode are listed in Table 30.41. 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 30.41 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).

30.12.5 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 30.42.

Table 30.42 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

30.13 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.

30.14 Usage Notes

30.14.1 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.

30.14.2 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.

30.14.3 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.

30.14.4 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).

30.14.5 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.

30.14.6 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 30.82).

(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 30.82).

(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 30.82).

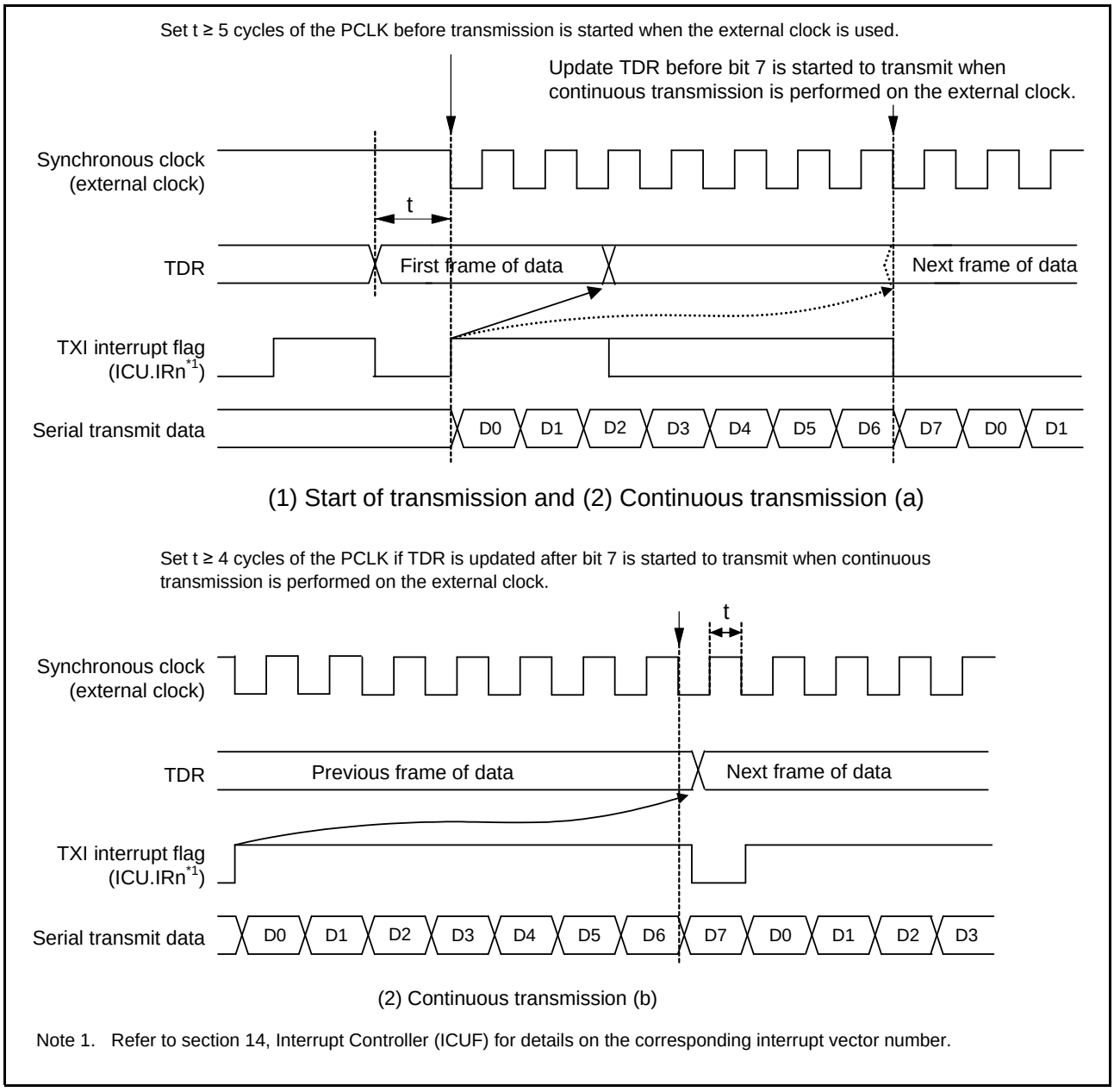


Figure 30.82 Restrictions on Use of External Clock in Clock Synchronous Transmission

30.14.7 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.

30.14.8 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 14, Interrupt Controller (ICUF).

- 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.

30.14.9 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 30.83 shows a sample flowchart for transition to software standby mode during transmission. Figure 30.84 and Figure 30.85 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 30.86 shows a sample flowchart for transition to software standby mode during reception.

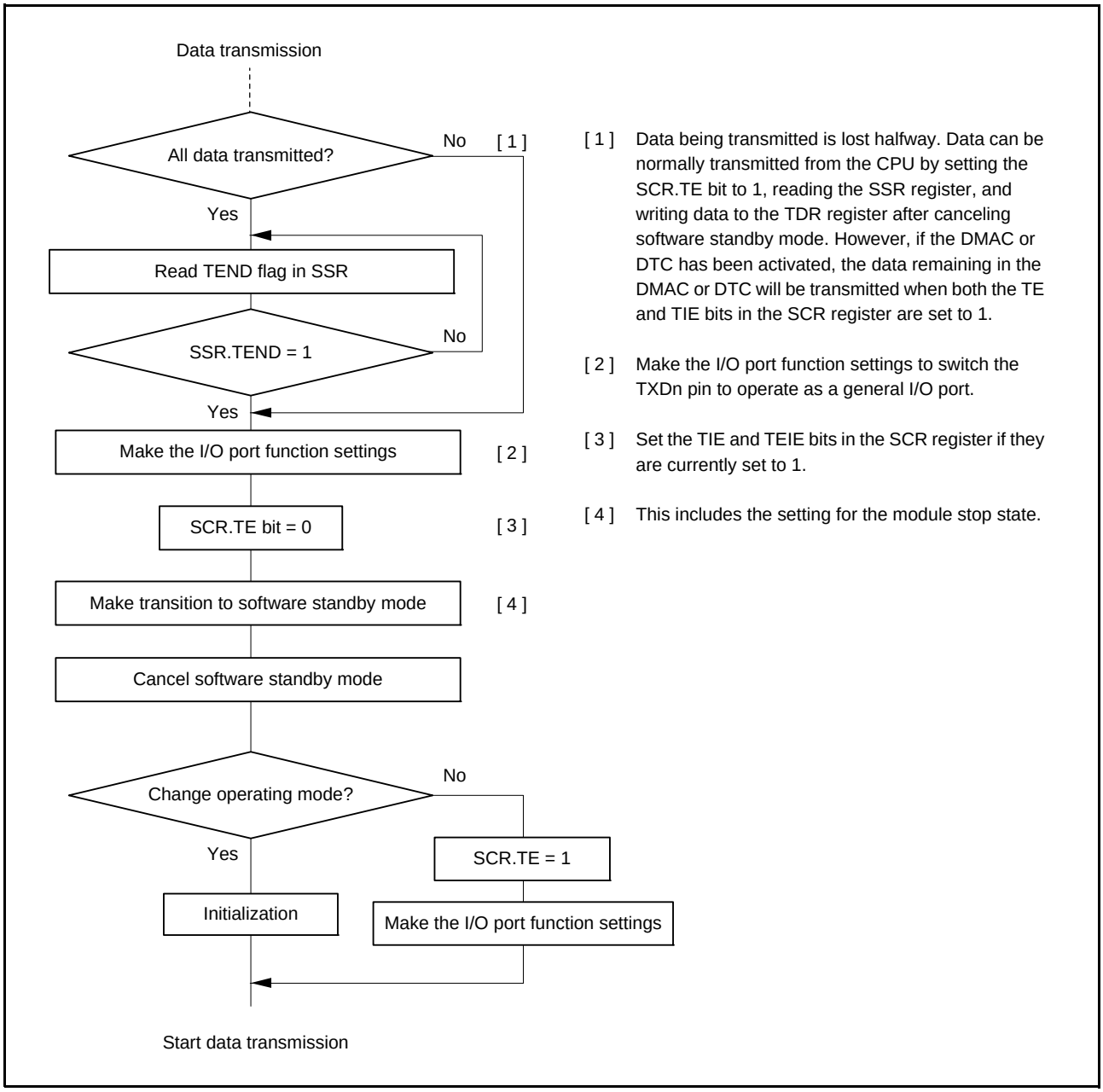


Figure 30.83 Example of Flowchart for Transition to Software Standby Mode during Transmission

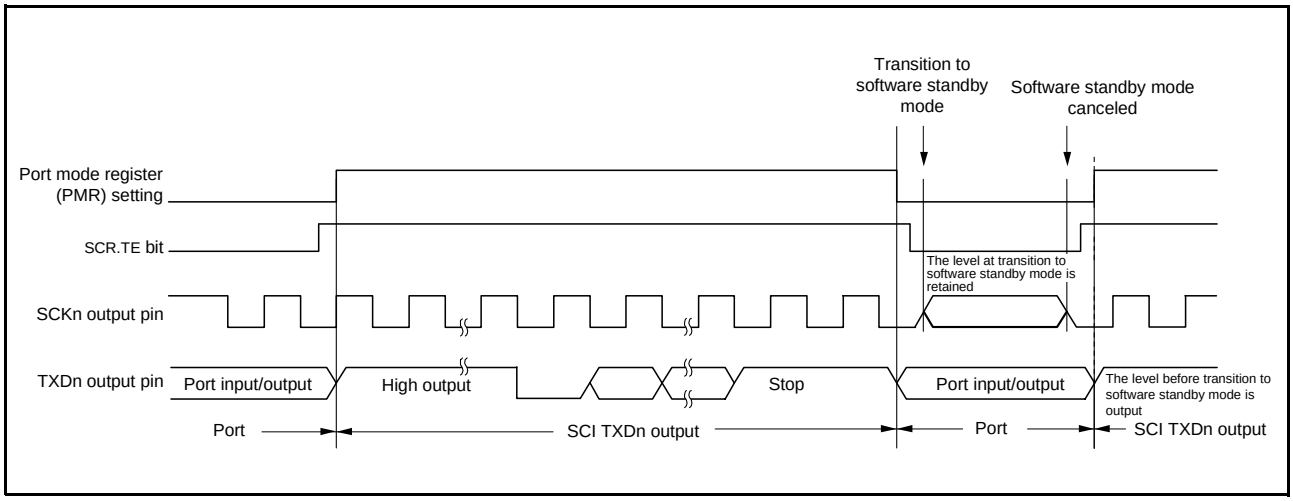


Figure 30.84 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

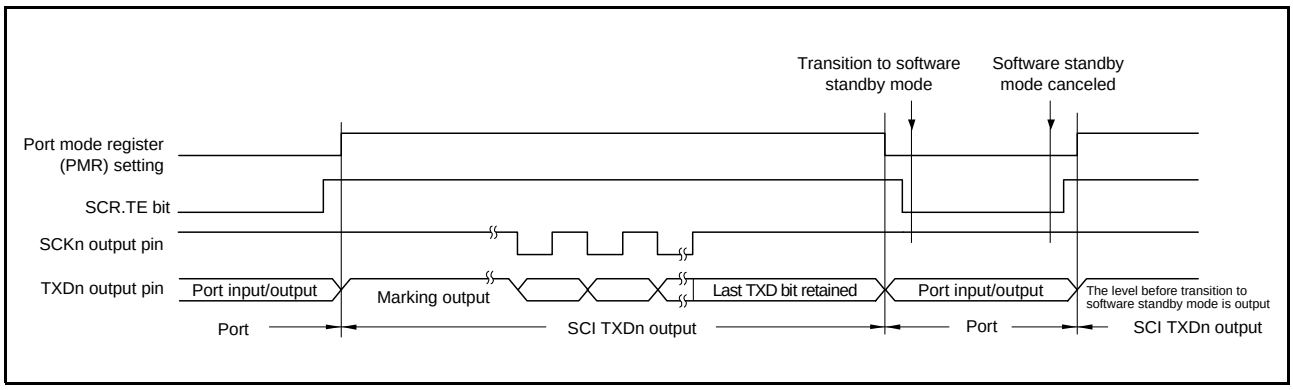


Figure 30.85 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

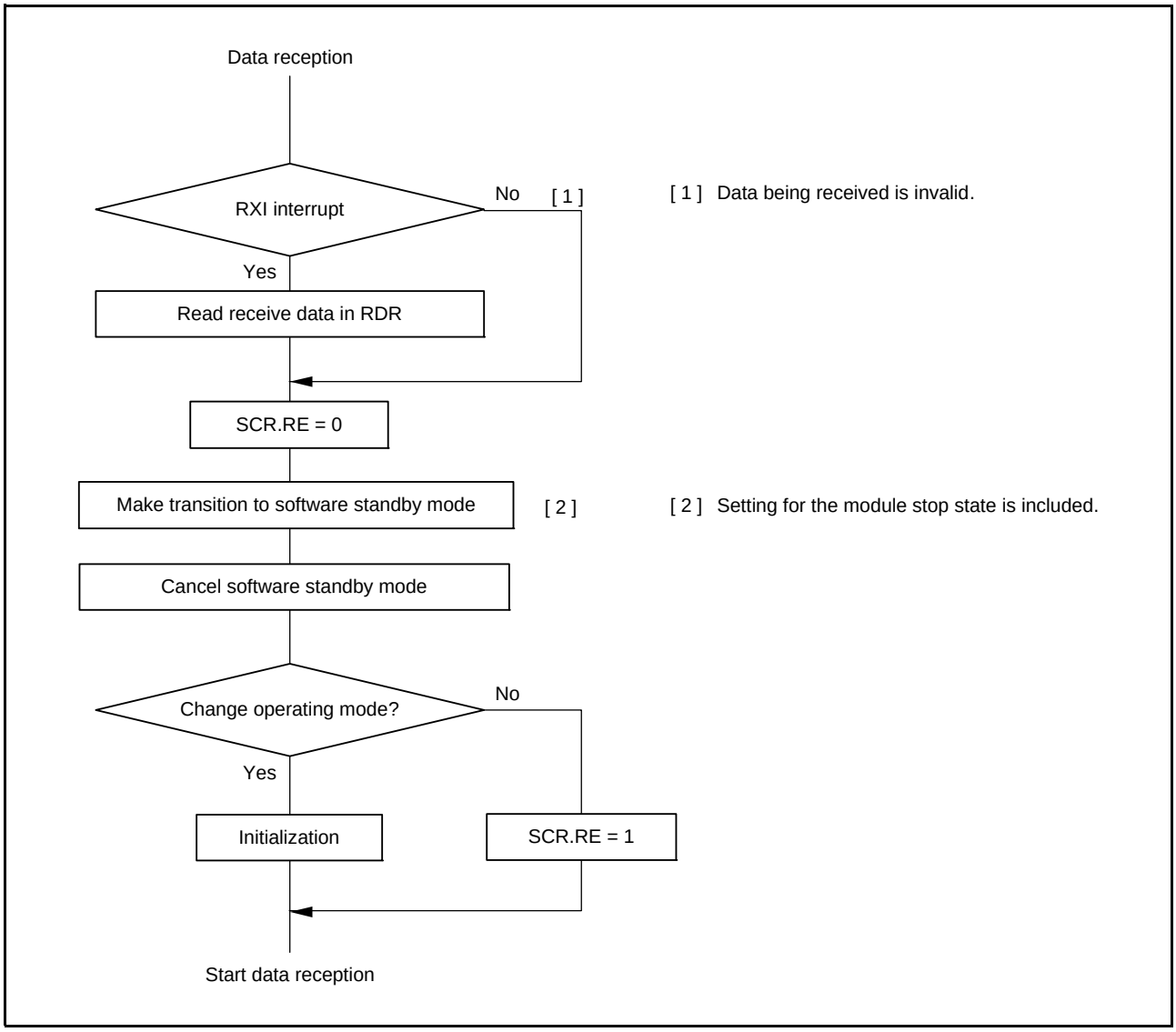


Figure 30.86 Example of Flowchart for Transition to Software Standby Mode during Reception

(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.

30.14.10 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

30.14.11 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 30.87. 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.

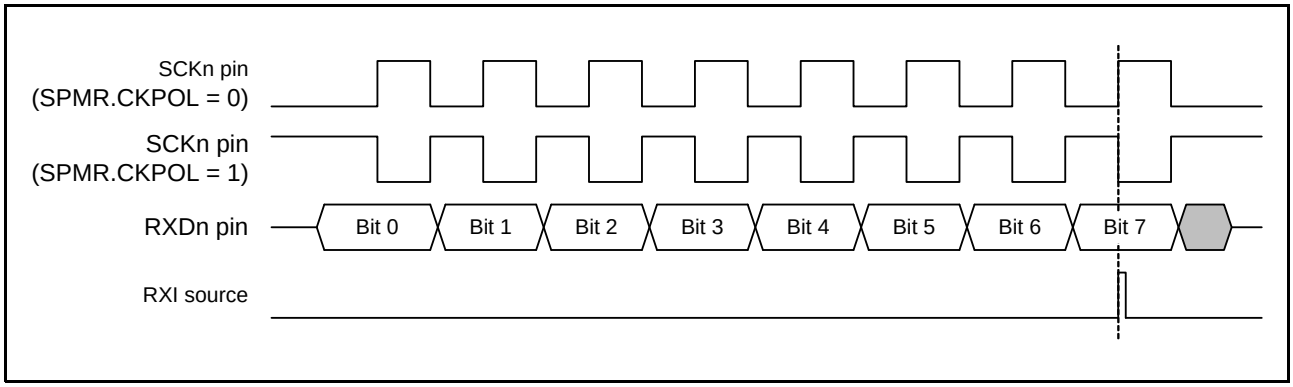


Figure 30.87 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(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.

30.14.12 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.

30.14.13 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.

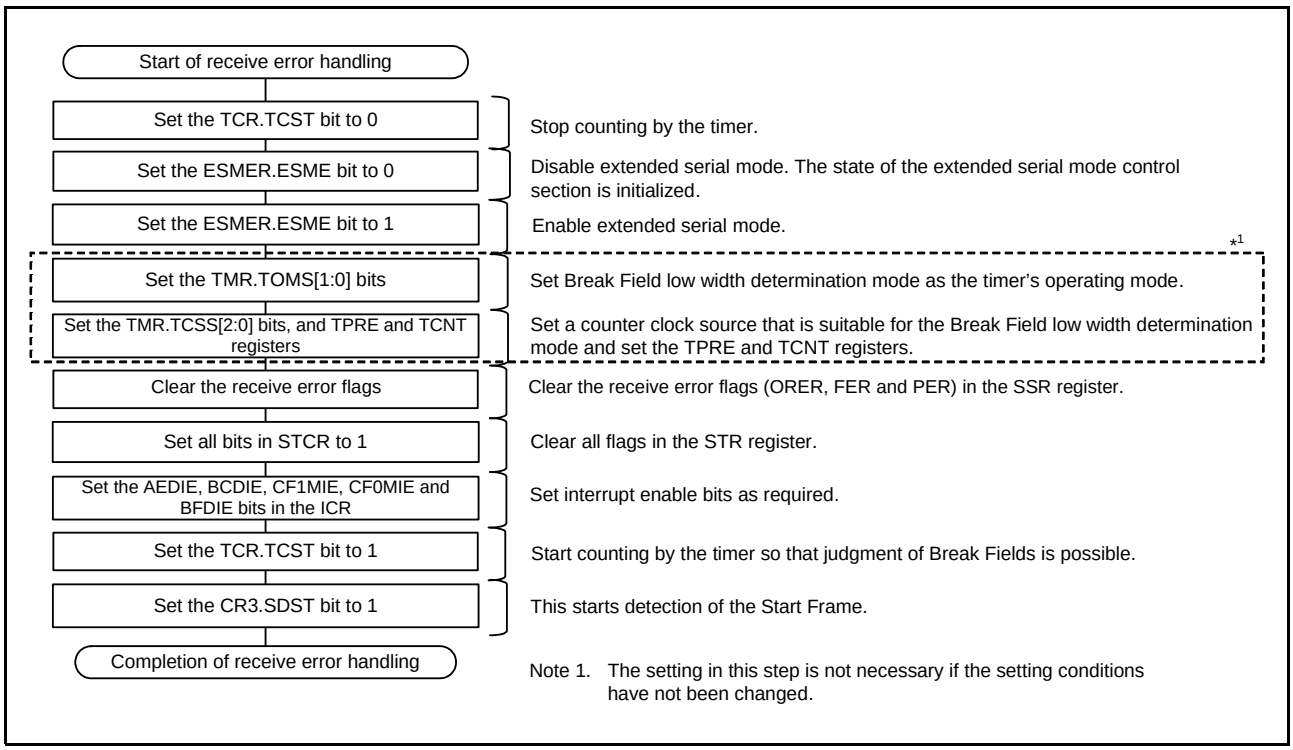


Figure 30.88 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

30.14.14 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 resistor to the TXDn line.

(2) Set the TE bit to 1*1 before changing the pin function to “TXDn”. Change the pin function to “general-purpose I/O port, output” before setting the TE bit to 0.

(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.

30.14.15 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.

31. Serial Communications Interface (RSCI)

31.1 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 31.1 lists the RSCI specifications.

Table 31.1 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 31.1 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 31.1 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 I^2C mode	Transfer format	I^2C-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.

Figure 31.1 shows RSCI Block Diagram (n = 010, 011) with all functions.

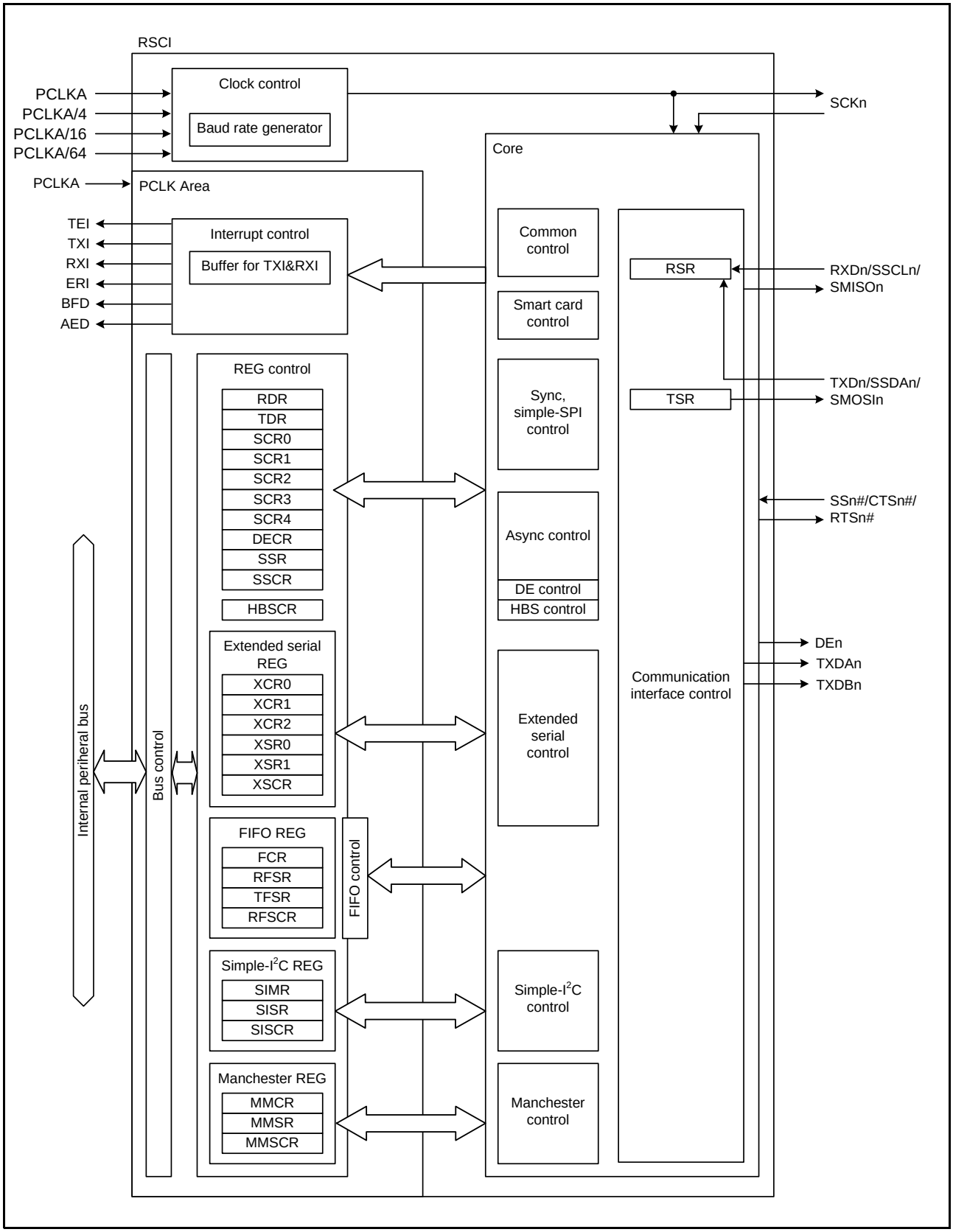


Figure 31.1 RSCI Block Diagram (n = 010, 011)

Table 31.2 to Table 31.5 list RSCI's input/output pins.

Table 31.2 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 31.3 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 31.4 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 31.5 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)

31.2 Register Descriptions

This chapter describes the RSCI registers, their functional specifications, and their operating specifications.

31.2.1 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.

31.2.2 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.

31.2.3 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.

31.2.4 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.

31.2.5 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 31.4, 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 31.3.6, 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 31.2.12, Manchester Mode Control Register (MMCR) and section 31.2.21, 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.

31.2.6 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 31.6 for the relationship between the CSREP and CTSE bit settings and the pin functions.

Table 31.6 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 31.7.

Table 31.7 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 31.7.2, Data Format (Except in Block Transfer Mode).

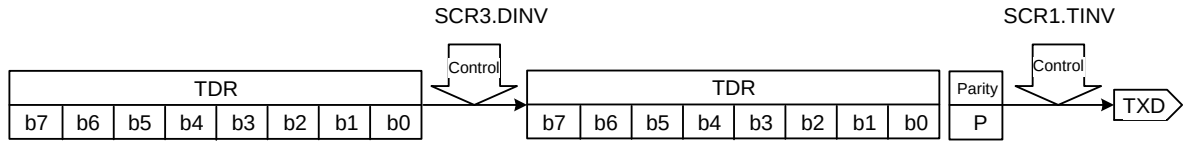
TINV Bit (Transmitter Output Invert), RINV Bit (Receiver Input Invert)

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 31.2 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.

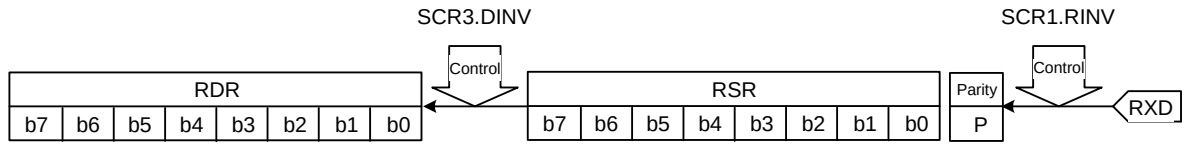
The receive/transmit data control (Data size = 8bits, Even parity, MSB first)

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 31.2 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 31.16, 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.

31.2.7 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 31.7.4, 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 31.8 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 31.9 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 31.9 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 31.10 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 31.11 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 31.12 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 31.13 and Table 31.14 list examples of N settings in BRR[7:0] in asynchronous mode and manchester mode. Table 31.15 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 31.17. Examples of BRR[7:0] bits (N) settings in smart card interface mode are listed in Table 31.19. Examples of BRR[7:0] bits (N) settings in simple I^2C mode are listed in Table 31.21. 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 31.7.4, Receive Data Sampling Timing and Reception Margin. Table 31.16 and Table 31.18 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 31.13 and Table 31.14. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 31.13 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 31.14 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 31.15 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 31.16 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 31.17 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 31.18 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 31.19 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 31.20 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 31.21 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 31.22 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 31.23.

The initial value of MDDR[7:0] bits is FFh. Bit 7 in this register is fixed to 1.

Table 31.23 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 I^2C*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 31.11, 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 I^2C mode satisfy the I^2C standard.

Table 31.24 and Table 31.25 list examples of N settings in BRR[7:0] bits and M settings in MDDR[7:0] bits in asynchronous mode.

Table 31.24 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency PCLKA (MHz)														
	8					9.8304					10				
Bit Rate (bps)	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

	Operating Frequency PCLKA (MHz)														
	12					12.288					14				
Bit Rate (bps)	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

	Operating Frequency PCLKA (MHz)														
	16					17.2032					18				
Bit Rate (bps)	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 31.25 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)

31.2.8 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 31.2 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 31.108 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 31.108 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 31.26 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 31.7.8, 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 31.7.6, Serial Data Transmission (Except in Block Transfer Mode) and section 31.7.8, Clock Output Control.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 31.7.3, Block Transfer Mode.

31.2.9 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 31.3.10, 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 31.10.7, 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 31.3.11, 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.

31.2.10 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 31.78 and Figure 31.79 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.

31.2.11 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.

31.2.12 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 31.5.7, 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 31.5.6, 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 31.5.9, 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 31.36 and Figure 31.37.

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 31.35.
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 31.35.
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.

31.2.13 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.).

31.2.14 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.

31.2.15 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.

31.2.16 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.

31.2.17 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 14.5.2, 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.

31.2.18 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

31.2.19 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 14.5.2, 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.

31.2.20 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.

31.2.21 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 14.5.2, 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 14.5.2, 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 14.5.2, 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 31.5.11, Errors in Manchester Mode.
[Clearing condition]
Write 1 to MMSCR.MCERC bit.
Refer to section 14.5.2, 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.

31.2.22 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.

31.2.23 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 31.73), 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.

31.2.24 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

31.2.25 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

31.2.26 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

31.2.27 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

31.2.28 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

31.2.29 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 31.6, HBS Support Mode.

31.3 Operation in Asynchronous Mode

Figure 31.3 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.

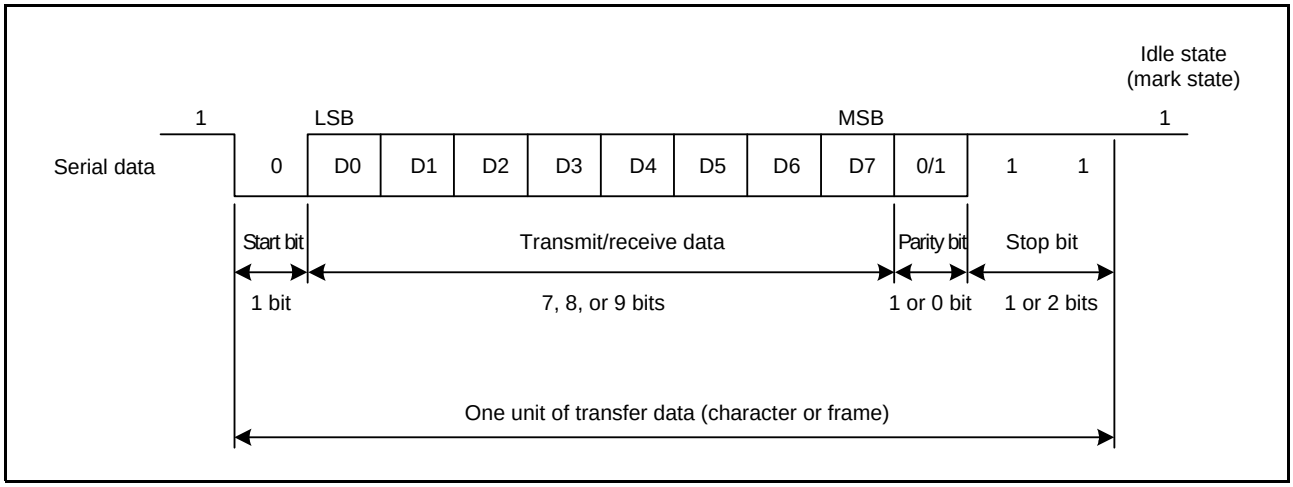


Figure 31.3 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

31.3.1 Serial Data Transfer Format

Table 31.27 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 31.4, Multi-Processor Communication Function.

Table 31.27 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

31.3.2 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 31.4. 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 31.4, 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.

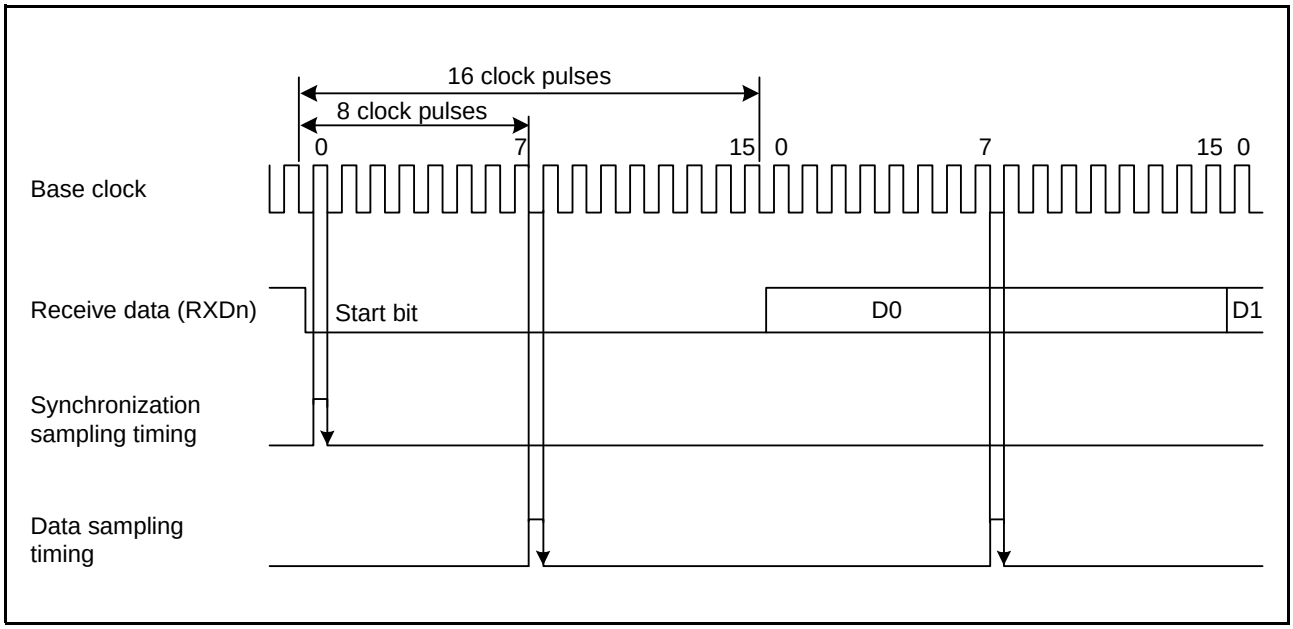


Figure 31.4 Receive Data Sampling Timing in Asynchronous Mode

31.3.3 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 31.5.

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.

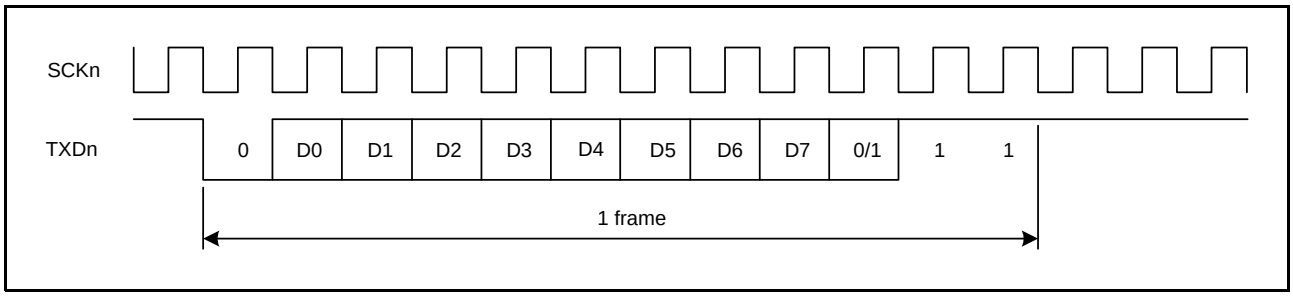


Figure 31.5 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SCR1.PE Bit = 1, SCR3.CHR[1:0] Bits = 10b, MP Bit = 0, STOP Bit = 1)

31.3.4 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 31.3.2, 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.

31.3.5 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

31.3.6 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 31.6 and Figure 31.7 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.

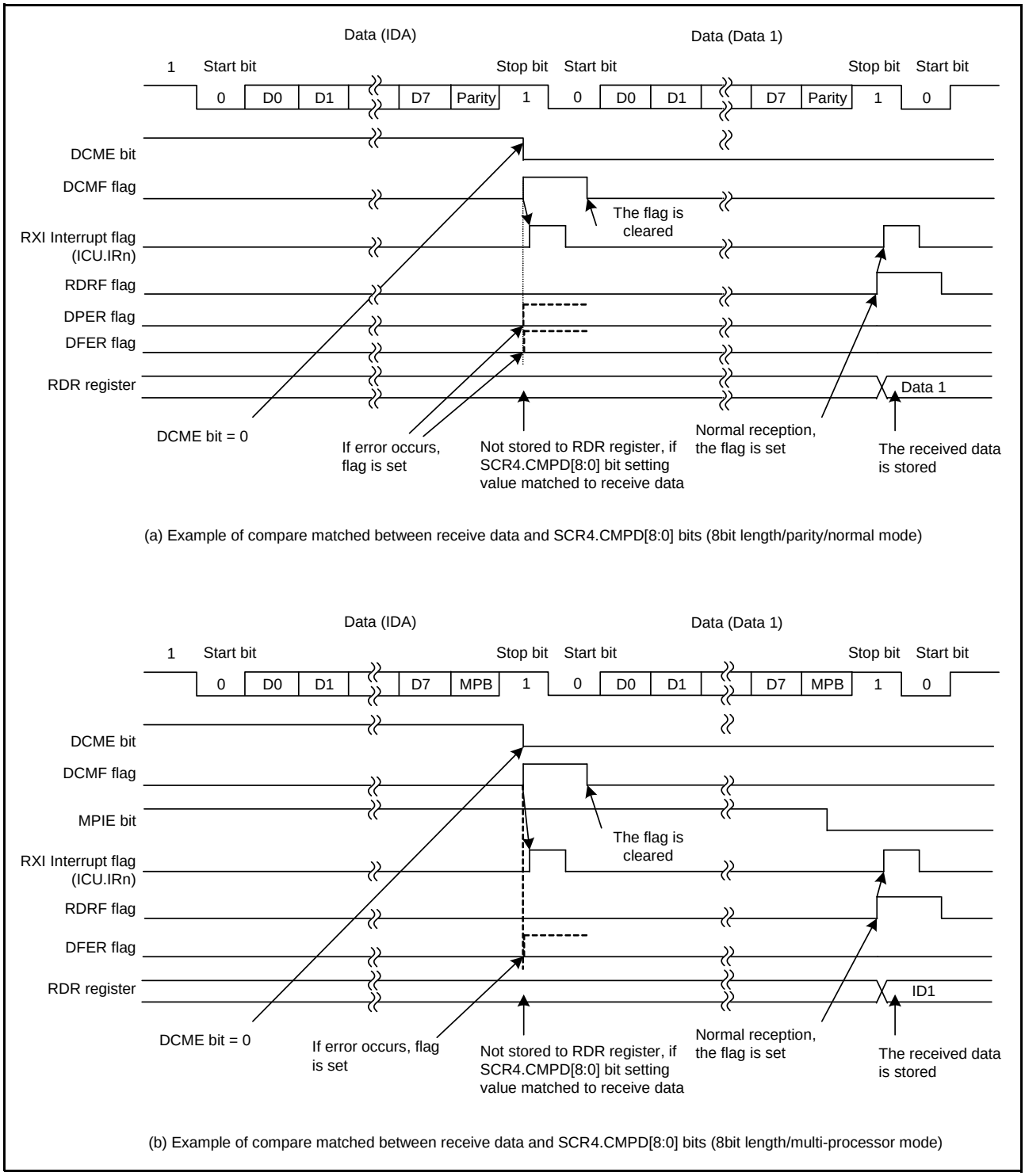
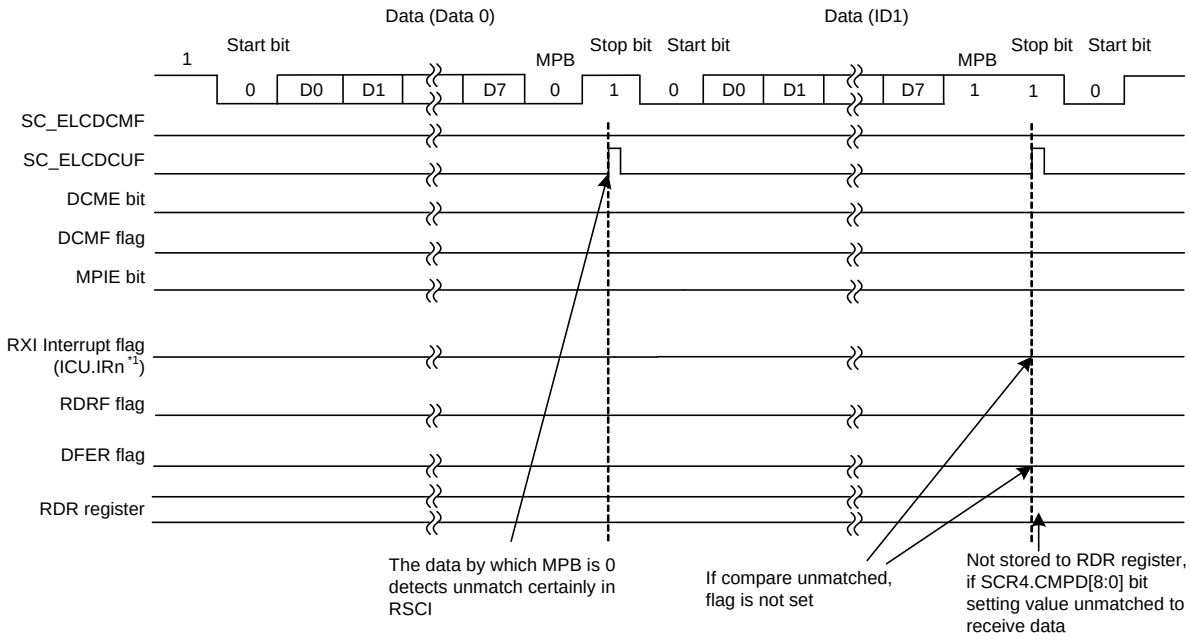
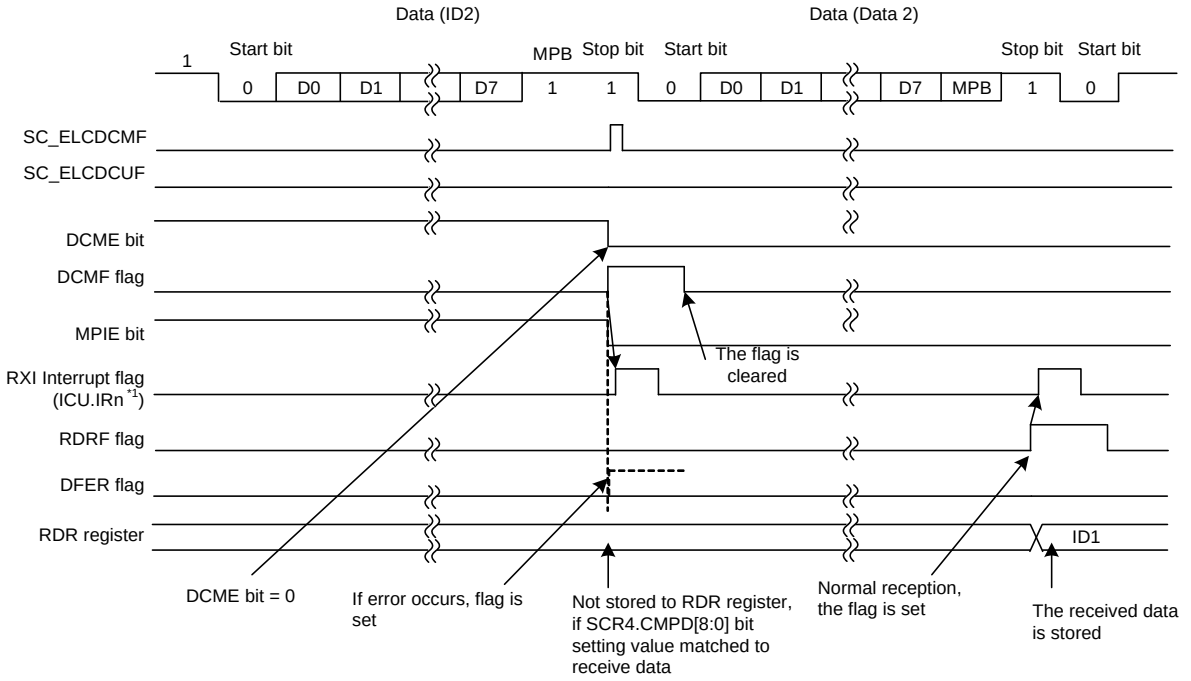


Figure 31.6 Example of Data Match Detection (1) 8bit-Data



(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 31.7 Example of Data Match Detection (2) Multi-Processor Mode/8bit-Data

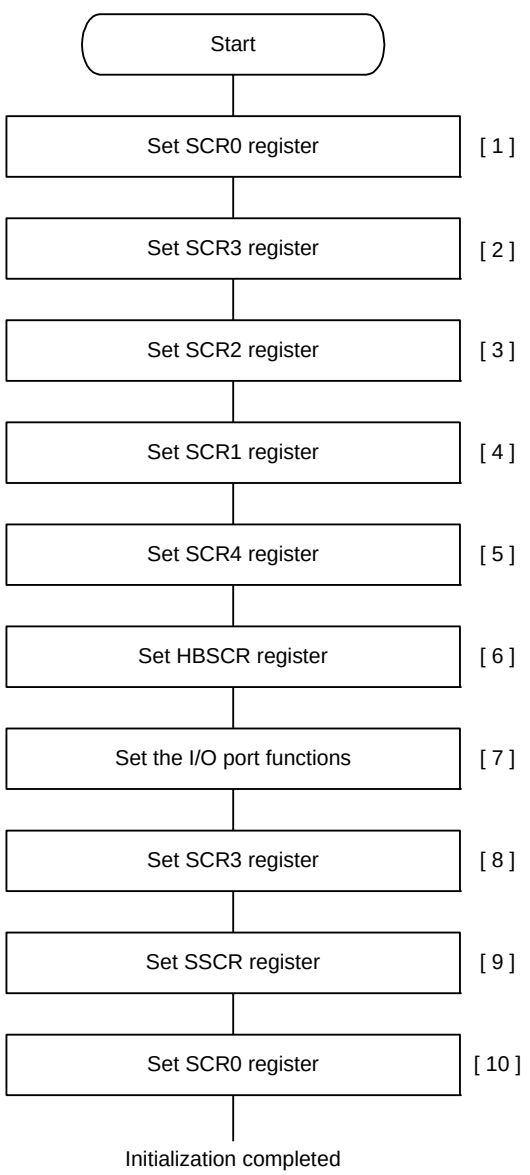
31.3.7 RSCI Initialization (Asynchronous Mode)

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 31.8 or Figure 31.9. 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.

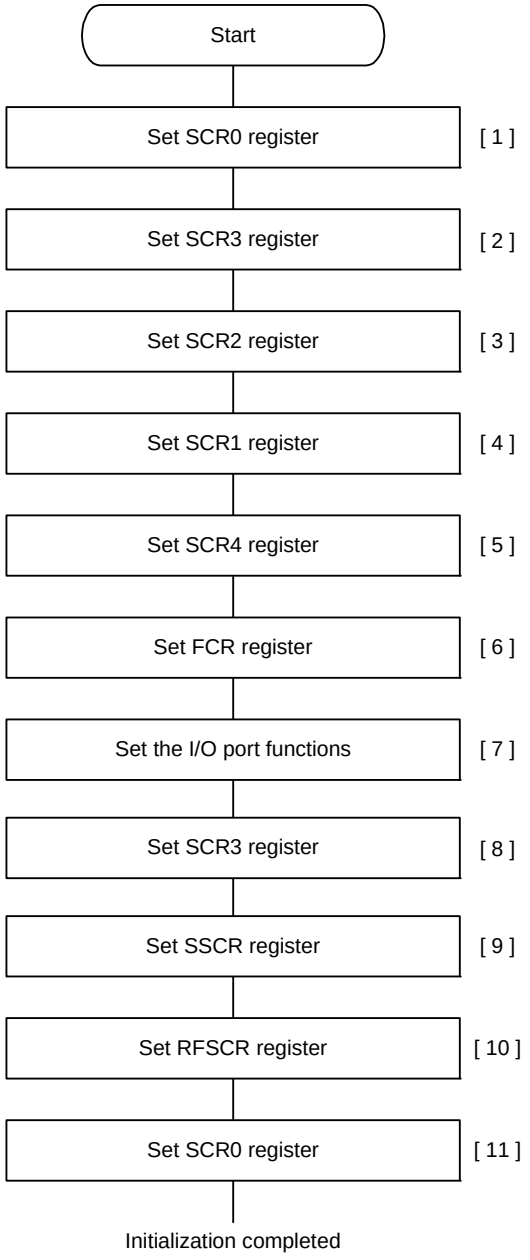
[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[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 31.8 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 31.9 Sample RSCI Initialization Flowchart (Asynchronous Mode, FIFO Mode)

Figure 31.10 shows an example of the timing when data is transmitted after reset is released, and RSCI is set to asynchronous mode according to Figure 31.8 or Figure 31.9. 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.

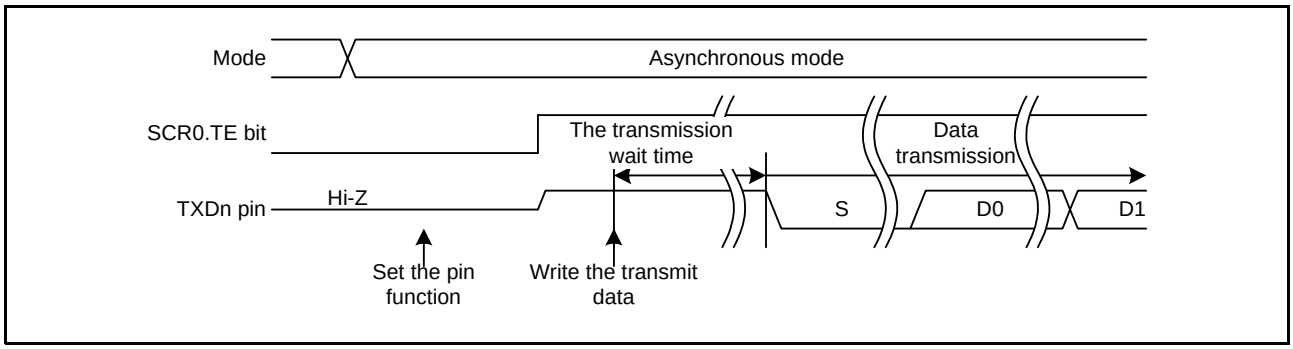


Figure 31.10 Example of Data Transmission Timing in Asynchronous Mode

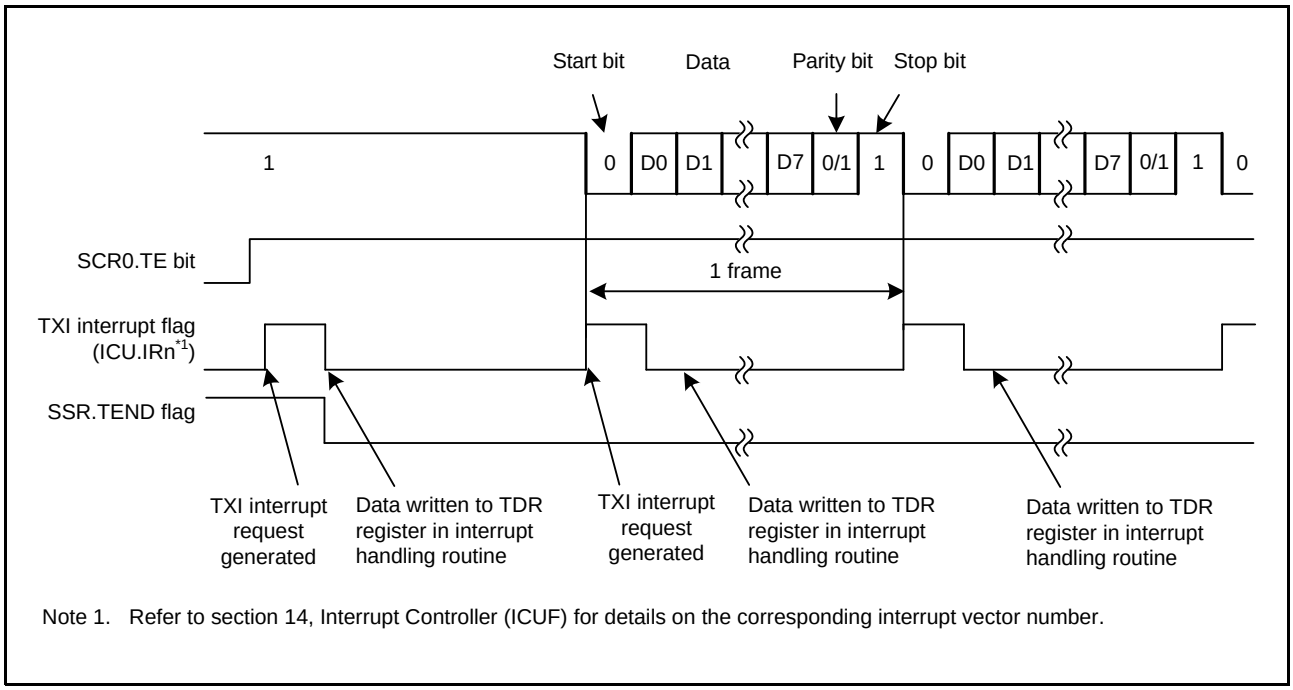
31.3.8 Serial Data Transmission (Asynchronous Mode)

(1) Non-FIFO Mode

Figure 31.11 to Figure 31.13 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the RSCI operates as described below.

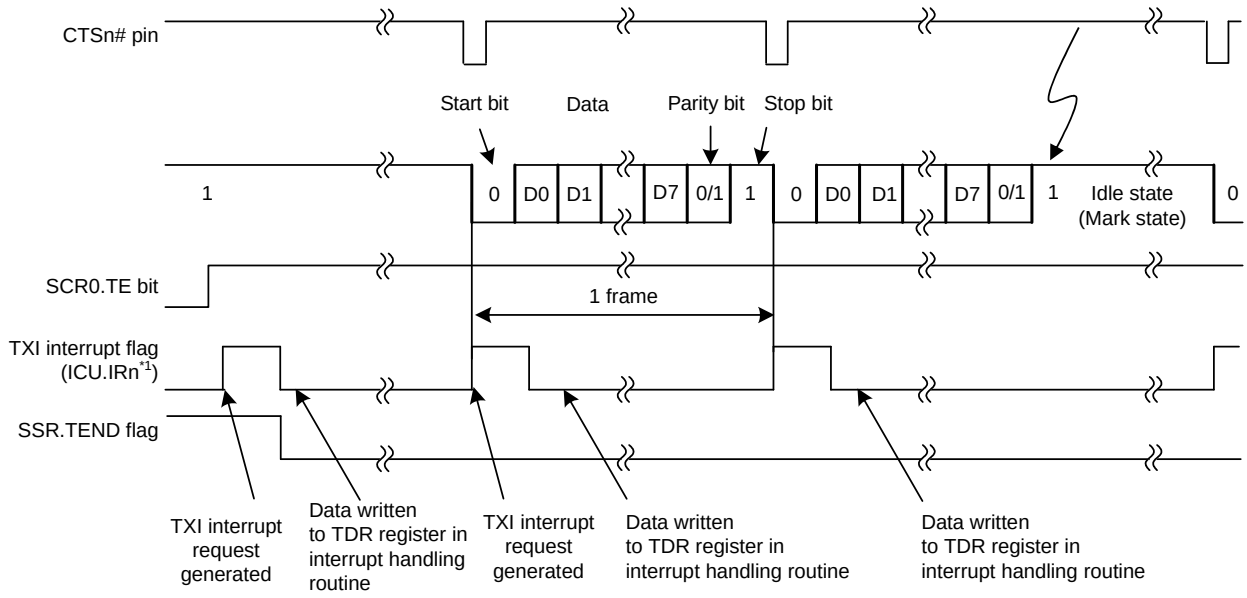
1. The RSCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. At the beginning of transmission, set 1 to the SCR0.TE bit and the SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.
2. Transmission 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 31.15 shows the example of Serial Transmission Flowchart.



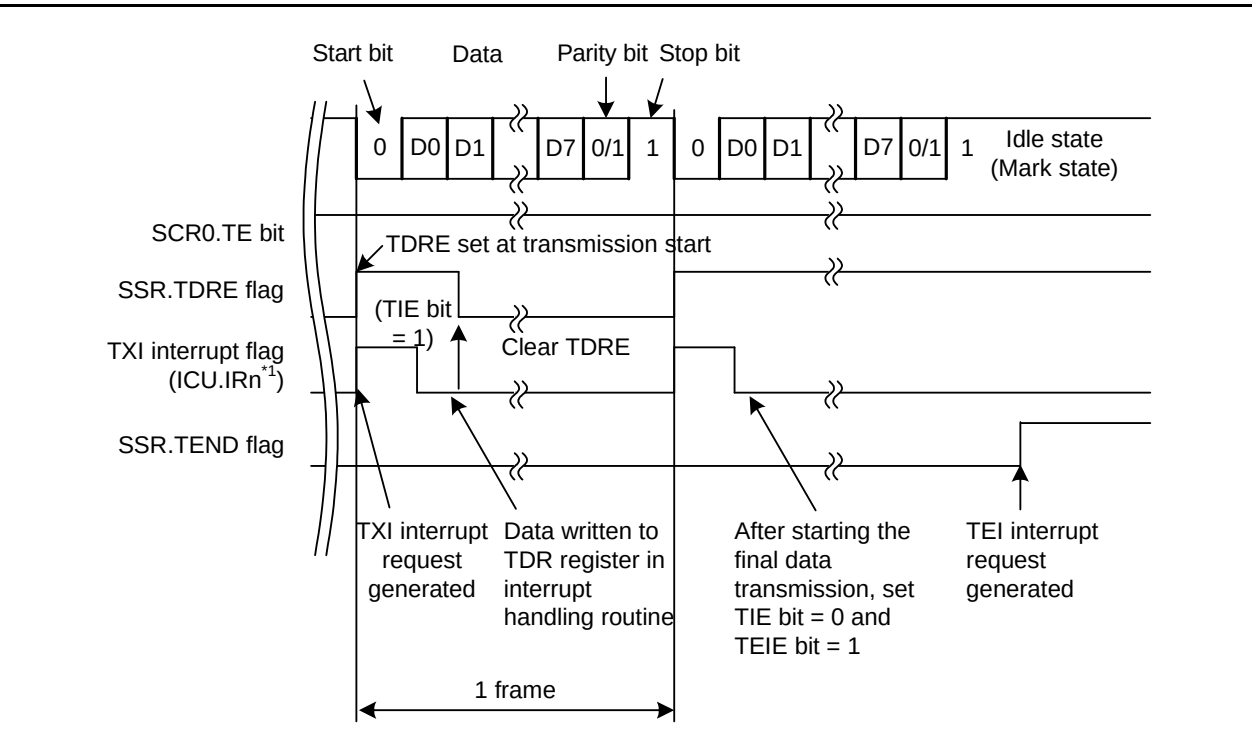
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.11 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)



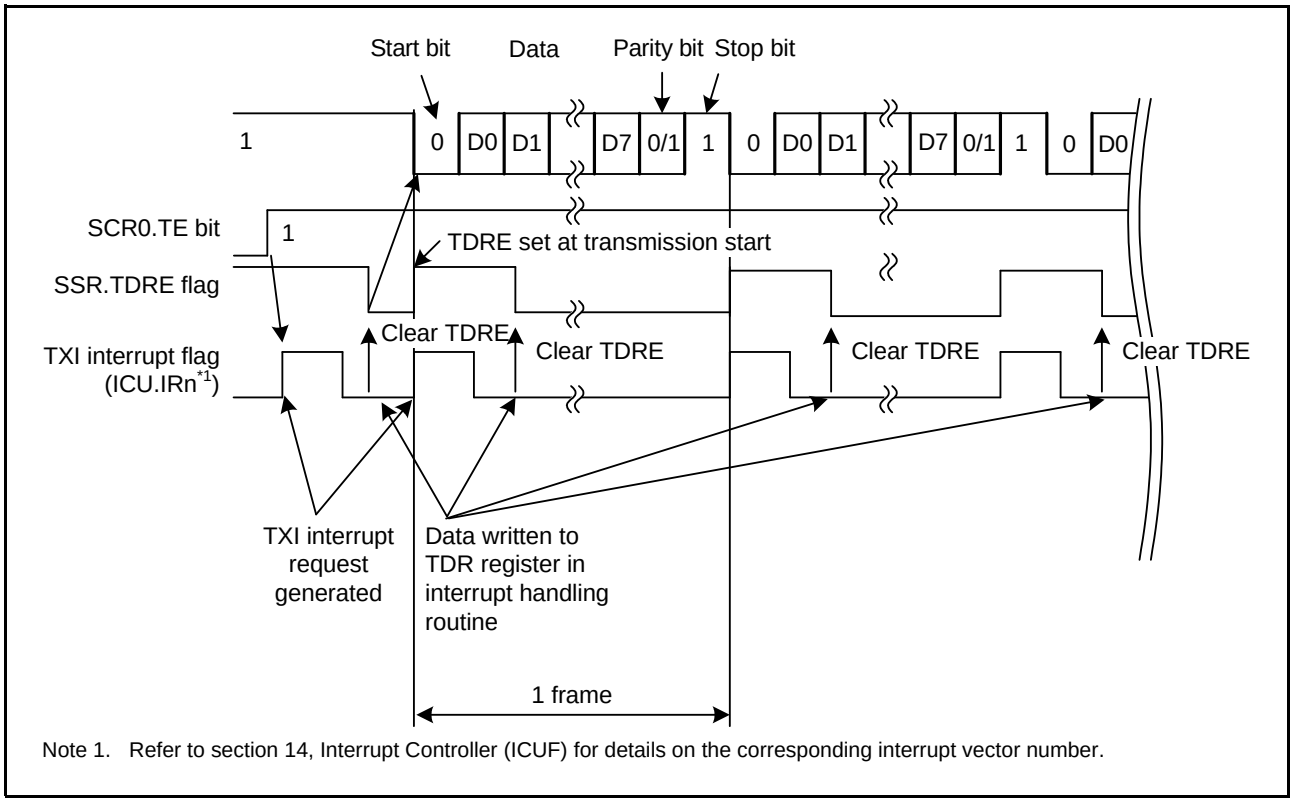
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.12 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.13 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.14 Example of Operation for Serial Transmission in Asynchronous Mode (4) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, during Transmission)

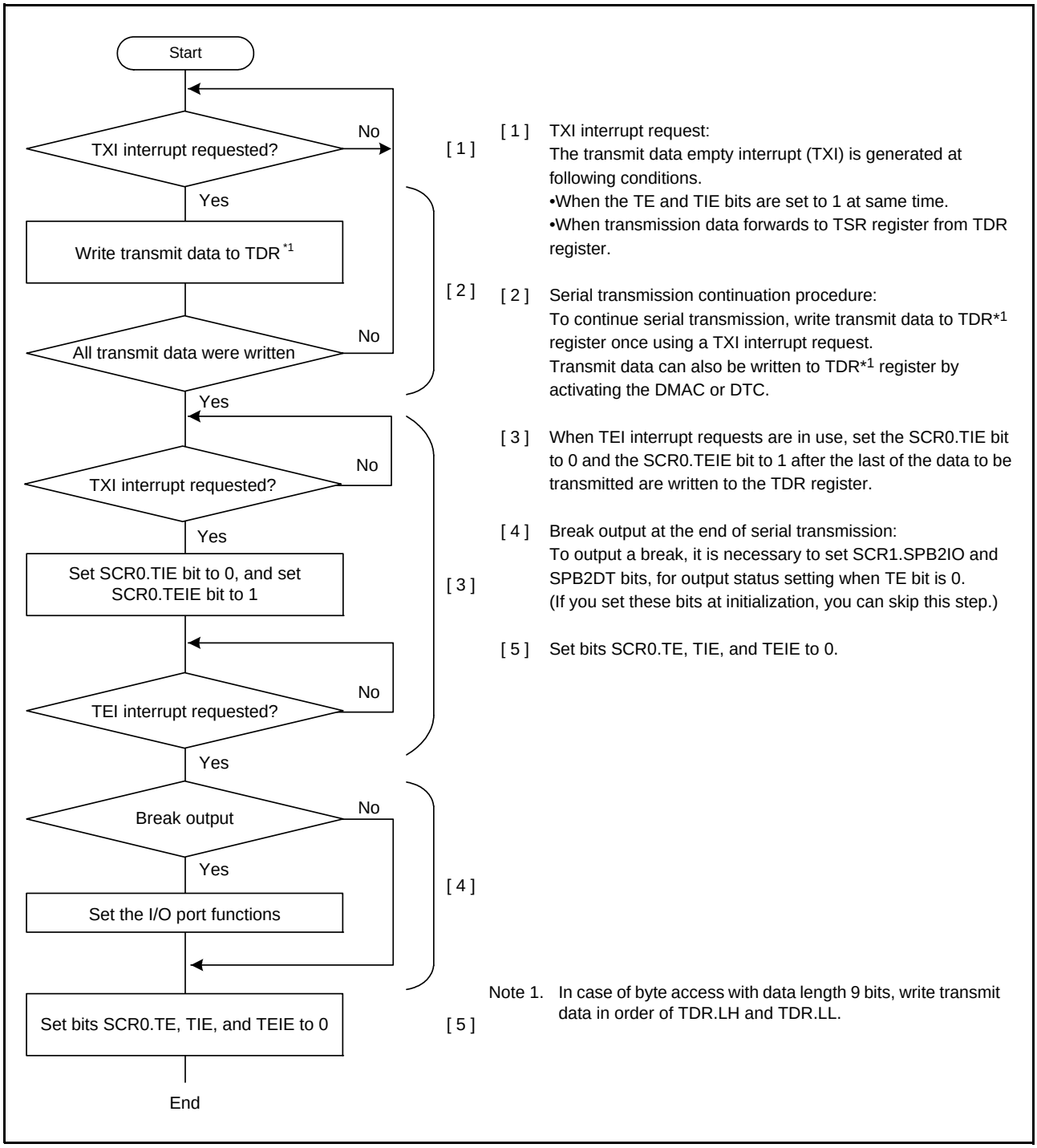


Figure 31.15 Example of Serial Transmission Flowchart in Asynchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Table 31.28 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 31.28 Data Format That is Written to Transmit FIFO (TDR) (FIFO Mode)

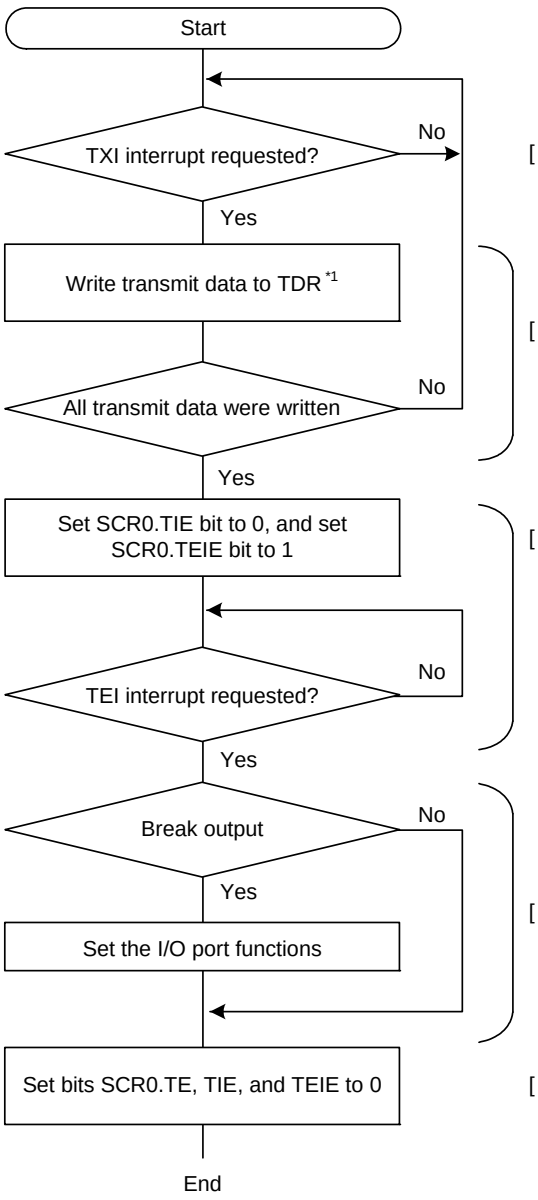
Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: 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 31.16 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.

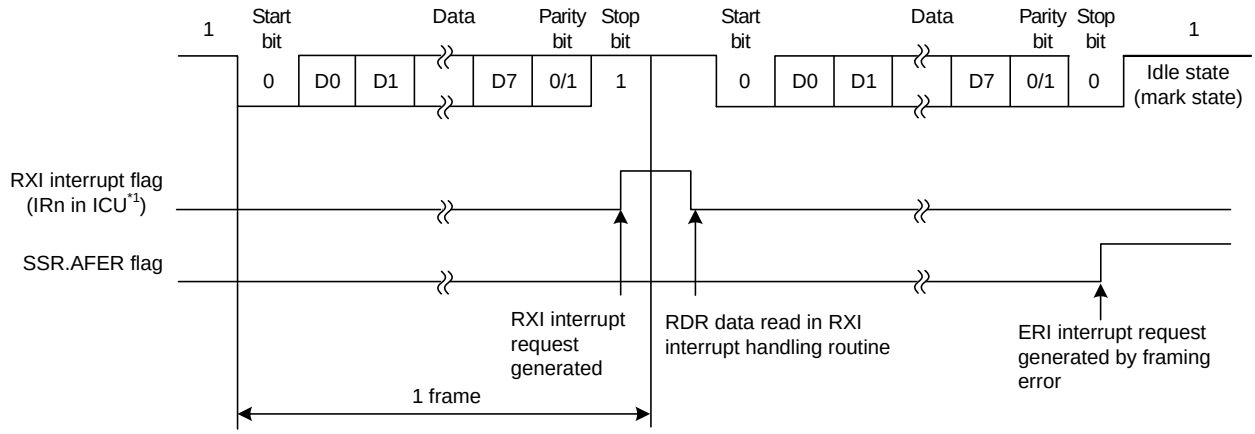
Figure 31.16 Example of Serial Transmission Flowchart in Asynchronous Mode (FIFO Mode)

31.3.9 Serial Data Reception (Asynchronous Mode)

(1) Non-FIFO Mode

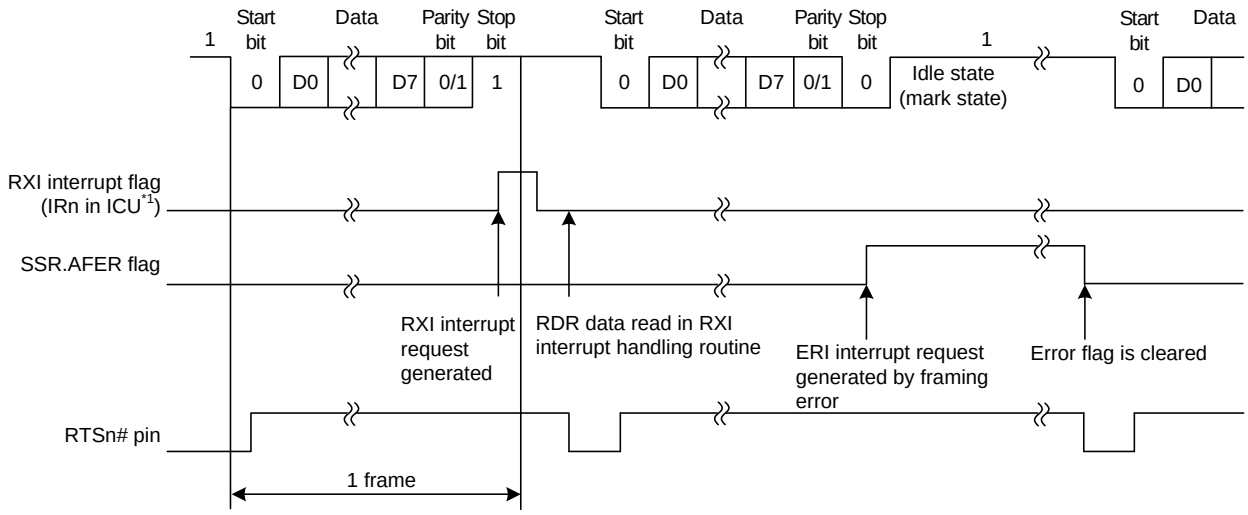
Figure 31.17 and Figure 31.18 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the RSCI operates as described below.

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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.17 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.18 Example of RSCI Operation for Serial Reception in Asynchronous Mode (2) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Used)

Table 31.29 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 31.19 and Figure 31.20 show samples of flowcharts for serial data reception.

Table 31.29 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

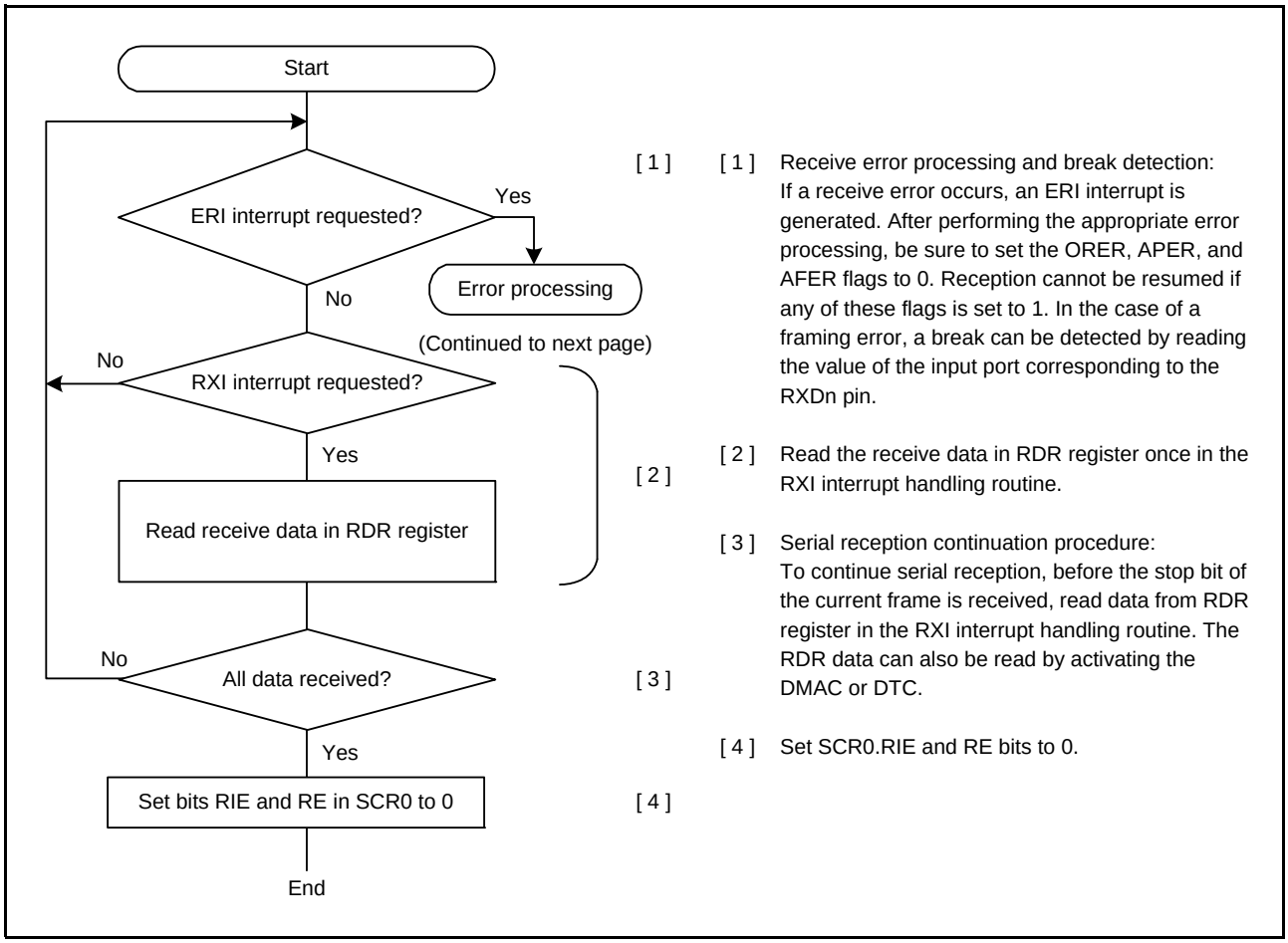


Figure 31.19 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (1)

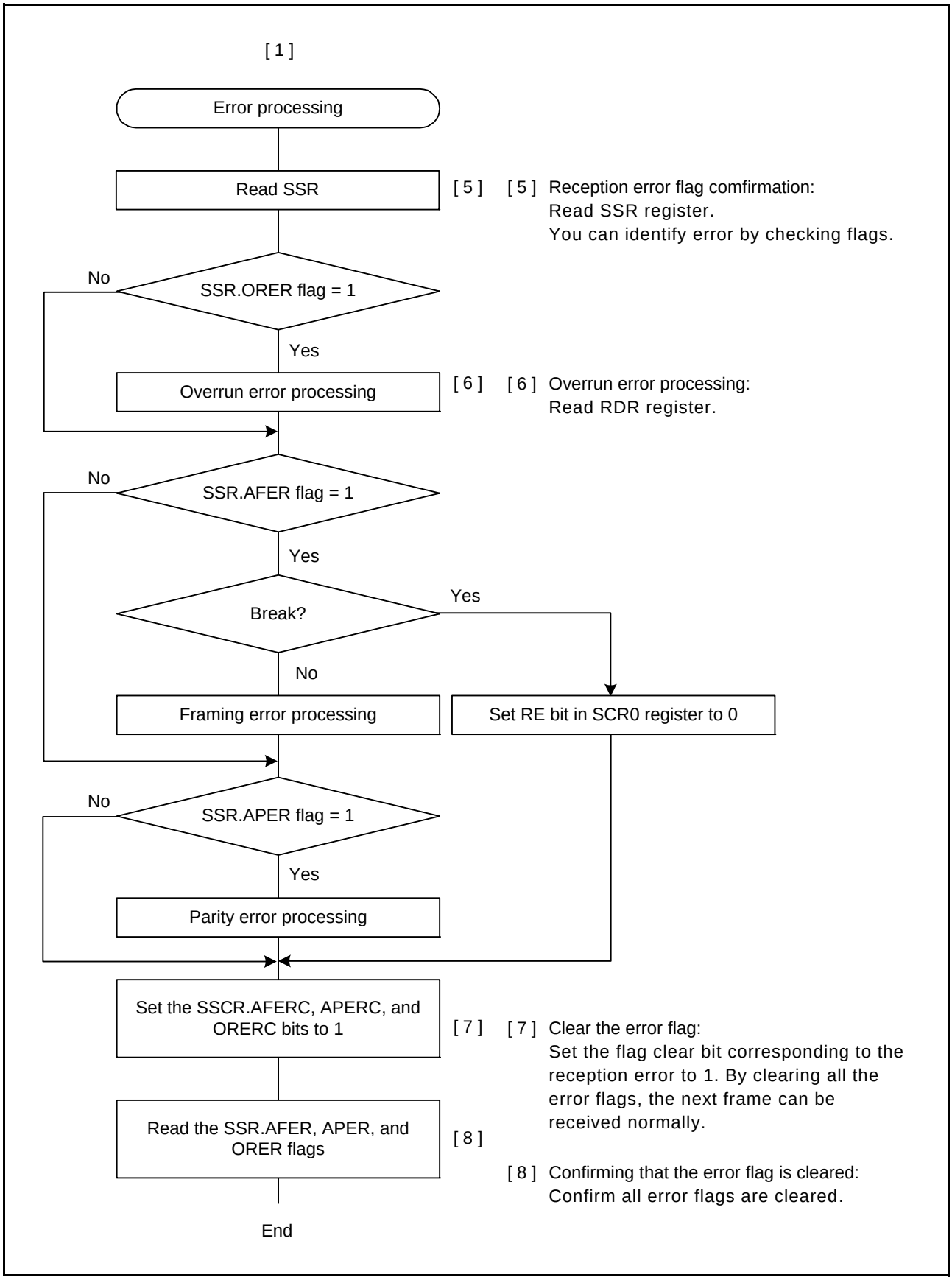


Figure 31.20 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (2)

(2) FIFO Mode

Table 31.30 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 31.30 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 31.31 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 31.21 and Figure 31.22 show samples of flowcharts for serial data reception in FIFO mode.
In serial data reception, the RSCI operates as described below.

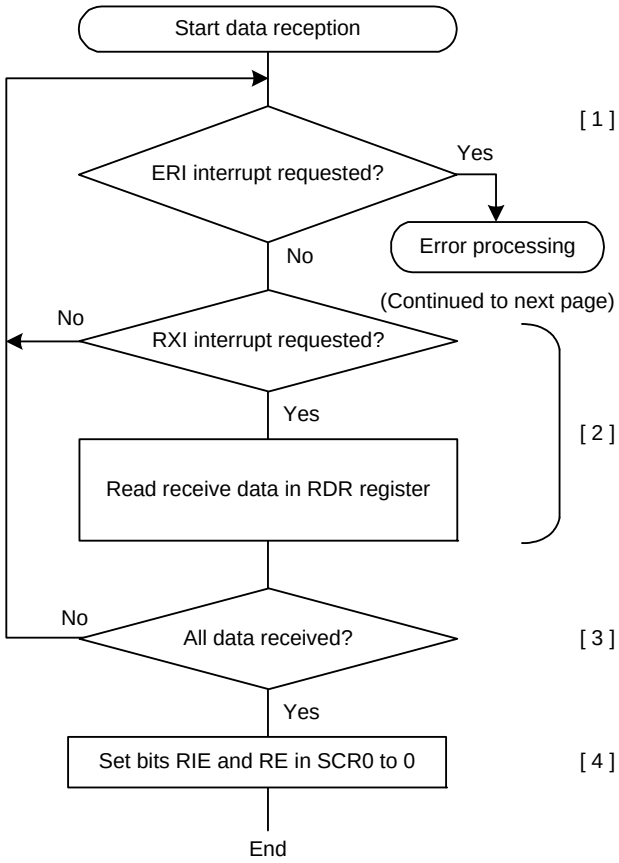
1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit 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 31.31 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 31.21 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (1)

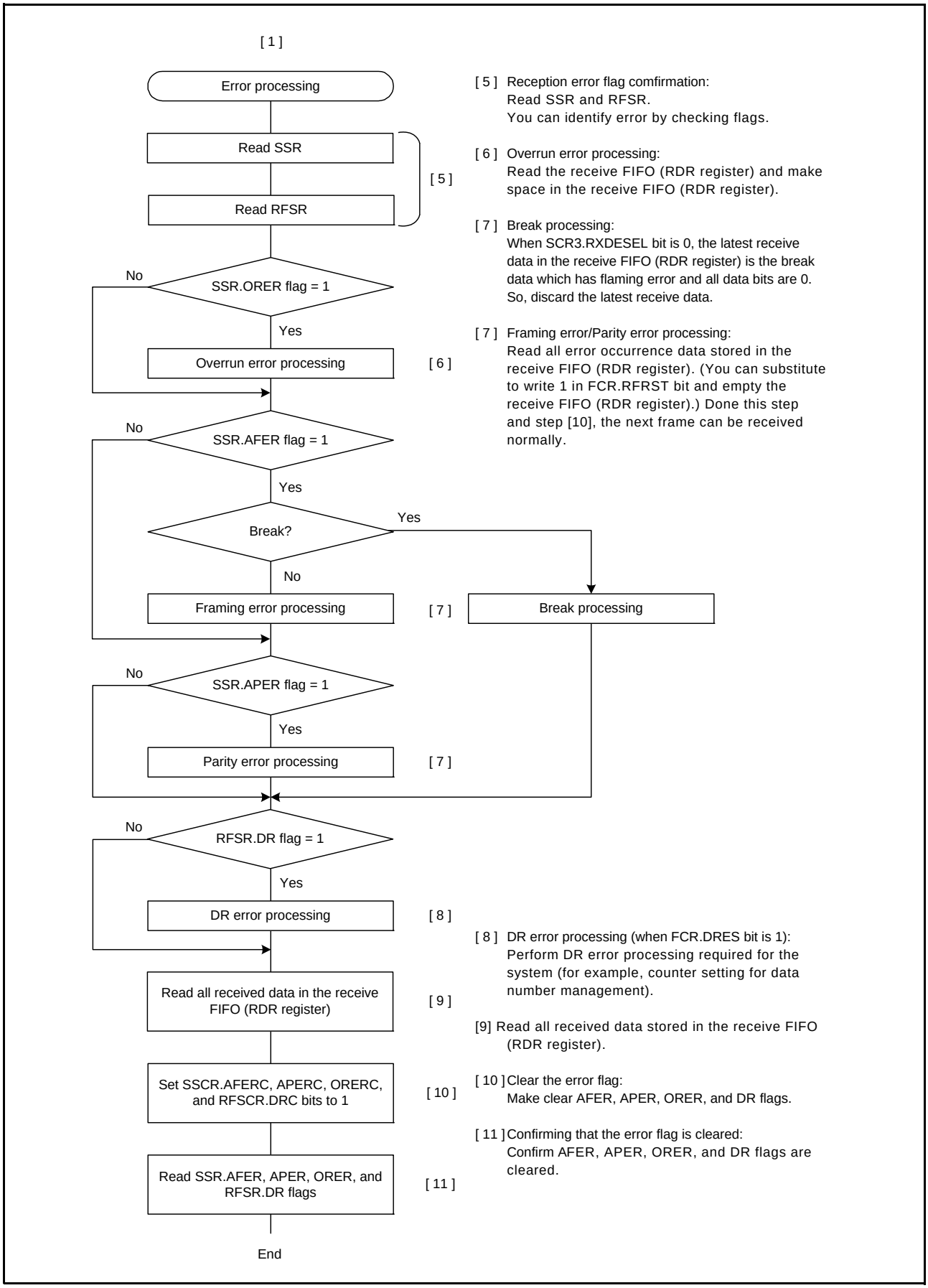


Figure 31.22 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (2)

31.3.10 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 31.23 shows an example of the sampling timing adjustment.

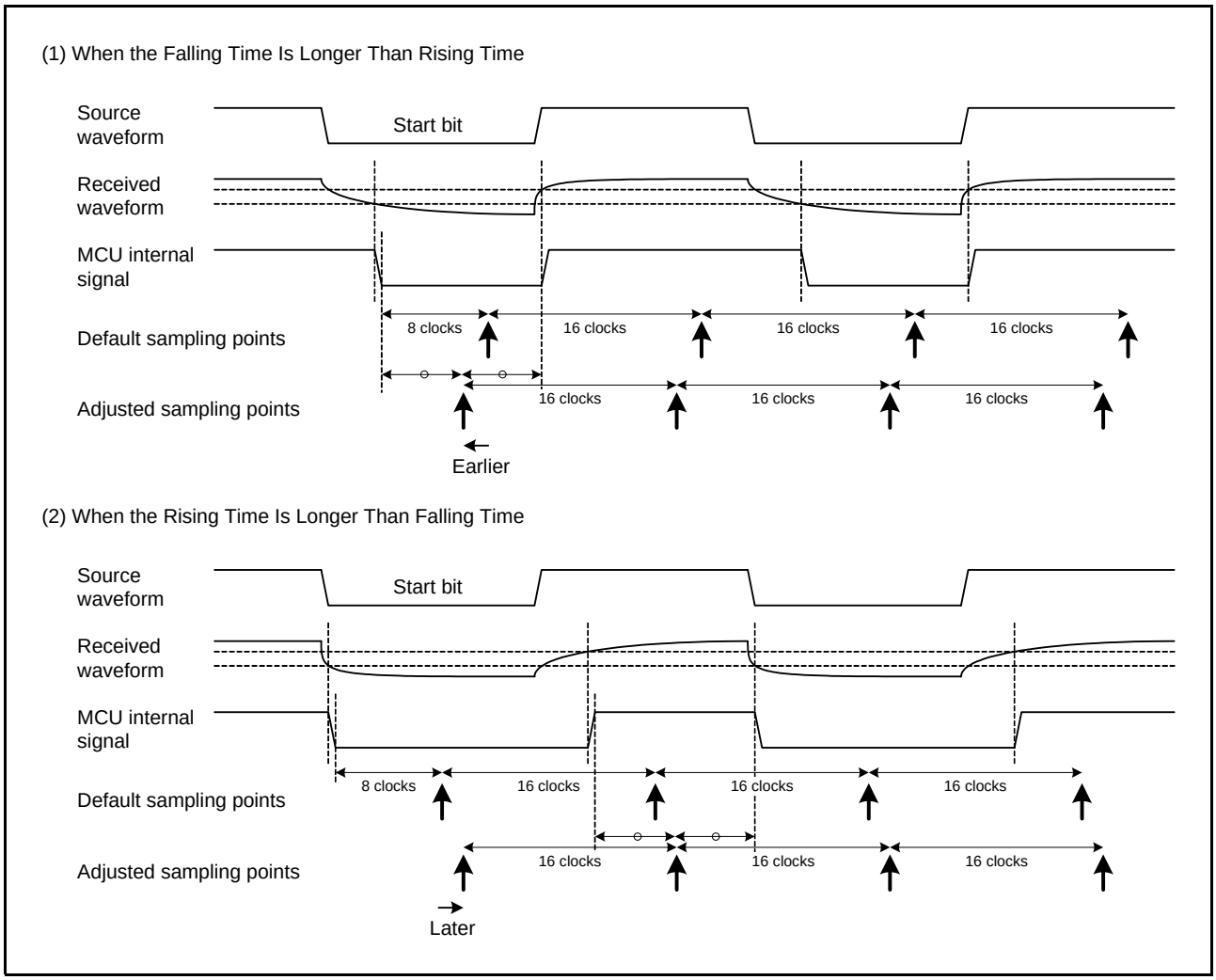


Figure 31.23 Example of Sampling Timing Adjustment (SCR2.ABCSE Bit = 0 and SCR2.ABCS Bit = 0)

31.3.11 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 31.24 shows an example of the transition timing adjustment.

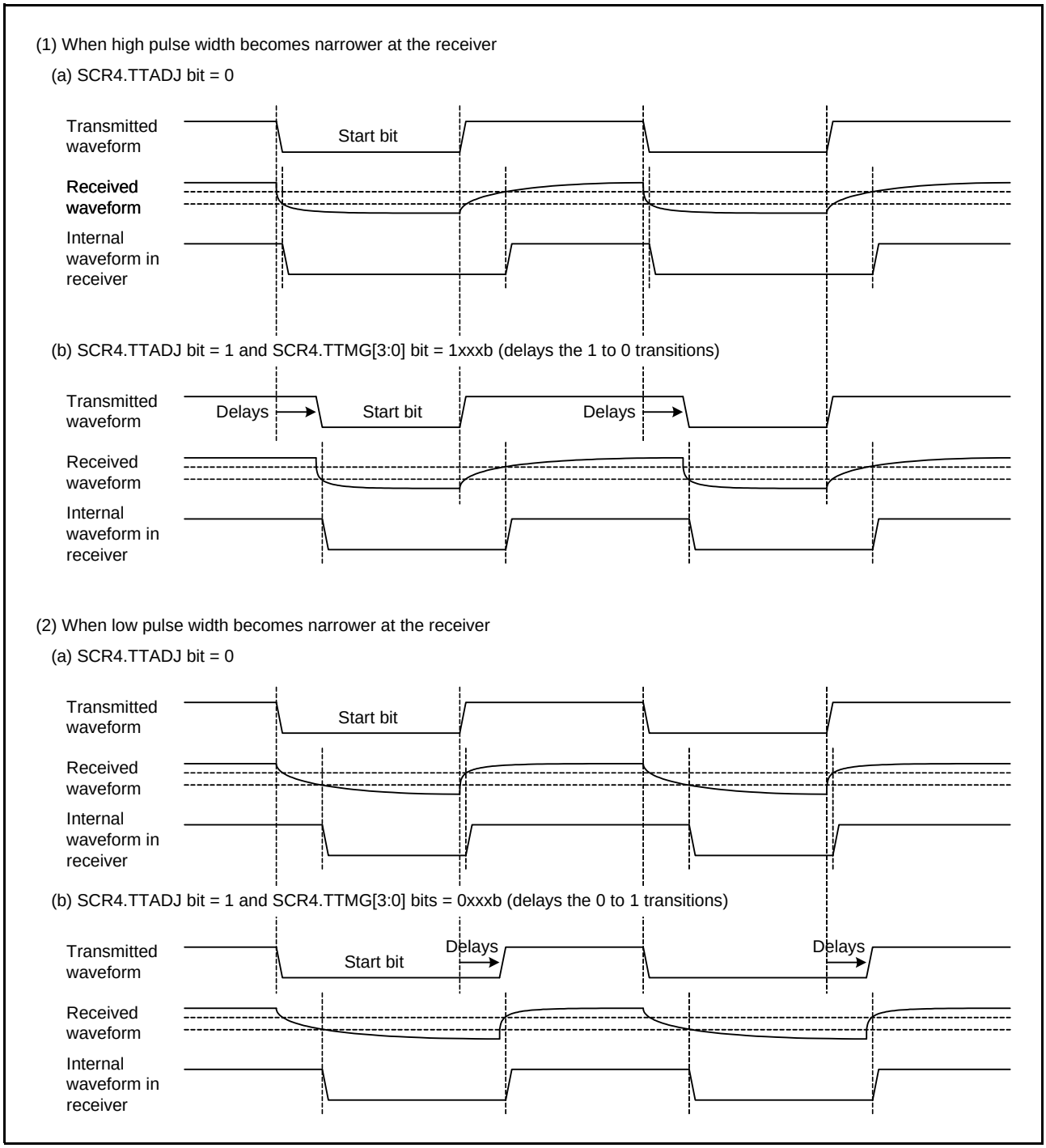


Figure 31.24 Example of Transition Timing Adjustment

31.4 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 31.25 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.

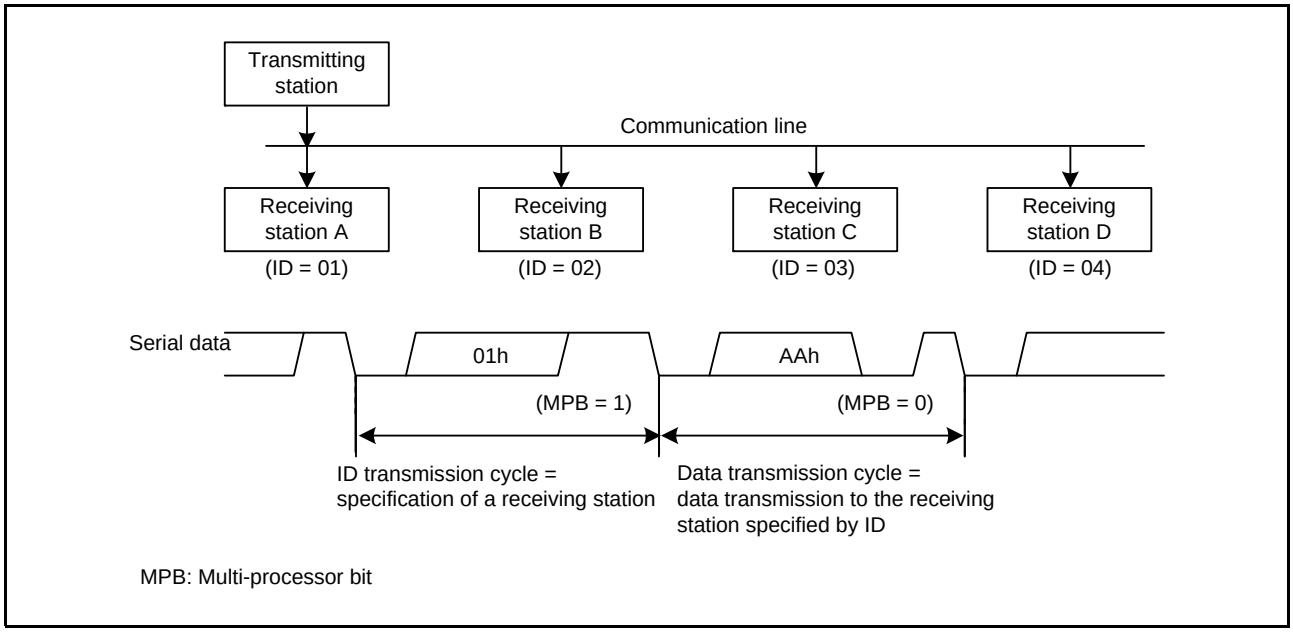


Figure 31.25 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

(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.

31.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO Mode

Figure 31.26 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.

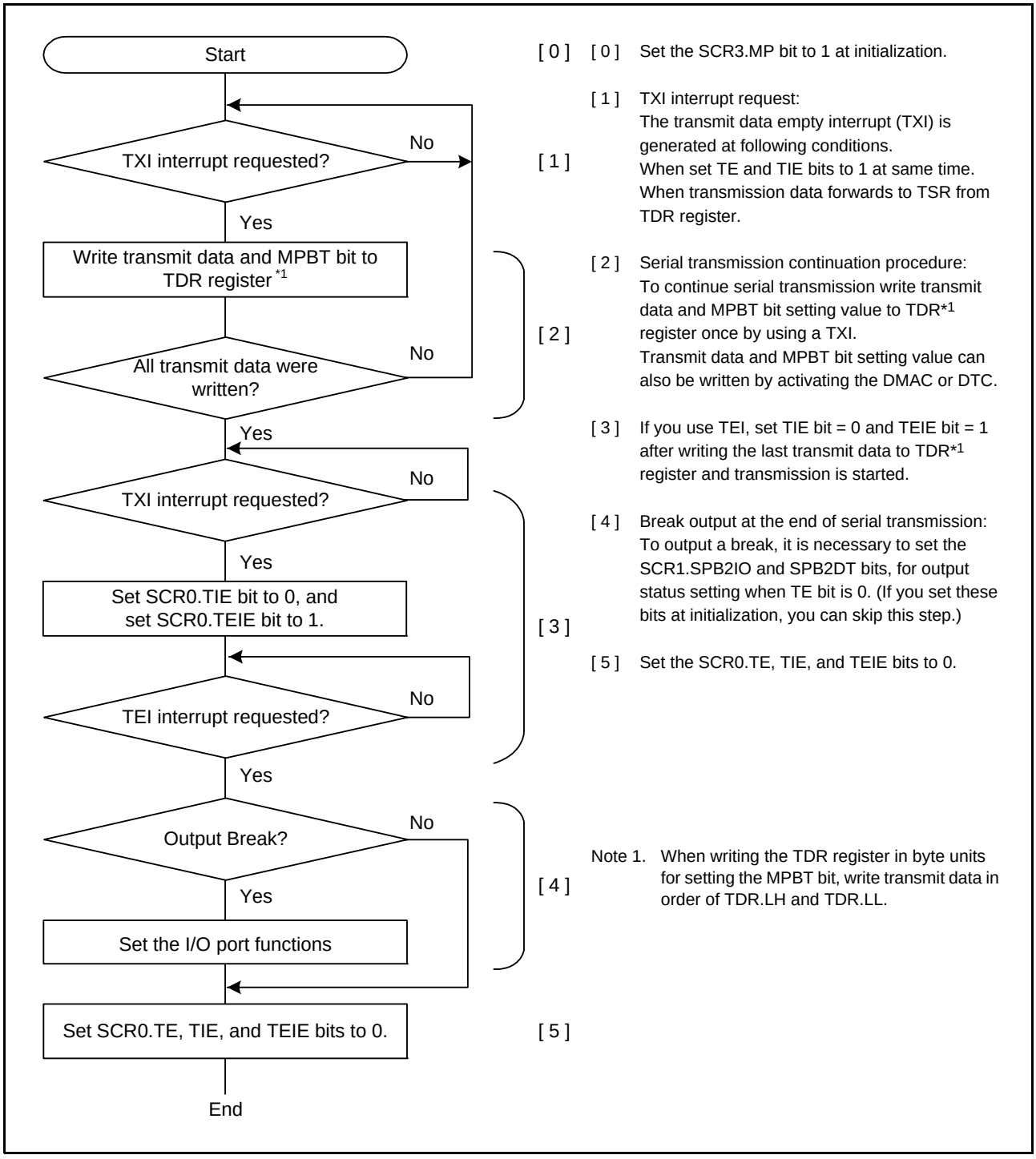


Figure 31.26 Example of Multi-Processor Serial Transmission Flowchart (Non-FIFO Mode)

(2) FIFO Mode

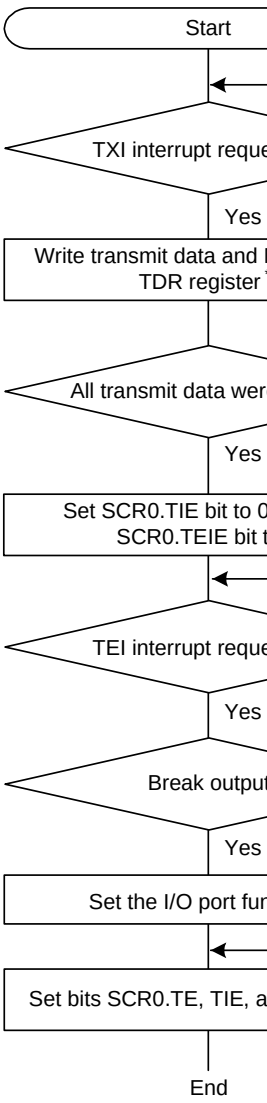
Table 31.32 shows an example of data format that is written to transmit FIFO (TDR register) in multi-processor mode. Write MPBT in bit9 of TDR. And write data to transmit FIFO (TDR register) corresponded to data length. It should write to 0 for unused bits.

Table 31.32 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 31.27 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 31.27 Example of Multi-Processor Serial Transmission Flowchart (FIFO Mode)

31.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO Mode

Figure 31.29 and Figure 31.30 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 31.28 is the example of operation for reception.

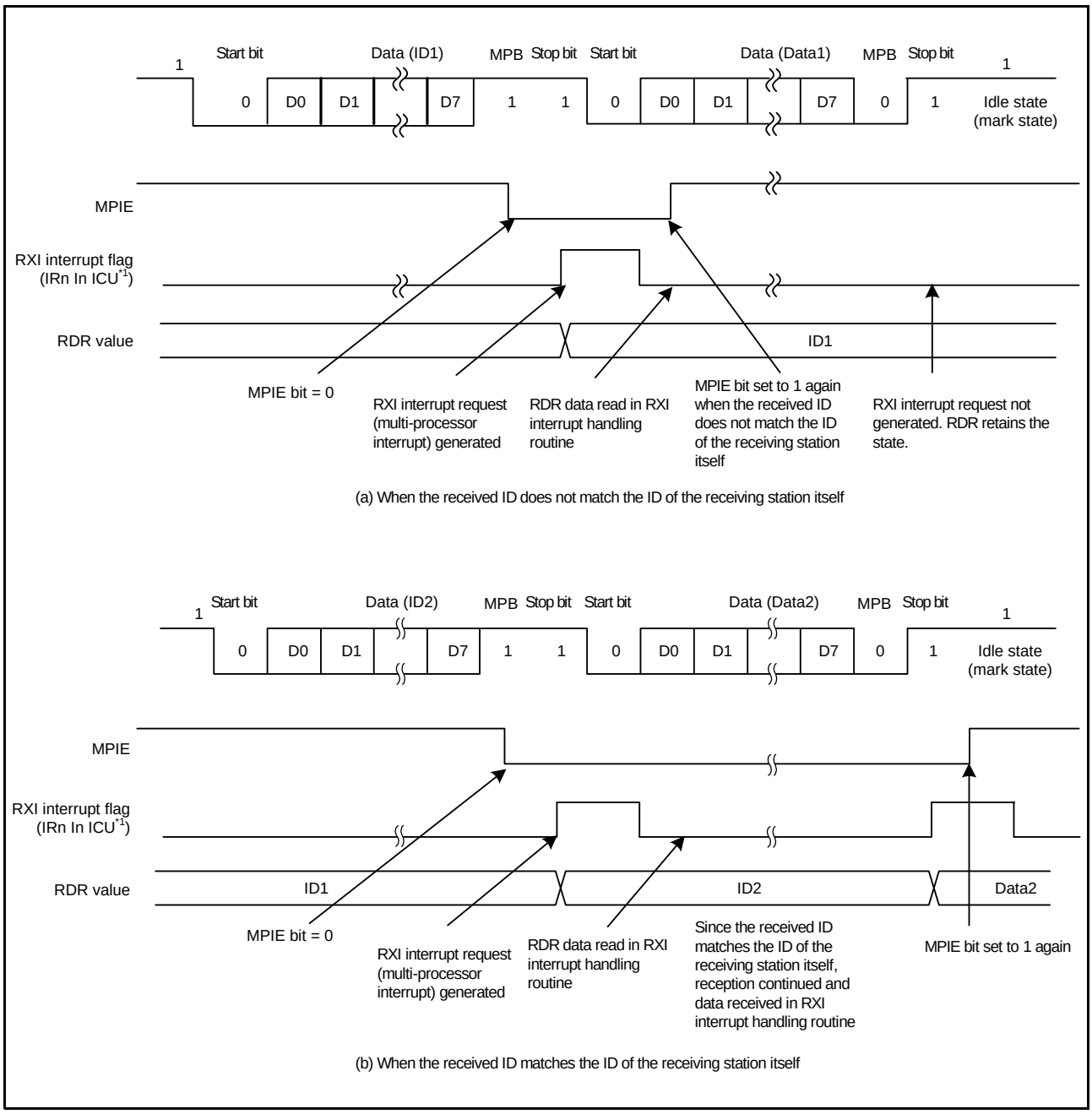
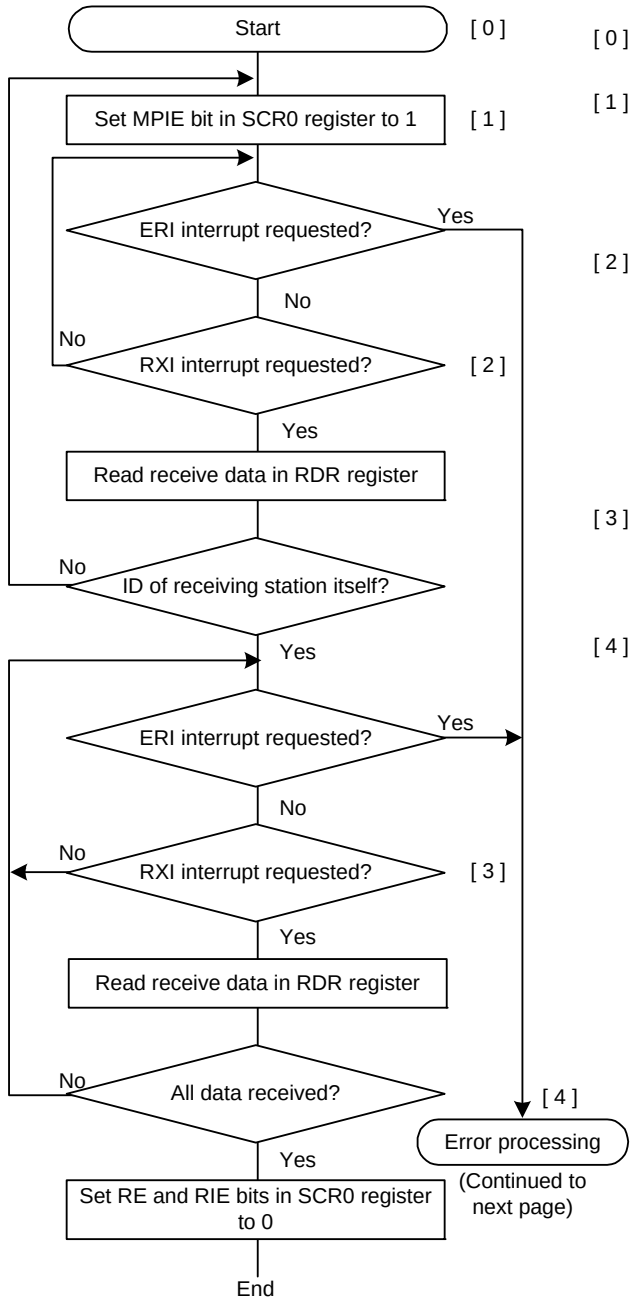


Figure 31.28 Example of RSCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)



[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 31.29 Example of Multi-Processor Serial Reception Flowchart (1) (Non-FIFO Mode)

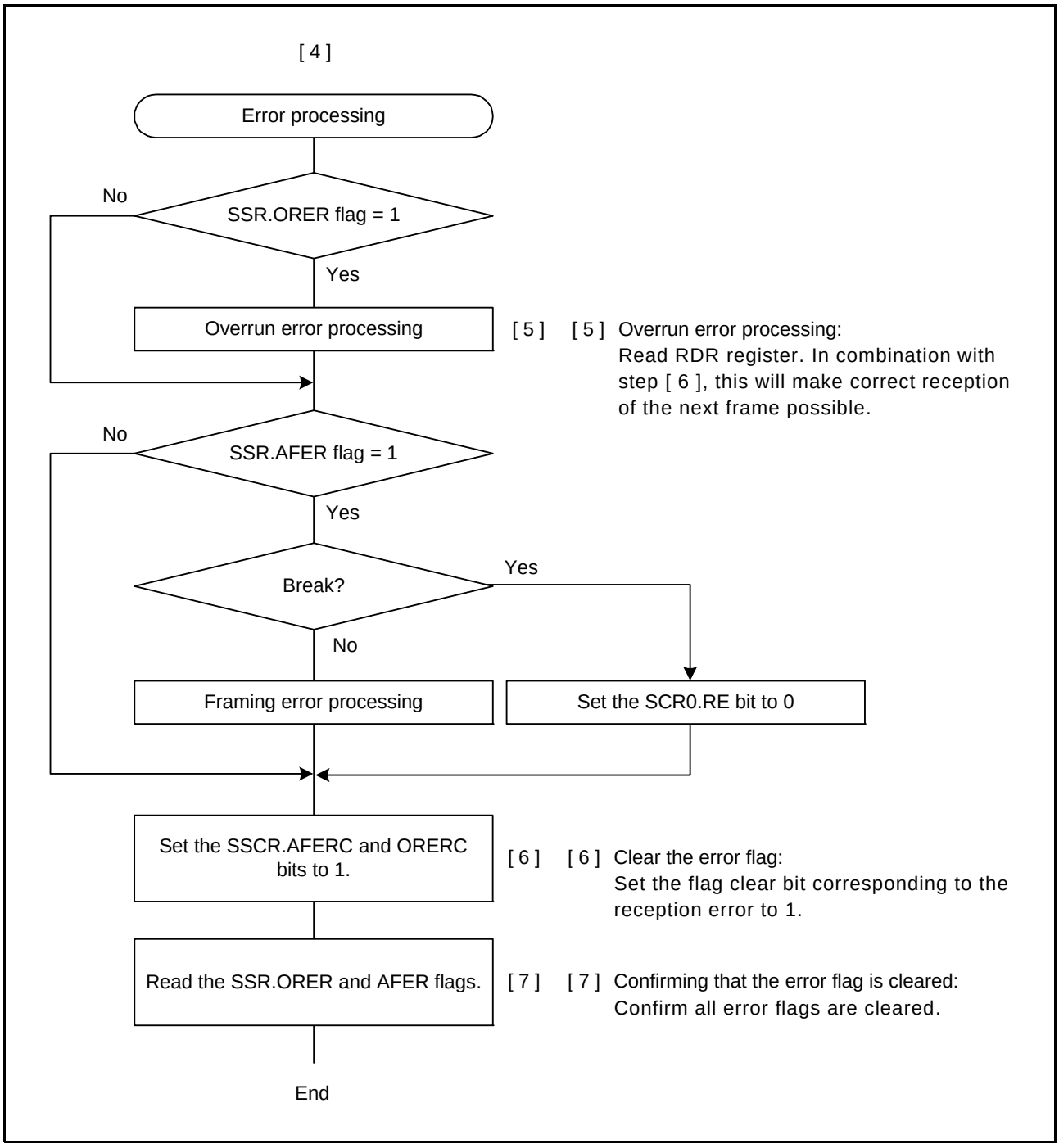


Figure 31.30 Example of Multi-Processor Serial Reception Flowchart (2) (Non-FIFO Mode)

(2) FIFO Mode

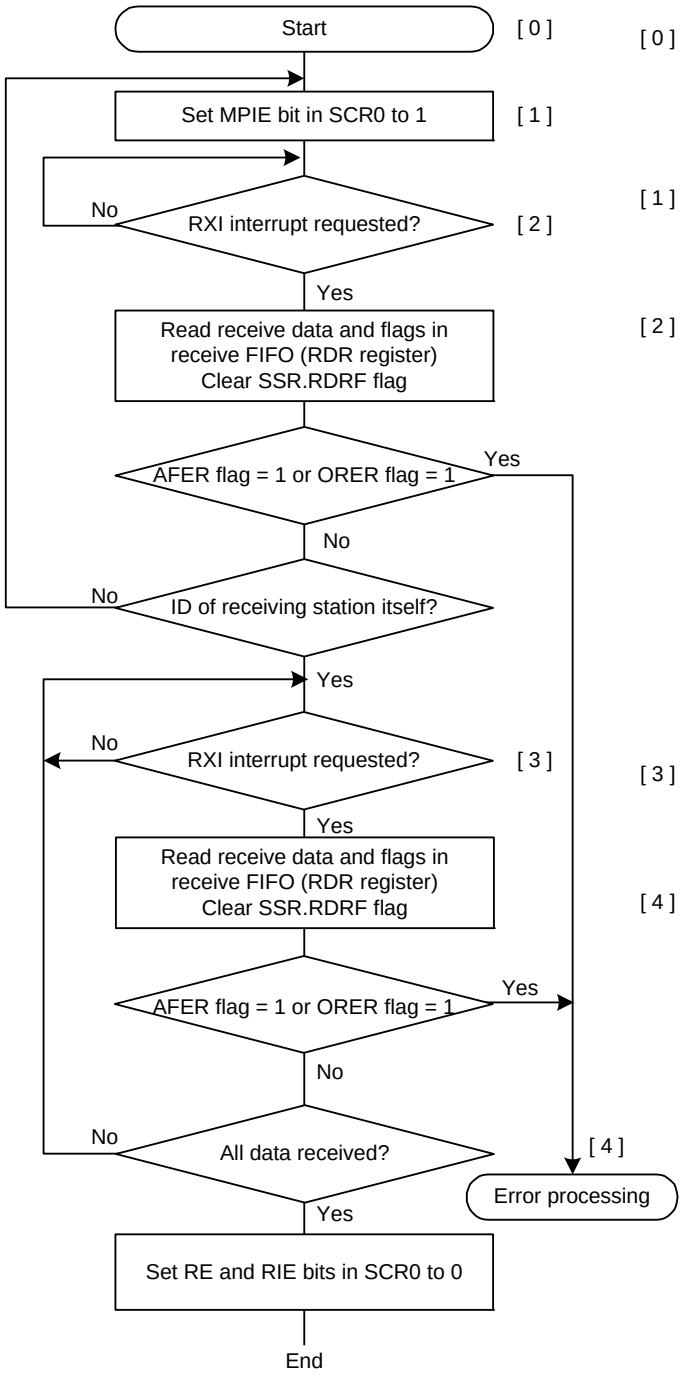
Table 31.33 shows an example of data format that is stored to receive FIFO (RDR register) in multi-processor mode. MPB flag is stored in bit9 of RDR register. 0 is stored to APER and PER flags. Data is stored to receive FIFO RDAT[8:0] bits corresponded to data length. It is stored to 0 for unused bits. 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 31.33 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 31.31 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 31.31 Example Flowchart of Serial Reception in Multi-Processor Mode (1) (FIFO Mode)

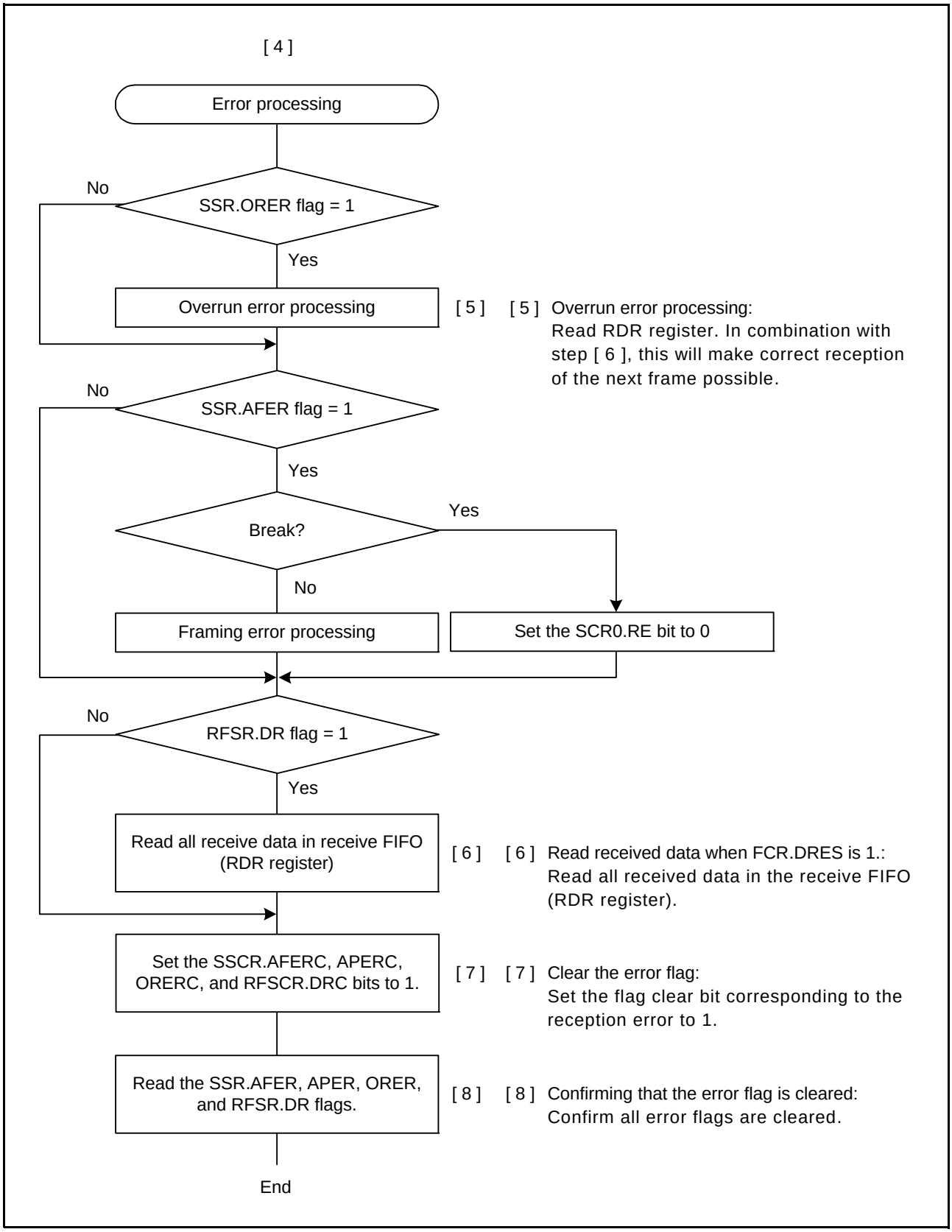


Figure 31.32 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO Mode)

31.5 Manchester Mode

In manchester mode, the transmit or receive serial data is coded in Manchester encoding.
Figure 31.33 shows the conceptual image of Manchester encoding.

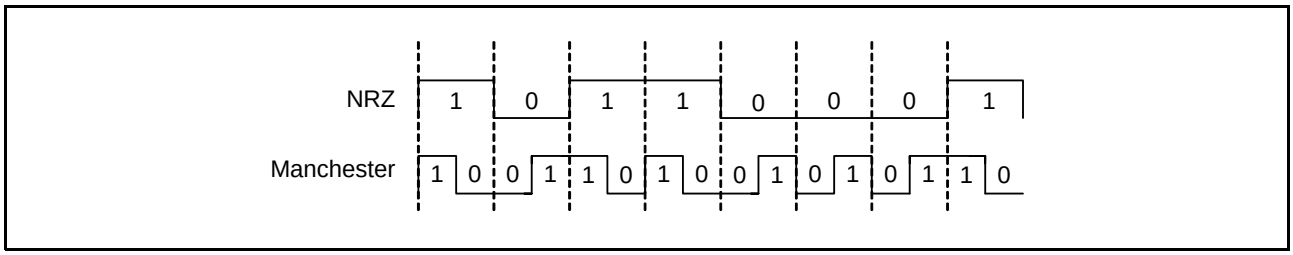


Figure 31.33 Example of Manchester Encoding

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 31.5.1, Frame Format.

31.5.1 Frame Format

Figure 31.34 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.

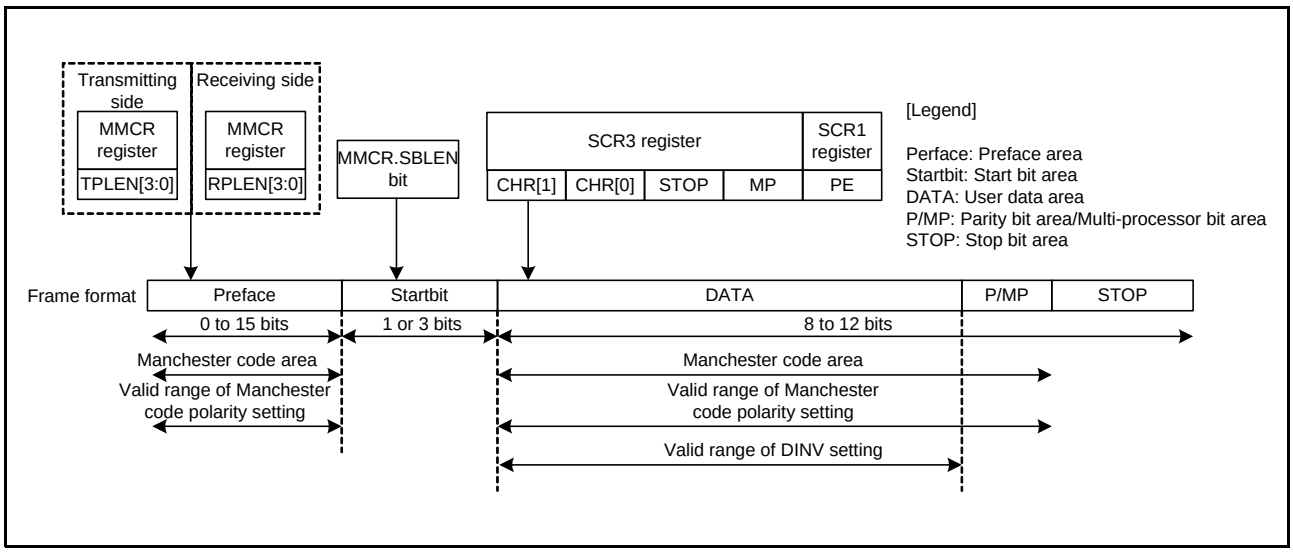


Figure 31.34 Frame Format in Manchester Mode

(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 31.35 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

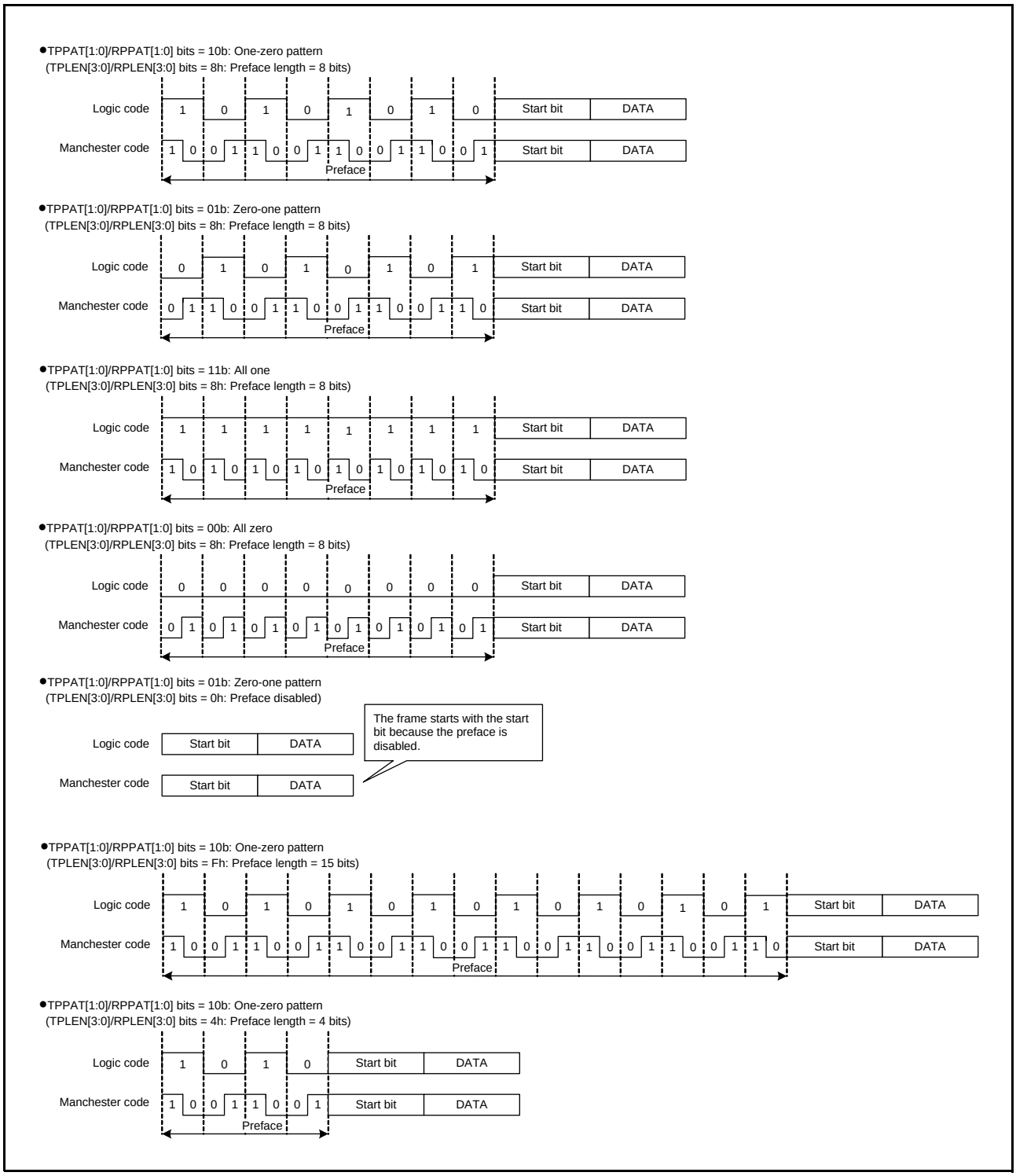
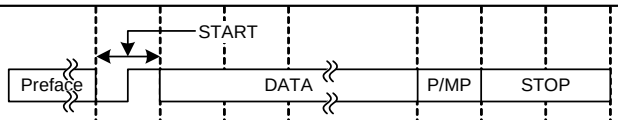
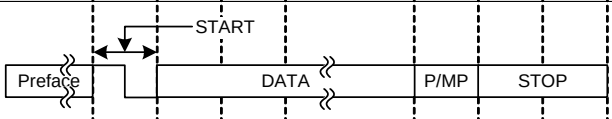
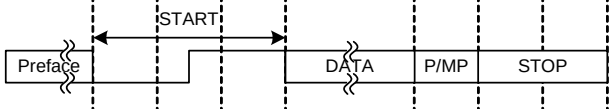
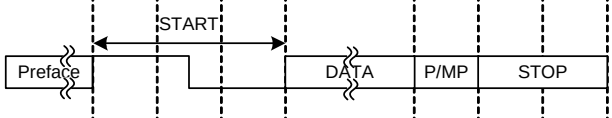
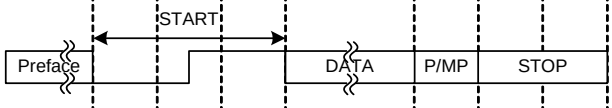
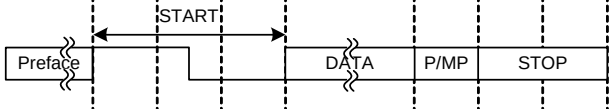


Figure 31.35 Preface Pattern Setting Example

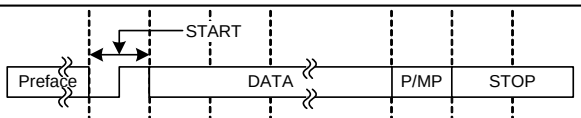
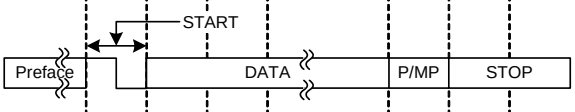
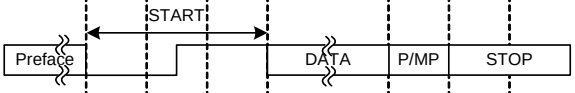
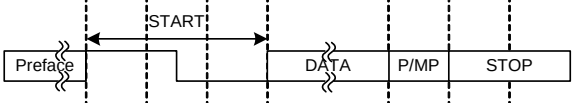
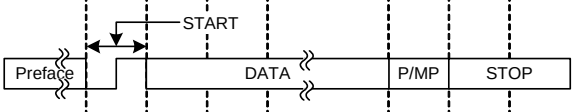
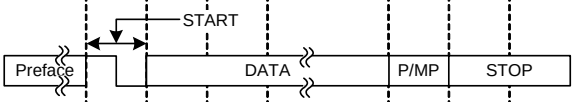
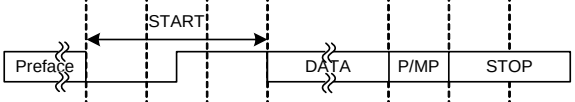
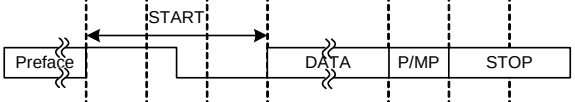
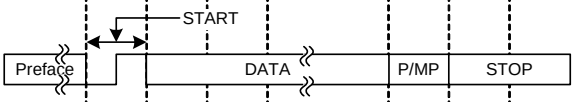
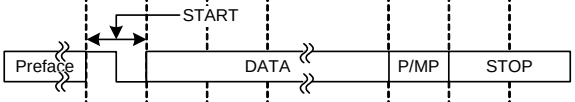
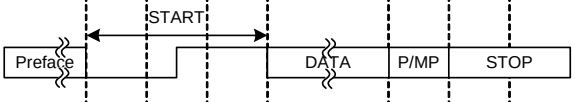
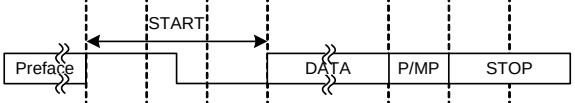
(2) Start Bit Area

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 31.36 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 31.37 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 31.36 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 31.37 Settings Related to and Judgment of the Start Bit Area at Reception

(3) Data

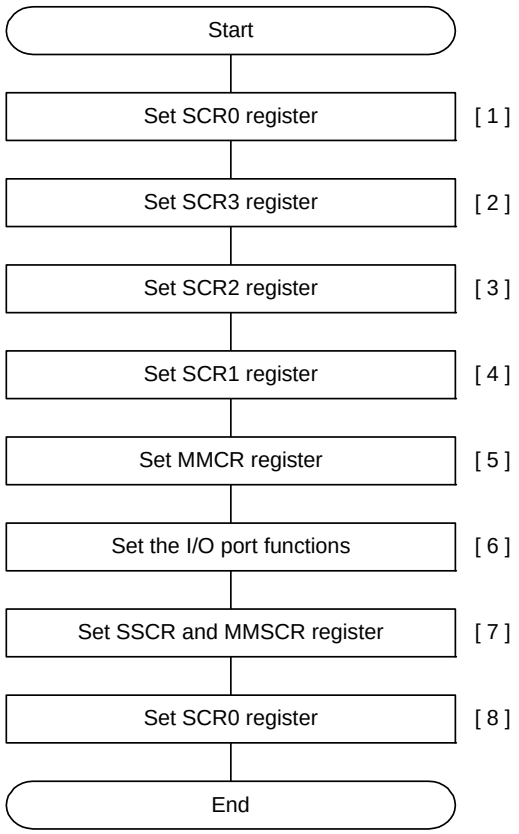
Since the format of the data area is the same as that of the asynchronous mode, see section 31.3.1, Serial Data Transfer Format.
As shown in Figure 31.34, Frame Format in Manchester Mode, the stop bit is not included in the manchester encoding range.

31.5.2 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.

31.5.3 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 31.38.
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 31.38 RSCI Initialization Flow in Manchester Mode

31.5.4 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.

31.5.5 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.

31.5.6 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 31.33)
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 31.5.1, Frame Format.
Figure 31.39 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 31.40 to Figure 31.42 show examples of the operation for serial transmission in manchester mode.

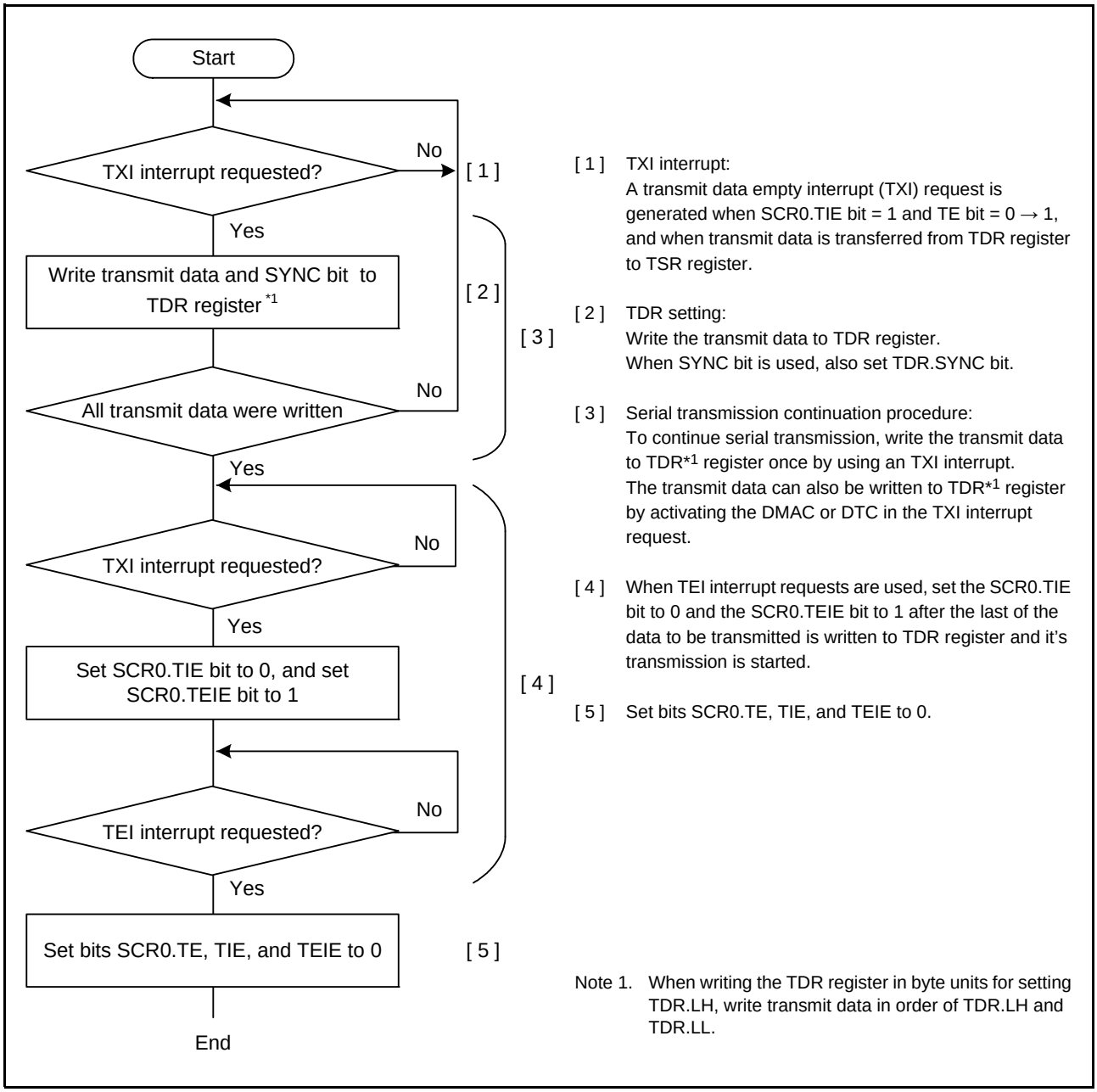
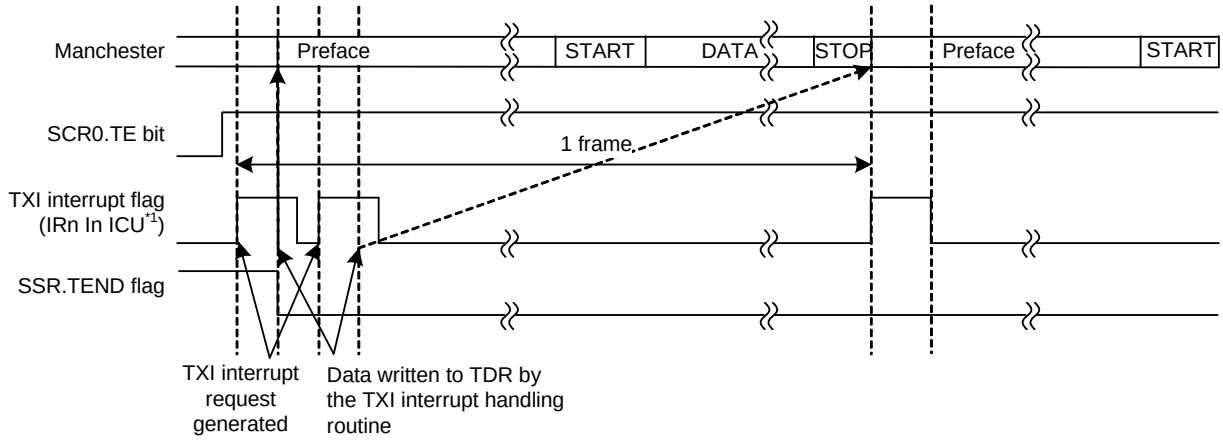
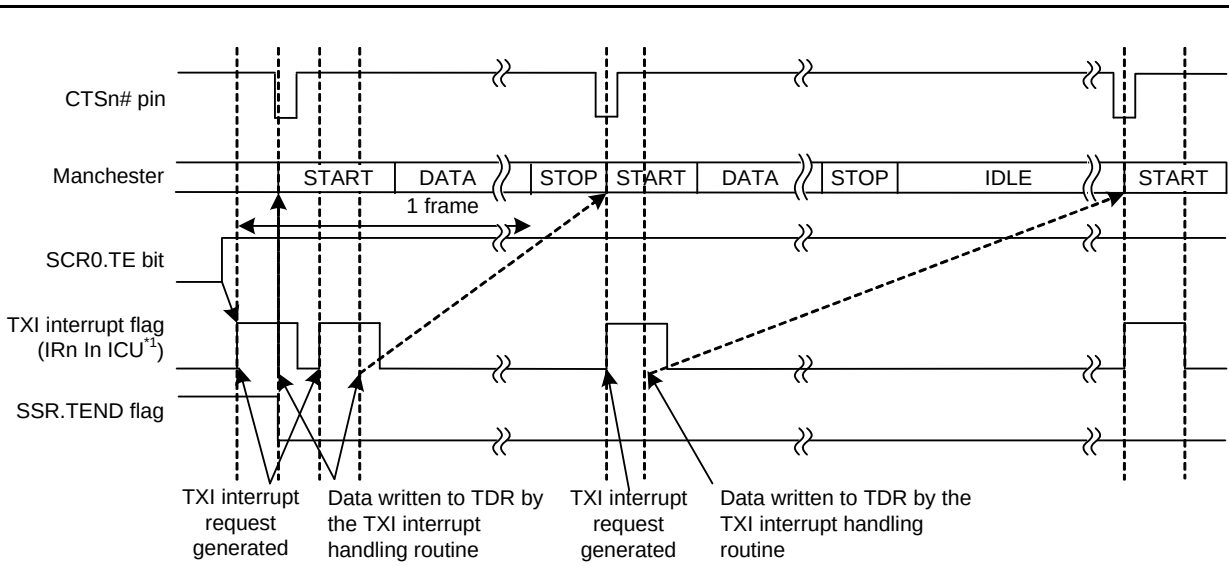


Figure 31.39 Serial Transmission Flowchart in Manchester Mode



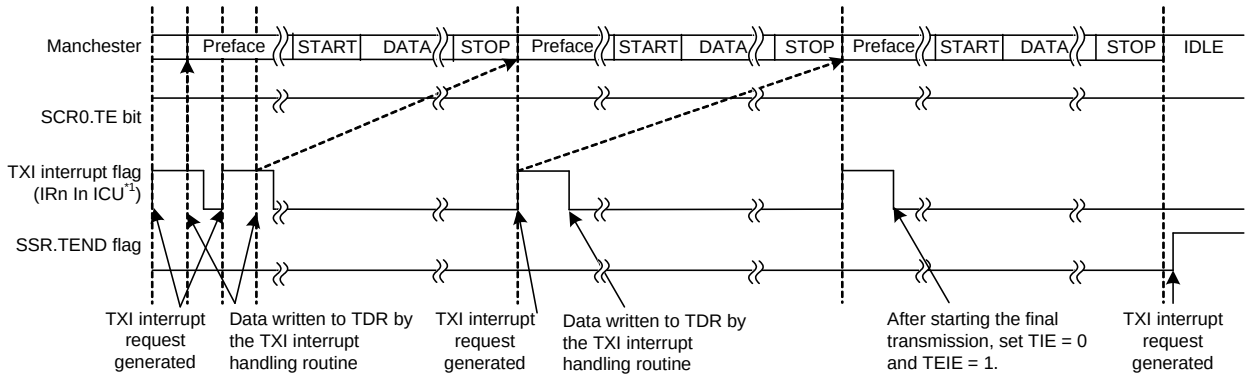
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.40 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)



Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.41 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (without Preface but with the CTS Function)



Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.42 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)

31.5.7 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 31.43, 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.

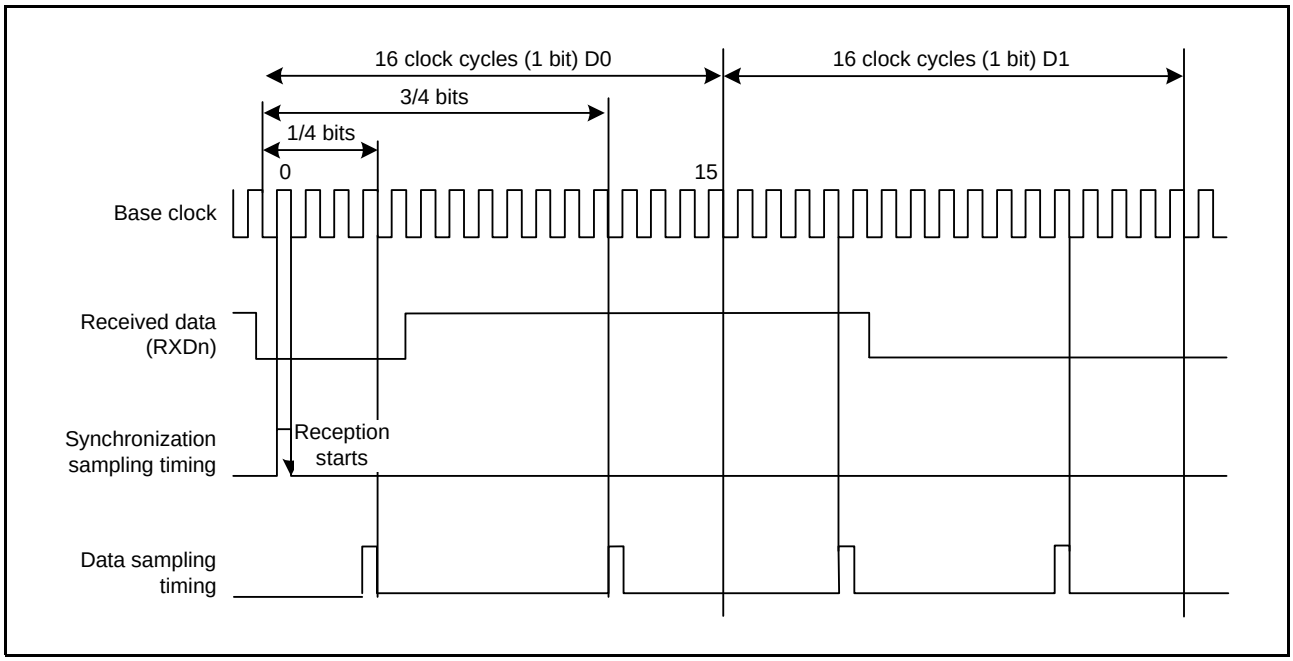


Figure 31.43 Data Reception Sampling Timing in Manchester Mode

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.

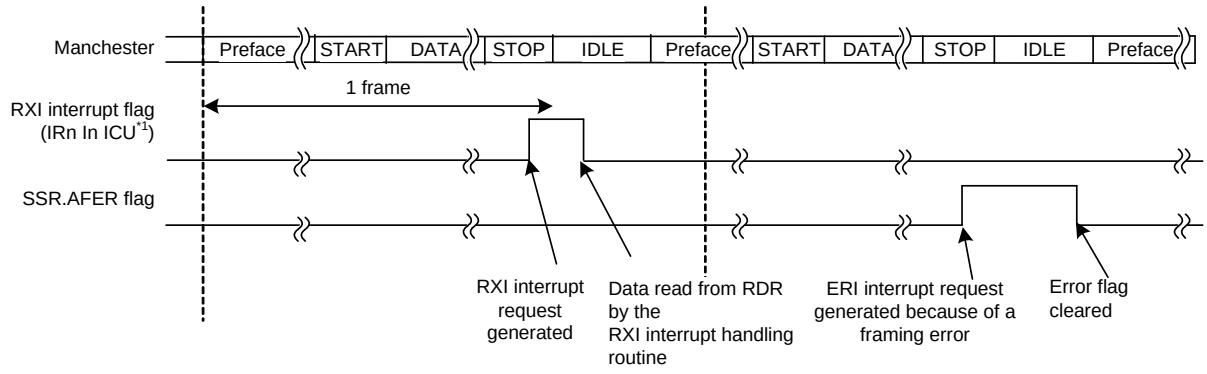
Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

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 31.5.11, 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 31.44 shows an example of the operation for serial data reception in manchester mode.

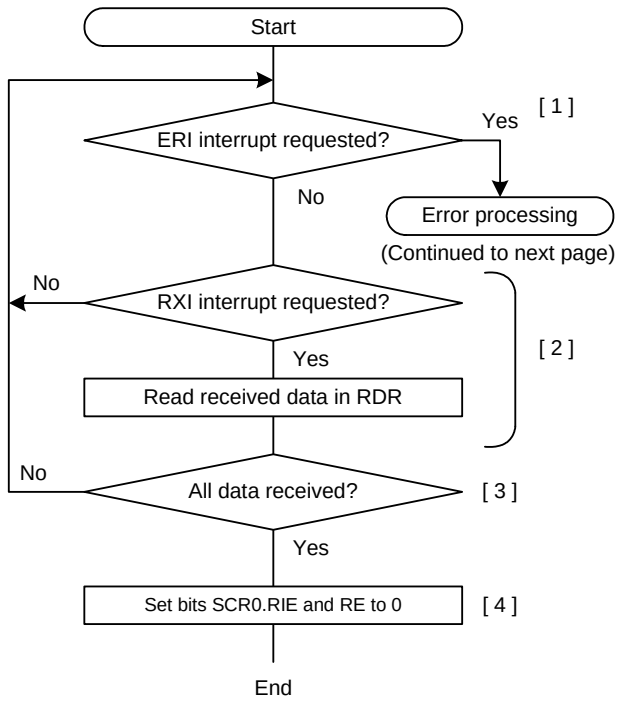


Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.44 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 31.5.11, 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 31.45 and Figure 31.46 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 31.45 Example of Serial Data Reception in Manchester Mode (Normal)

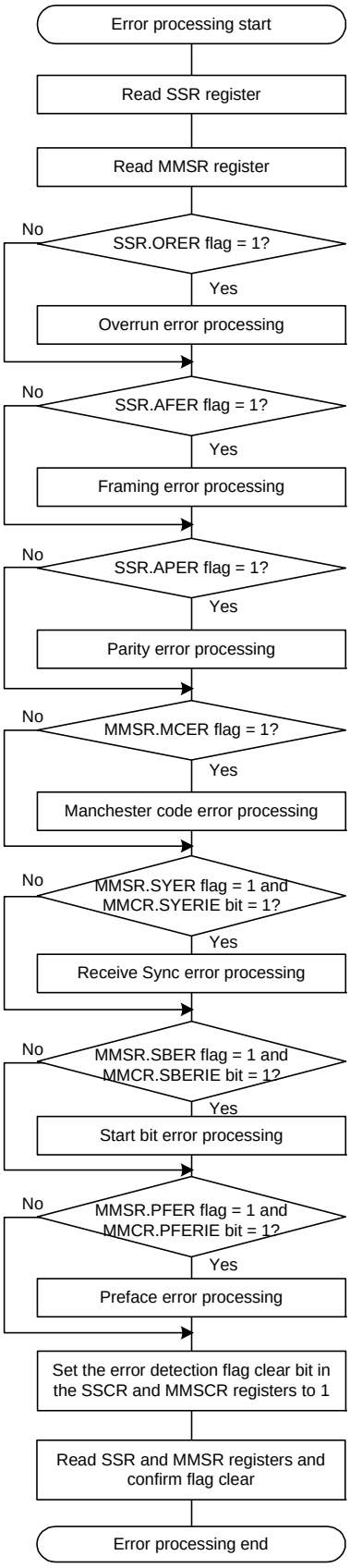


Figure 31.46 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

31.5.8 Operation When Multi-Processor Bit is Used

See section 31.4, 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 31.46 for error processing in manchester mode for the reception flowchart (Figure 31.29). Refer to Table 31.36 for the operation status when detecting various errors.

31.5.9 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 31.47 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.

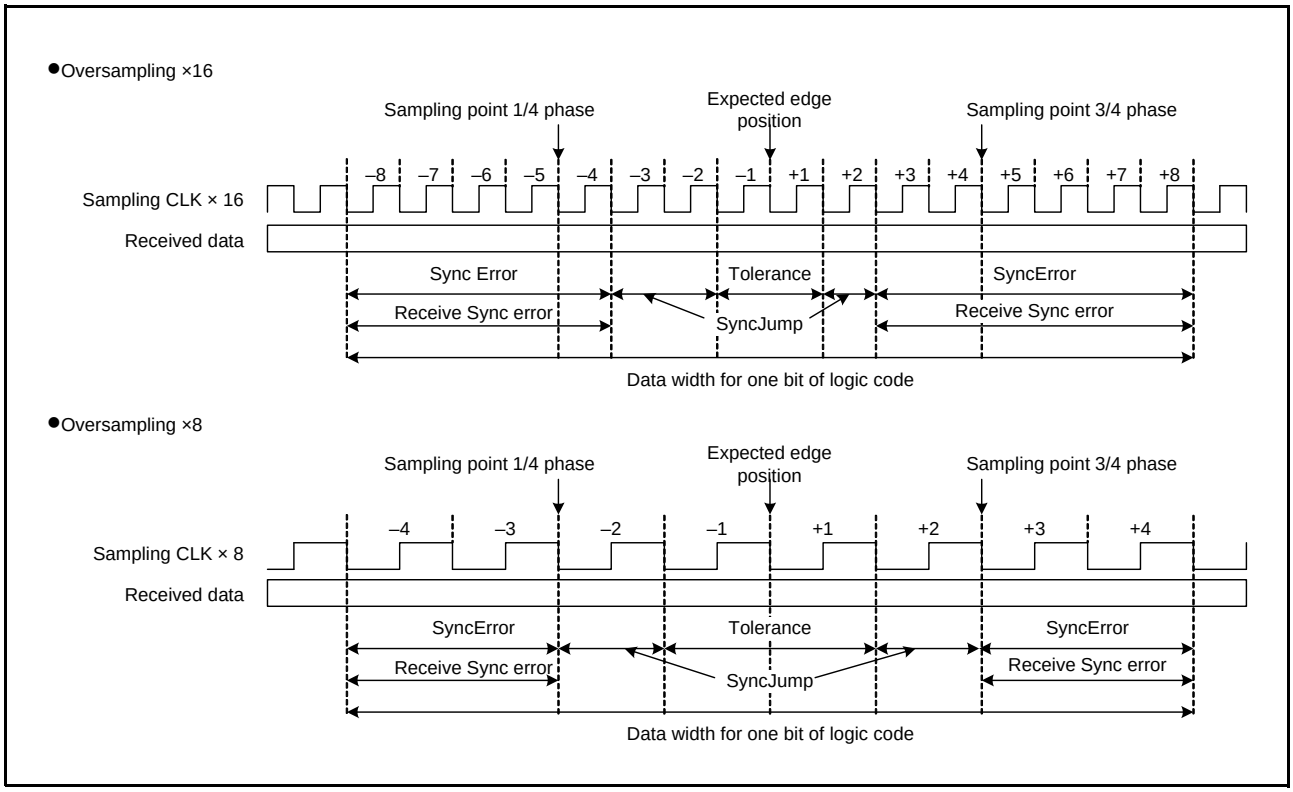


Figure 31.47 Conceptual Image of Reception Retiming Range

31.5.10 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 31.48 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 31.5.1, Frame Format (2).

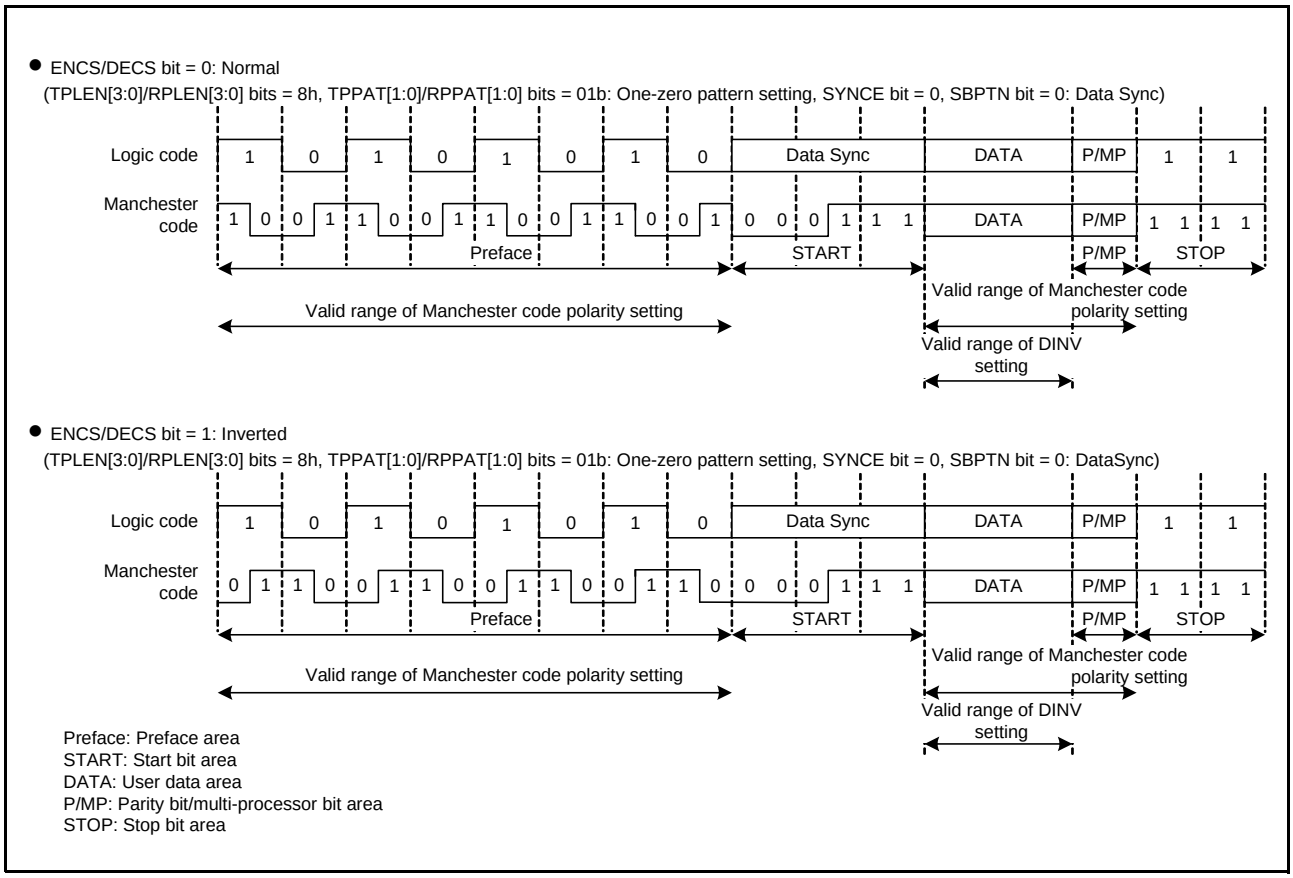


Figure 31.48 Valid Range of the Manchester Code Polarity Setting

31.5.11 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 31.3.9, 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 31.34 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR register.
Table 31.35 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 31.36 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. 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. 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, 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. 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, 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 31.5.9, Receive Retiming is enabled, the receive retiming operation is performed.
If no edges are detected within the receive retiming range (SyncError area in Figure 31.47) 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. 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, 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 31.34 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 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 is generated by SYER factor.
Note 3. If MMCR.PFERIE bit = 1 or MMCR.SBERIE bit = 1, an ERI interrupt is generated when the corresponding flag is set.

Table 31.35 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 31.5.11, Errors in Manchester Mode (7)

Table 31.36 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
	Preface	Start Bit	Data	Parity Bit	Stop Bit						
No error	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Not output
						1					Output
	No error	SBER, No SYER*1	—	—	—	—	0	—	Lost	Set SBER*1	Not output
							1				Output
	SYER, No PFER	No error	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output
								1	Lost		Output
	No error	SYER, No SBER	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output
								1	Lost		Output
	No error	No error	SYER		No error	—	—	0	Transfer to RDR	Set SYER	Not output
								1			Output
	No error	No error	MCER		No error	—	—	—	Transfer to RDR	Set MCER	Output
	No error	No error	—	APER	No error	—	—	—	Transfer to RDR	Set APER	Output
	No error	No error	—	—	AFER	—	—	—	Transfer to RDR	Set AFER	Output
	There is some error				ORER	—	—	—	Lost	Set some flags*2	Output
	No error	No error	No error	No error	No error, ORER	—	—	—	Lost	Set ORER	Output
Some error*3, *5	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Output*4
						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. For MMSR.PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 31.37 Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 1)

MPB*1	Each Area of the Frame					PFE RIE	SBE RIE	SYE RIE	Received Data	Error Flag	Interrupt Request
	Preface	Start Bit	Data	Parity Bit	Stop Bit						
1	No error	No error	—	—	—	—	—	—	Transfer to RDR	Set some flags	Output
	No PFER, SYER	No SBER, SYER	—	—	—	—	—	0			
								1	Lost	don't set any flags	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 is output, and if it is detected, ERI interrupt request 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.

31.6 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.

31.6.1 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.

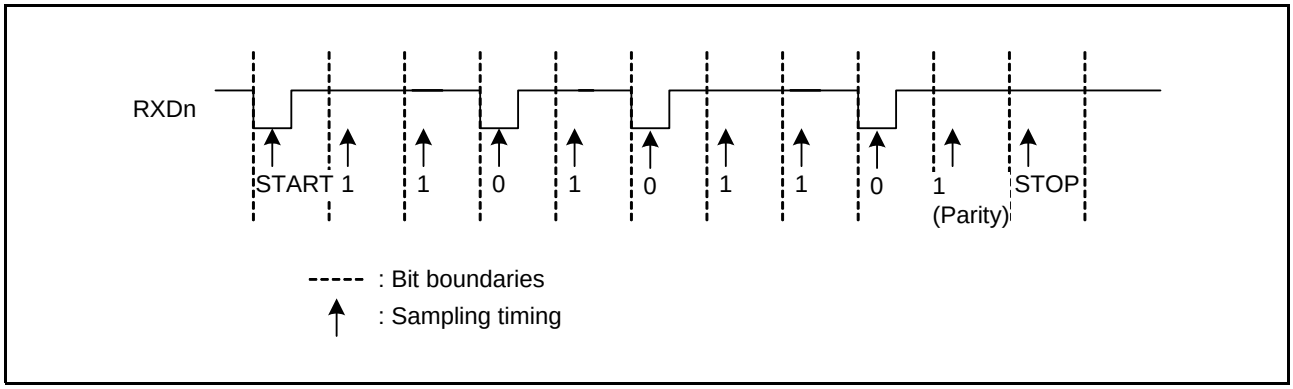


Figure 31.49 HBS Support Mode Reception Timing Chart

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 31.3.2, 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.

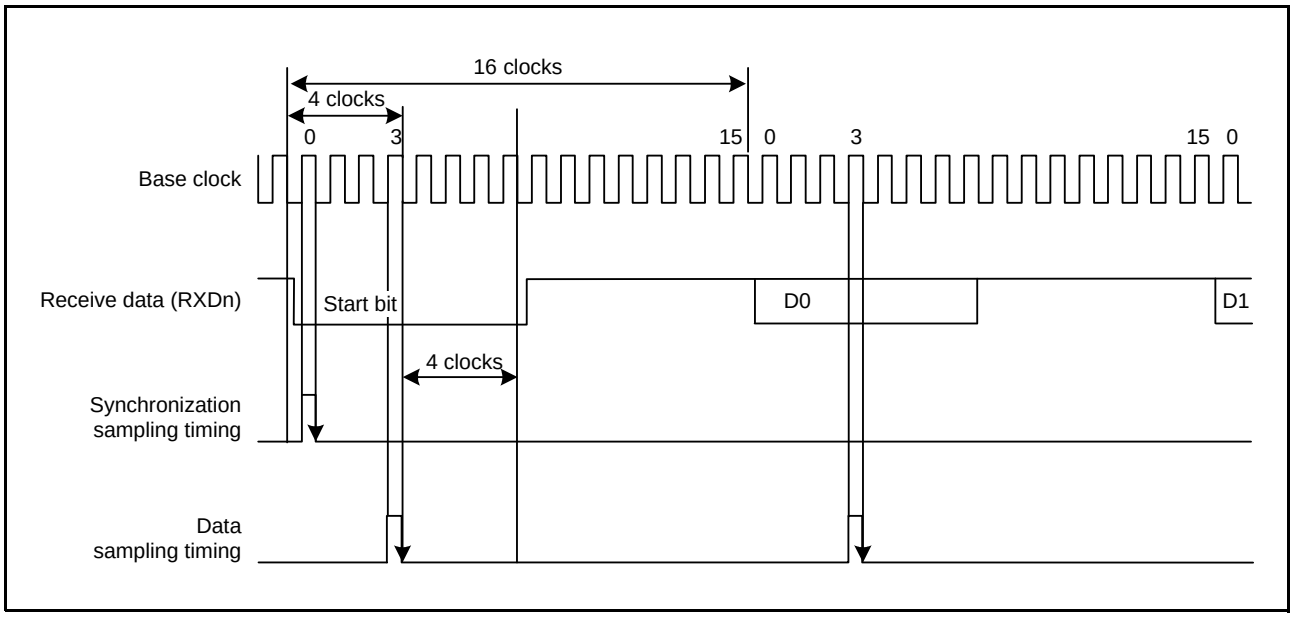


Figure 31.50 Details of Reception Sampling Timing in HBS Support Mode

31.6.2 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.

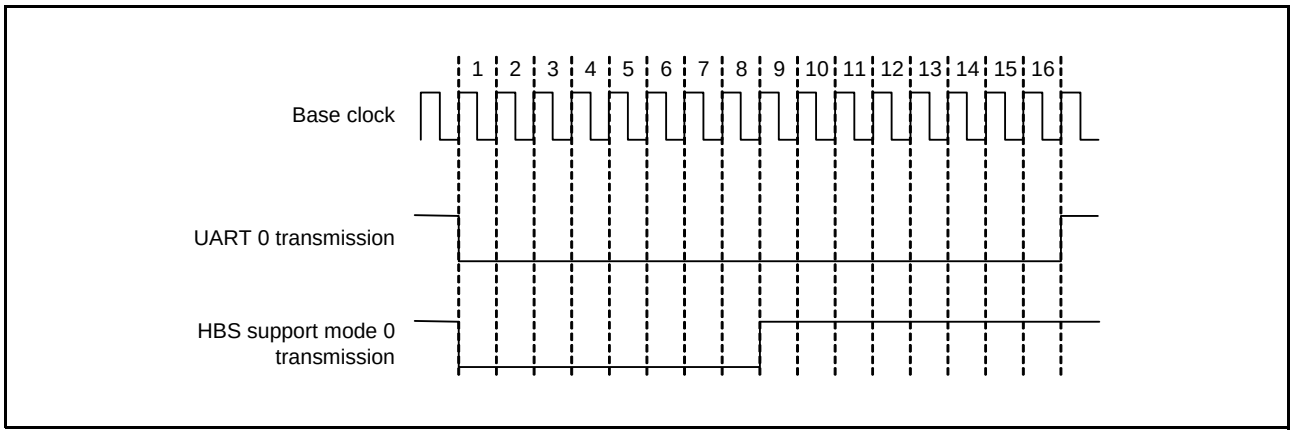


Figure 31.51 Transmission Waveform of HBS Support Mode

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.

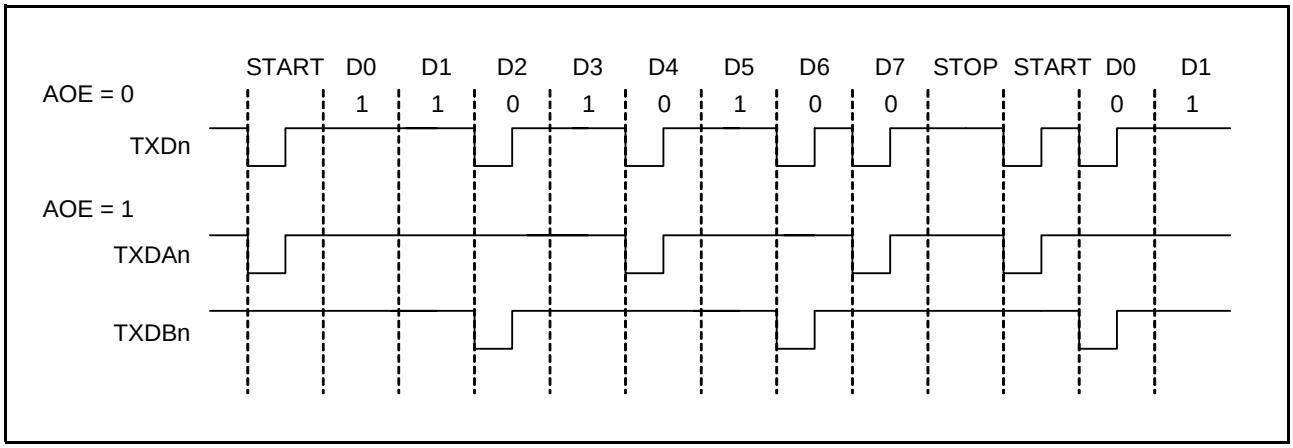


Figure 31.52 Difference in Transmission Waveform Depending on AOE Bit (When LPS Bit = 0)

Figure 31.52 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.

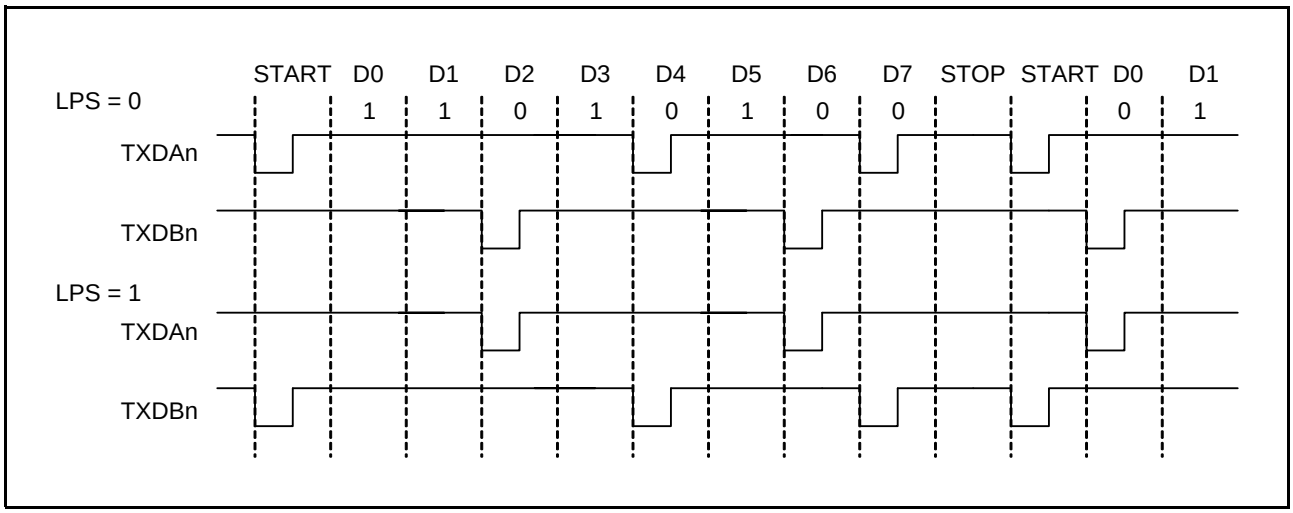


Figure 31.53 Difference in Transmission Waveform Depending on LPS Bit (When AOE Bit = 1)

Figure 31.53 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.

31.6.3 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 31.38 before use. Register bits not described can be set in the same way as the asynchronous mode.

Table 31.38 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.

31.7 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.

31.7.1 Sample Connection

Figure 31.54 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.

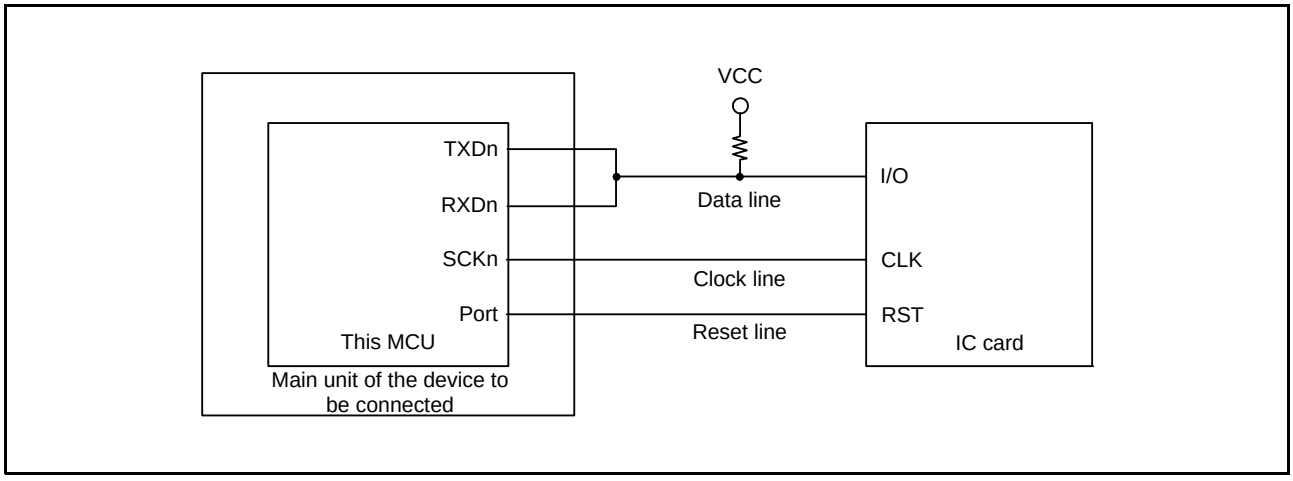


Figure 31.54 Sample Connection with a Smart Card (IC Card)

31.7.2 Data Format (Except in Block Transfer Mode)

Figure 31.55 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.

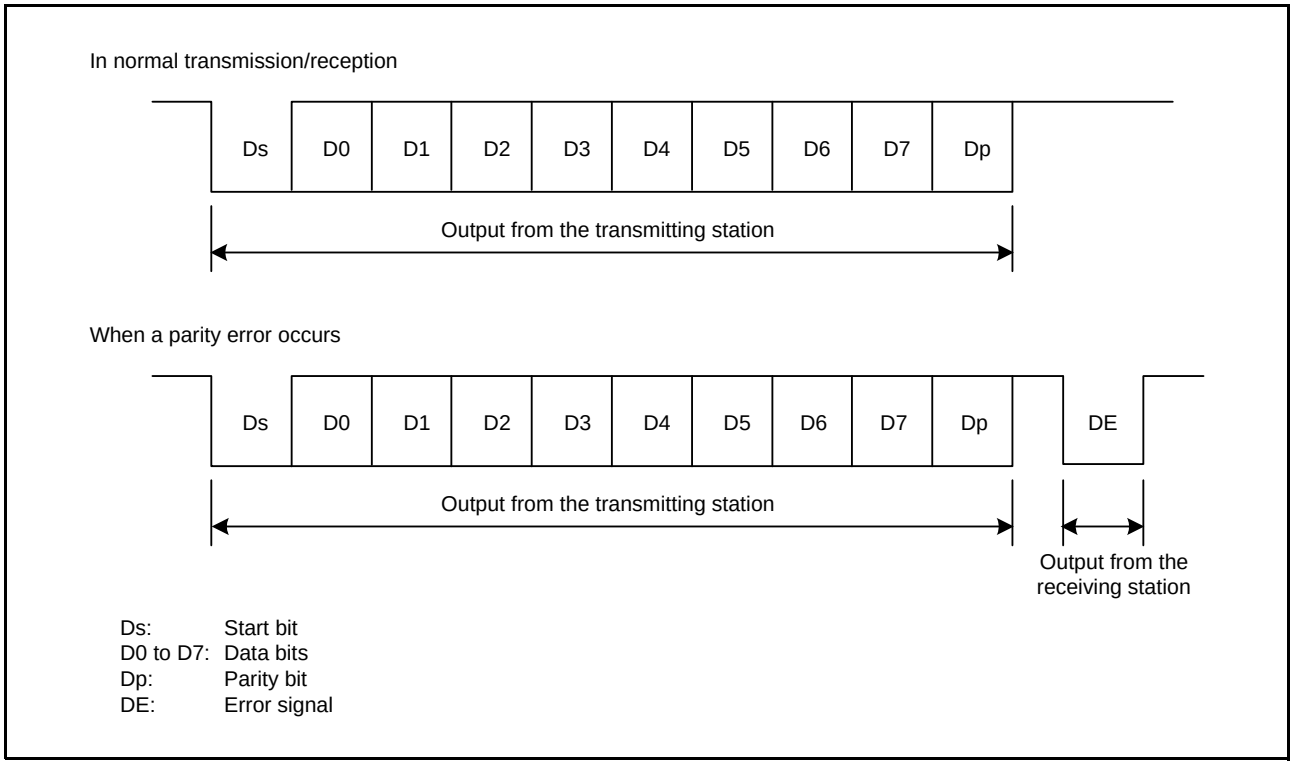


Figure 31.55 Data Formats in Smart Card Interface Mode

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 31.56. 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.

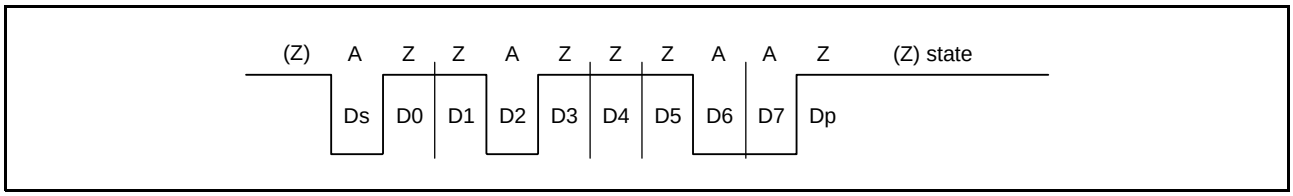


Figure 31.56 Direct Convention (SCR3.DDIR Bit = 1, SCR3.DINV Bit = 0, SCR1.PM Bit = 0)

(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 31.57. 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.

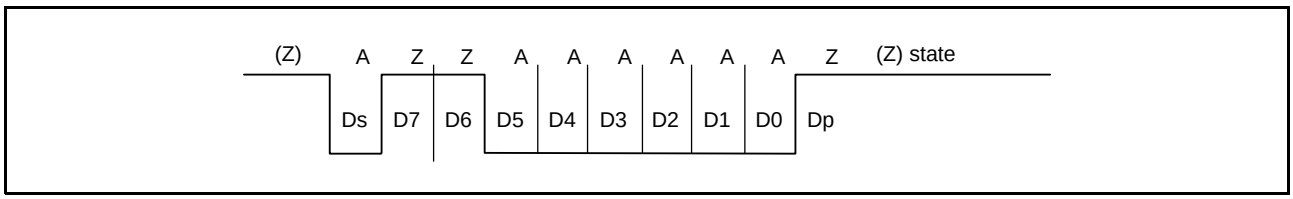


Figure 31.57 Inverse Convention (SCR3.DDIR Bit = 0, SCR3.DINV Bit = 1, SCR1.PM Bit = 1)

31.7.3 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.

31.7.4 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 31.58. 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 (%)

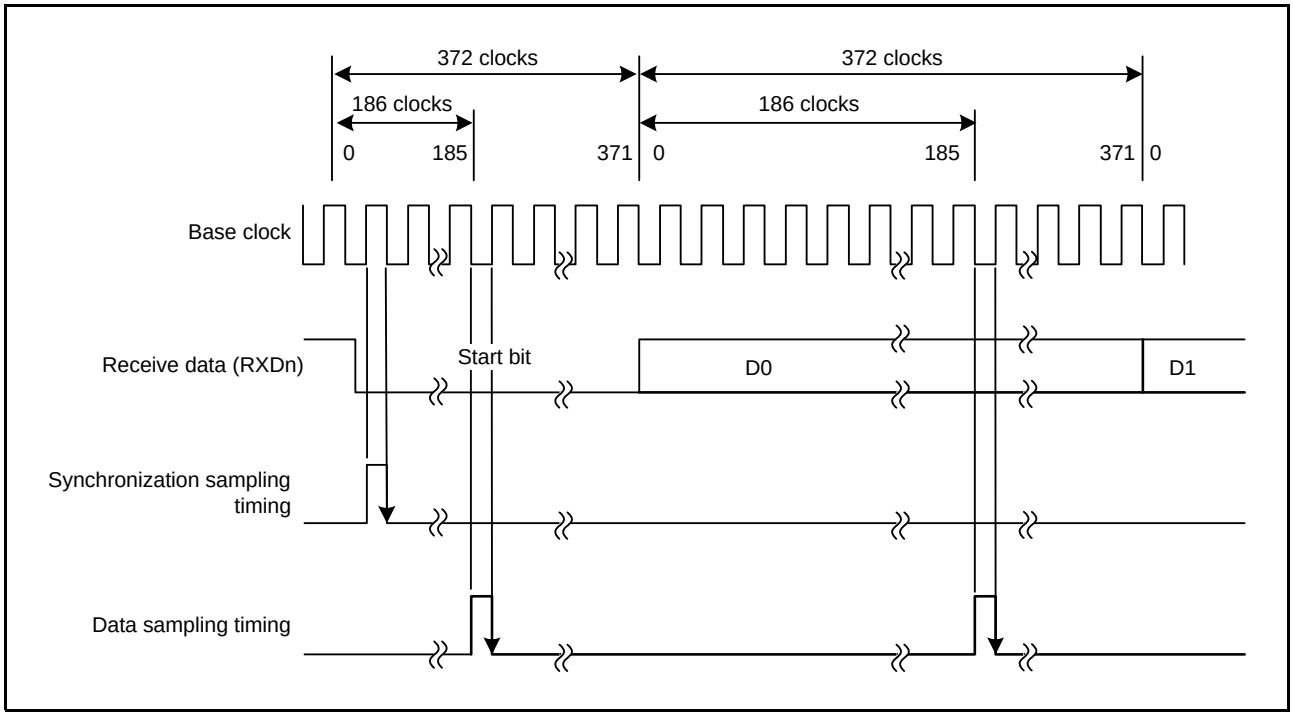


Figure 31.58 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

31.7.5 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 31.59.

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.

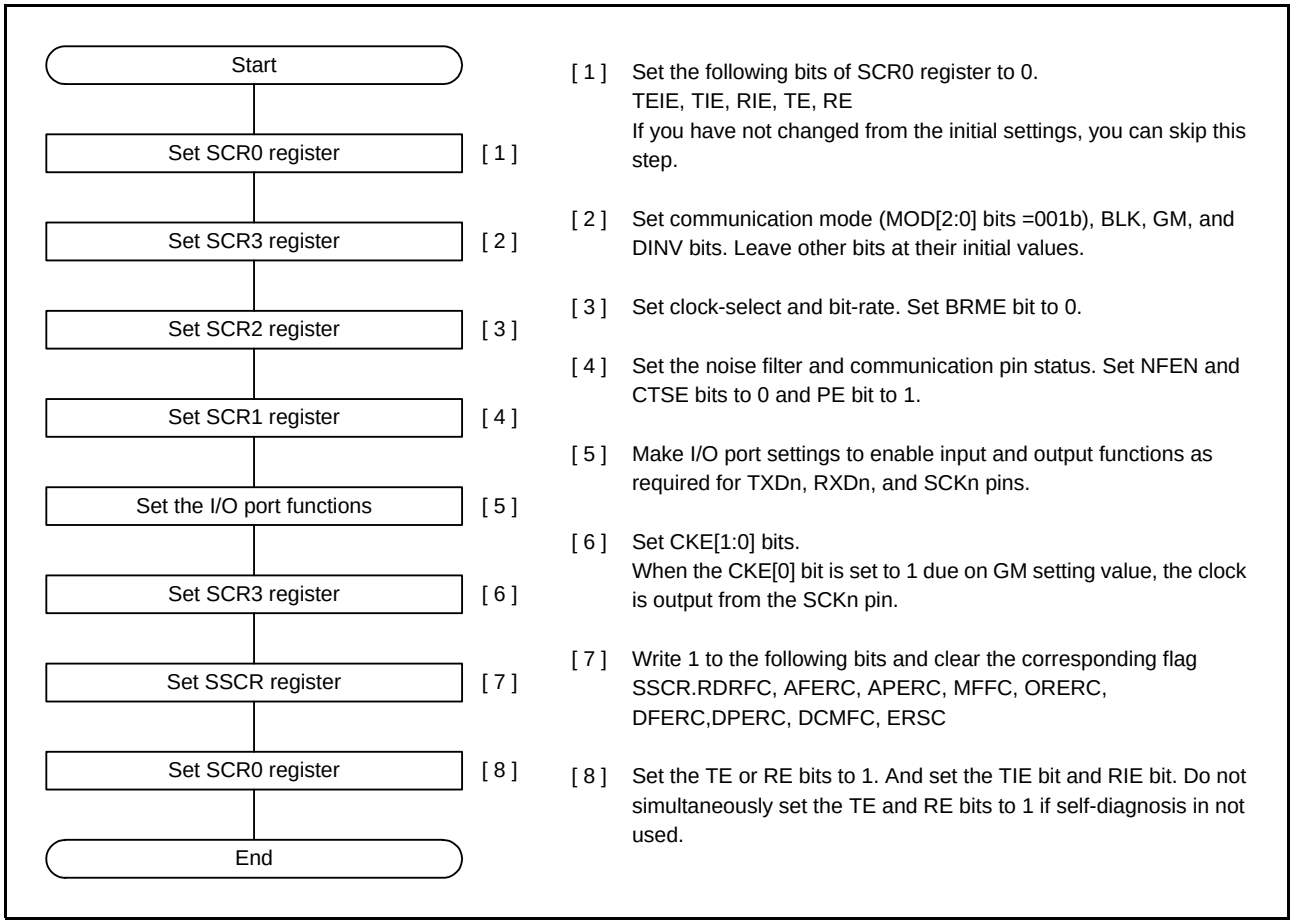


Figure 31.59 Example of RSCI Initialization Flowchart (Smart Card Interface Mode)

Figure 31.60 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.

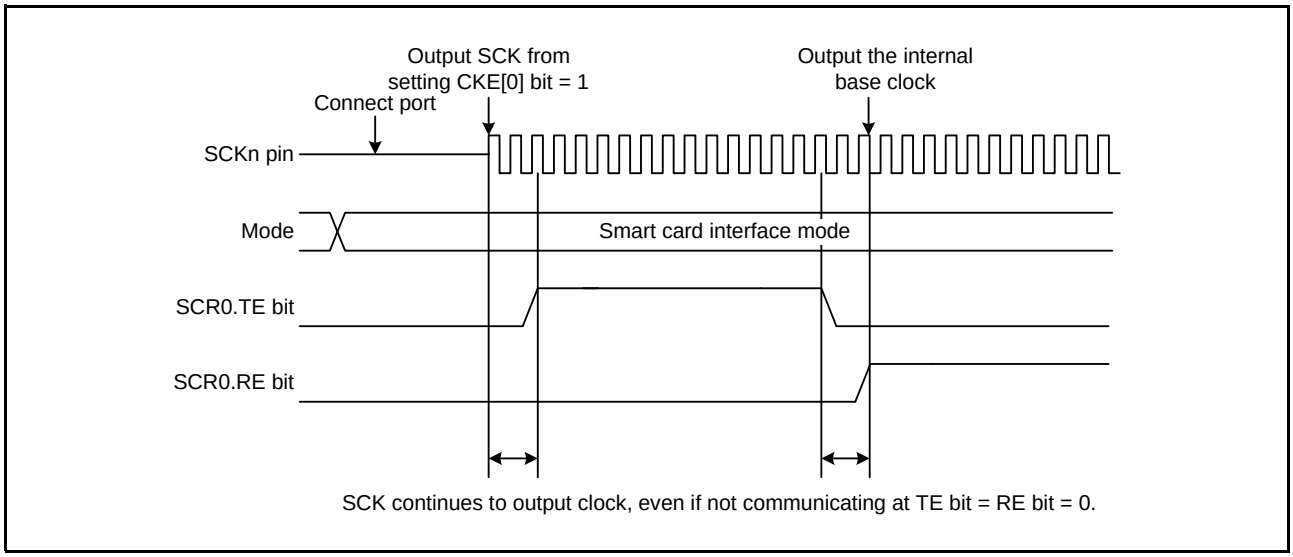
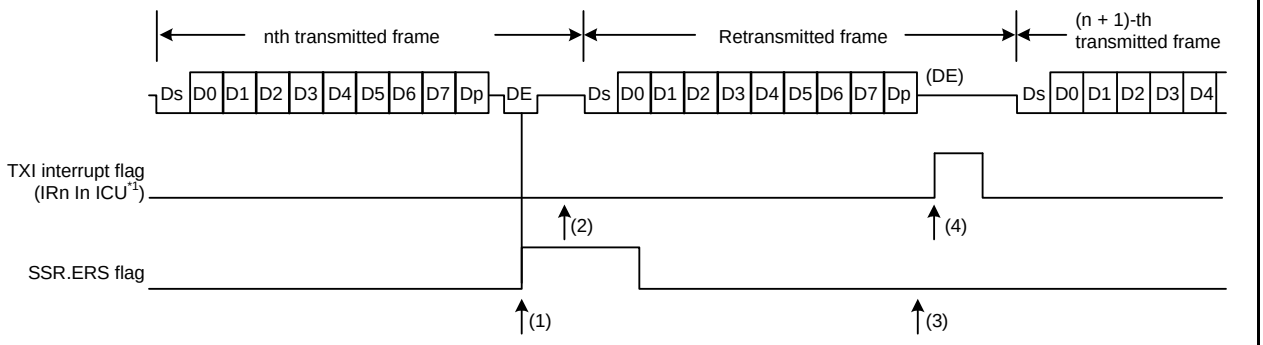


Figure 31.60 Example of Data Transmission Timing in Smart Card Interface Mode

31.7.6 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 31.61 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.61 Data Retransmit Operation in RSCI Transmit Mode

Figure 31.63 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 17, DMA Controller (DMACAa) and section 18, Data Transfer Controller (DTCb).
Note that SSR.TEND flag is set in different timings depending on the SCR3.GM bit setting. Figure 31.62 shows the TEND flag generation timing.

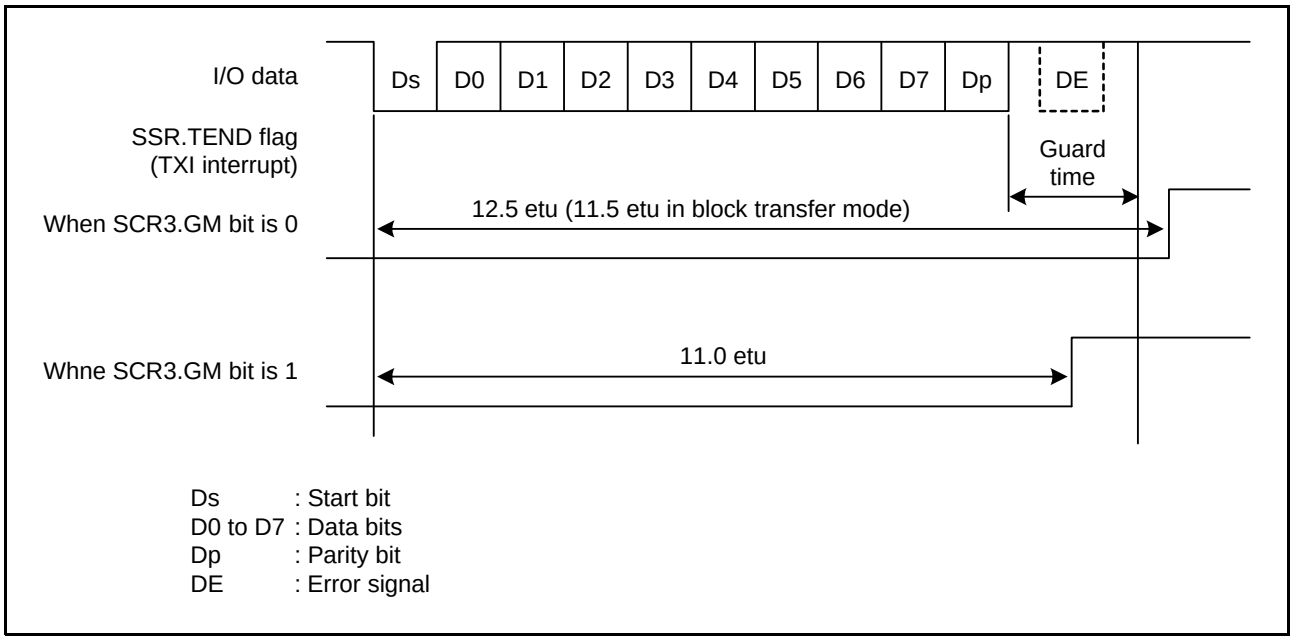


Figure 31.62 SSR.TEND Flag Generation Timing during Transmission

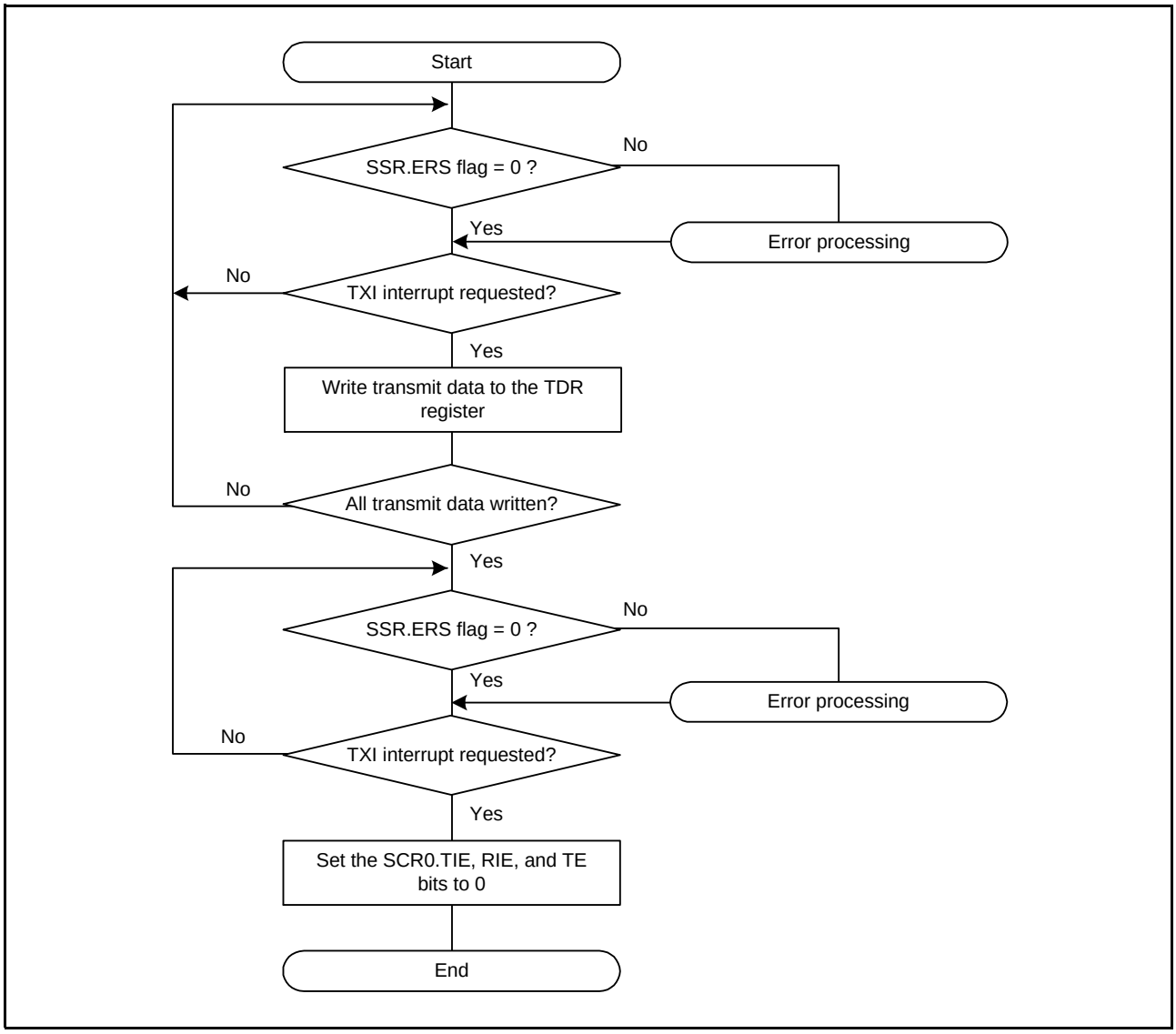
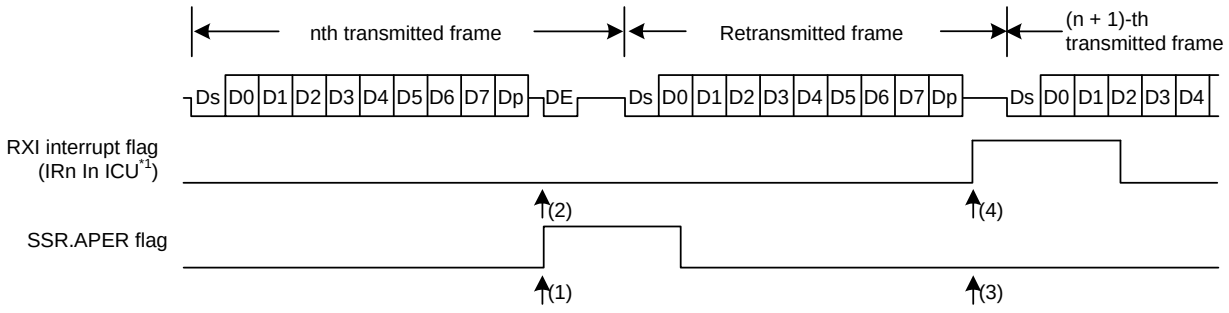


Figure 31.63 Sample Smart Card Interface Transmission Flowchart

31.7.7 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 31.64 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.64 Data Retransmit Operation in RSCI Receive Mode (Data Retransmit Operation during Reception)

Figure 31.65 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 31.3, Operation in Asynchronous Mode.

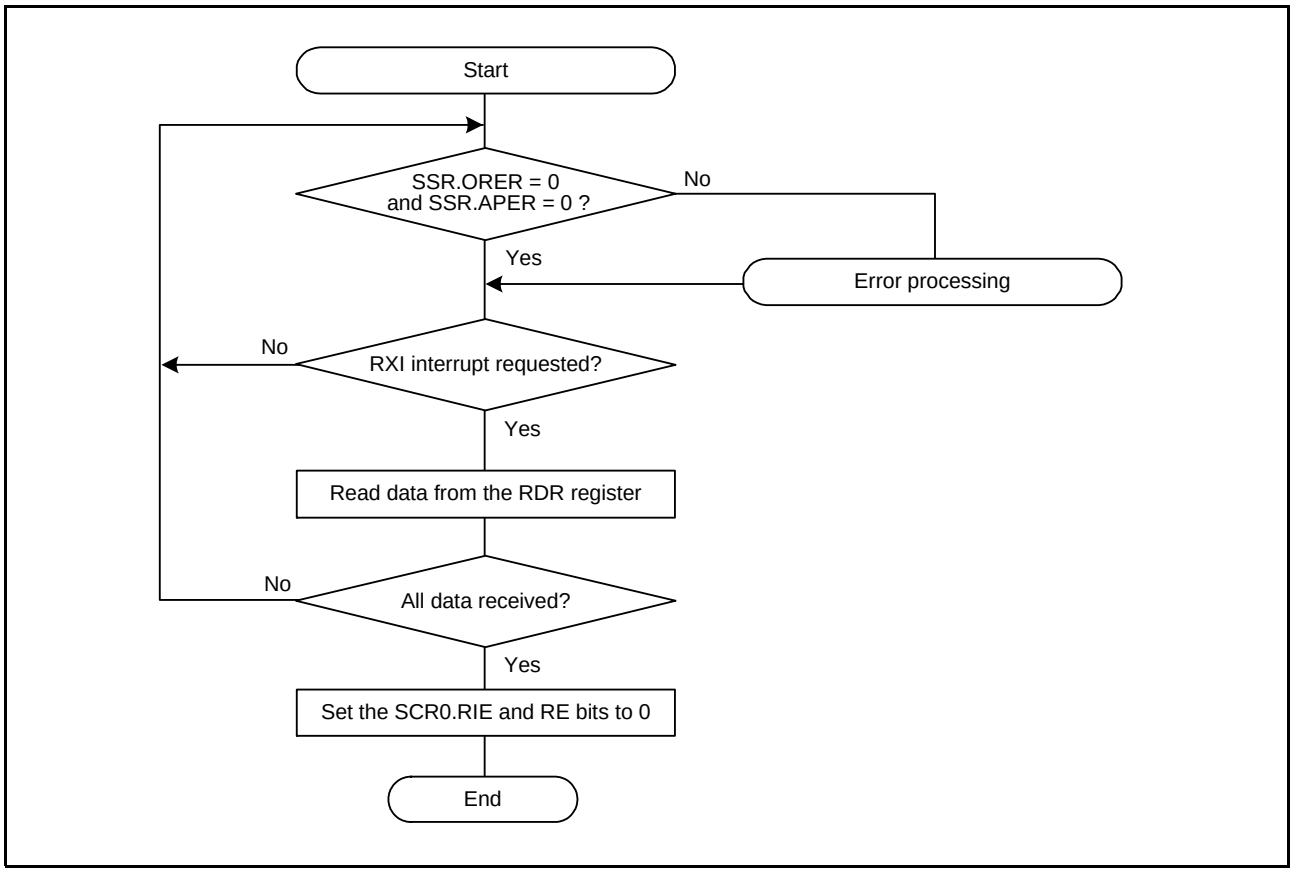


Figure 31.65 Sample Smart Card Interface Reception Flowchart

31.7.8 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 31.2.8, Control Register 3 (SCR3). When setting the clock output, the base clock described in section 31.7.4, 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 31.2.7, 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 31.66 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.

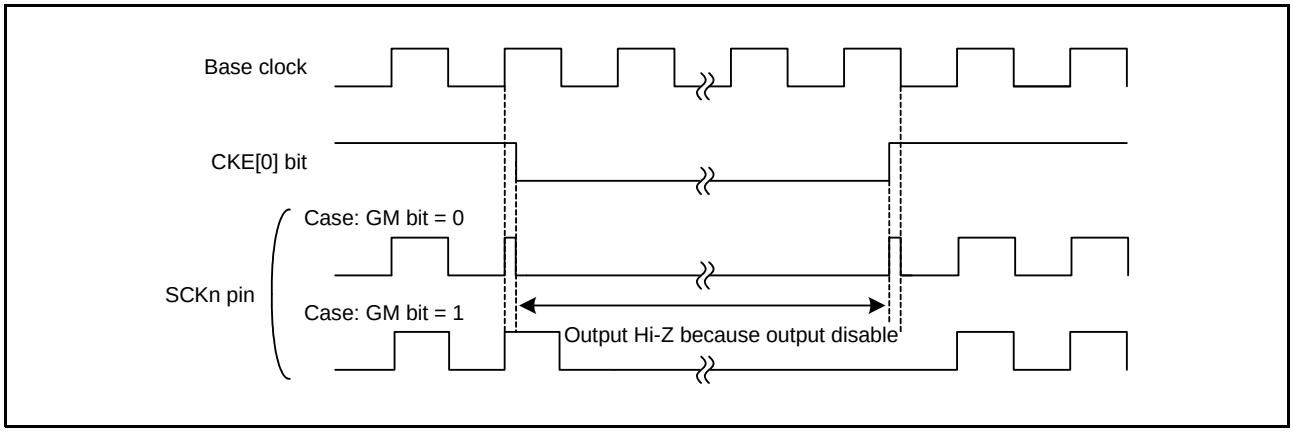


Figure 31.66 Clock Control Timing Chart by SCR3.GM Bit

31.8 Extended Serial Mode

31.8.1 Serial Transfer Protocol

As an extended function of the RSCI, the RSCI supports the serial communication protocol (Figure 31.67) 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.

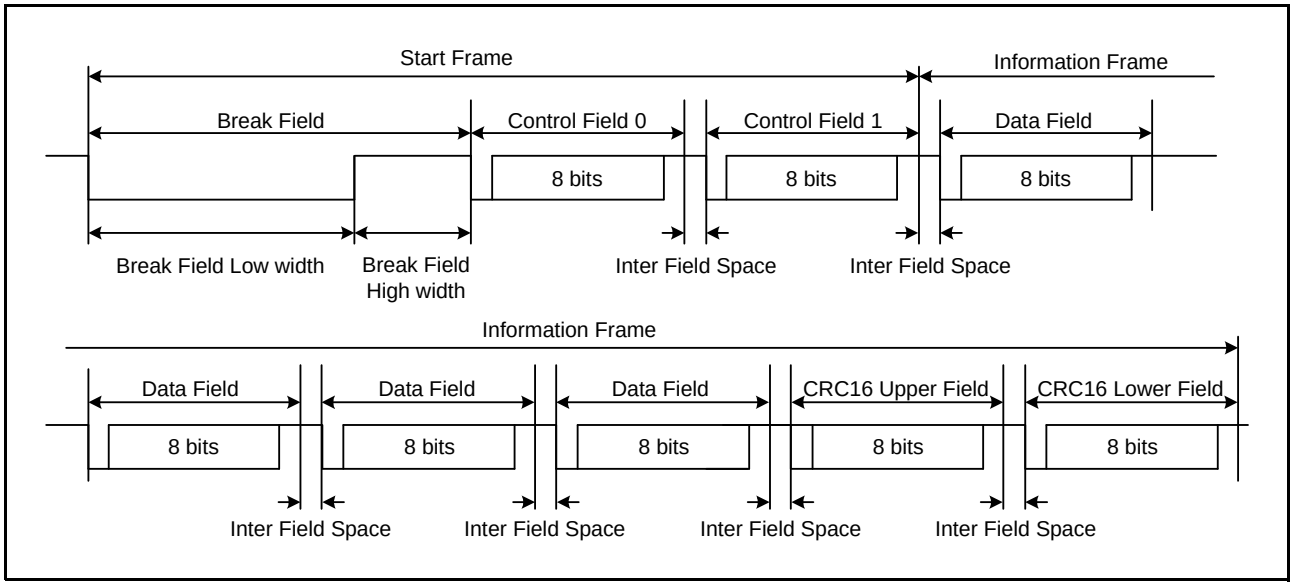


Figure 31.67 Protocol for Serial Transfer by the Extended Serial Mode

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.

31.8.2 Transmitting a Start Frame

Figure 31.68 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 31.69 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 31.8) 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)

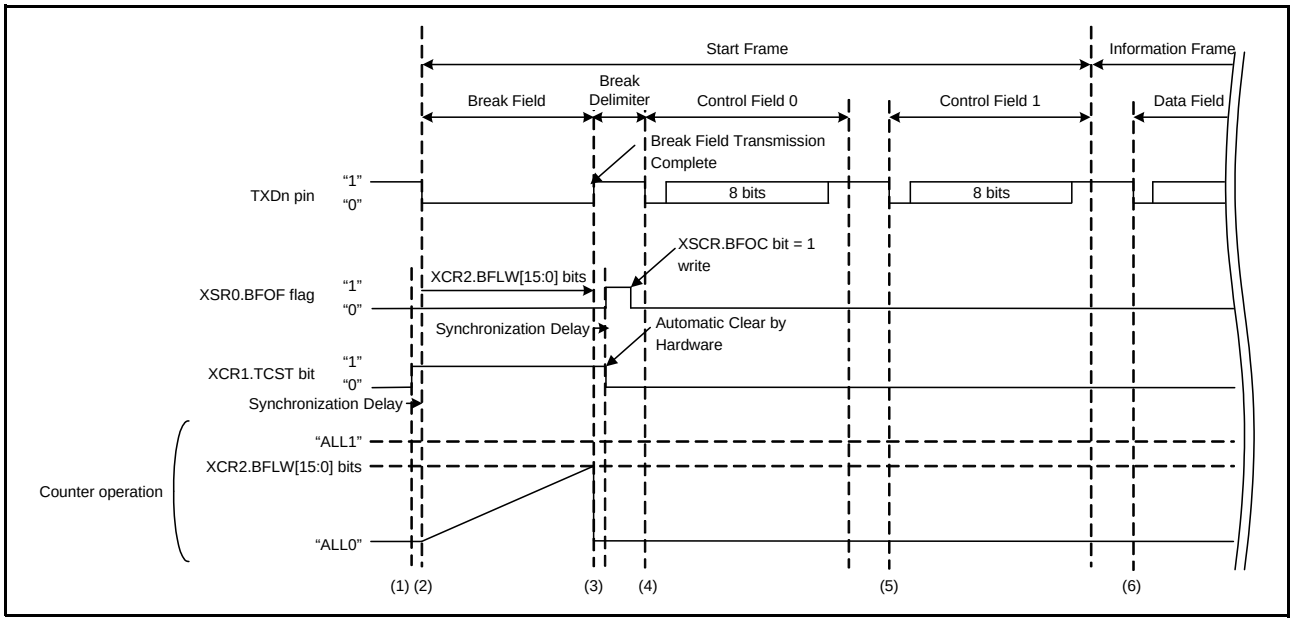


Figure 31.68 Example of Operations When Transmitting a Start Frame

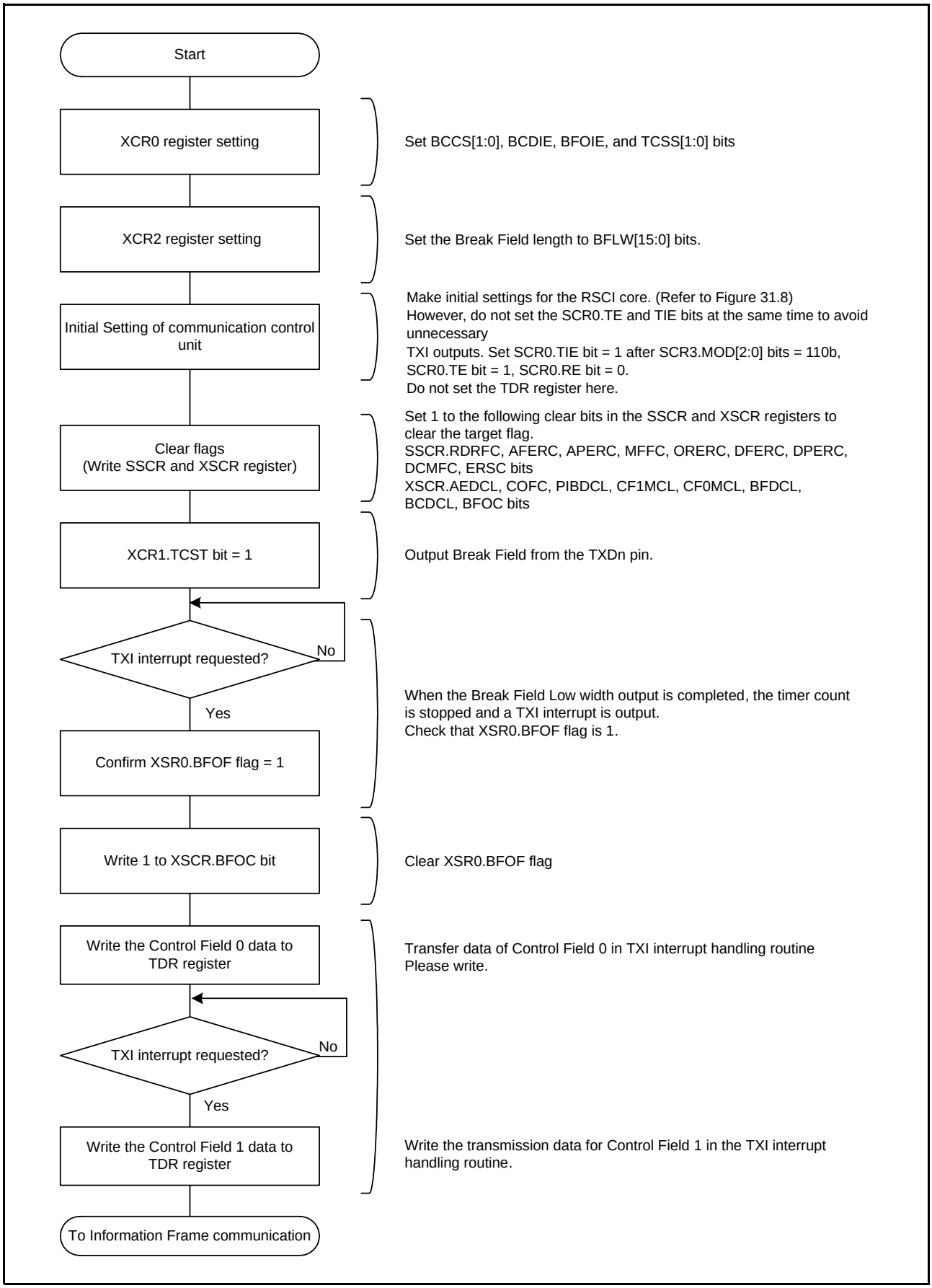


Figure 31.69 Example of Start Frame Transmission

31.8.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 31.39.

Table 31.39 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

31.8.3.1 Extended Serial Normal Reception of Start Frame (PIB not Used)

Figure 31.70 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 31.71 shows an example of reception to detect the Break Field during Control Field 1. Figure 31.72 shows a flowchart to receive the Start Frame, and Figure 31.73 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 31.71).
(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.

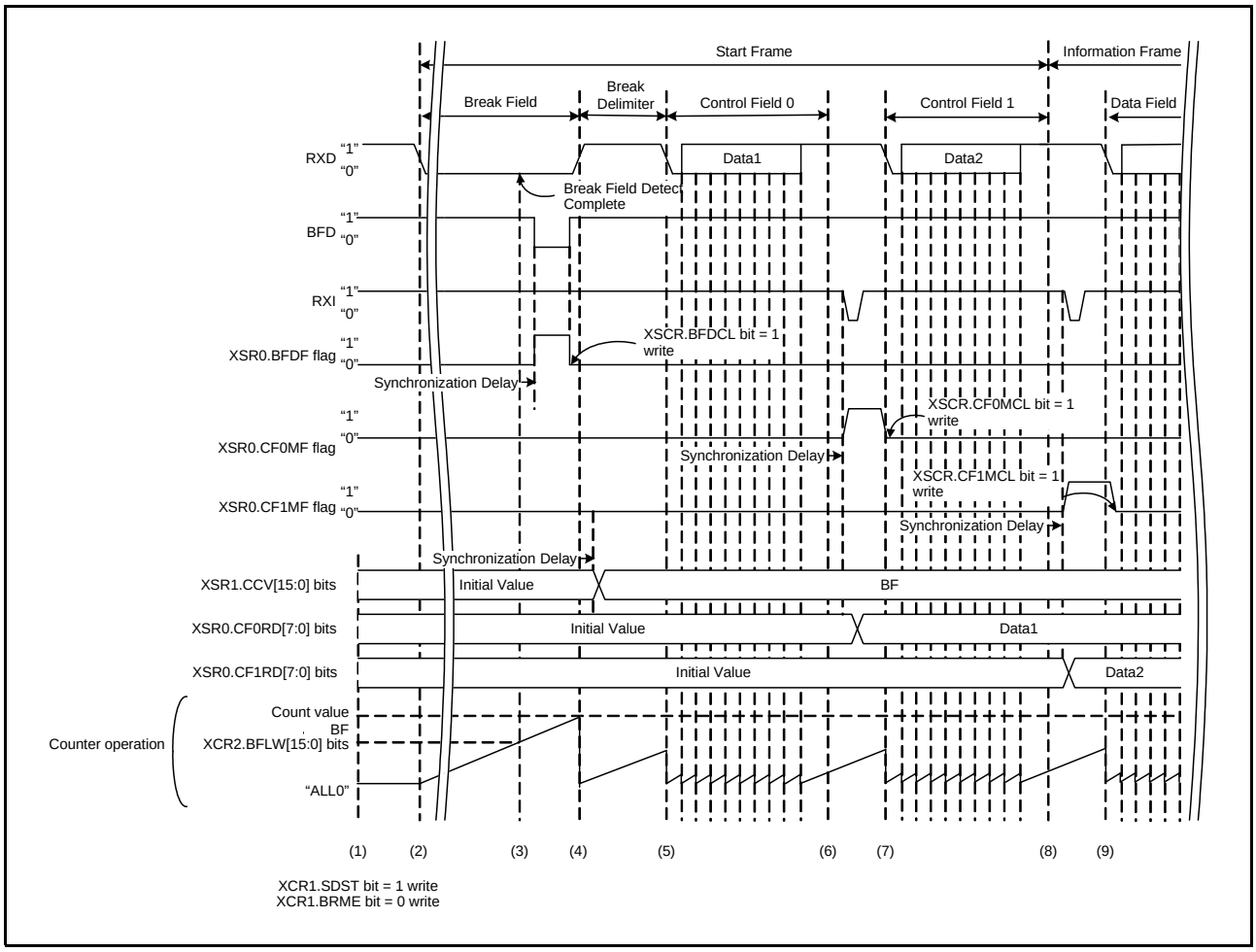


Figure 31.70 Normal Reception Example of Start Frame (PIB Not Used)

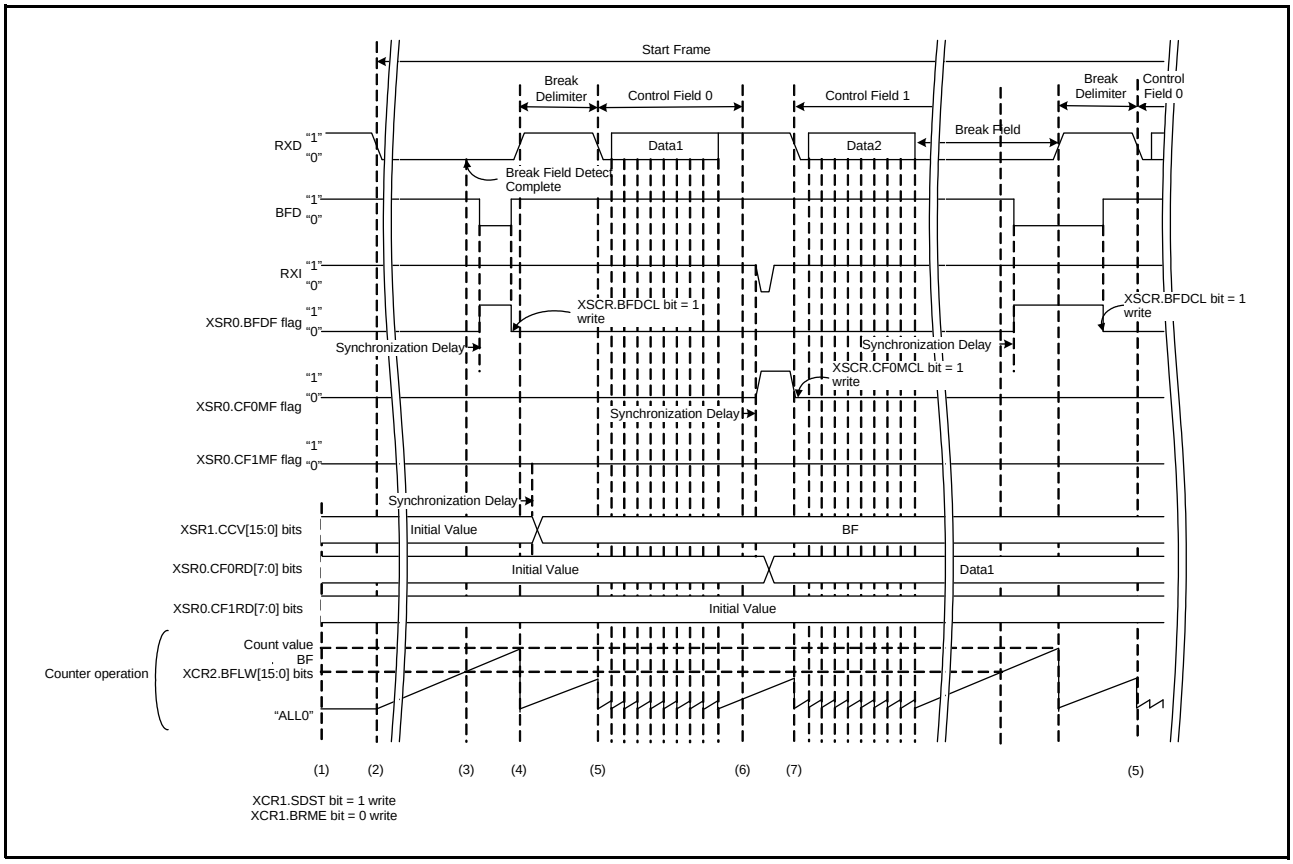


Figure 31.71 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

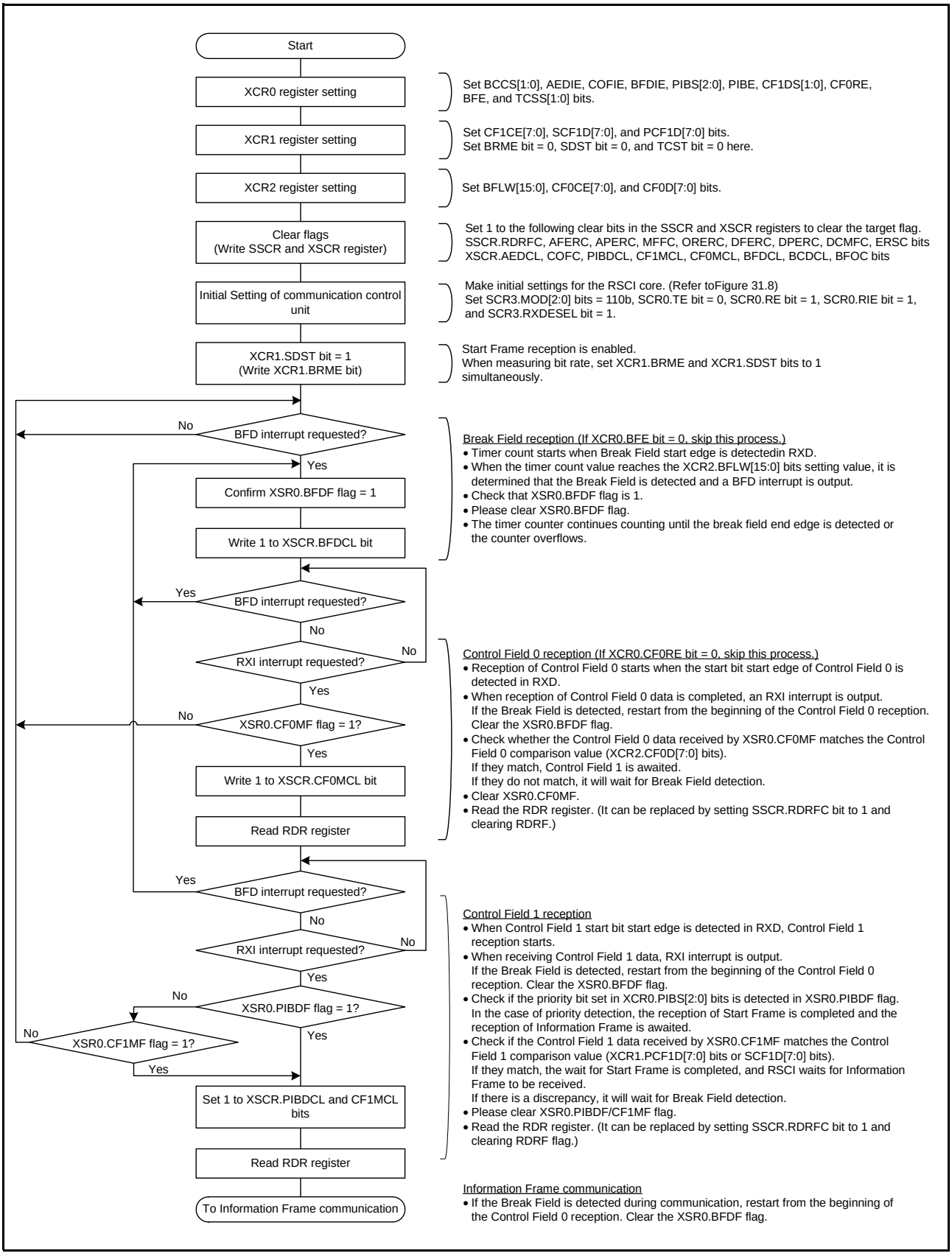


Figure 31.72 Sample Flowchart for Reception of a Start Frame

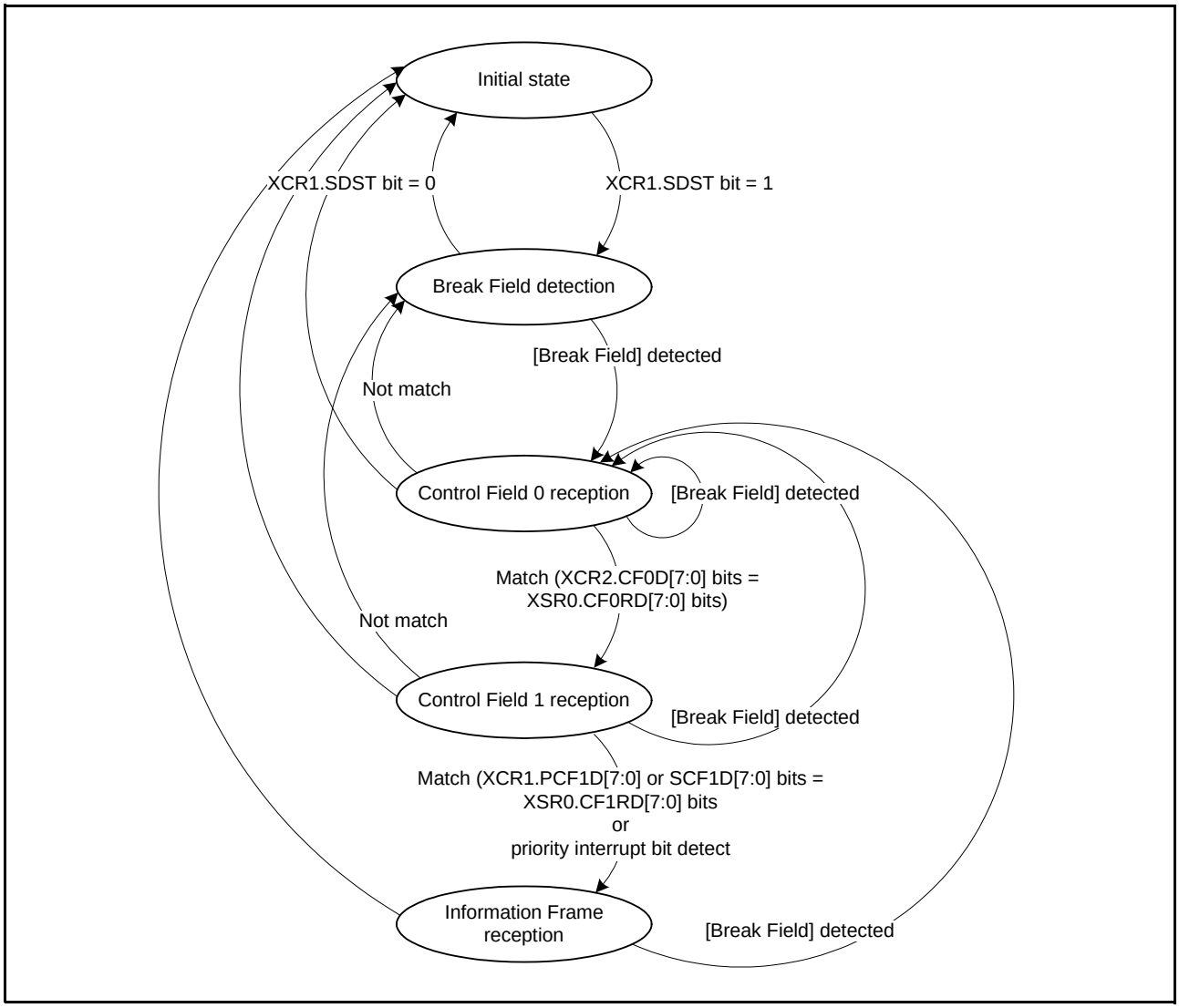


Figure 31.73 State Transition Diagram of Start Frame Reception

31.8.3.2 Priority Interrupt Bit

Figure 31.74 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 31.70, 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.

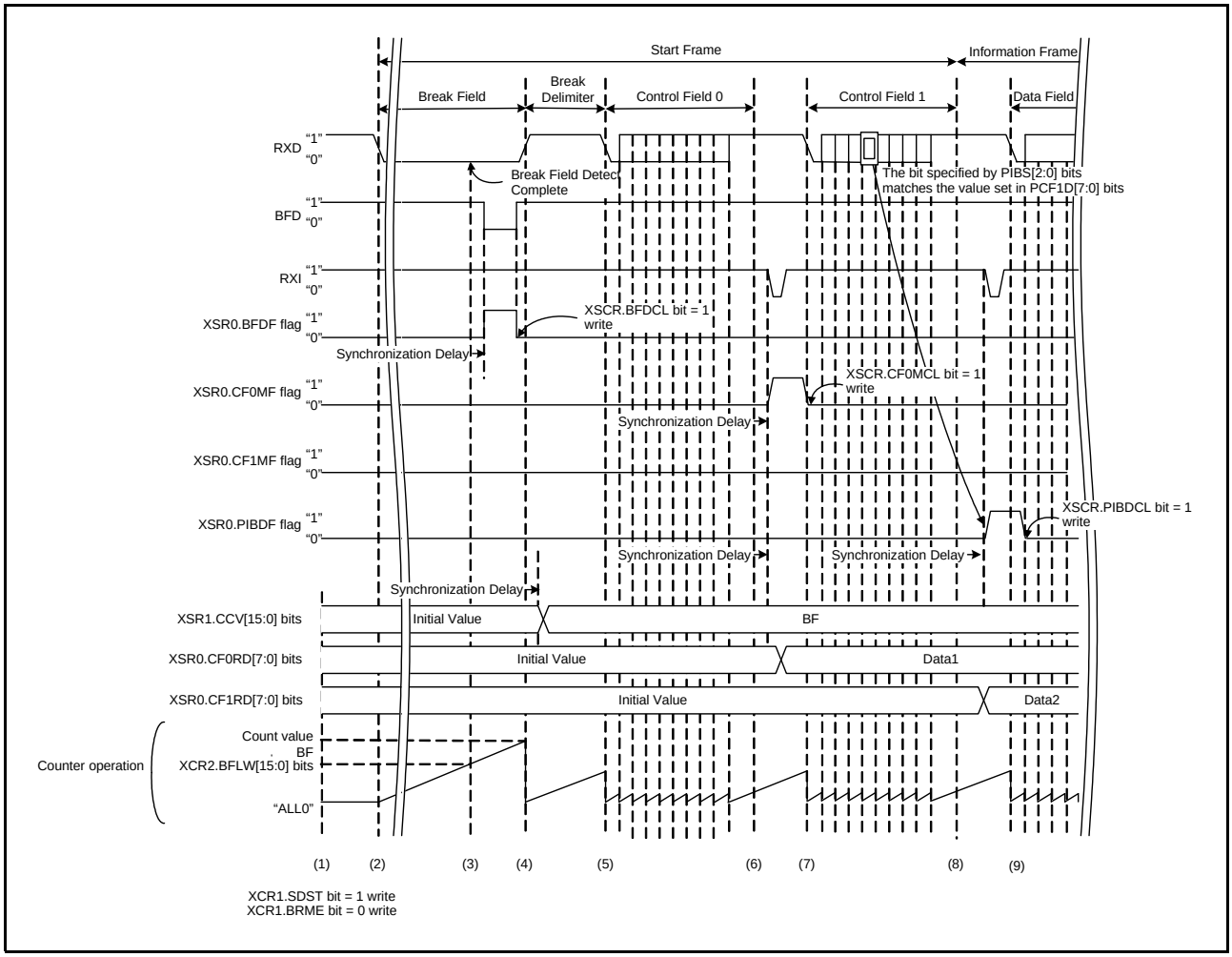


Figure 31.74 Start Frame Reception Example (Priority Interrupt Bit Used)

31.8.4 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 31.75 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 31.76. Check the bus state to decide whether to resume transmission.

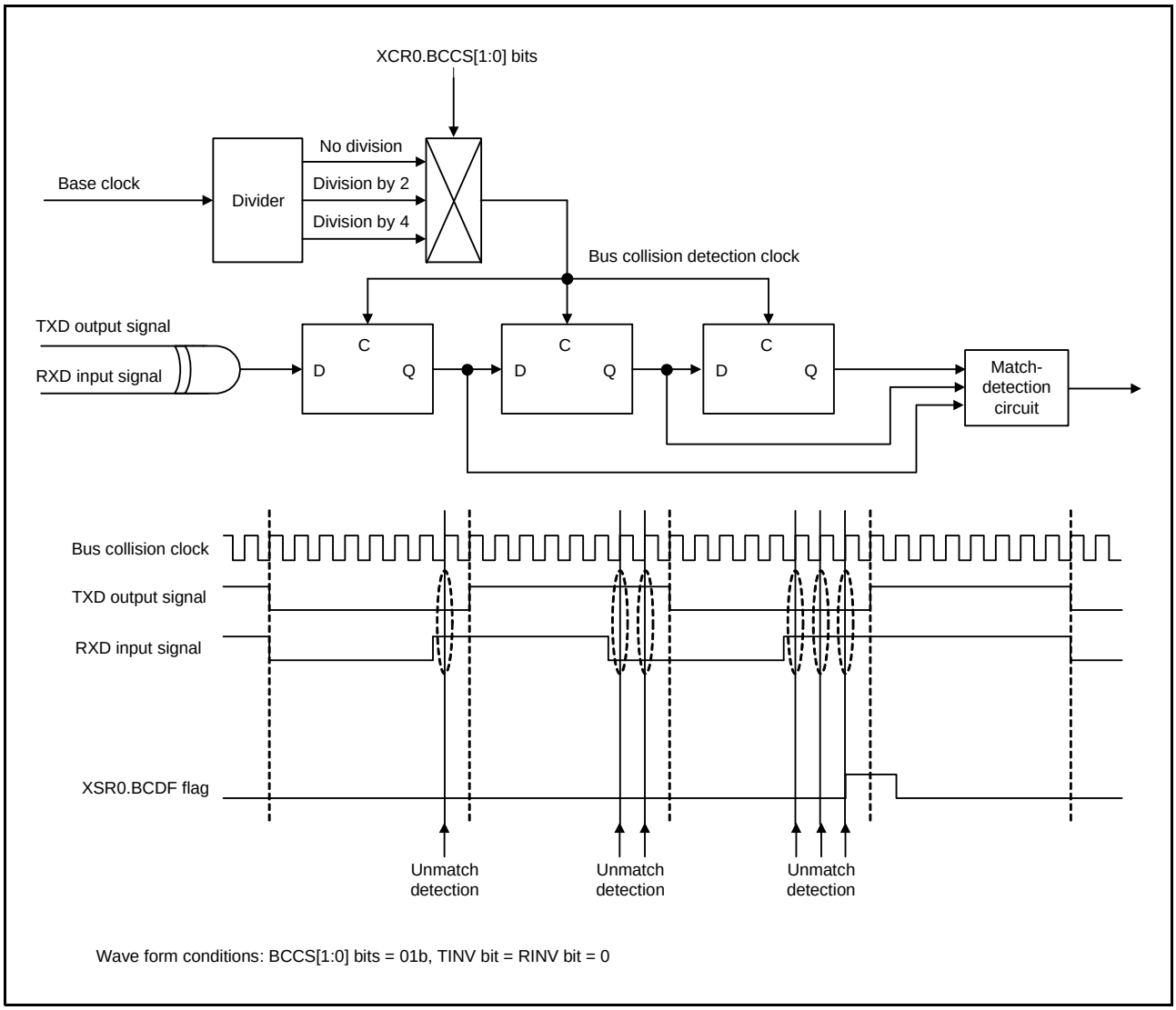


Figure 31.75 Example of Operations with Bus Collision Detection

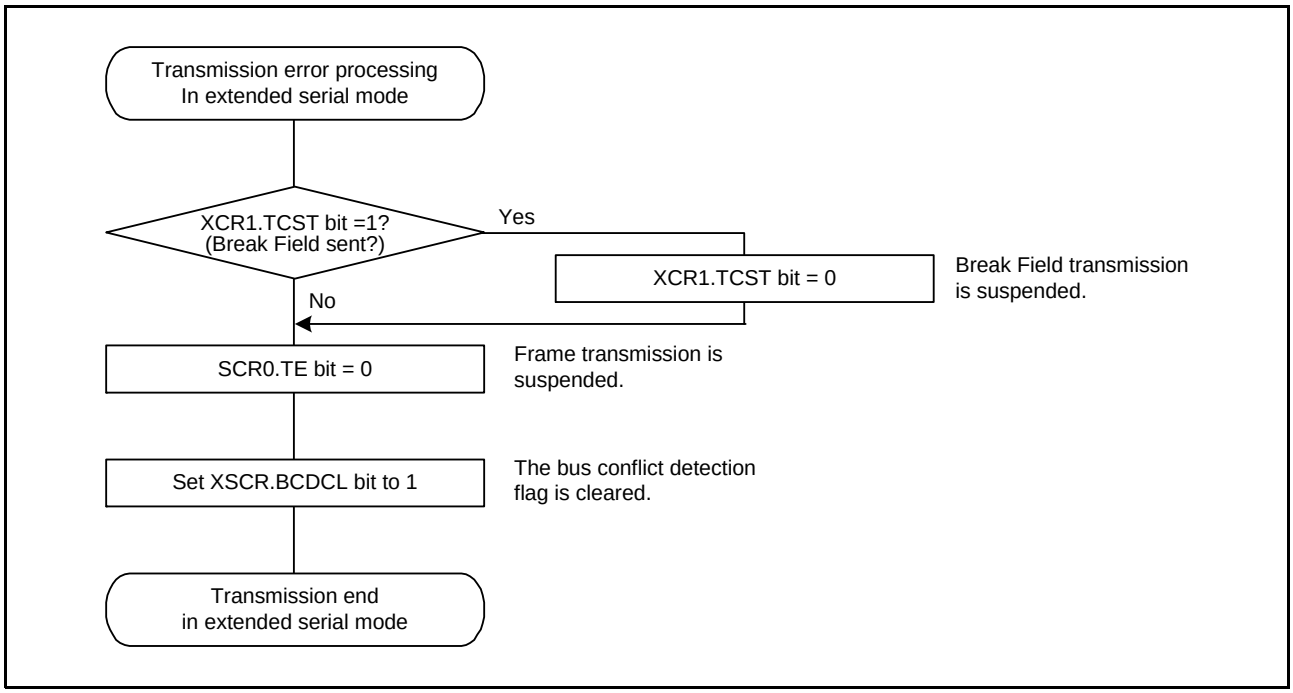


Figure 31.76 ERI Interrupt Handling Flow at Transmission in Extended Serial Mode

31.8.5 Bit Rate Measurement

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 31.77 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.

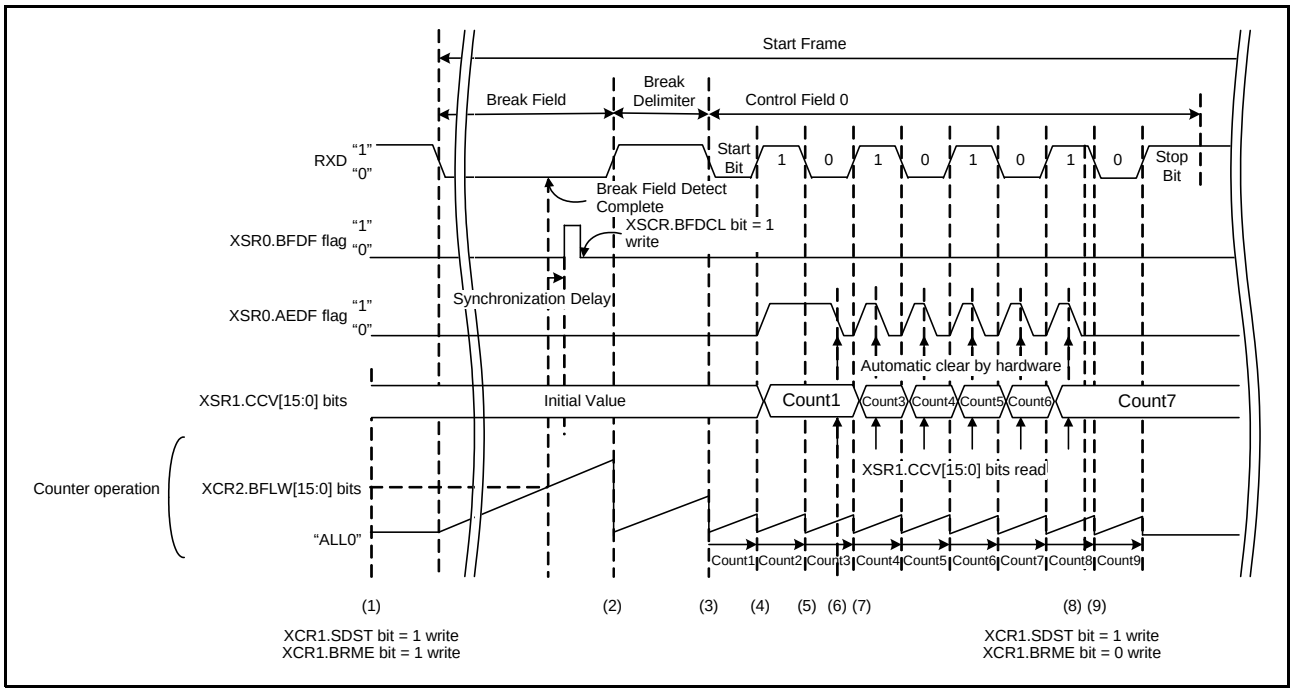


Figure 31.77 Operation Example of the Bit Rate Measurement Function

31.9 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 31.78 shows I²C bus format, and Figure 31.79 shows the timing of I²C.

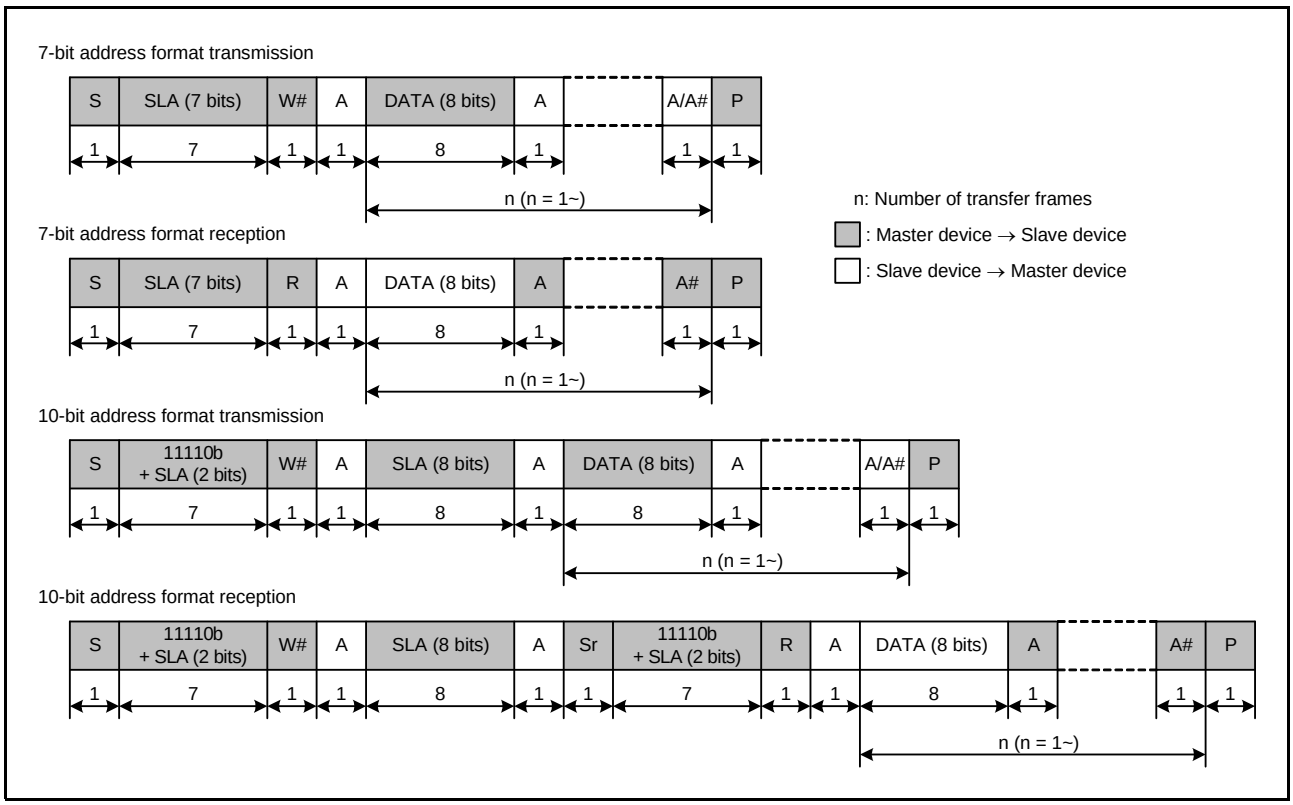


Figure 31.78 I²C-Bus Format

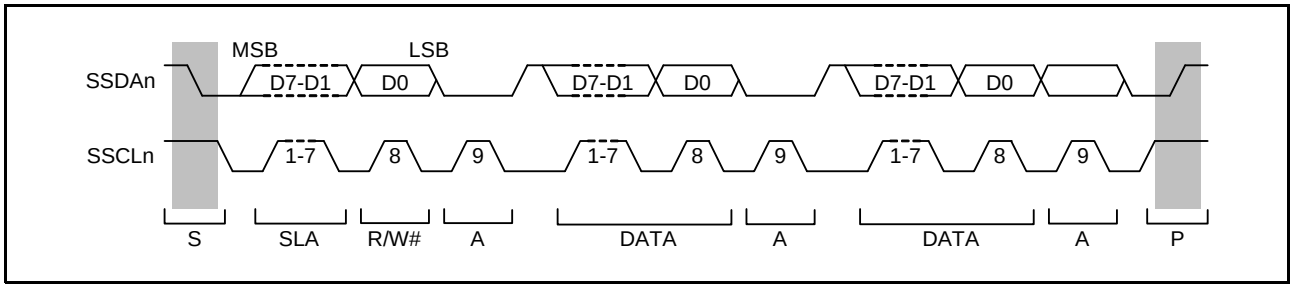


Figure 31.79 I²C-Bus Timing (When SLA is 7 Bits)

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.

31.9.1 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 31.80 shows the timing of operations in the generation of start, restart and stop conditions.

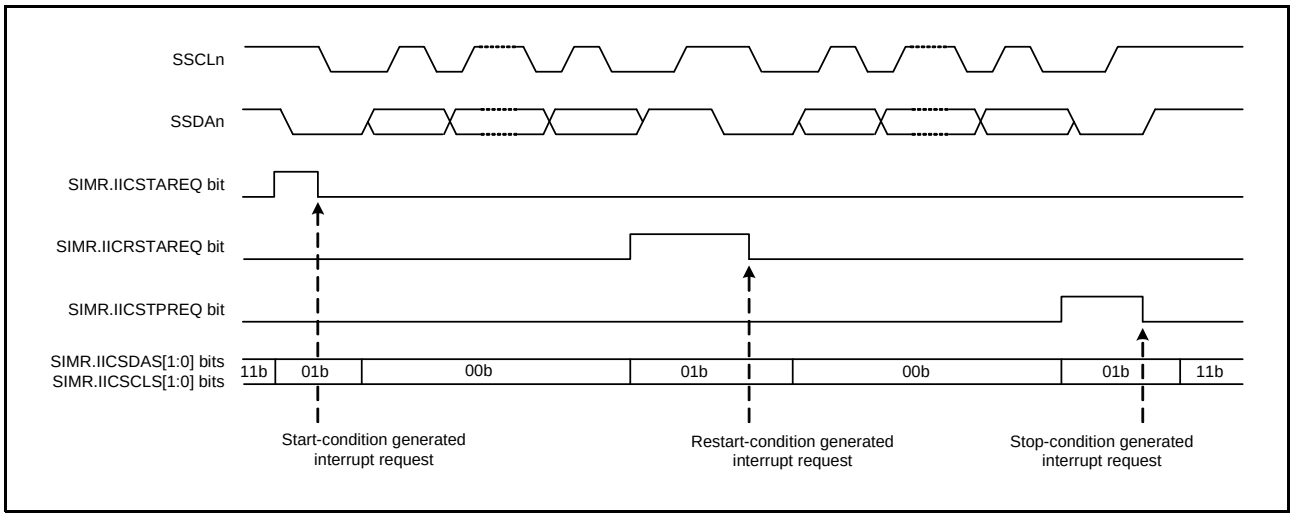


Figure 31.80 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

31.9.2 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 31.81 shows an example of operations to synchronize the clocks.

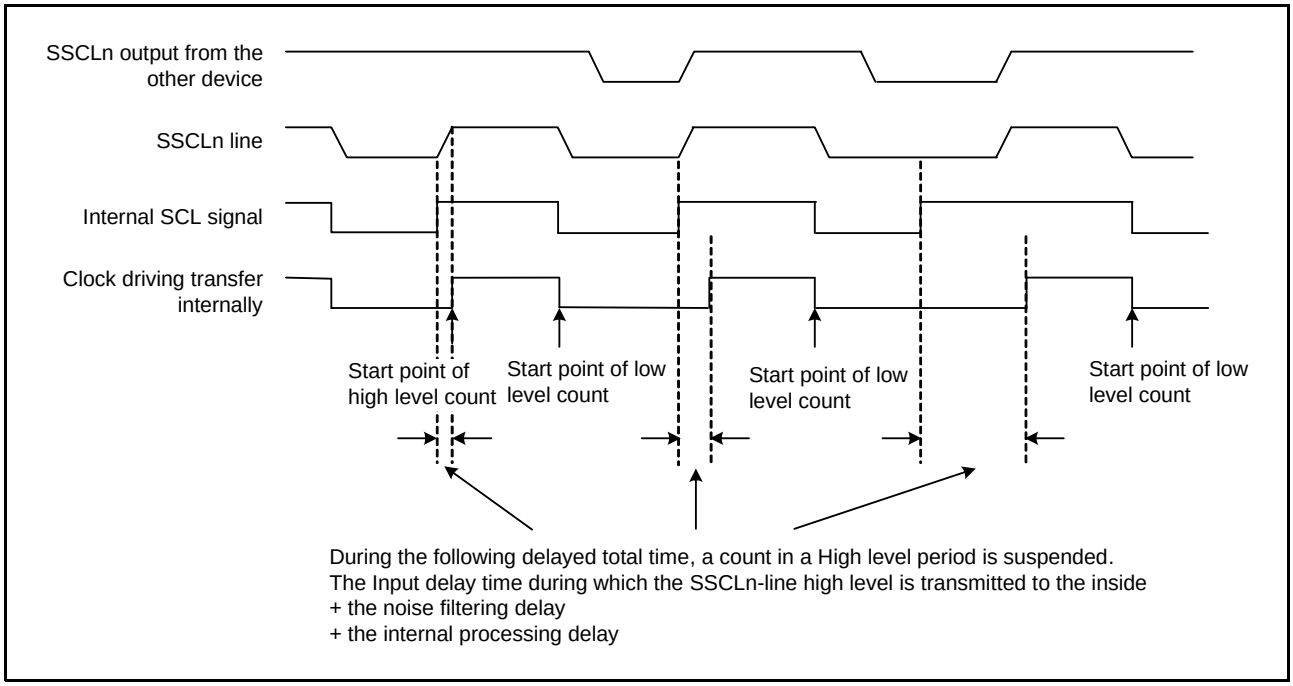


Figure 31.81 Example of Operations for Clock Synchronization

31.9.3 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 31.82 shows the timing of delays in SDA output.

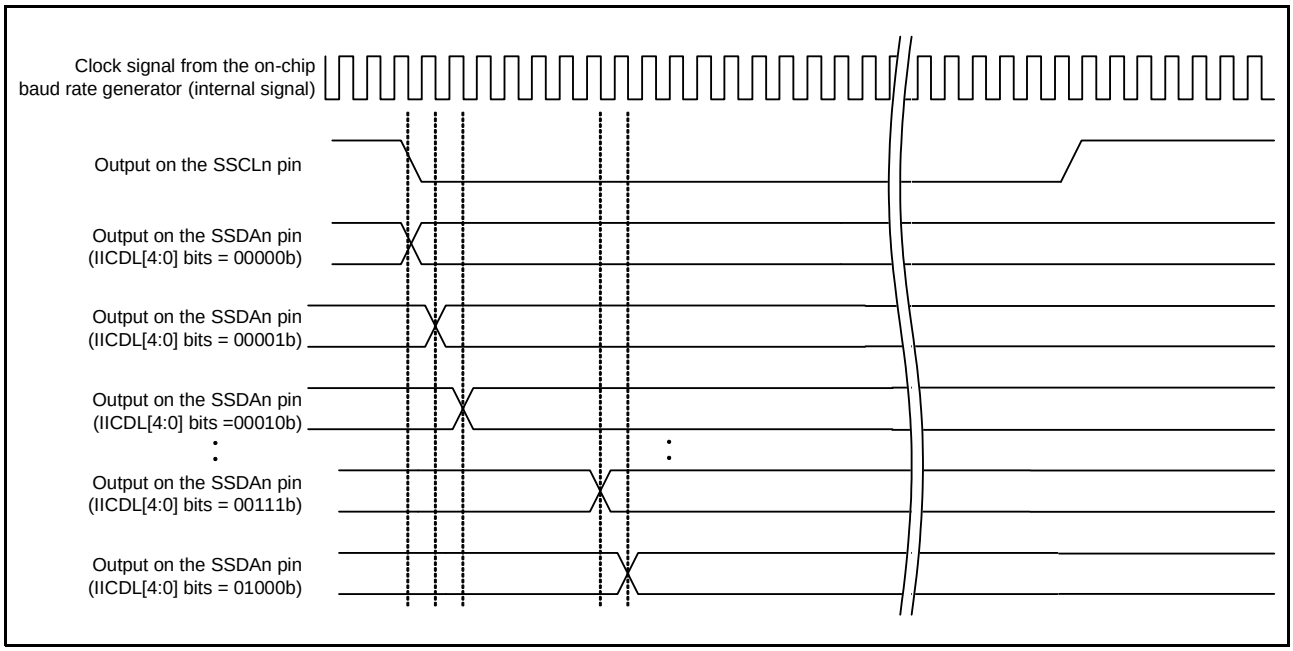
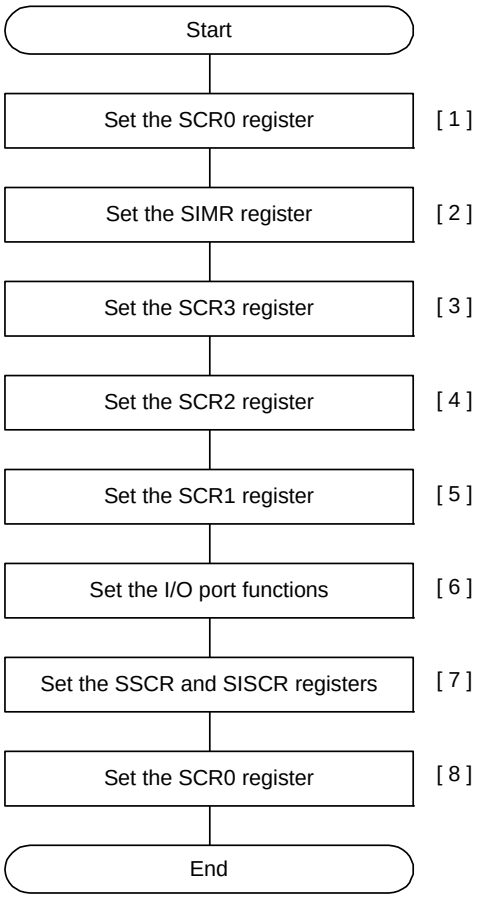


Figure 31.82 Timing of Delays in SDA Output

31.9.4 RSCI Initialization (Simple I2C Mode)

Write initial value (0000_0000h) to SCR0, then initialize RSCI according to Figure 31.83.
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.

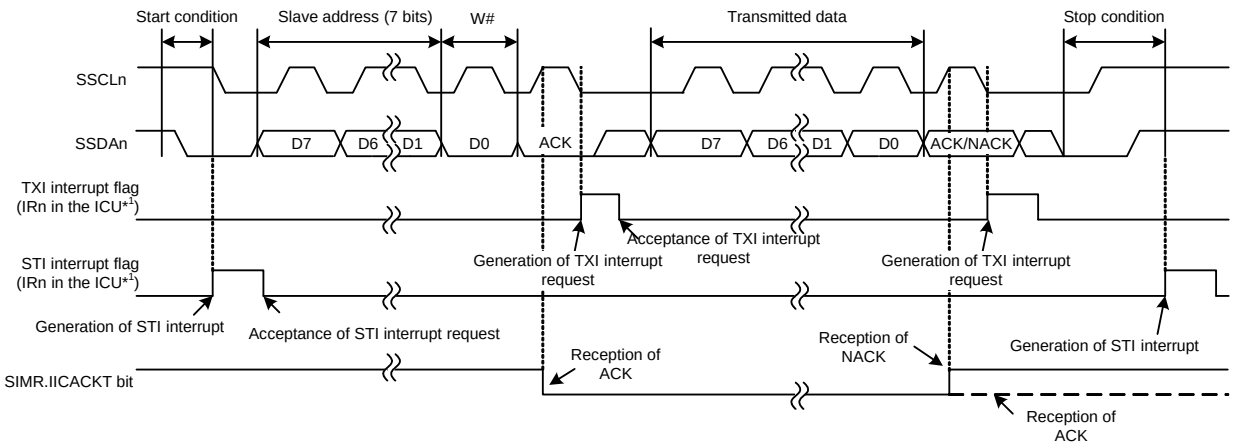
Figure 31.83 Example of the Flowchart of RSCI Initialization (for Simple I2C Mode)

31.9.5 Operation in Master Transmission (Simple I2C Mode)

Figure 31.84 and Figure 31.85 show examples of operations in master transmission, Figure 31.86 to Figure 31.88 show the example flowcharts. See Table 31.46 about the STI interrupt.

Figure 31.84 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 31.86 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 31.87 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

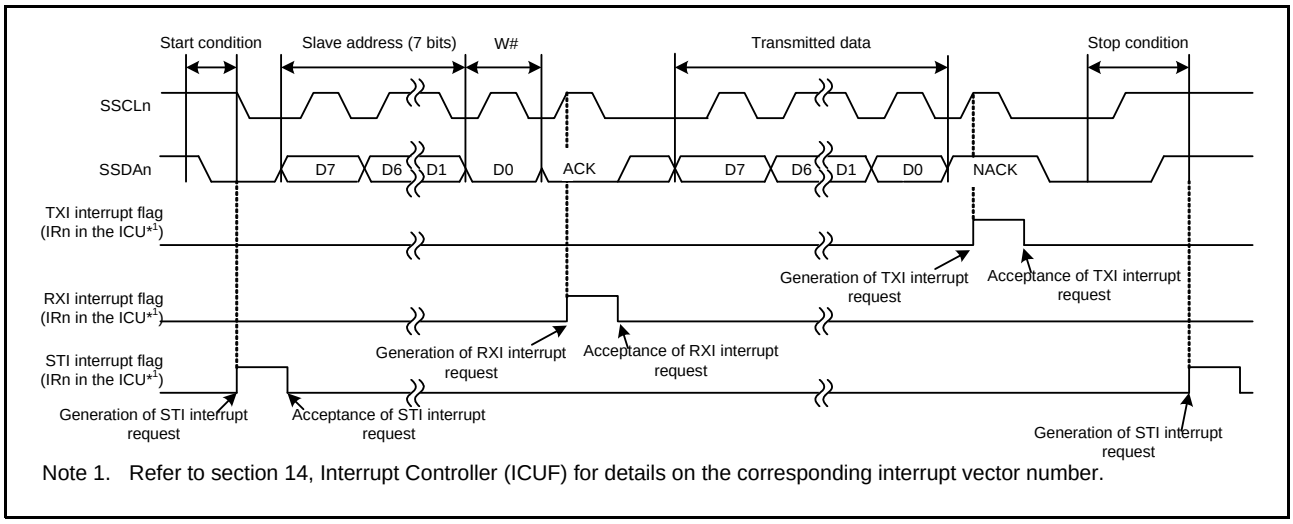
Figure 31.84 Example 1 of Operations for Master Transmission in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))

Figure 31.85 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 31.88 shows a flow chart of SIMR.IICINTM bit is 0

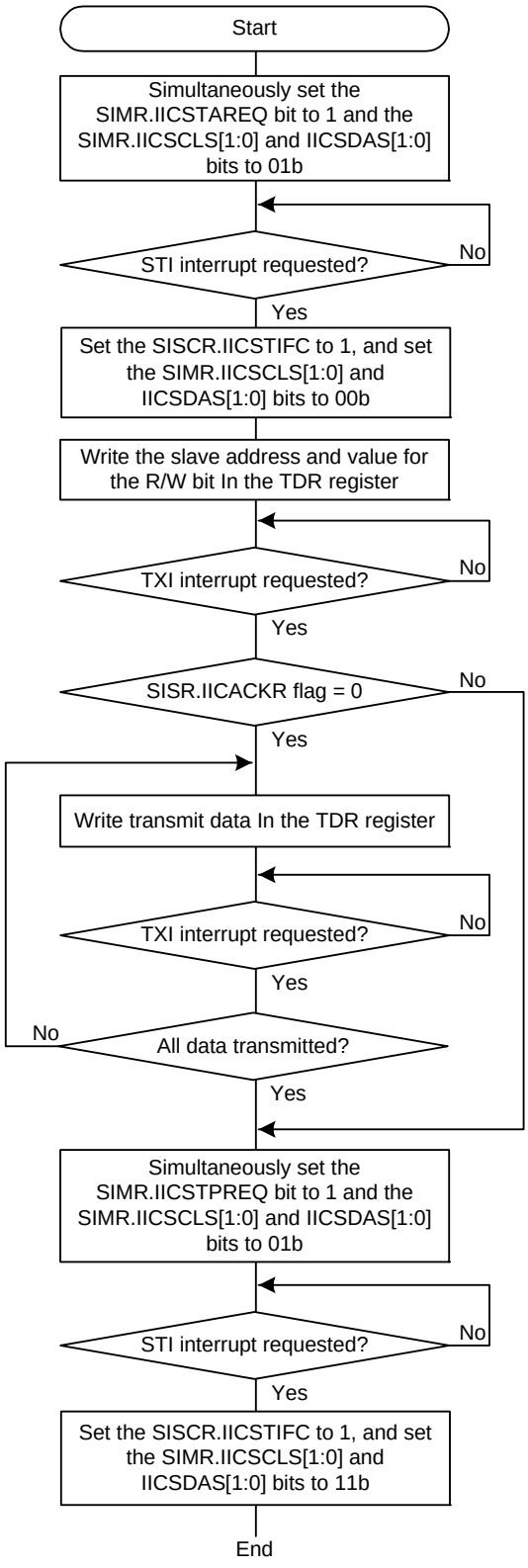
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.85 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] 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.

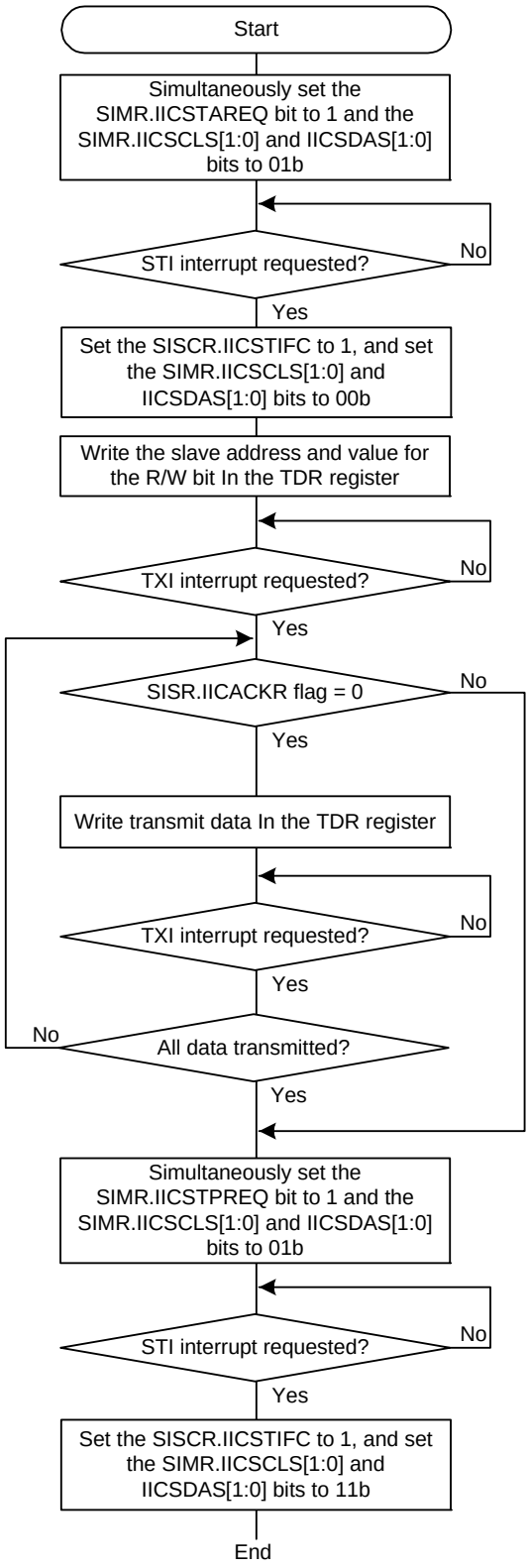
If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[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 31.86 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] [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. To checking whether ACK or NACK in every transmission, you cannot use DMA or DTC.

[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 31.87 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)

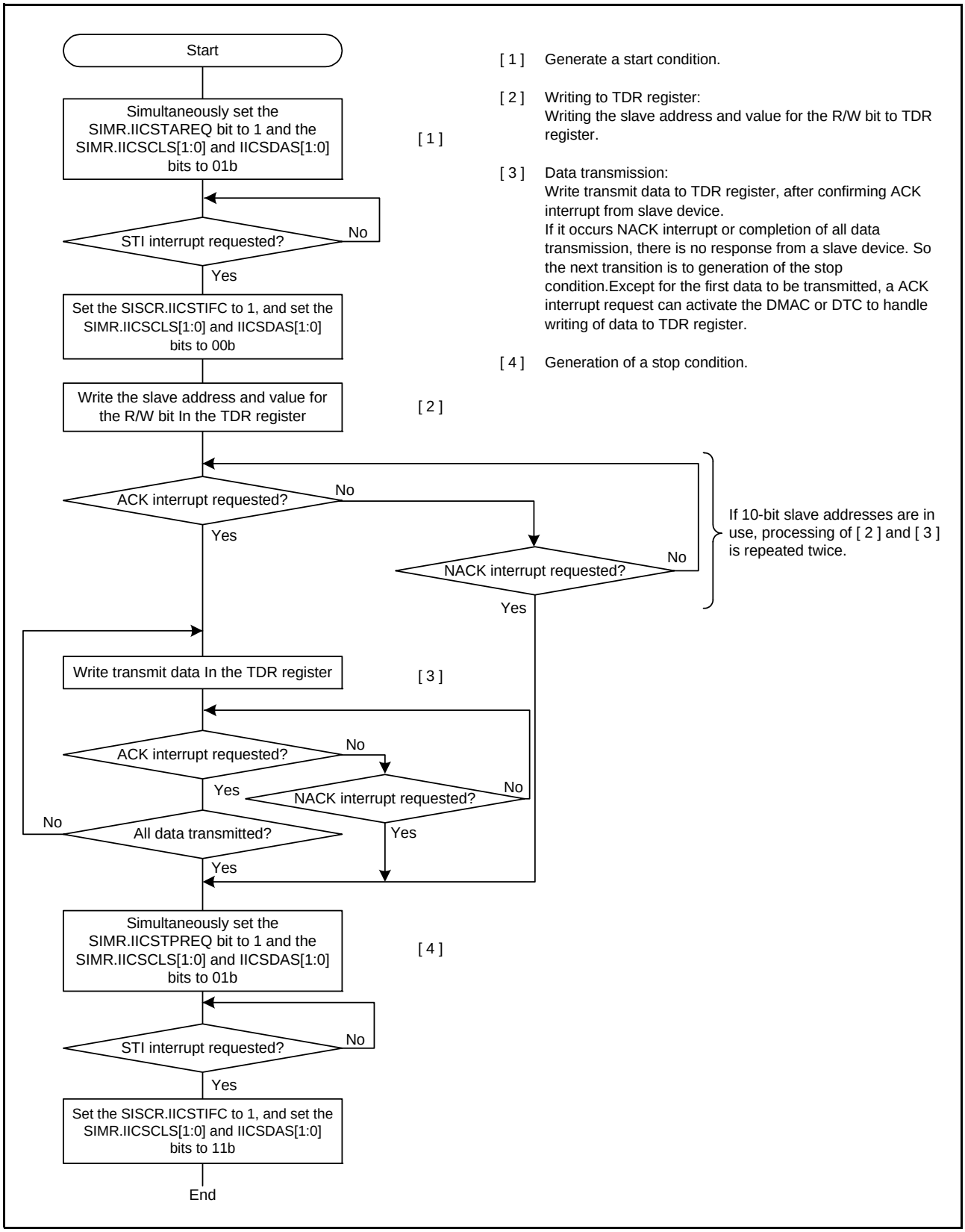
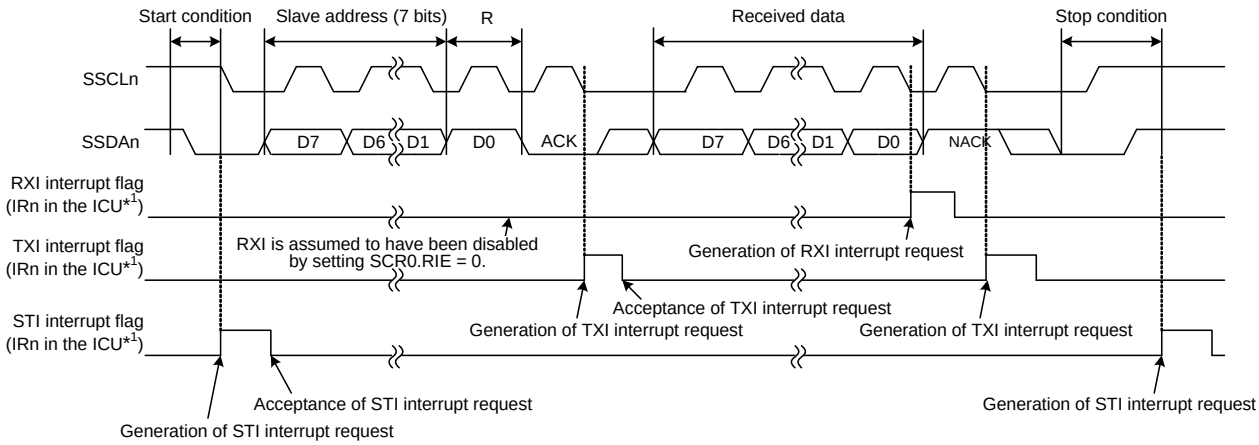


Figure 31.88 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (when SIMR.IICINTM Bit is 0)

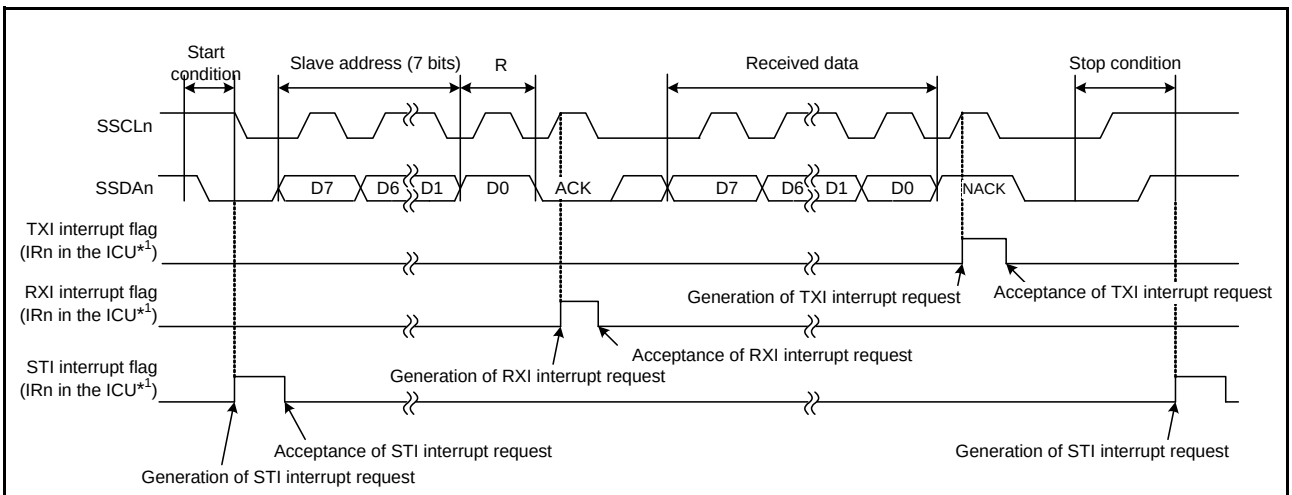
31.9.6 Master Reception (Simple I²C Mode)

Figure 31.89 and Figure 31.90 show example of operations in simple I²C mode master reception. Figure 31.91 and Figure 31.92 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.89 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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.90 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))

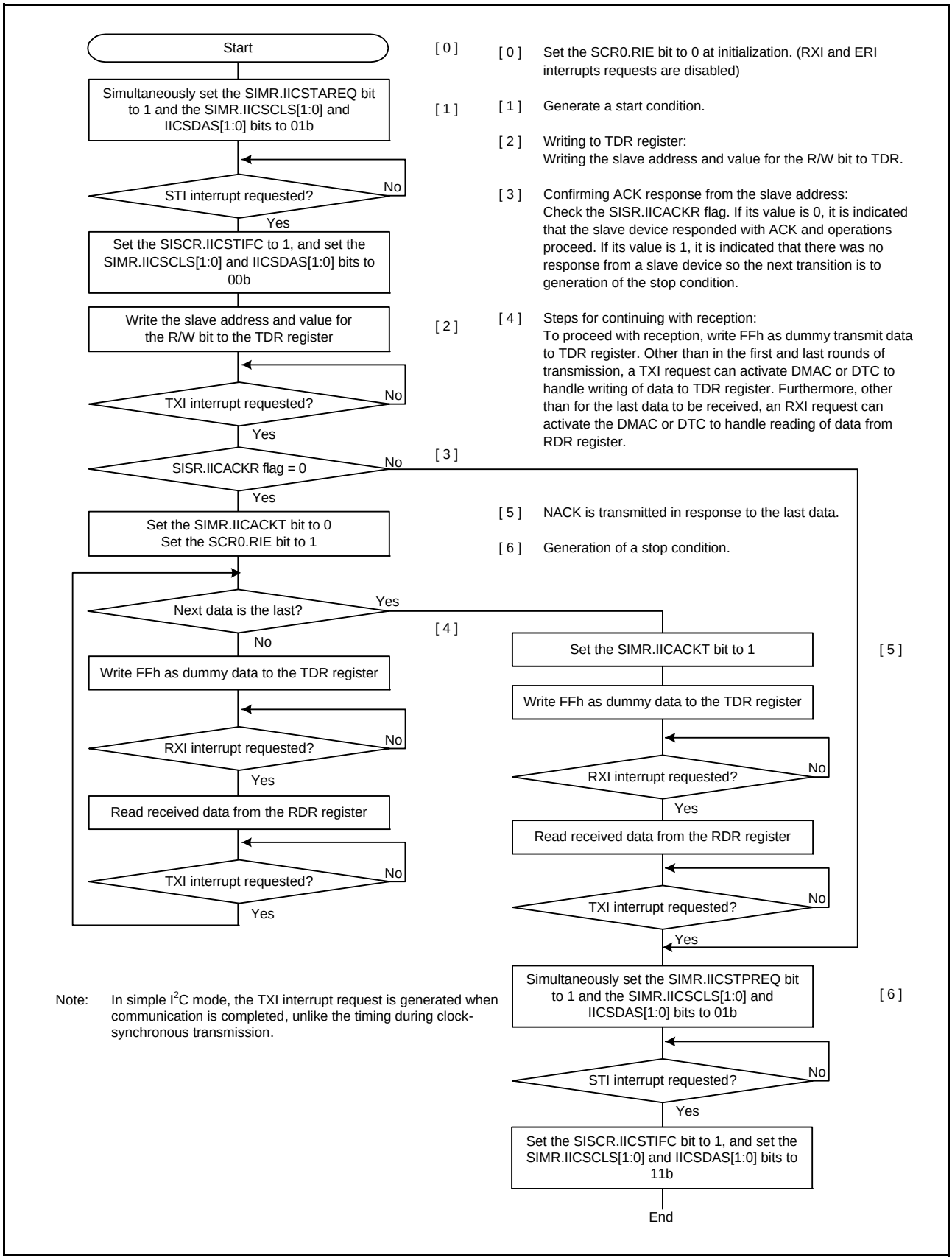


Figure 31.91 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 1, and Transmission Interrupts and Reception Interrupts in Use)

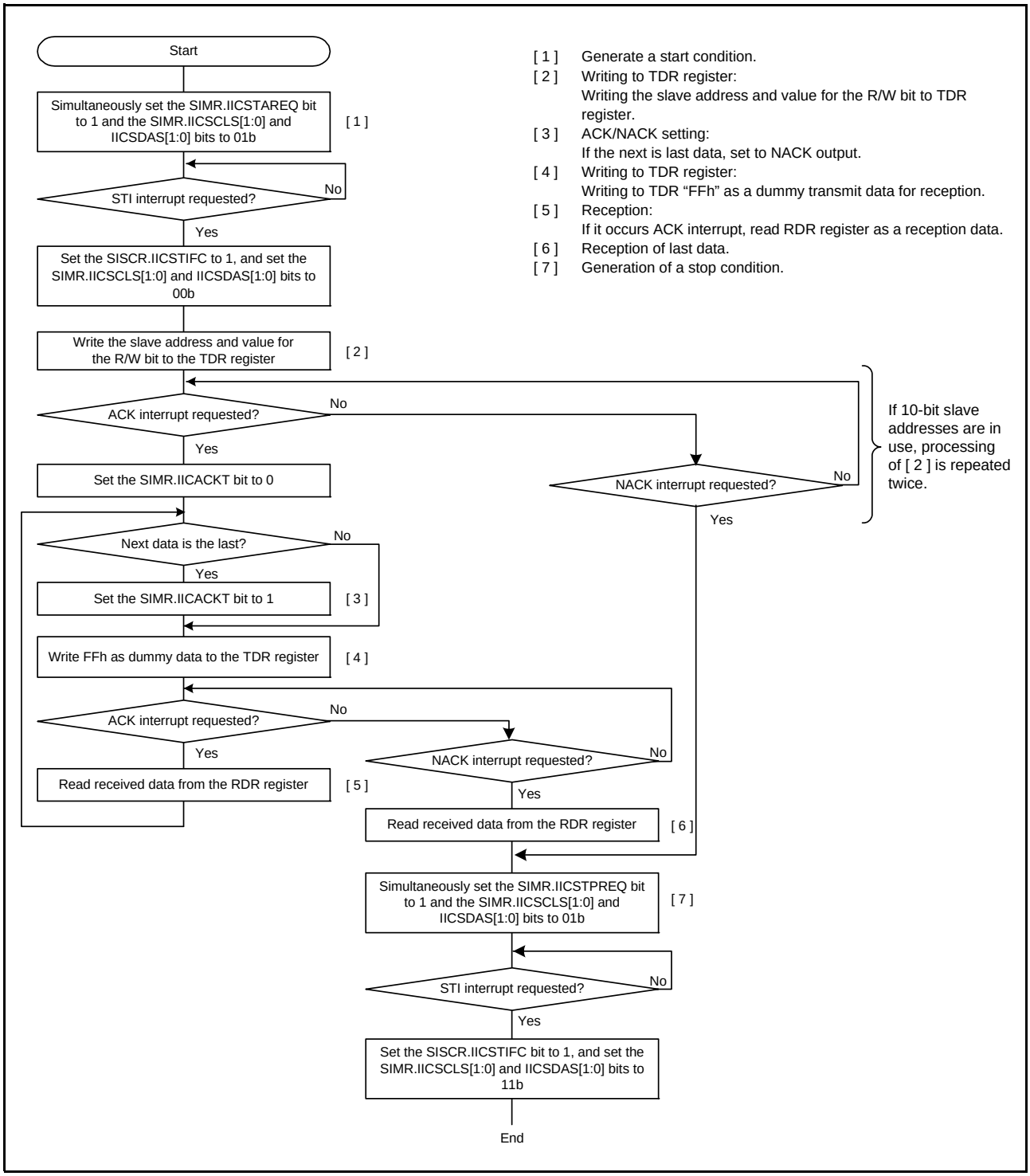


Figure 31.92 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 0, and ACK Interrupts and NACK Interrupts are in Use)

31.10 Operation in Clock Synchronous Mode

Figure 31.93 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.

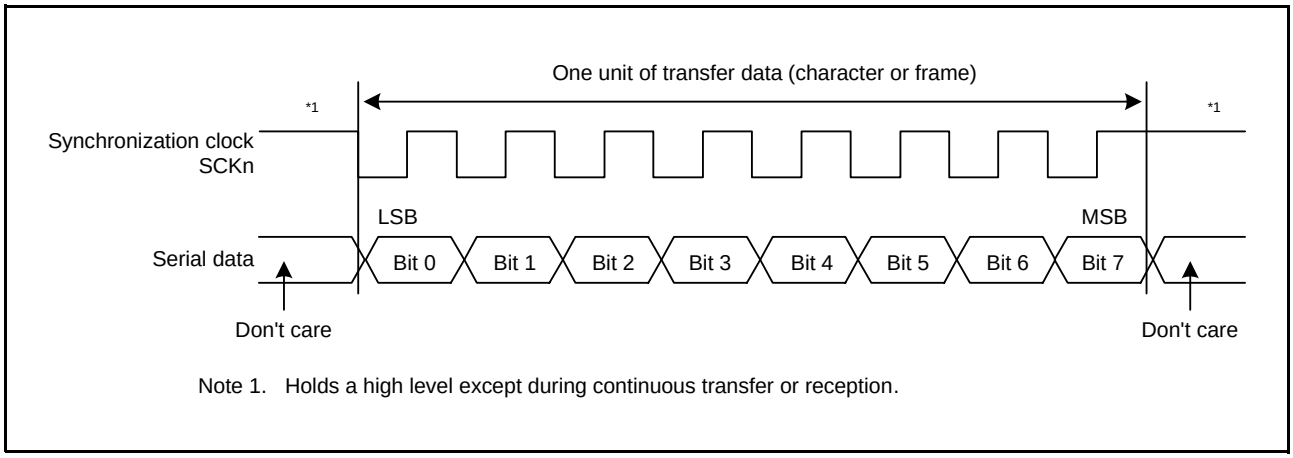


Figure 31.93 Data Format in Clock Synchronous Serial Communications (LSB First, CPHA Bit = 1, CPOL Bit = 1)

31.10.1 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.

31.10.2 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

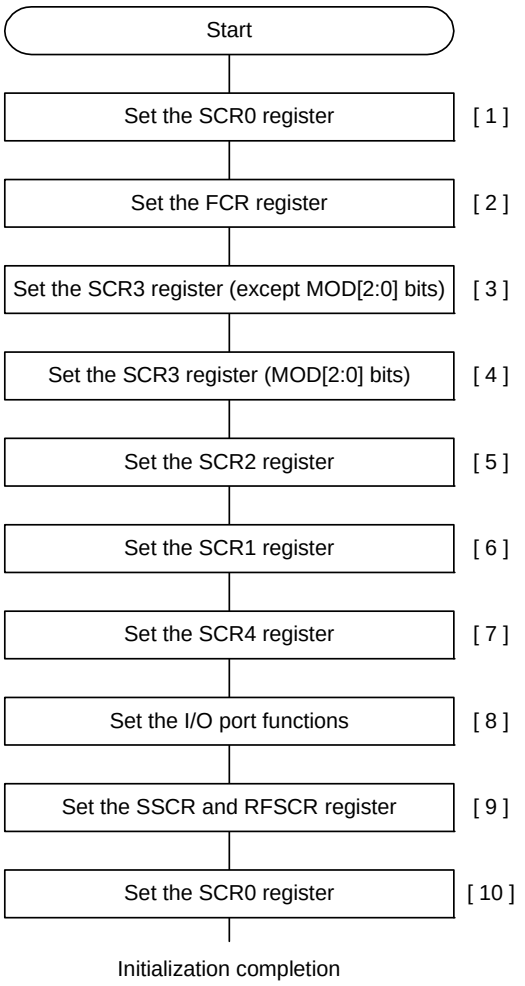
- The SCR0.RE or SCR0.TE bit is 1
- The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when SCR0.RE bit = 1)
- Data written in the transmit FIFO (TDR register) is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]

When the conditions for low-level output are not satisfied

31.10.3 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 31.94.
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 31.94 Example of RSCI Initialization Flowchart (Clock Synchronous Mode)

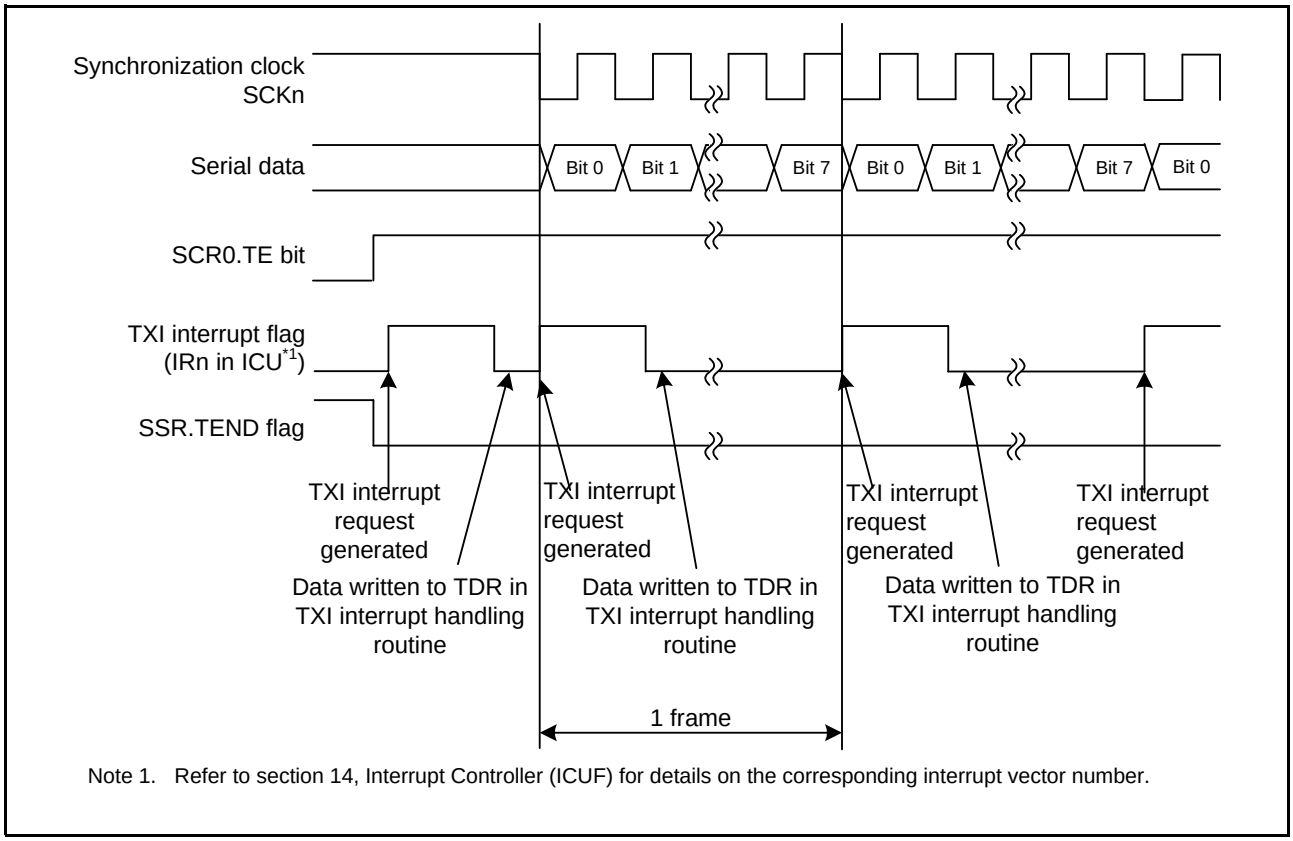
31.10.4 Serial Data Transmission (Clock Synchronous Mode)

(1) Non-FIFO Mode

Figure 31.95 to Figure 31.97 show an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the RSCI operates as described below.

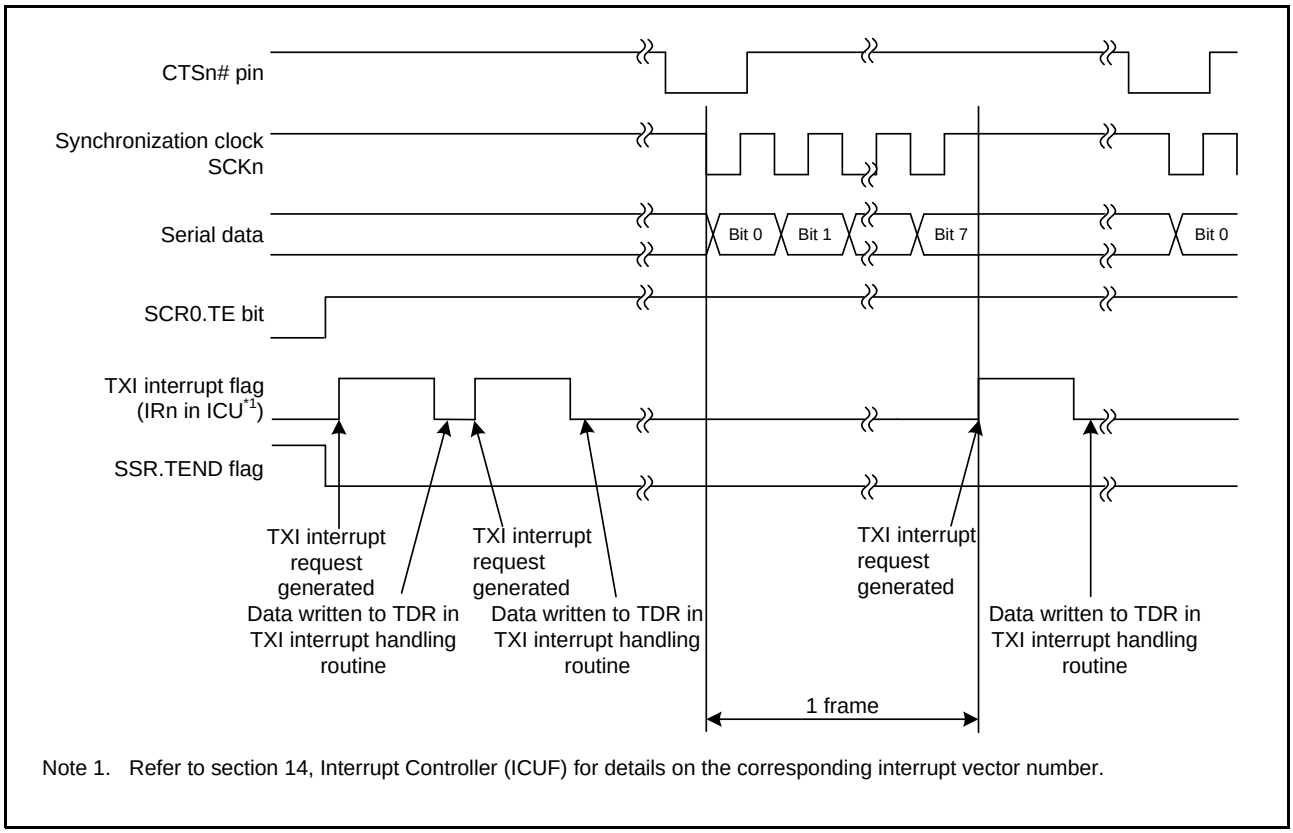
1. When data is written to the TDR register in the TXI interrupt routine, the RSCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the SCR0.TIE bit and the SCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
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 31.98 shows a sample flowchart of serial data transmission.



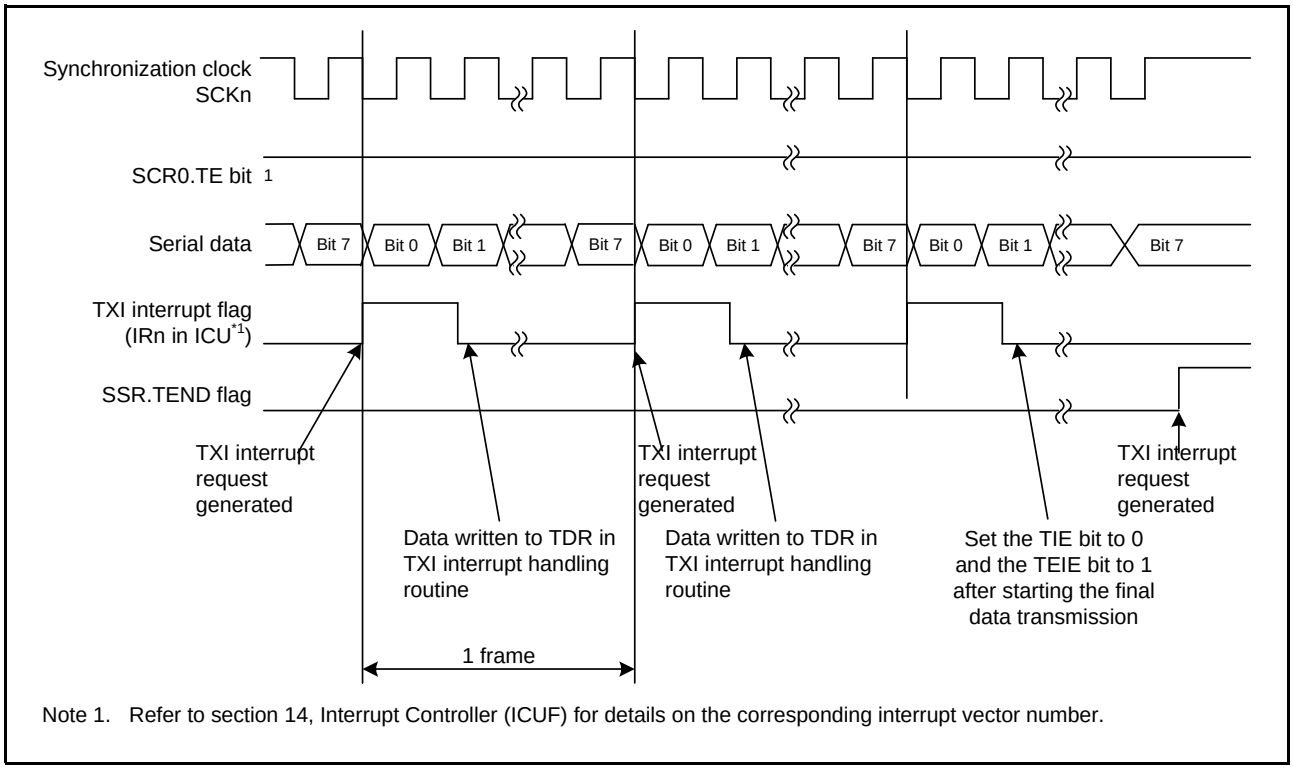
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.95 Serial Transmission Example in Clock Synchronous Mode (1) (CTS Function Not Used/ Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



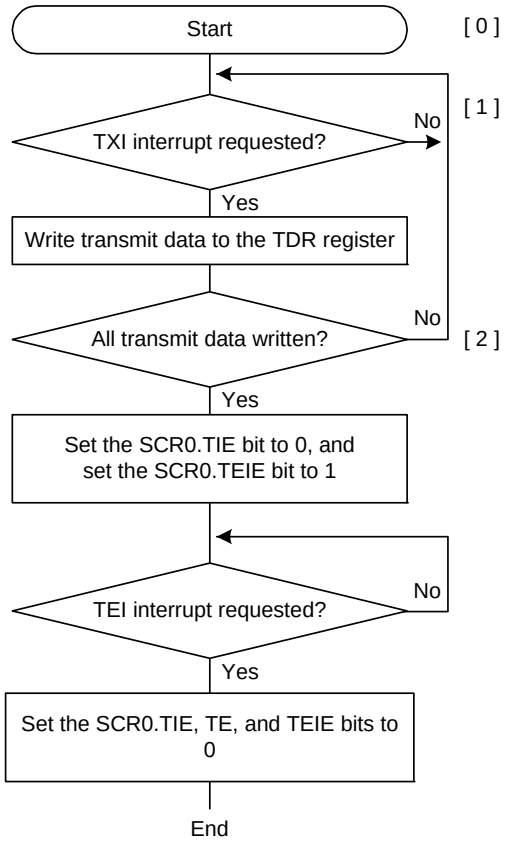
Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.96 Serial Transmission Example in Clock Synchronous Mode (2) (CTS Function Used/Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.97 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.

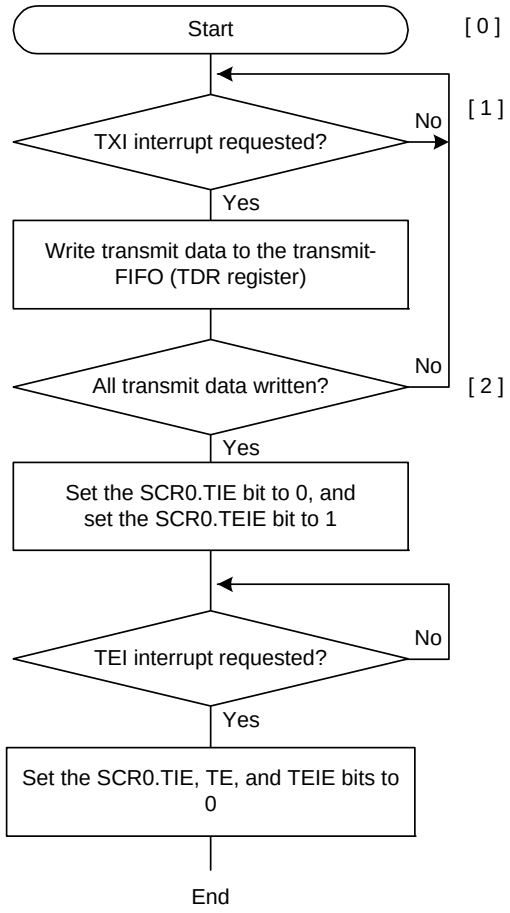
Figure 31.98 Example Flowchart of Serial Transmission in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 31.99 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.

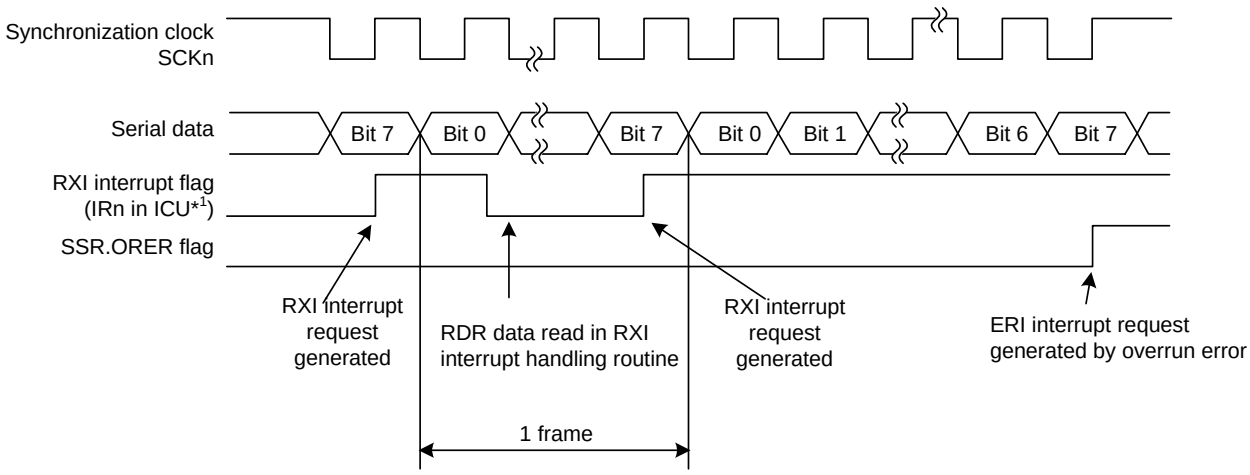
Figure 31.99 Example Flowchart of Serial Transmission in Clock Synchronous Mode (FIFO Mode)

31.10.5 Serial Data Reception (Clock Synchronous Mode)

(1) Non-FIFO Mode

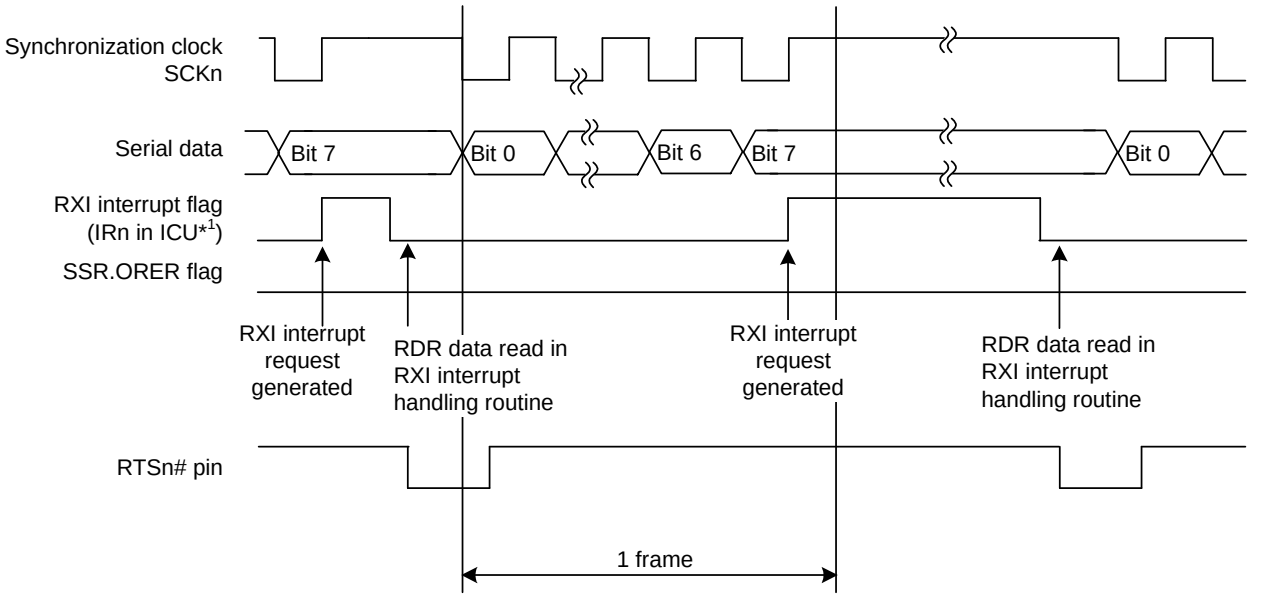
Figure 31.100 and Figure 31.101 show operation examples of serial data reception in clock synchronous mode.
The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.100 Example of Operation for Serial Reception in Clock Synchronous Mode (1)
(When RTS Function is Not Used/CPHA Bit = 1, CPOL Bit = 1)

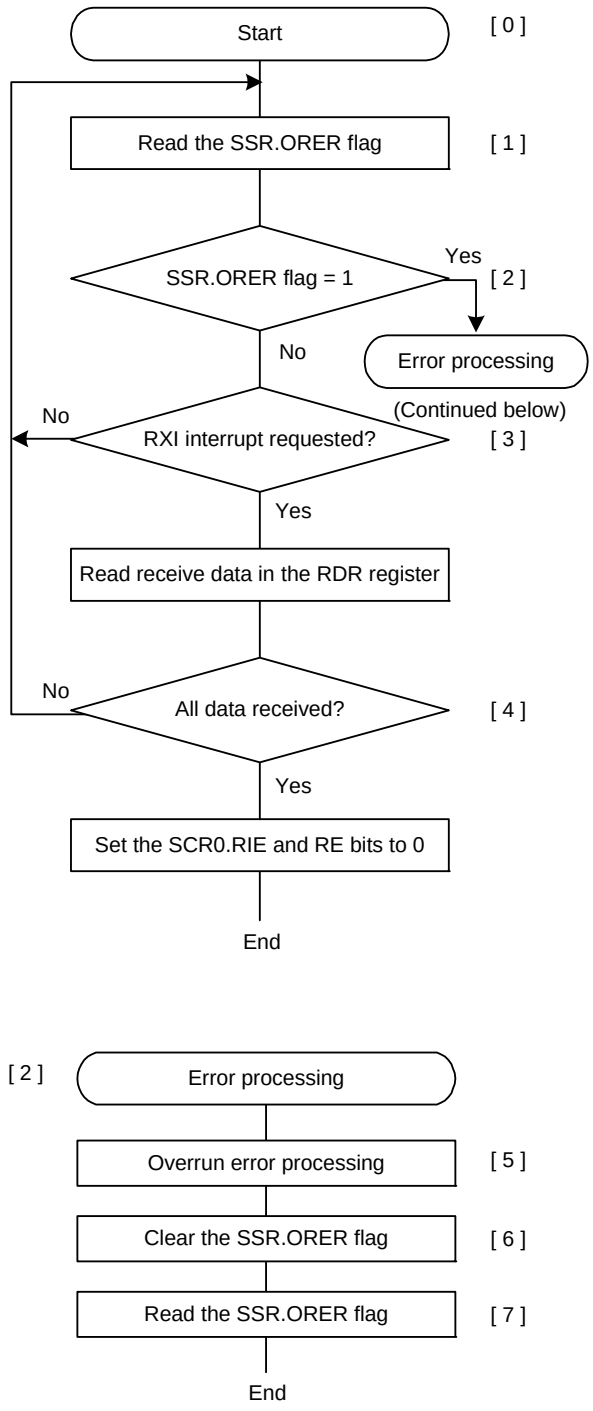


Note 1. Refer to section 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.101 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 31.102 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.

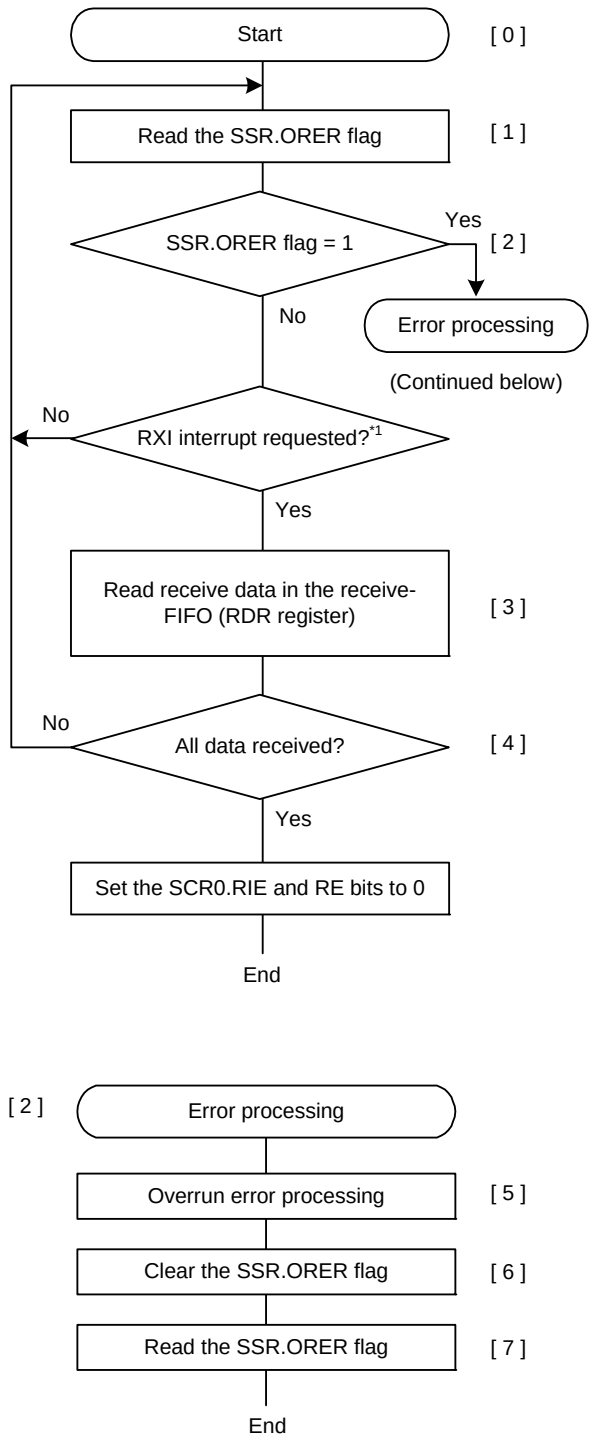
Figure 31.102 Example Flowchart of Serial Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 31.103 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 31.103 Example Flowchart of Serial Reception in Clock Synchronous Mode (FIFO Mode)

31.10.6 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 31.104 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.

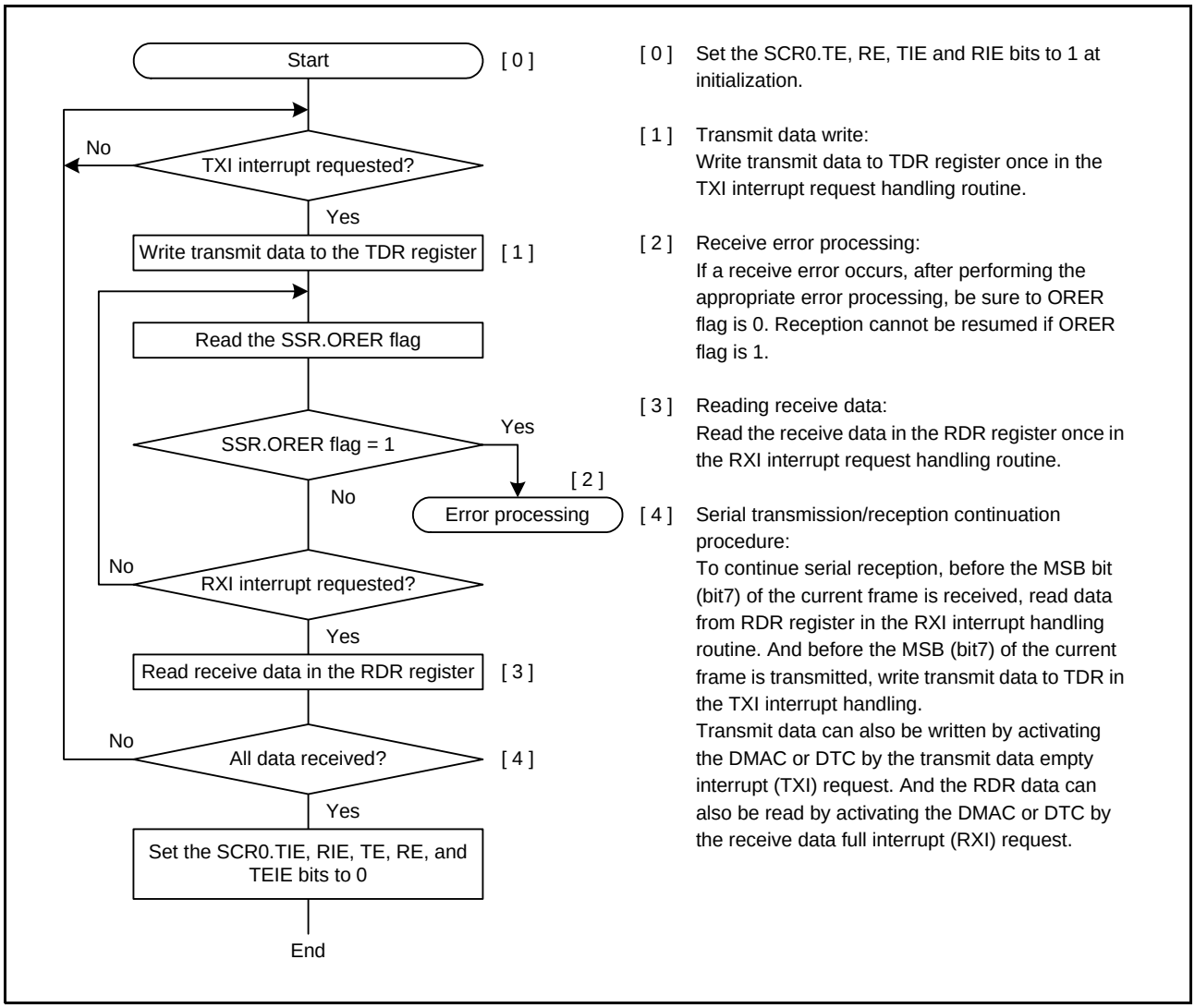


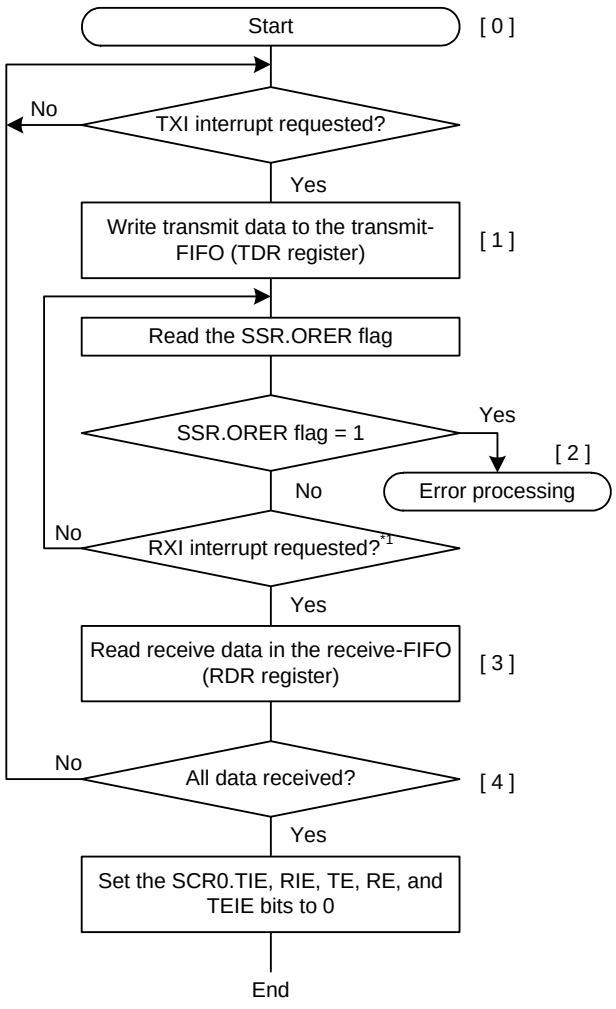
Figure 31.104 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 31.105 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.

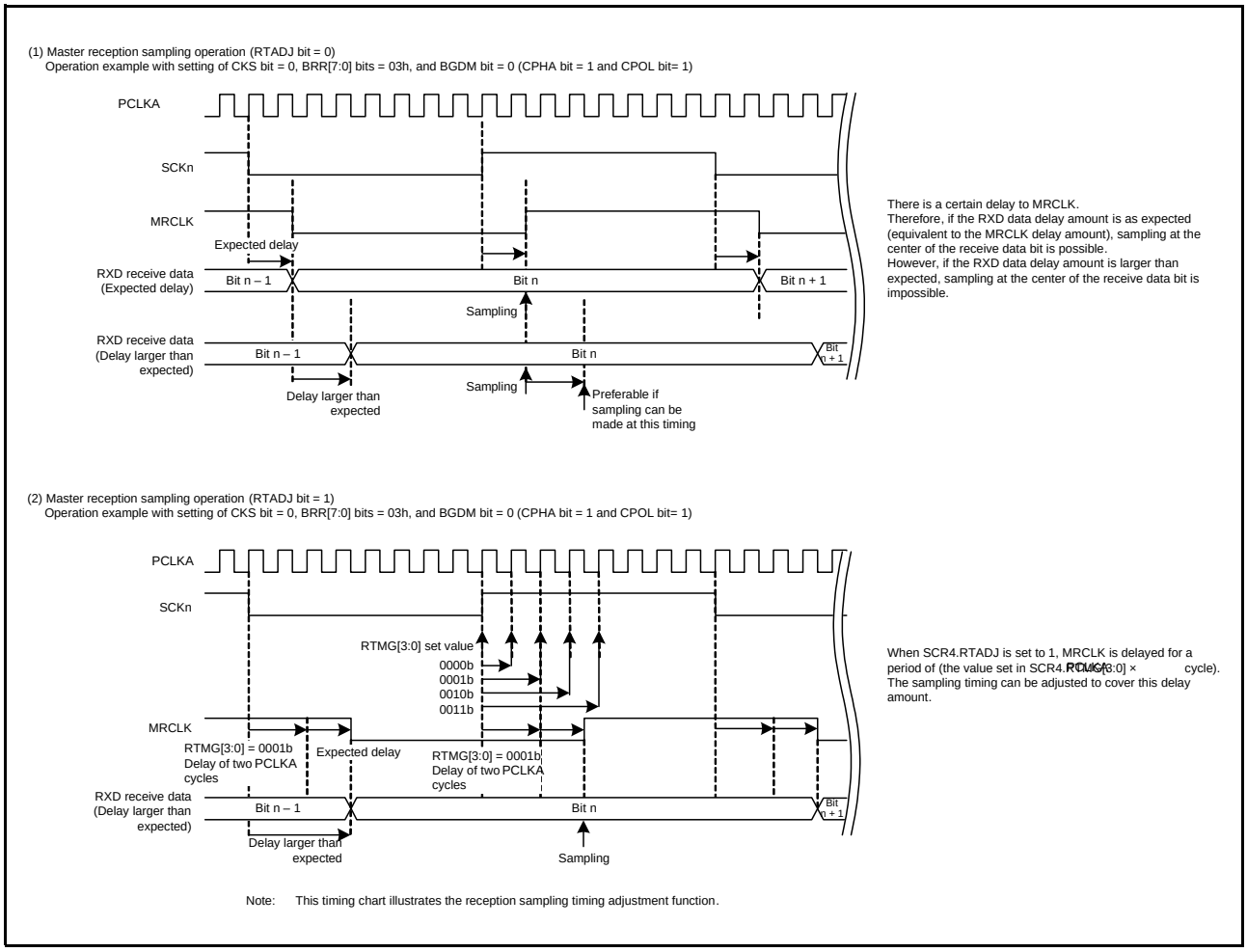
Figure 31.105 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (FIFO Mode)

31.10.7 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 31.106 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master) and Simple SPI Mode (Master)

31.11 Operation in Simple SPI Mode

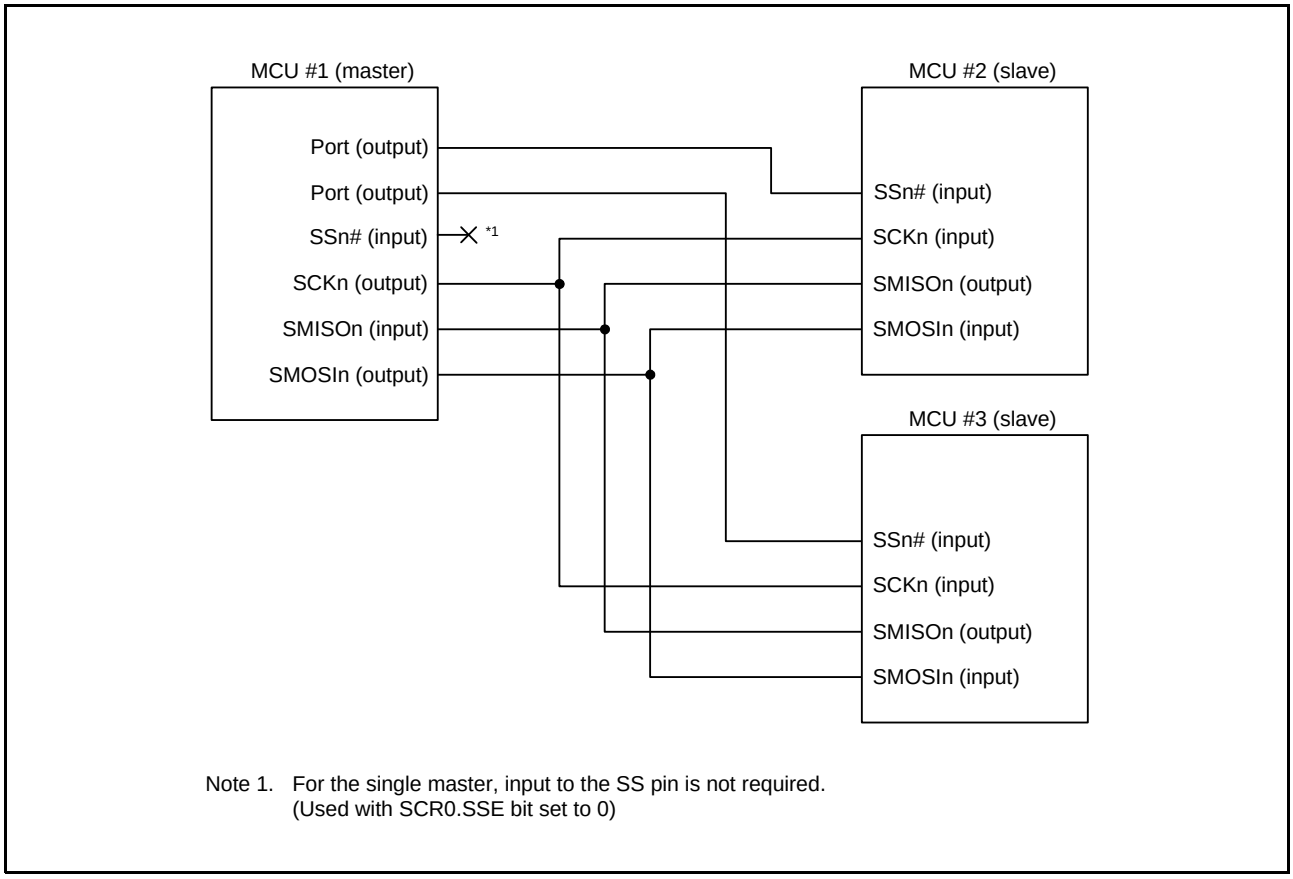
As an extended function, the RSCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

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 31.107 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 31.107 Example of Connections via a Simple SPI Mode

31.11.1 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 31.40 lists the states of pins according to the mode and the level on the SSn# pin.

Table 31.40 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).

31.11.2 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.

31.11.3 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.

31.11.4 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 31.108 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

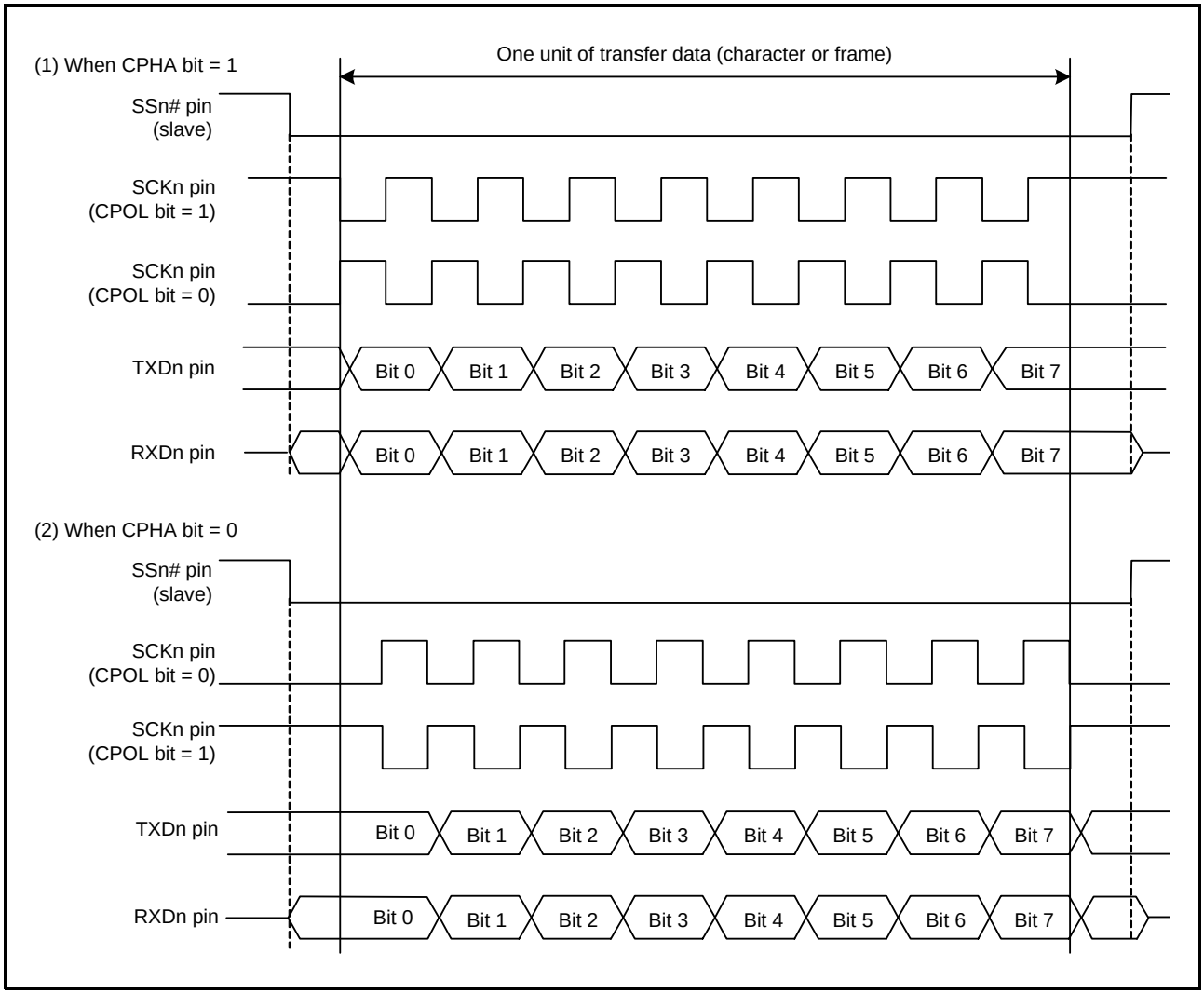


Figure 31.108 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

31.11.5 RSCI Initialization (Simple SPI Mode)

The RSCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 31.94, 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.

31.11.6 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.

31.11.7 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 31.10.7, Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

31.12 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 31.109 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.

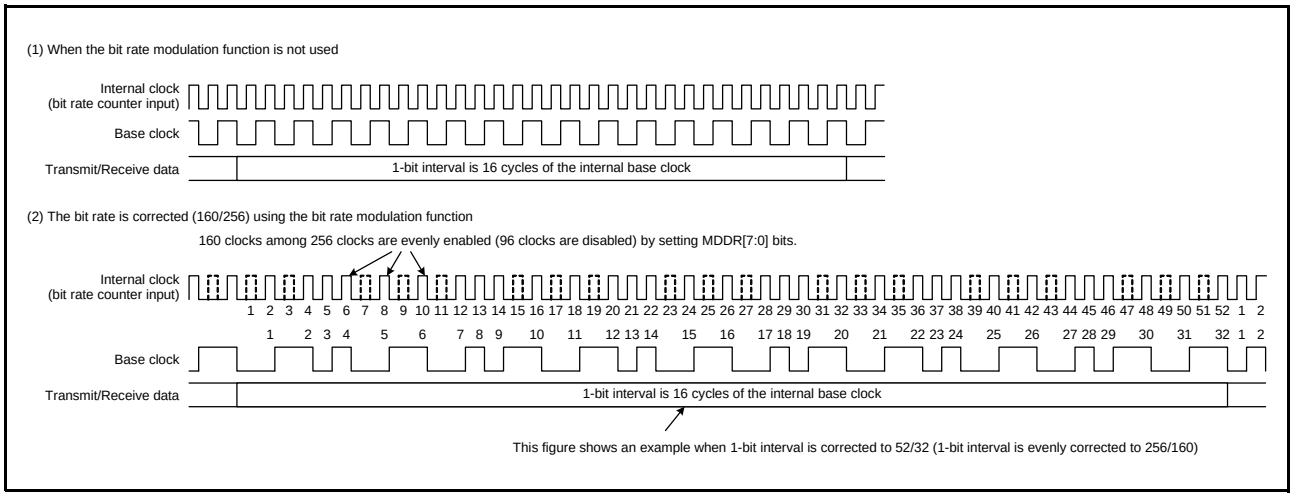


Figure 31.109 Example of Base Clock when Using the Bit Rate Modulation Function

31.13 Noise Cancellation Function

Figure 31.110 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.

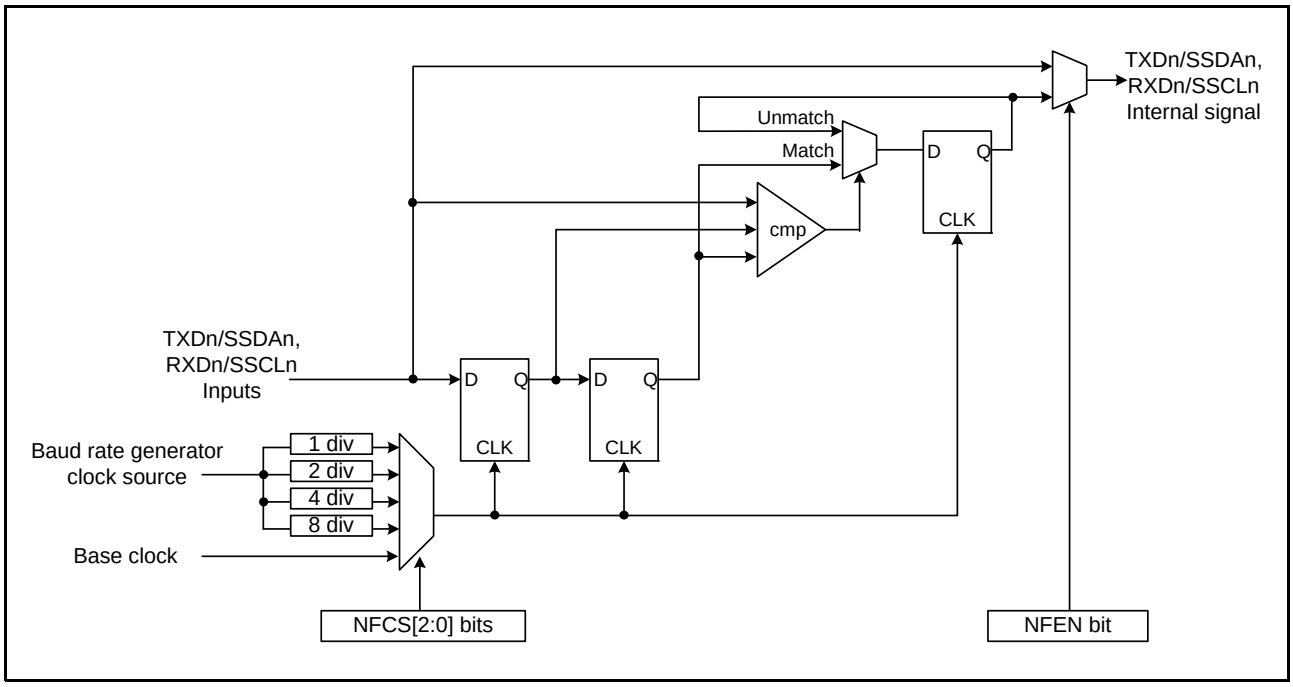


Figure 31.110 Block Diagram of Digital Noise Filter

31.14 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 31.2.13, 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.

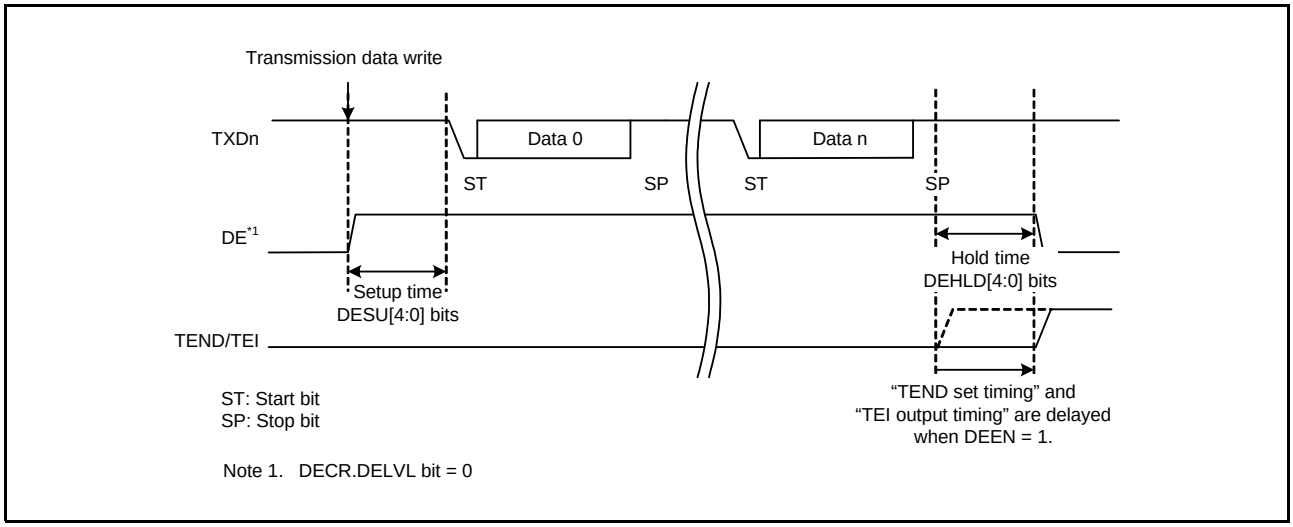


Figure 31.111 Image Waveform for RS-485 Driver Control DE Signal Output

31.15 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 31.41 shows the relationship between the TINV and LOOP bit settings and the received data.

Table 31.41 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 31.112 shows the configuration of the shift register input/output path in loopback mode.

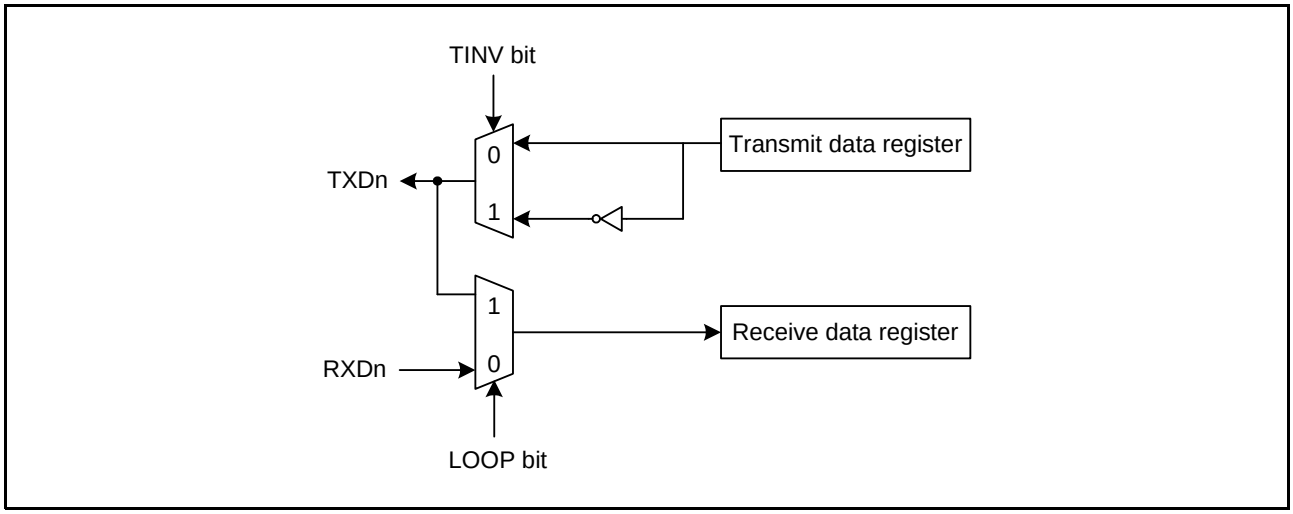


Figure 31.112 Shift Register Input/Output Configuration Image in Loopback Mode

31.16 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.

31.17 Interrupt Signal

Table 31.42 lists RSCI interrupt signals.

The interrupt explanation corresponding to each operation mode is described in sections 31.17.2 to 31.17.5. Also, TXI and RXI have an interrupt buffer function. Refer to section 31.17.1, 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 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa) for how to set DTC or DMAC.

Table 31.42 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

31.17.1 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.

31.17.2 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 31.43 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 31.44 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 31.43 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 31.44 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

31.17.3 Interrupt in Smart Card Interface Mode

Table 31.45 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 31.45 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 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa).

31.17.4 Interrupts in Simple I2C Mode

Table 31.46 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 31.46 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.

31.17.5 Interrupts in Extended Serial Mode

Table 31.47 lists interrupt sources in extended serial mode.

Table 31.47 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 31.72), 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).

31.18 Usage Notes

31.18.1 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.

31.18.2 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 31.113 shows a sample flowchart for transition to software standby mode during transmission. Figure 31.114 and Figure 31.115 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 31.116 shows a sample flowchart for reception to software standby mode during reception.

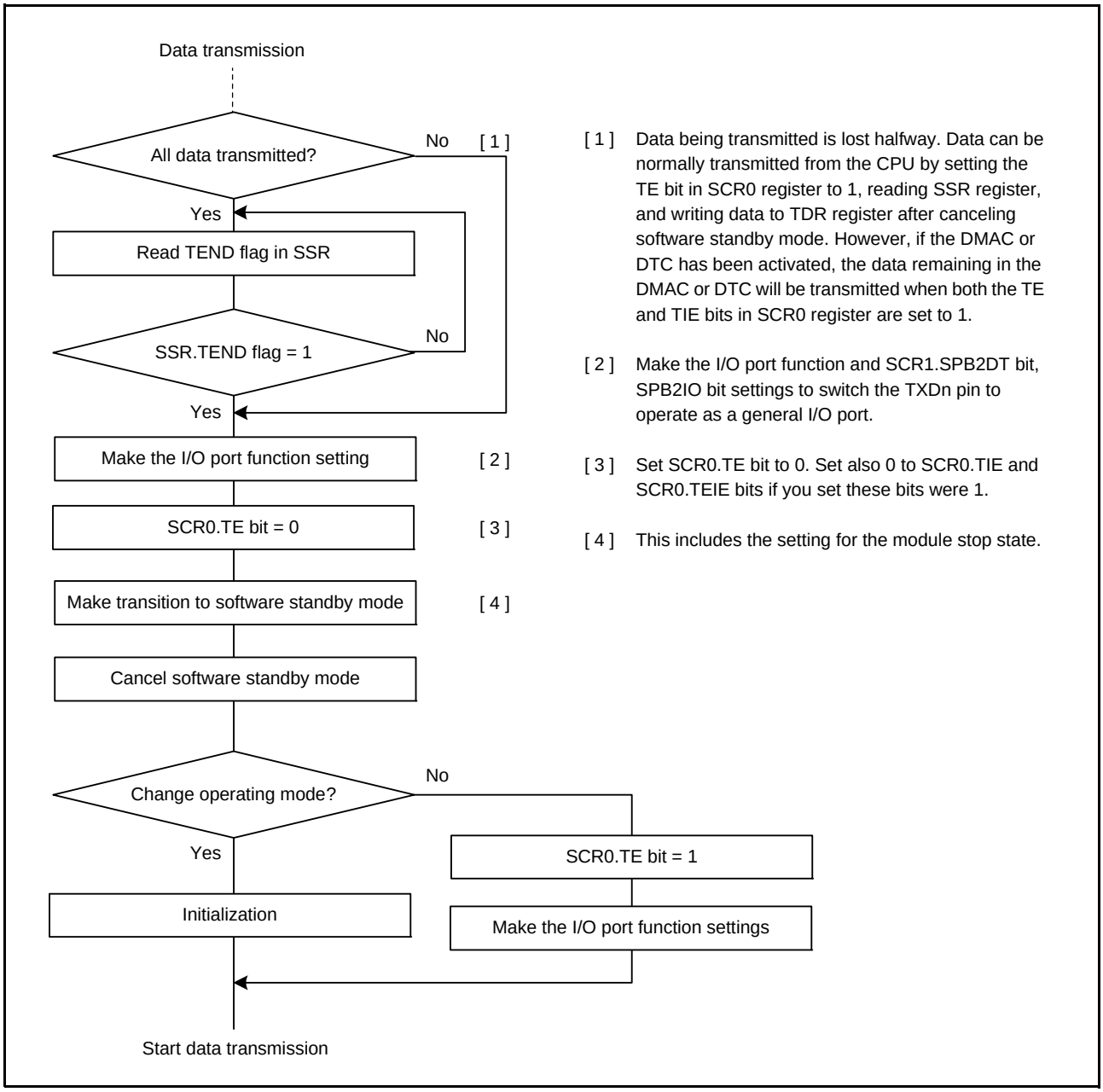


Figure 31.113 Example of Flowchart for Transition to Software Standby Mode during Transmission

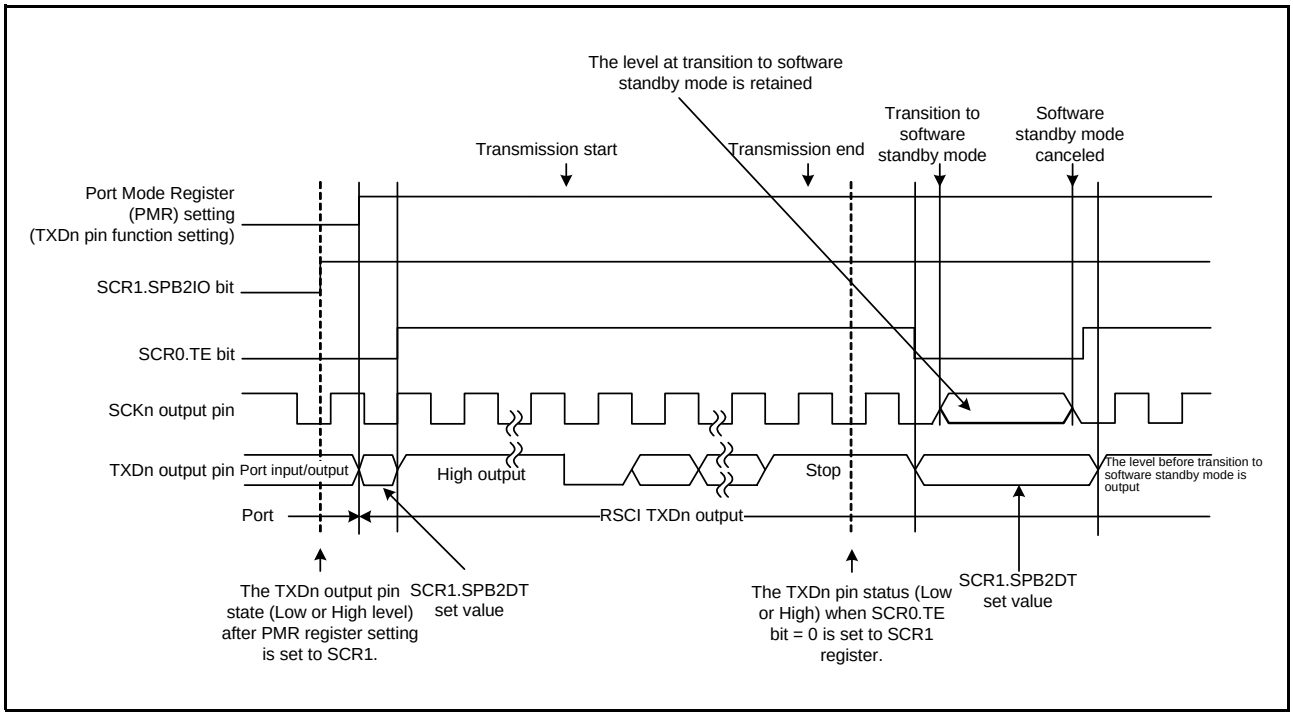


Figure 31.114 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

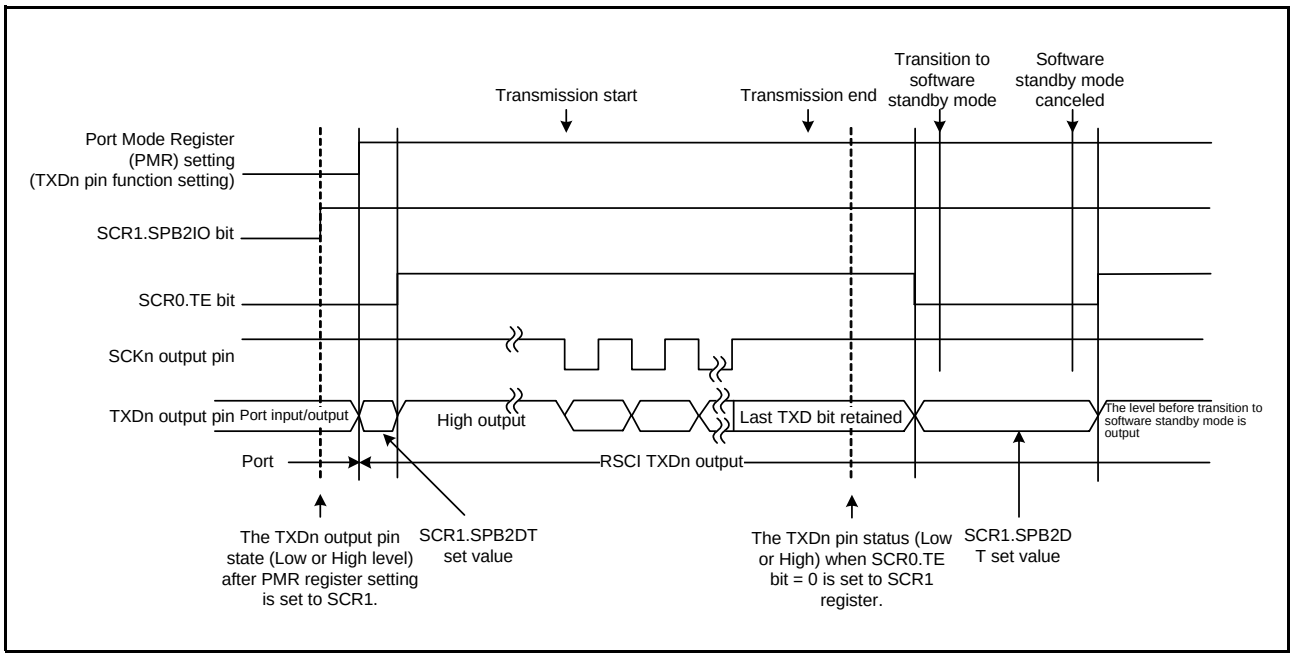


Figure 31.115 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

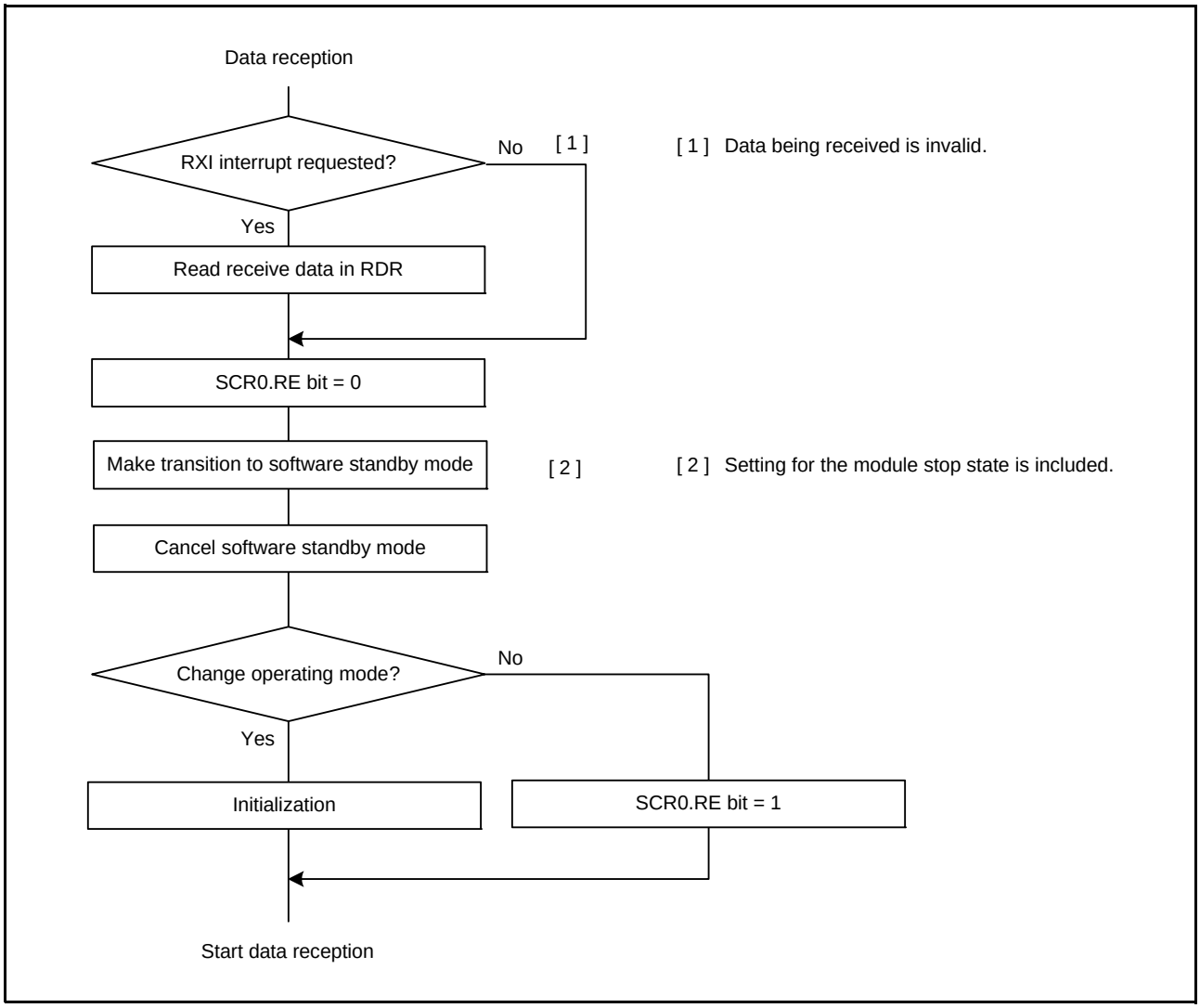


Figure 31.116 Example of Flowchart for Transition to Software Standby Mode during Reception

31.18.3 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.

31.18.4 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.

31.18.5 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).

31.18.6 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.

31.18.7 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.

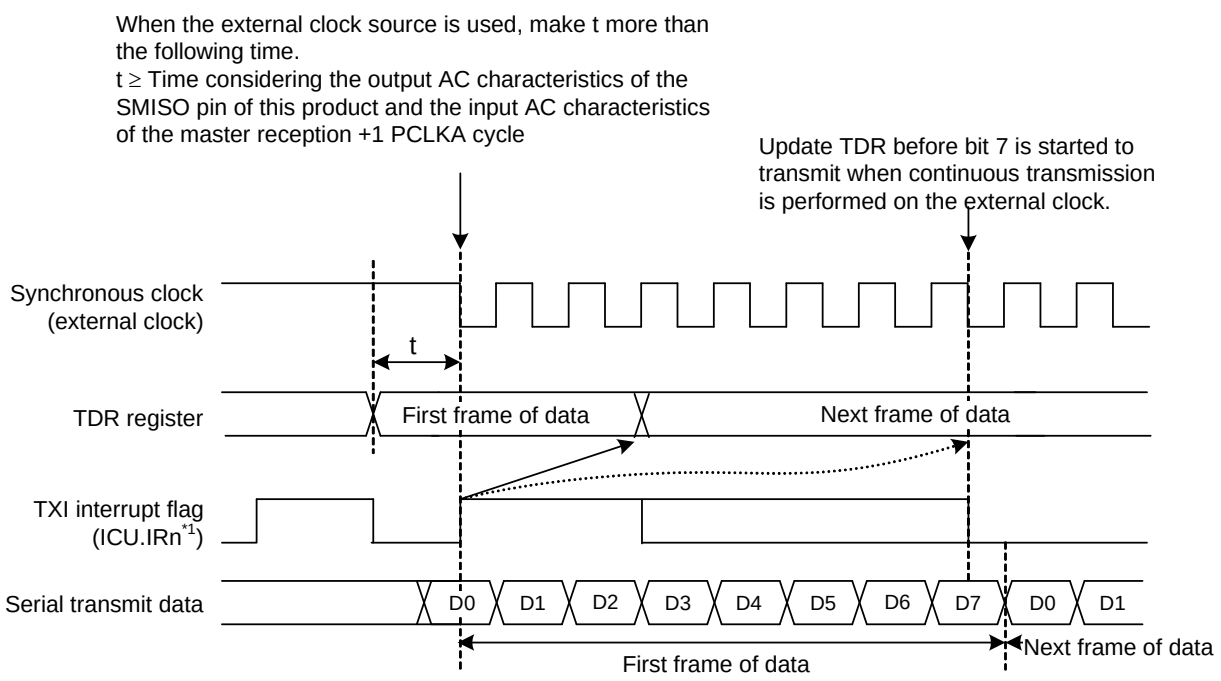
(1) Start of Transmission

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 31.117)
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.

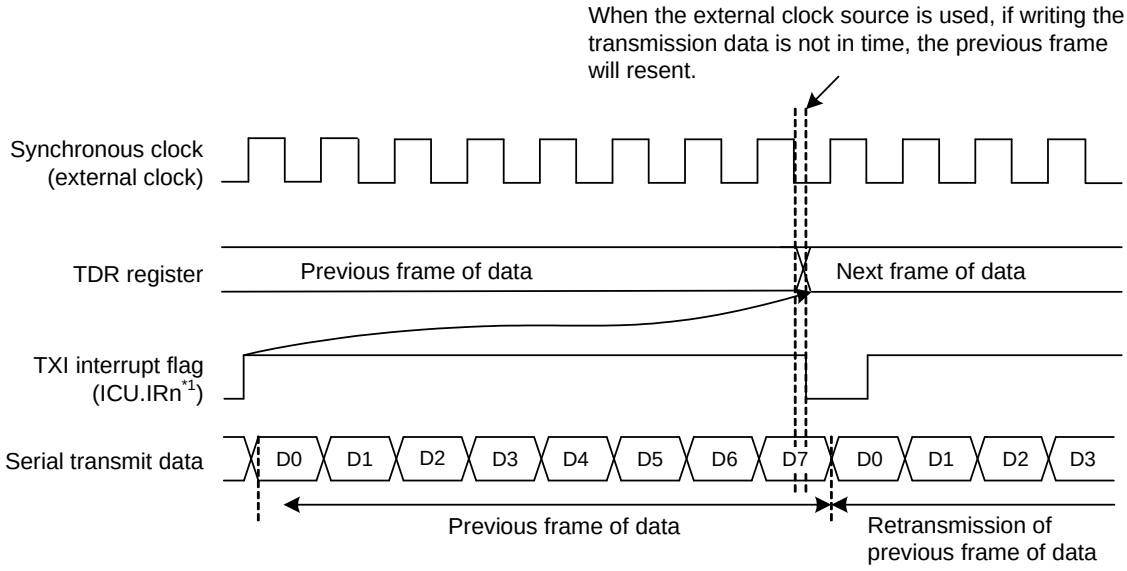
(2) Continuous Transmission

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 31.117)

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 14, Interrupt Controller (ICUF) for details on the corresponding interrupt vector number.

Figure 31.117 Restrictions on Use of External Clock in Clock Synchronous Transmission

31.18.8 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.

31.18.9 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 14, Interrupt Controller (ICUF).

- 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.

31.18.10 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 31.118. 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.

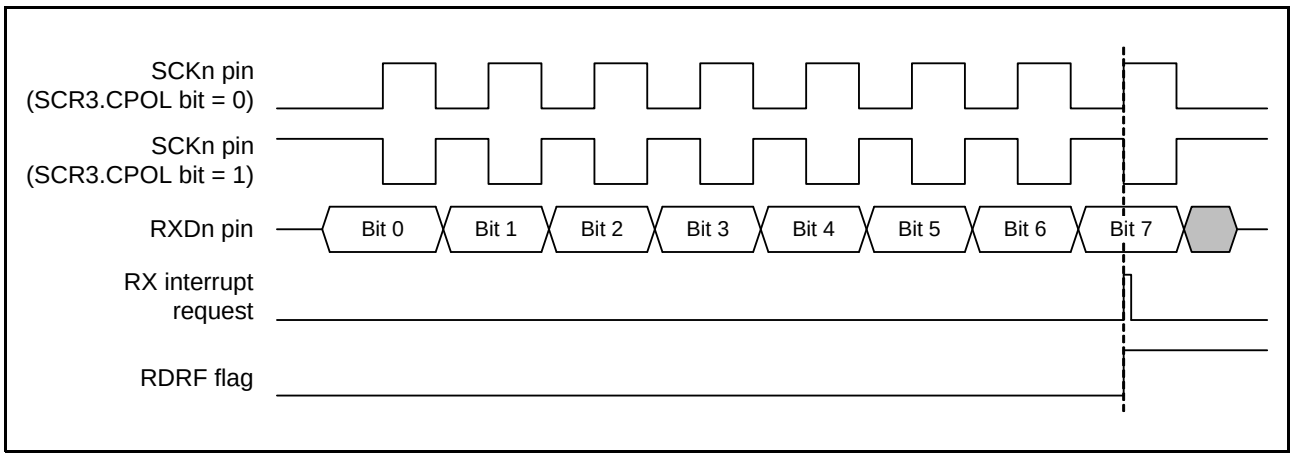


Figure 31.118 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(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.

31.18.11 Note on Transmit Enable Bit (TE Bit)

In initial register value, when SCR0.TE bit is 0, and the pin function is TXDn, the pin outputs high impedance.
So please make sure that the TXDn line won't be high impedance by the following one of ways.

(a) The pull-up resistance is connected to the TXDn line.
(b) Before SCR0.TE bit is 0, the function of the pin is changed to general-purpose input port or output port. And after SCR0.TE bit is 1, the function of the pin is changed o "TXDn".
(c) In asynchronous mode and manchester mode, you can set SCR1 register and decided level of TXDn pin during TE is 0.

In the Simple SPI mode slave operation, the RXDn pin operates in the same way as the above TXDn pin, so please deal with (a) or (b) in the same way. ((c) can not be used.)

31.18.12 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

31.18.13 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.

31.18.14 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).

31.18.15 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 related to reception, and then write 0 to the SCR0.RE bit.

32. I2C-bus Interface (RIICa)

This MCU has a two-channel I2C-bus interface (RIIC0, RIIC2).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

32.1 Overview

Table 32.1 lists the specifications of the RIIC, Figure 32.1 shows a block diagram of the RIIC. Table 32.2 lists the I/O pins of the RIIC.

Table 32.1 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 is supported (up to 400 kbps)
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 32.1 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

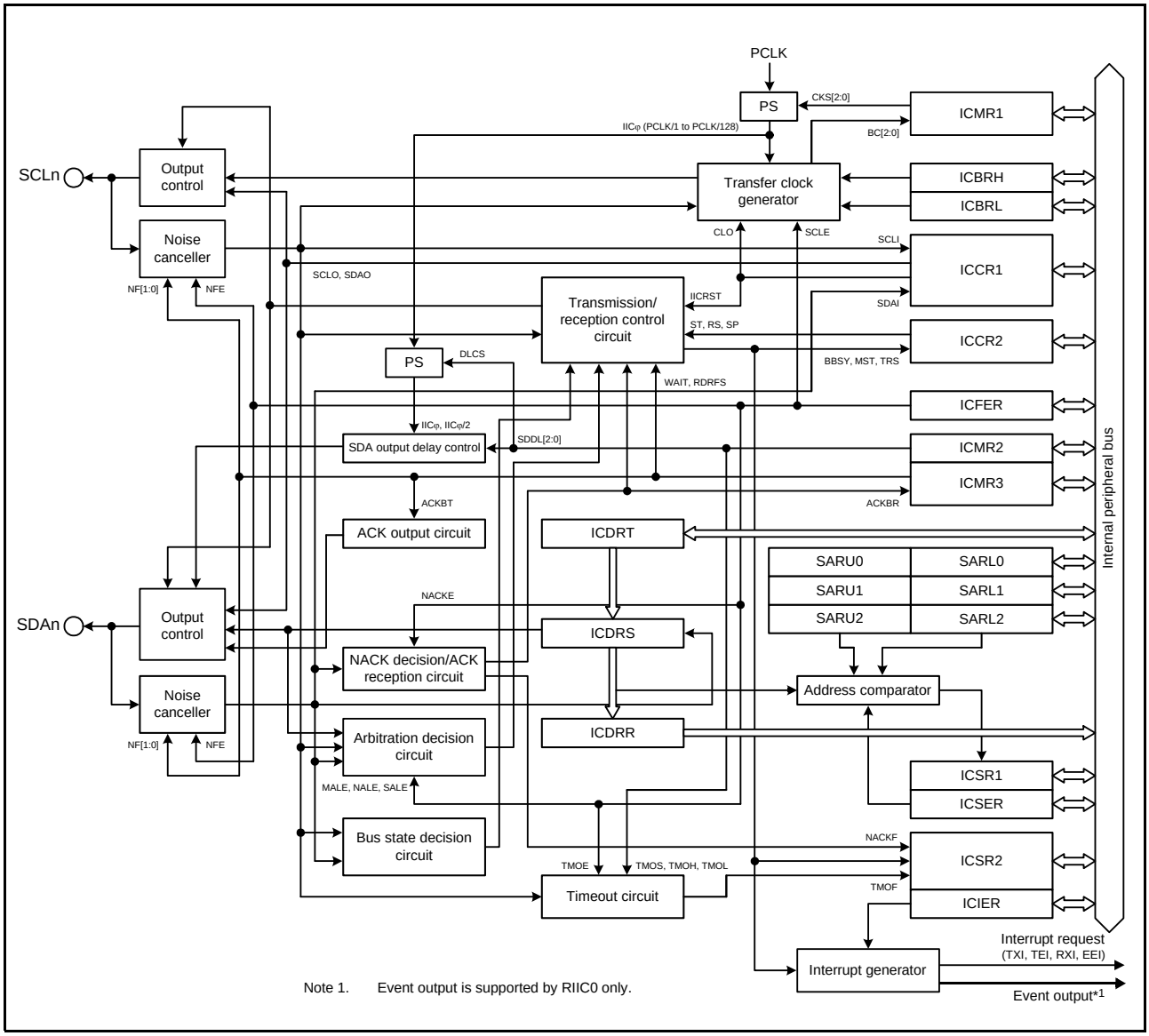


Figure 32.1 RIIC Block Diagram (n = 0, 2)

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 32.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin
RIIC2	SCL2	I/O	RIIC2 serial clock I/O pin
	SDA2	I/O	RIIC2 serial data I/O pin

32.2 Register Descriptions

32.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h, RIIC2.ICCR1 0008 8340h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDAn line is low. 1: SDAn line is high.	R
b1	SCLI	SCL Line Monitor	0: SCLn line is low. 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDAn pin low. 1: The RIIC has released the SDAn pin. • Write: 0: The RIIC drives the SDAn pin low. 1: The RIIC releases the SDAn pin. (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 32.11.2, 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 32.3 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 32.14, 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 32.3 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 32.3, 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.

32.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h, RIIC2.ICCR2 0008 8341h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition 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 32.10, 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 32.10, 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 32.10, 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)

32.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h, RIIC2.ICMR1 0008 8342h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock (IICφ) source for the RIIC. b6 b4 0 0 0: PCLK/1 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.

32.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h, RIIC2.ICMR2 0008 8343h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCLn line is 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 0: No output delay 0 0 1: 1 IICφ cycle 0 1 0: 2 IICφ cycles 0 1 1: 3 IICφ cycles 1 0 0: 4 IICφ cycles 1 0 1: 5 IICφ cycles 1 1 0: 6 IICφ cycles 1 1 1: 7 IICφ cycles • When ICMR2.DLCS bit is 1 (IICφ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICφ cycles 0 1 0: 3 or 4 IICφ cycles 0 1 1: 5 or 6 IICφ cycles 1 0 0: 7 or 8 IICφ cycles 1 0 1: 9 or 10 IICφ cycles 1 1 0: 11 or 12 IICφ cycles 1 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16-bit counter. In short mode, the counter functions as a 14-bit counter. While the SCLn line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, refer to section 32.11.1, 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 32.5, 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))

32.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h, RIIC2.ICMR3 0008 8344h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	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 32.6, 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.

32.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC2.ICFER 0008 8345h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

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 32.11.1, 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 32.8.2, 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.

32.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h, RIIC2.ICSER 0008 8346h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.

When this bit is set to 1, the slave address set in registers SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.

When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.

For details on the device-ID address detection, refer to section 32.7.3, 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.

32.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h, RIIC2.ICIER 0008 8347h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	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.

32.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h, RIIC2.ICSR1 0008 8348h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

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

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL 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

32.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h, RIIC2.ICSR2 0008 8349h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	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 32.4 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

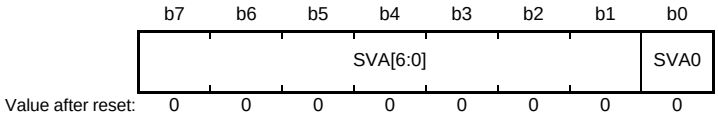
[Clearing conditions]

- 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.

32.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC2.SARL0 0008 834Ah, RIIC0.SARL1 0008 830Ch, RIIC2.SARL1 0008 834Ch, RIIC0.SARL2 0008 830Eh, RIIC2.SARL2 0008 834Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	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.

32.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC2.SARU0 0008 834Bh, RIIC0.SARU1 0008 830Dh, RIIC2.SARU1 0008 834Dh, RIIC0.SARU2 0008 830Fh, RIIC2.SARU2 0008 834Fh

	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)

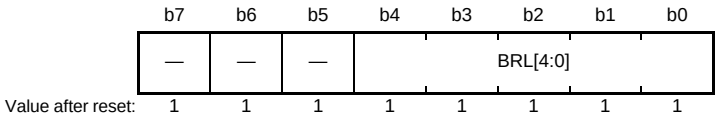
This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting 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.

32.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h, RIIC2.ICBRL 0008 8350h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low 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 32.8, 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))

32.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h, RIIC2.ICBRH 0008 8351h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High 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 32.5 lists examples of ICBRH/ICBRL settings.

Table 32.5 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)

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)

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)

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)

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
SCLn line fall time (tf): 400 kbps or less (Sm/Fm): 300 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.

32.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h, RIIC2.ICDRT 0008 8352h

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

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
The ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

32.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h, RIIC2.ICDRR 0008 8353h

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

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
The ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL line low one cycle before the RDRF flag is set to 1 next.

32.2.17 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.

32.3 Operation

32.3.1 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 32.2 shows the I2C-bus format, and Figure 32.3 shows the I2C-bus timing.

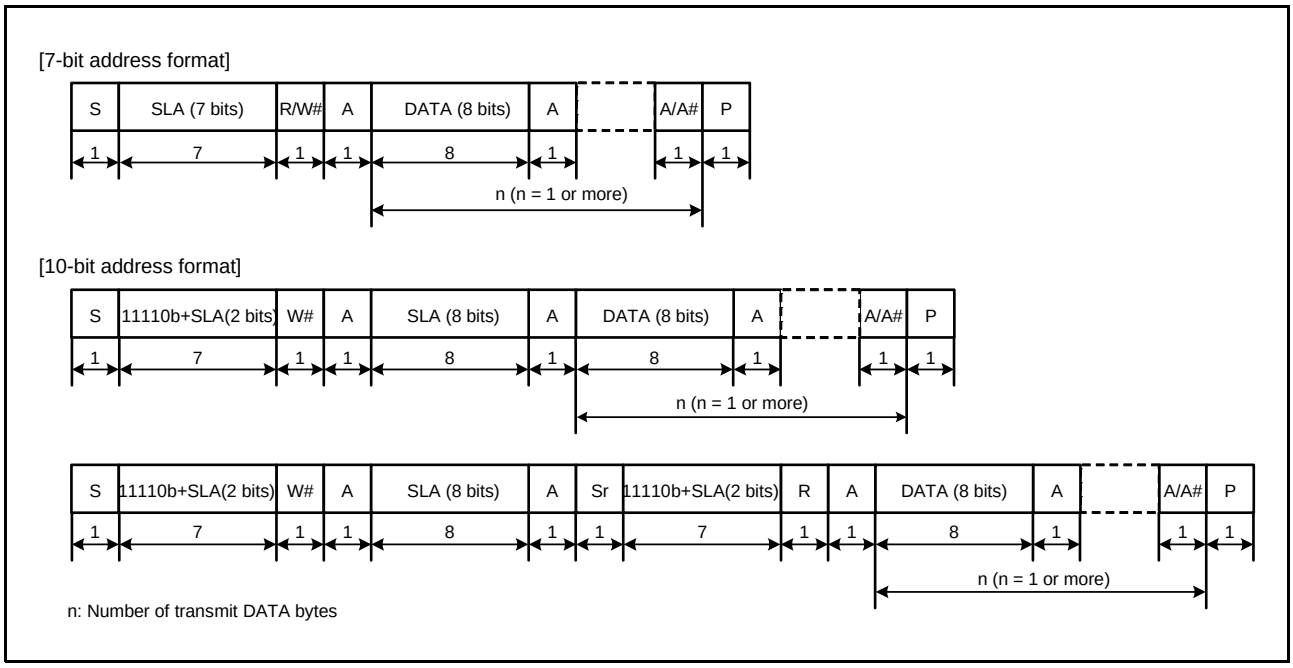


Figure 32.2 I2C-bus Format

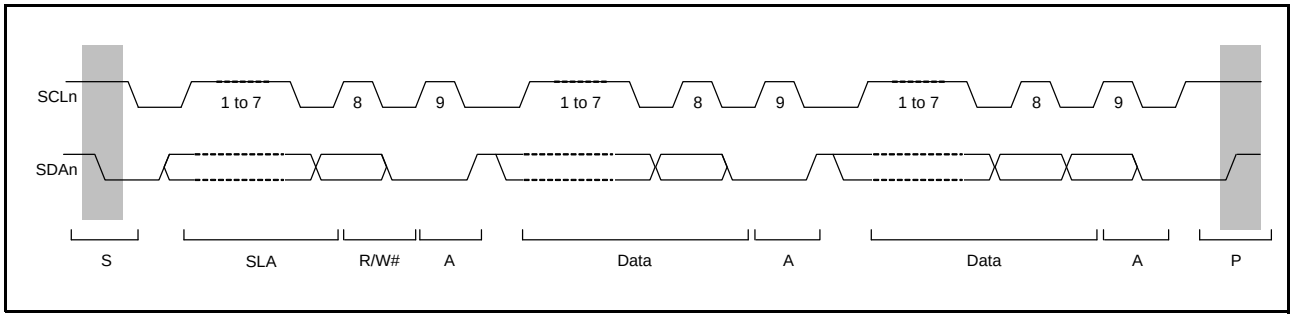


Figure 32.3 I2C-bus Timing (SLA = 7 Bits)

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.

32.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 32.4.
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 32.4). 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.

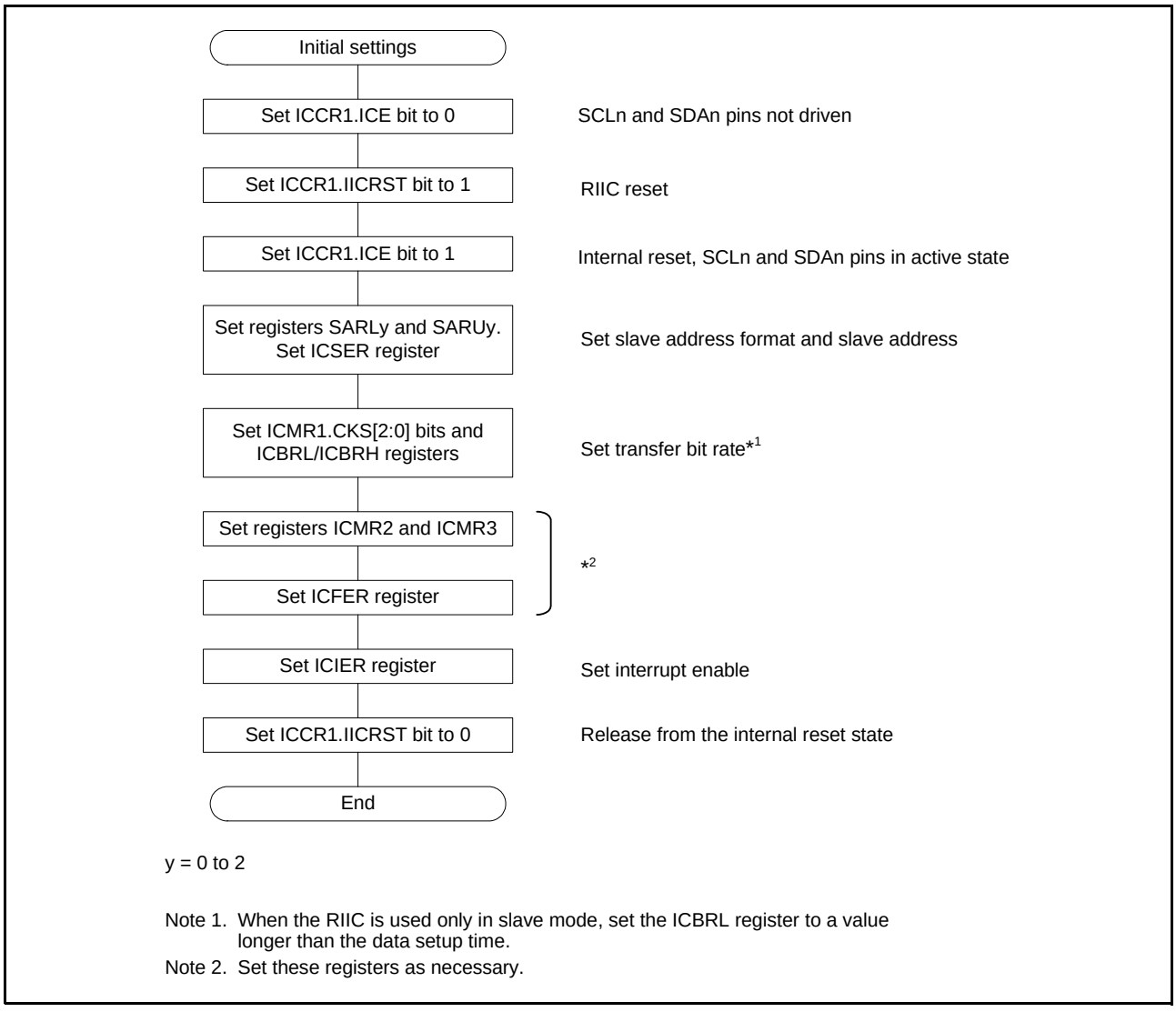


Figure 32.4 Example of RIIC Initialization Flowchart

32.3.3 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 32.5 shows an example of usage of master transmission and Figure 32.6 to Figure 32.8 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 32.3.2, 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.

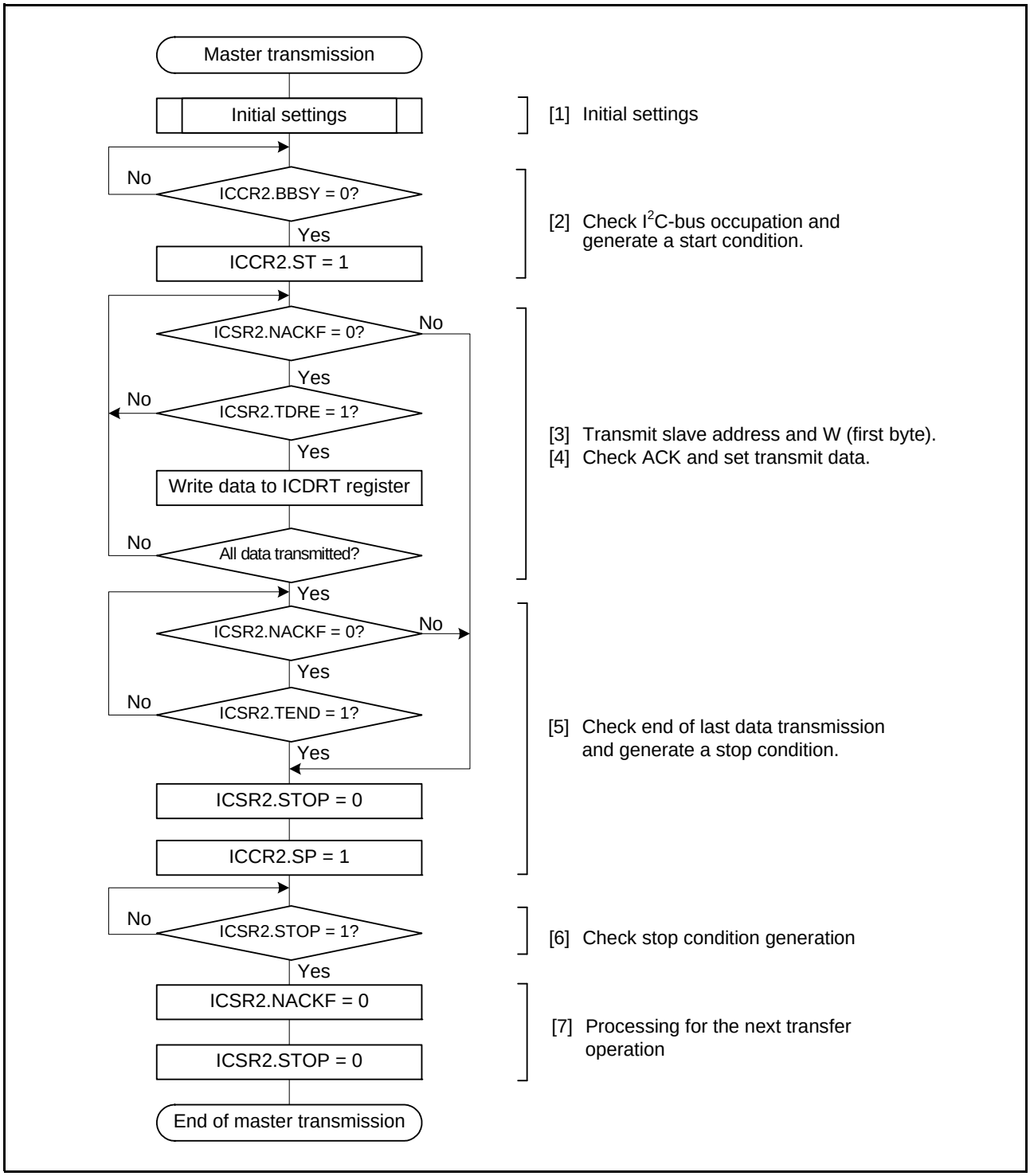


Figure 32.5 Example of Master Transmission Flowchart

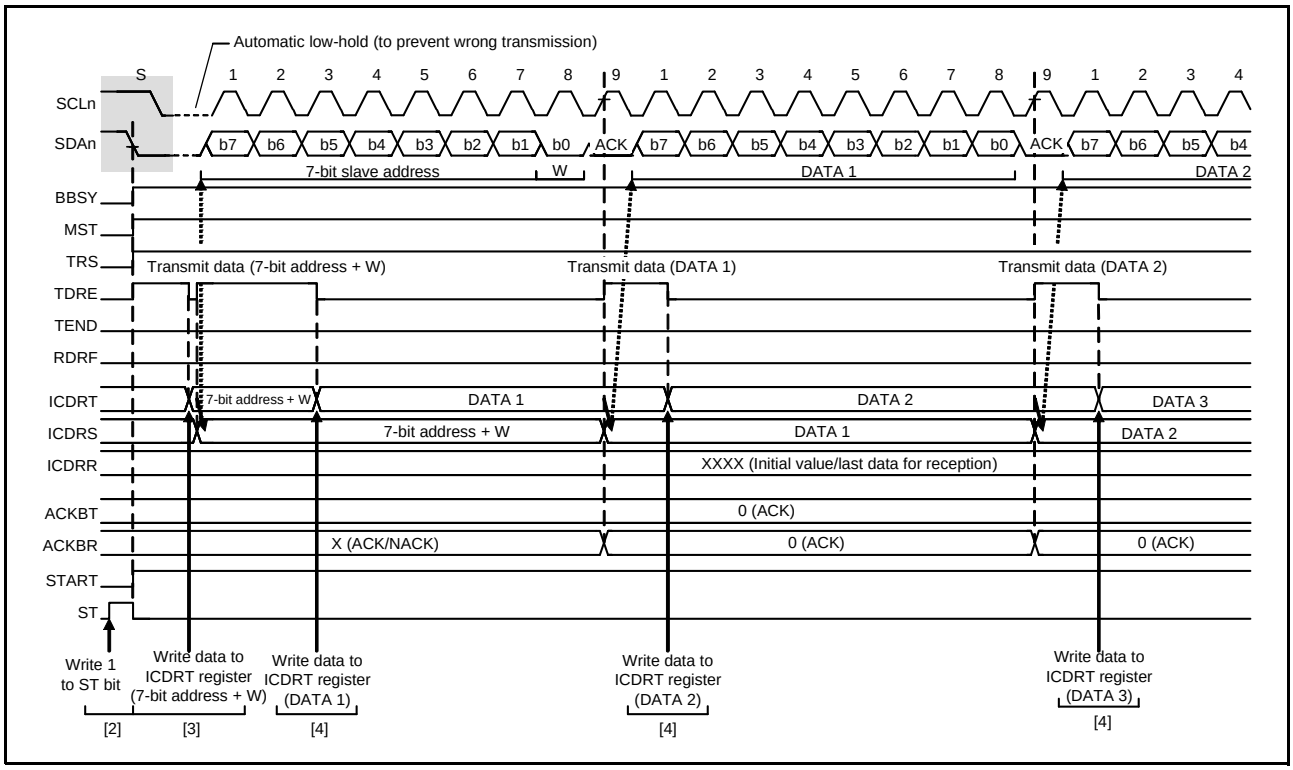


Figure 32.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

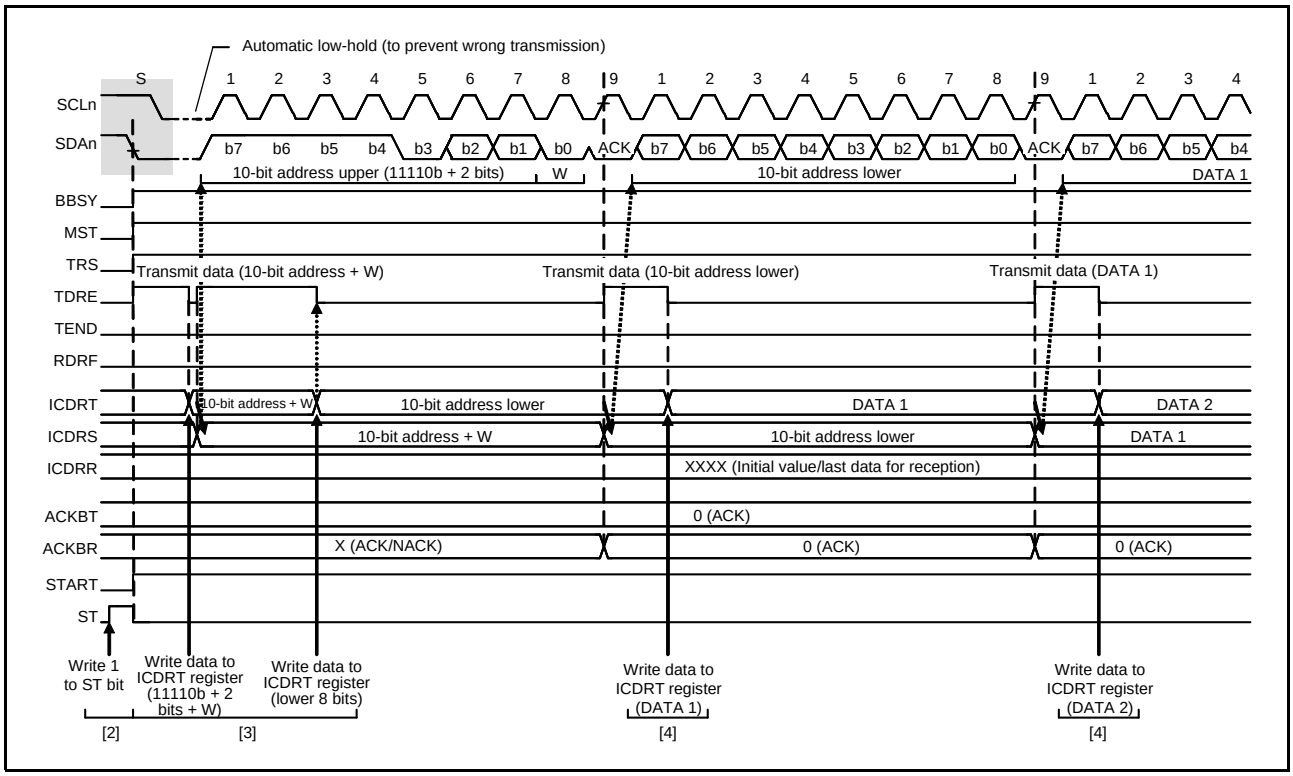


Figure 32.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

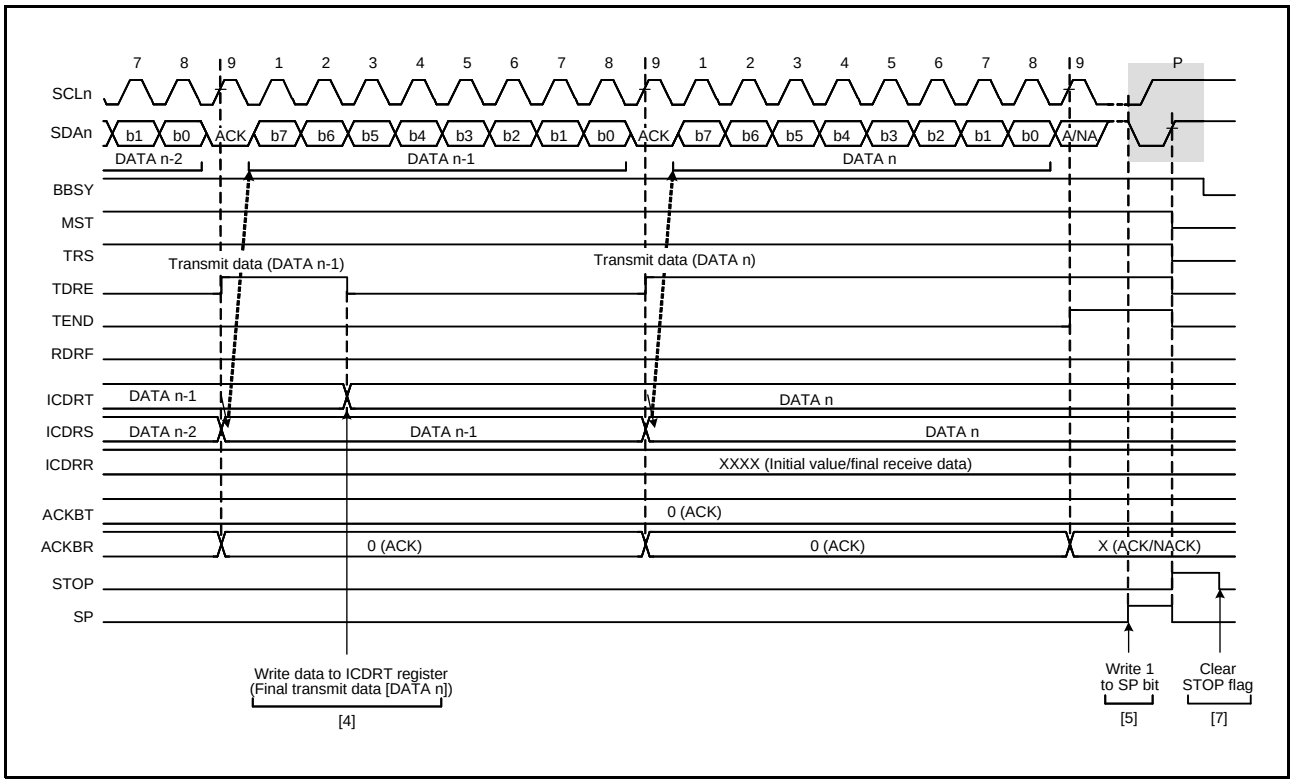


Figure 32.8 Master Transmit Operation Timing (3)

32.3.4 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 32.9 and Figure 32.10 show examples of usage of master reception (7-bit address format) and Figure 32.11 to Figure 32.13 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 32.3.2, 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.

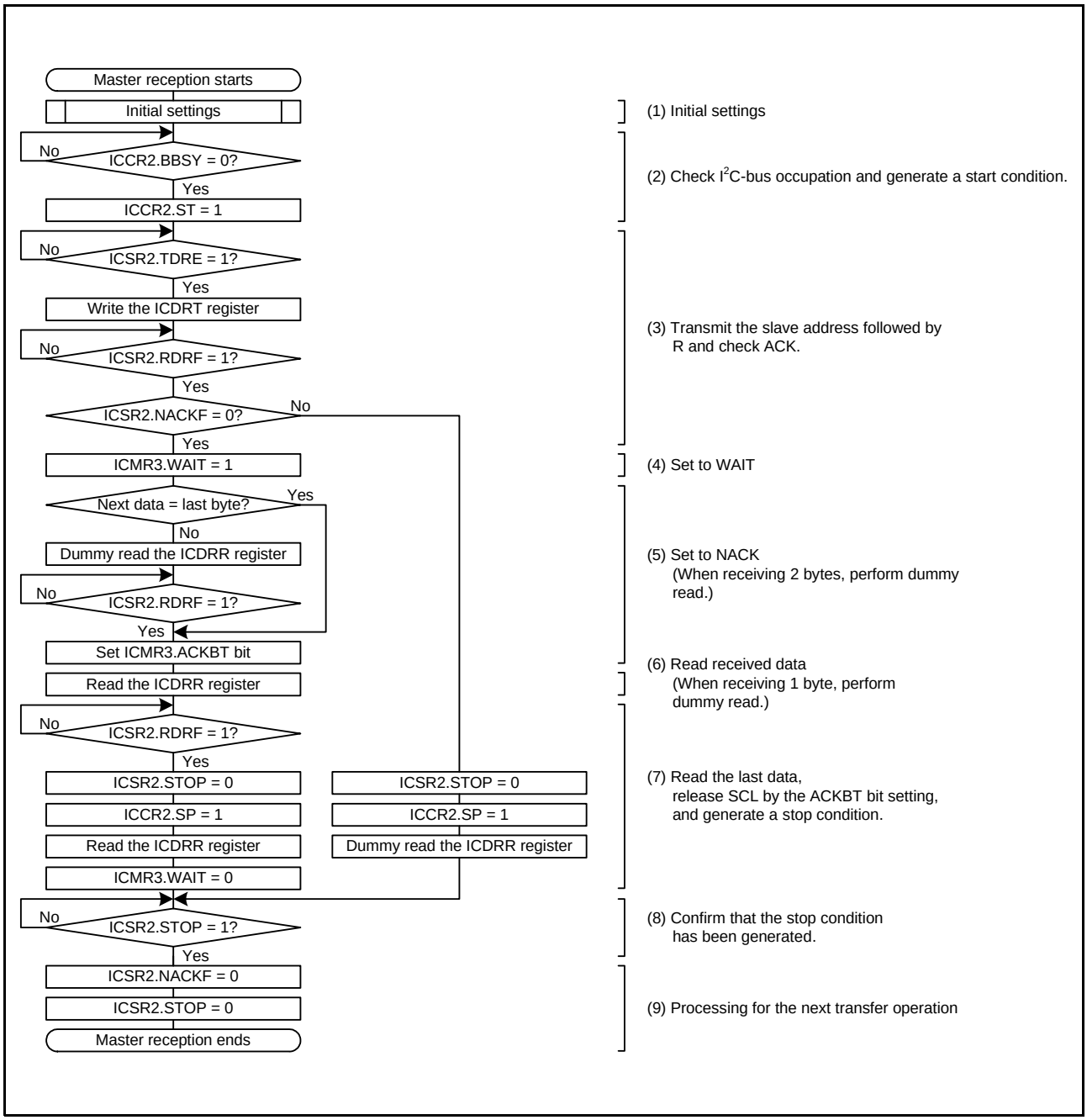


Figure 32.9 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

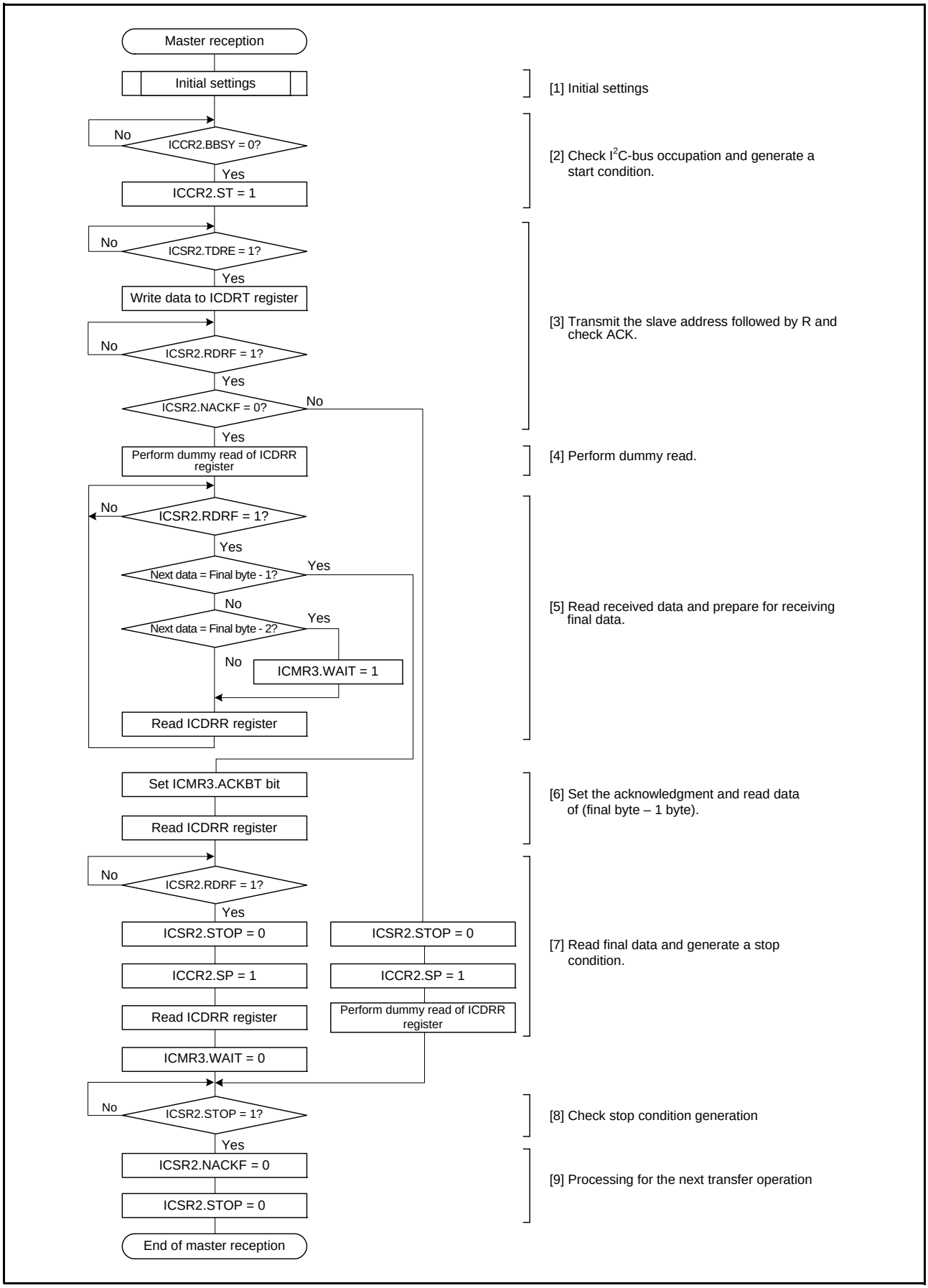


Figure 32.10 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

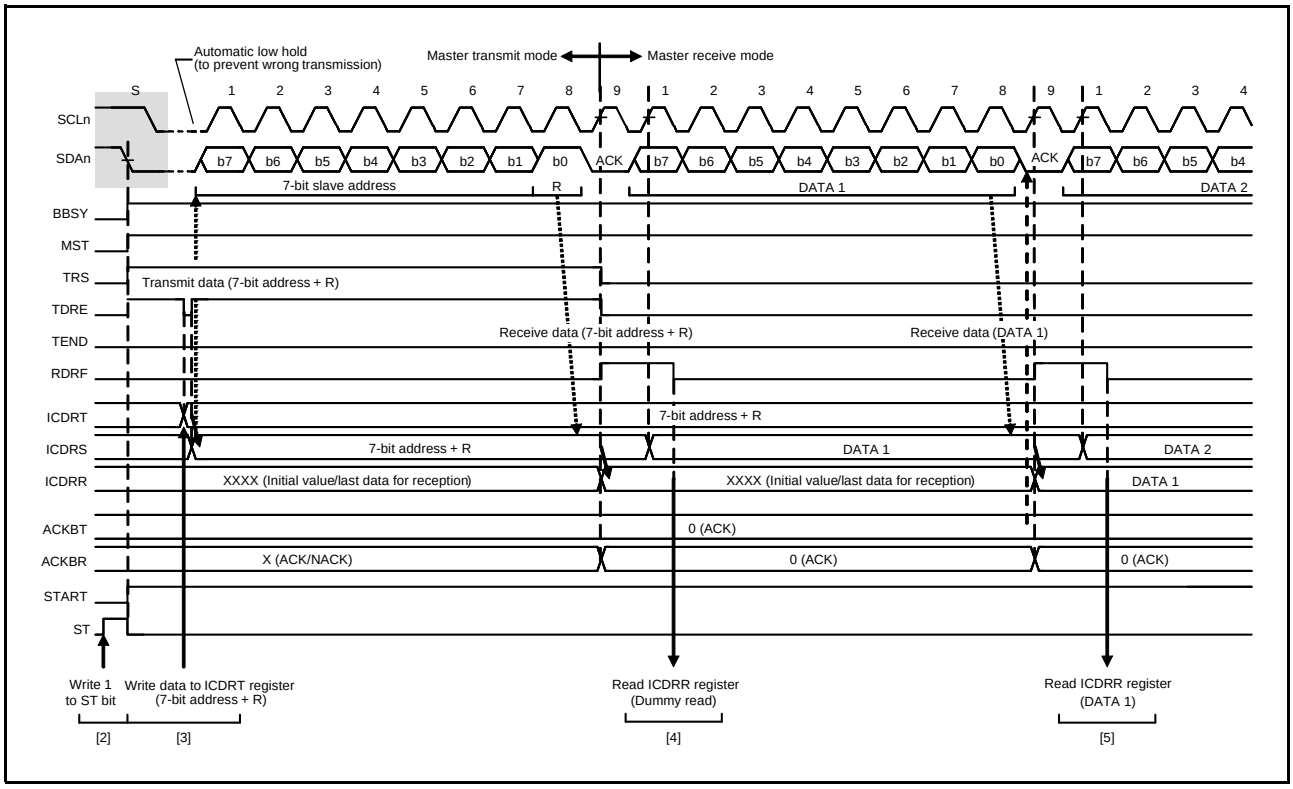


Figure 32.11 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

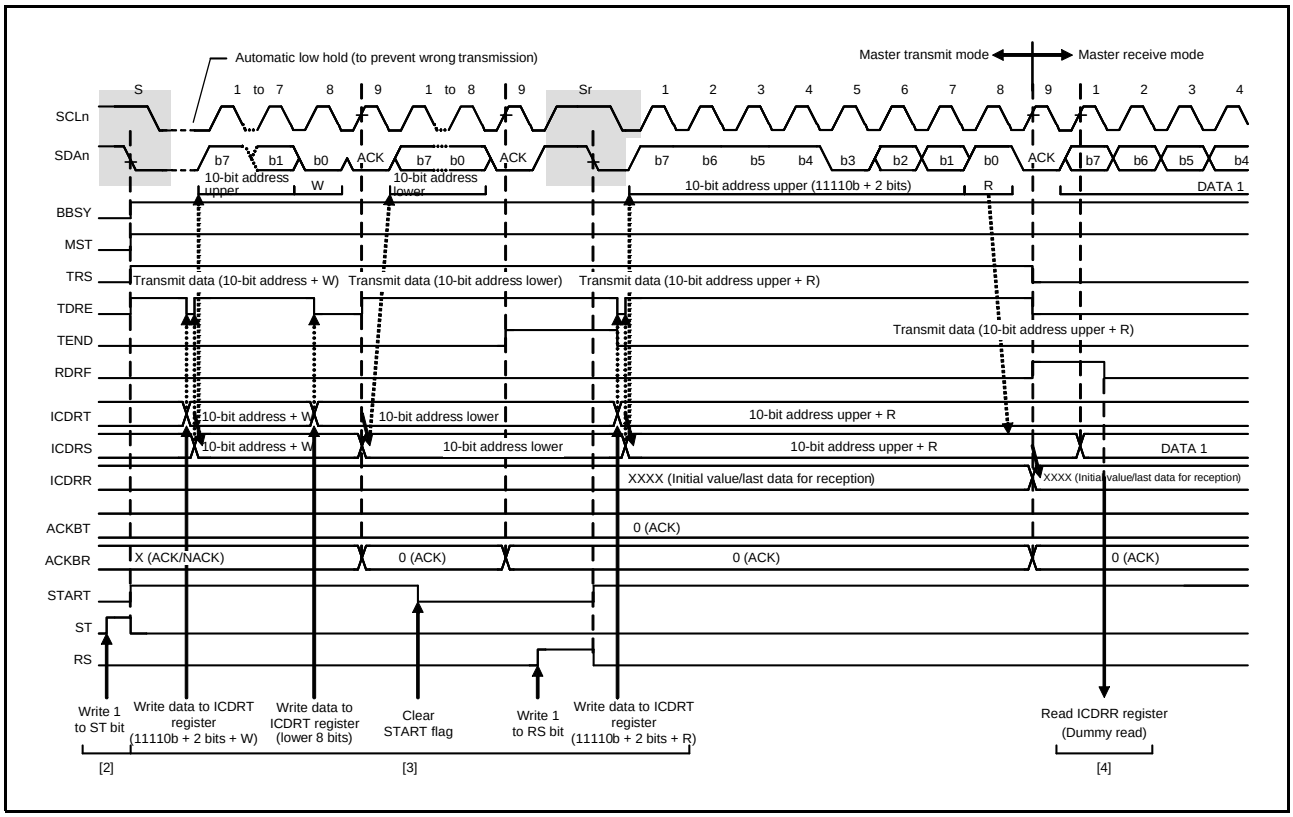


Figure 32.12 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

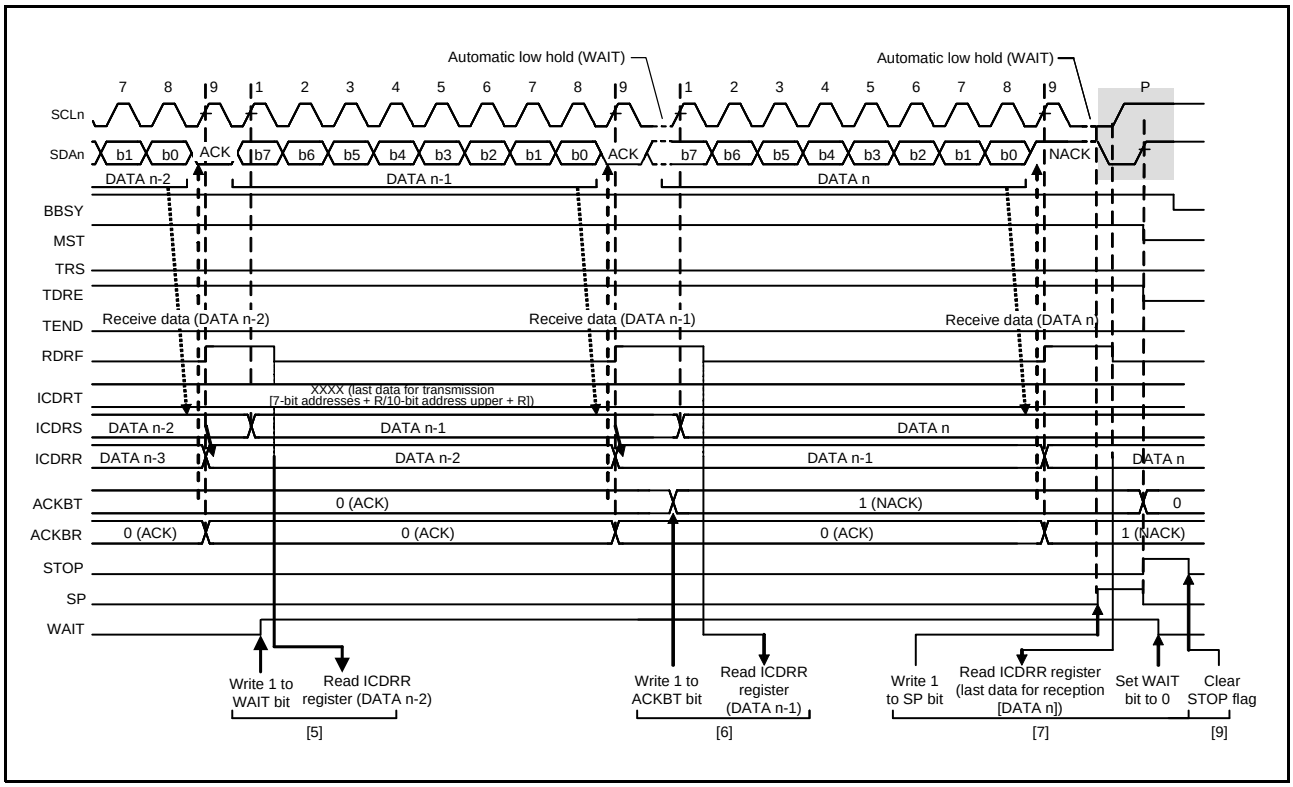


Figure 32.13 Master Receive Operation Timing (3) (When RDRFS bit is 0)

32.3.5 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 32.14 shows an example of usage of slave transmission and Figure 32.15 and Figure 32.16 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 32.3.2, 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.

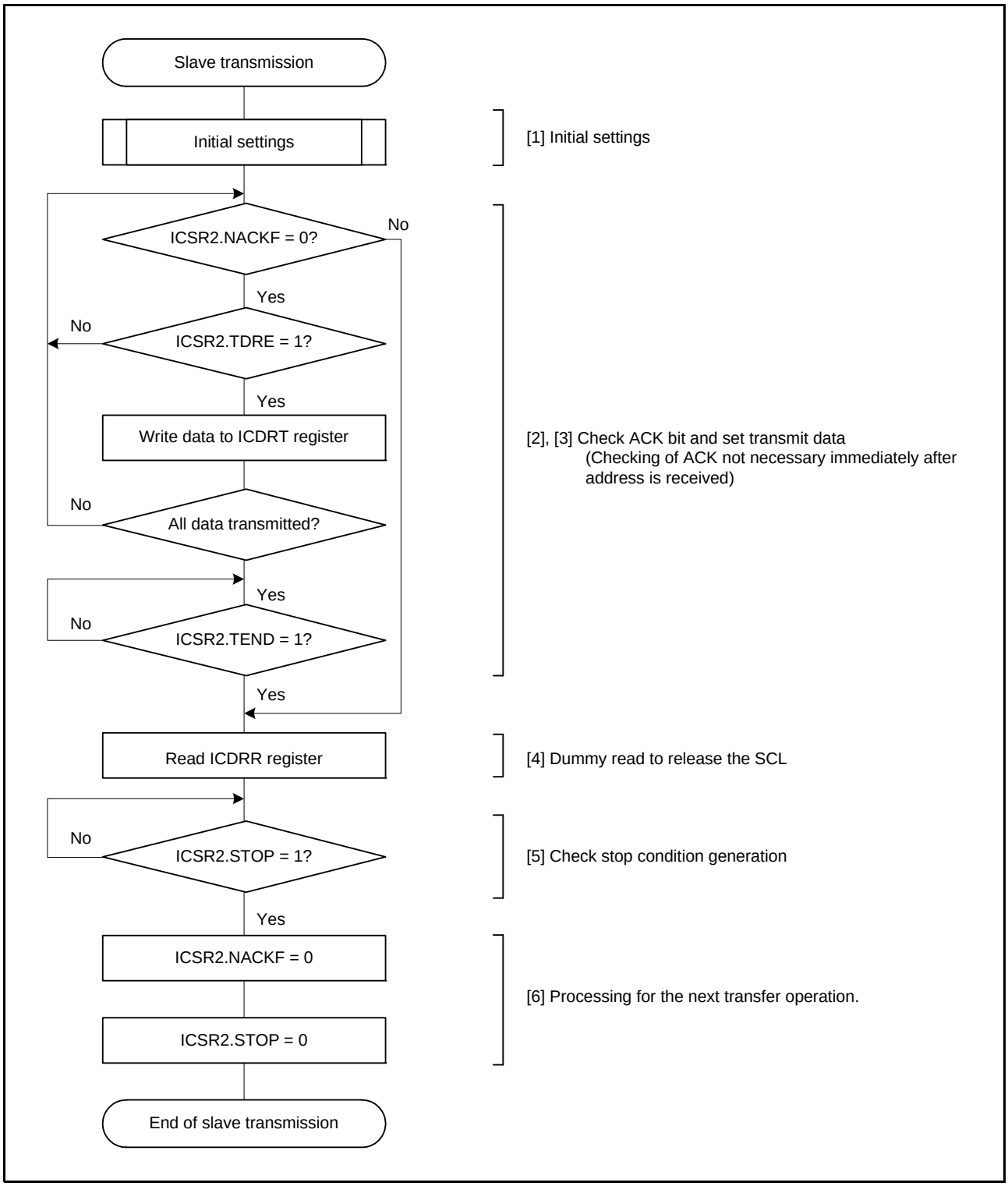


Figure 32.14 Example of Slave Transmission Flowchart

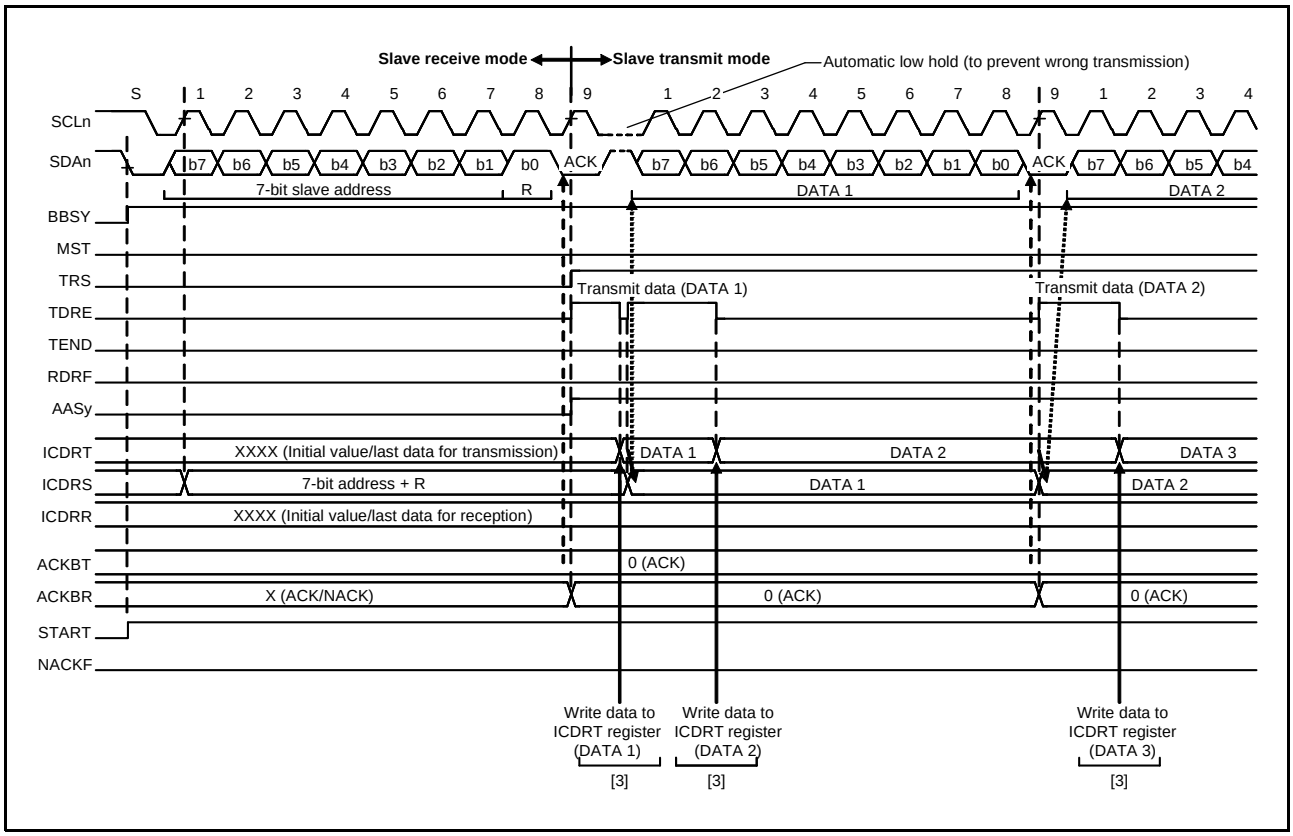


Figure 32.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

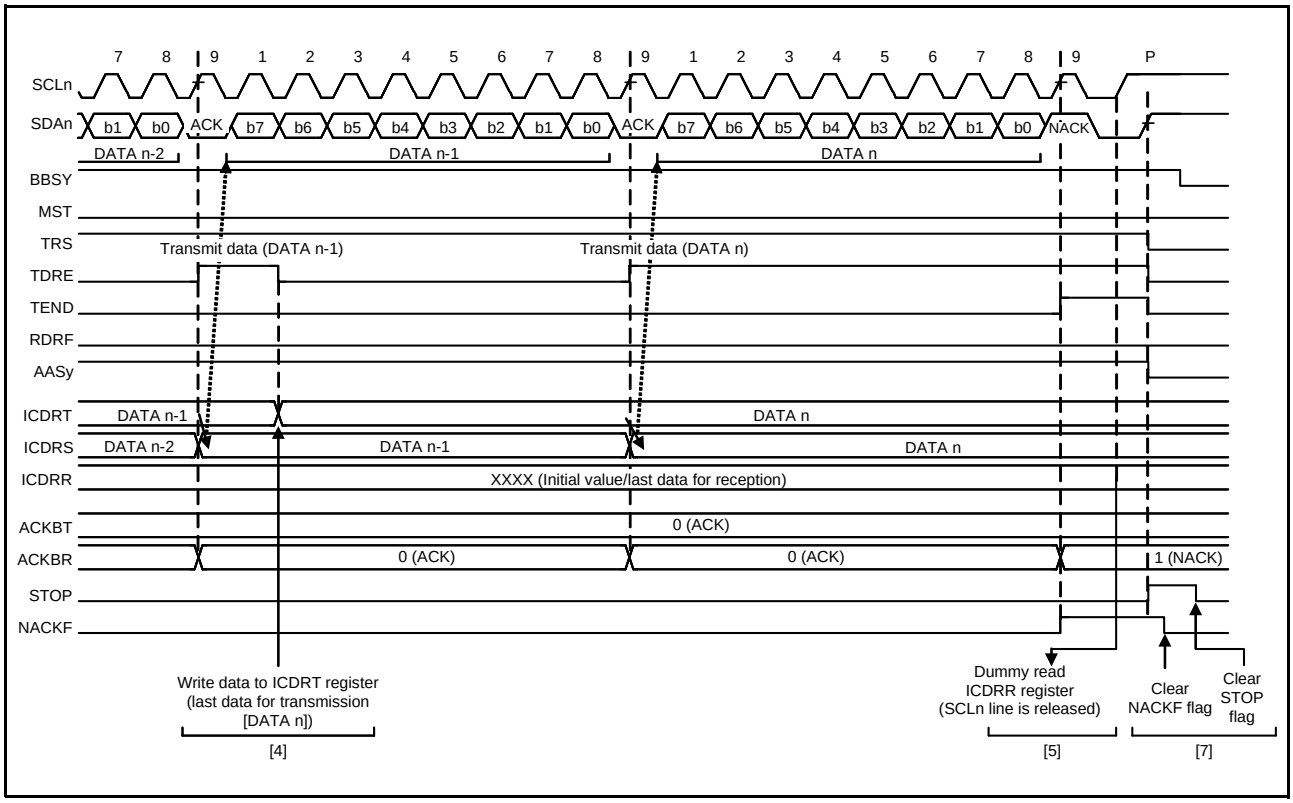


Figure 32.16 Slave Transmit Operation Timing (2)

32.3.6 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 32.17 shows an example of usage of slave reception and Figure 32.18 and Figure 32.19 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 32.3.2, 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.

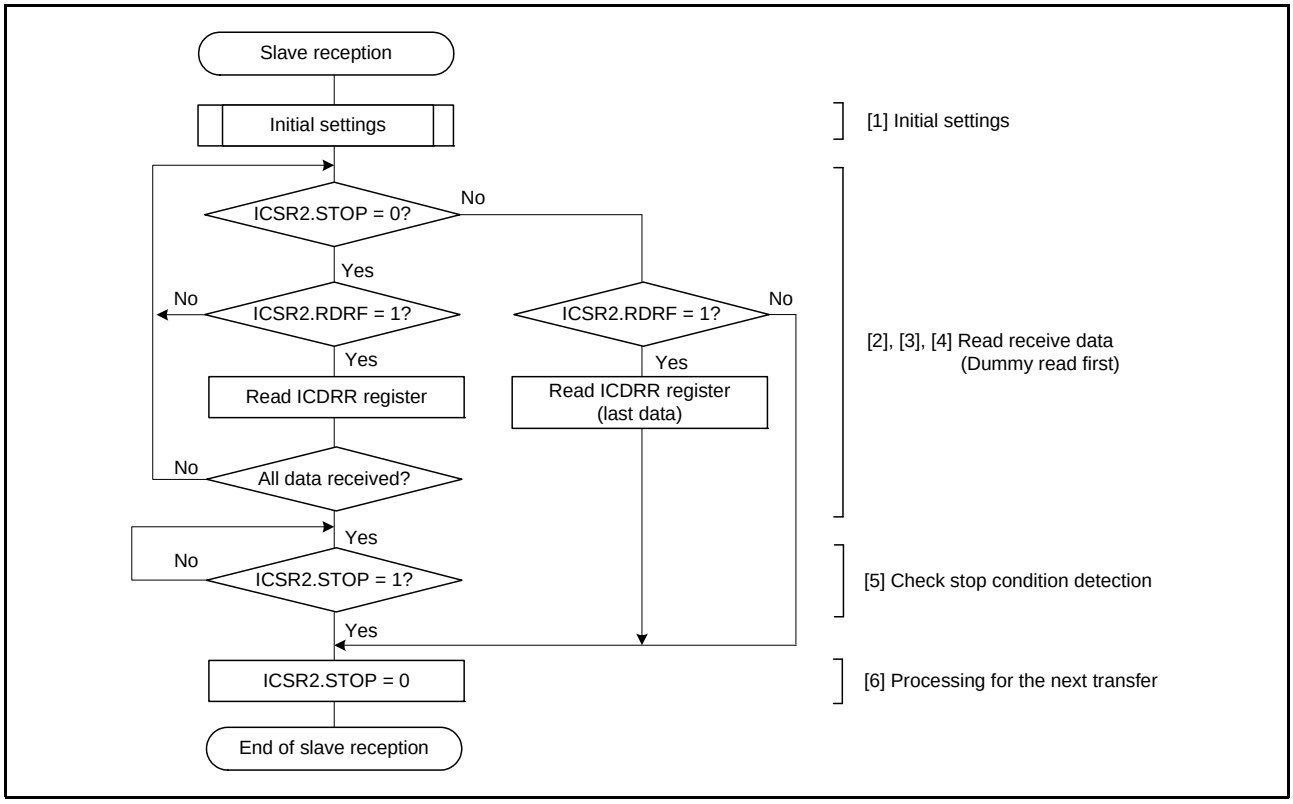


Figure 32.17 Example of Slave Reception Flowchart

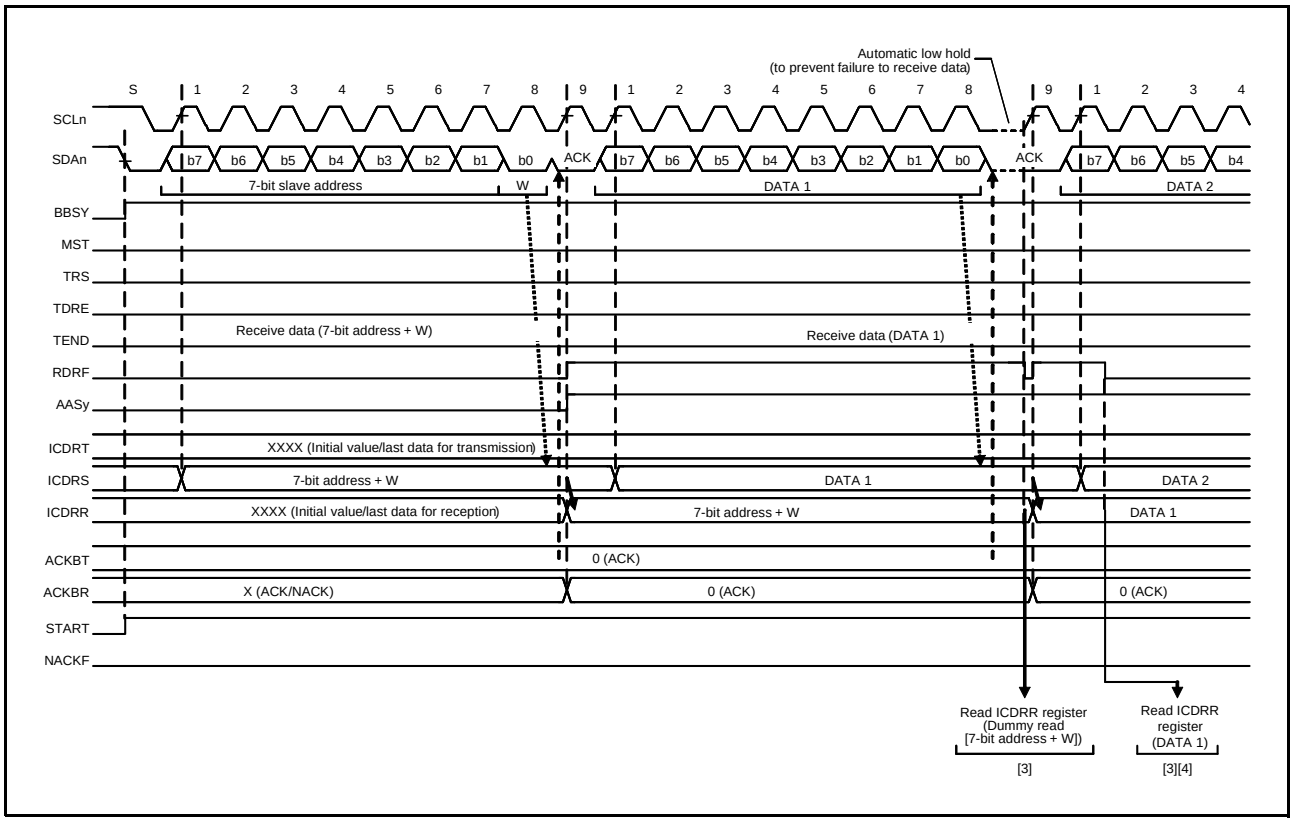


Figure 32.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

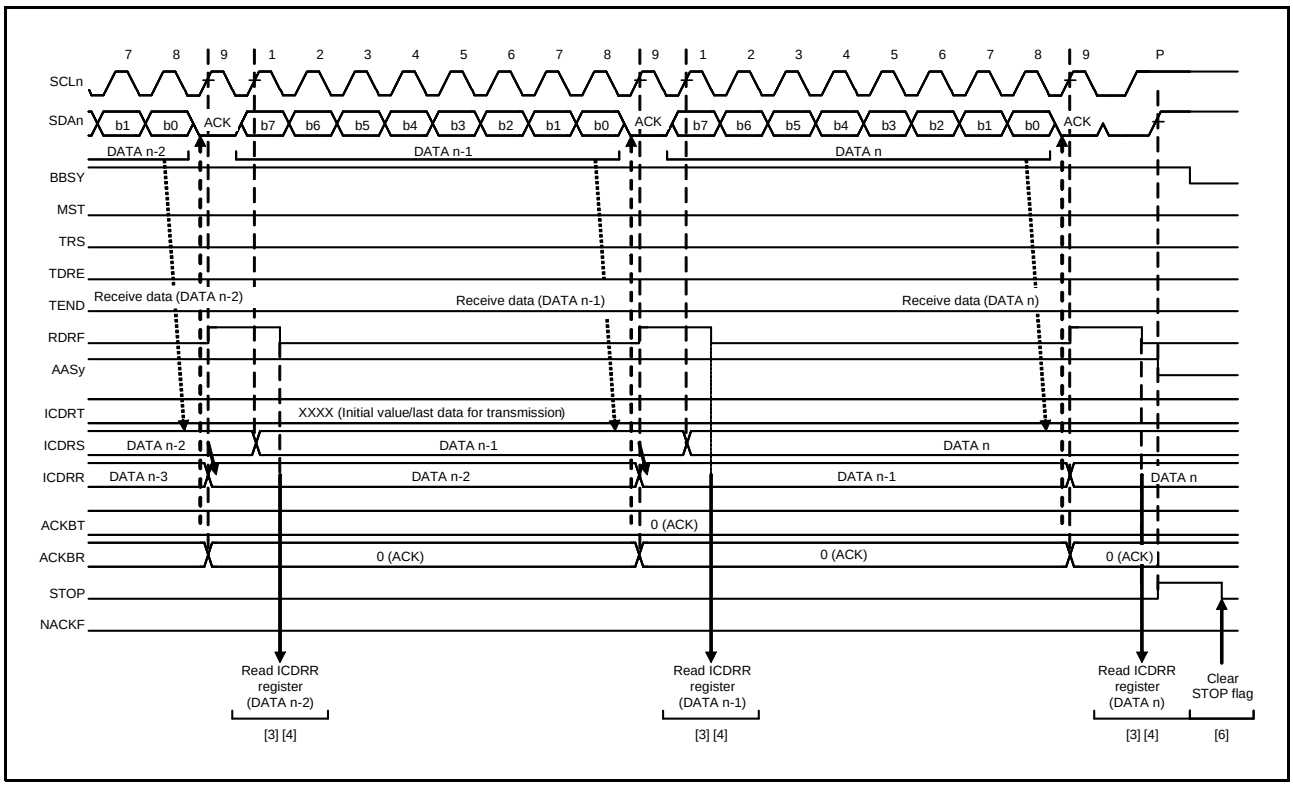


Figure 32.19 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

32.4 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.

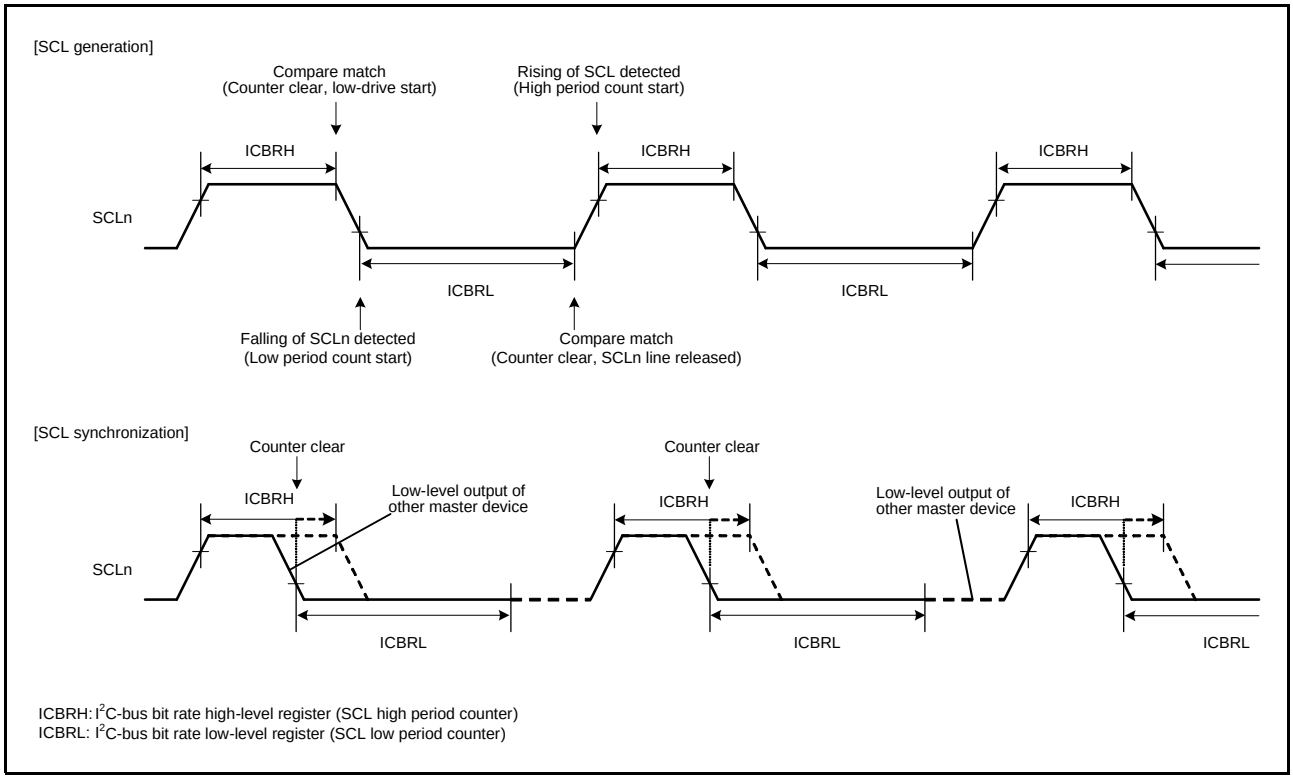


Figure 32.20 Generation and Synchronization of the SCL Signal from the RIIC

32.5 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).

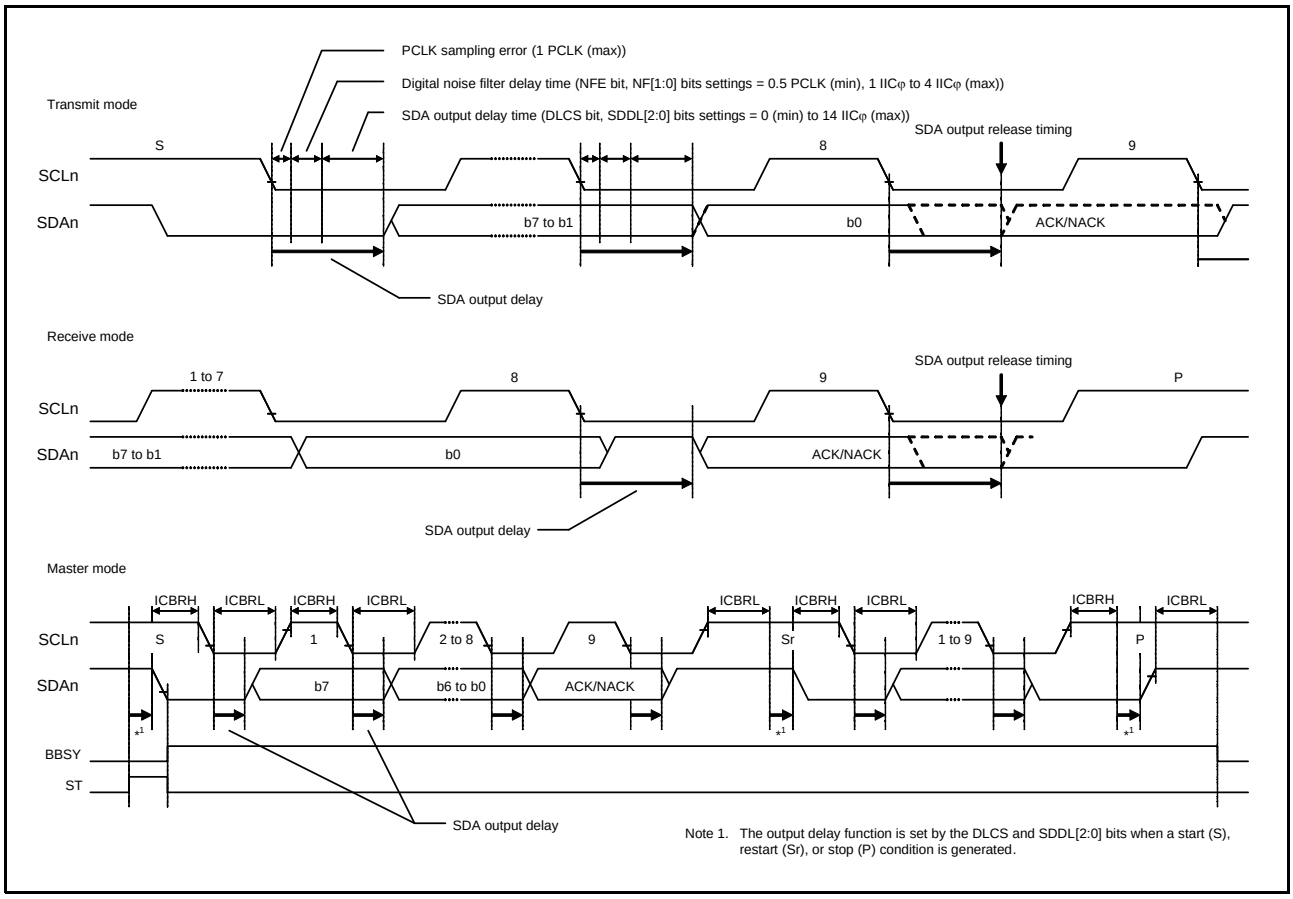


Figure 32.21 SDA Output Delay Function

32.6 Digital Noise Filters

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through digital noise filters. Figure 32.22 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.

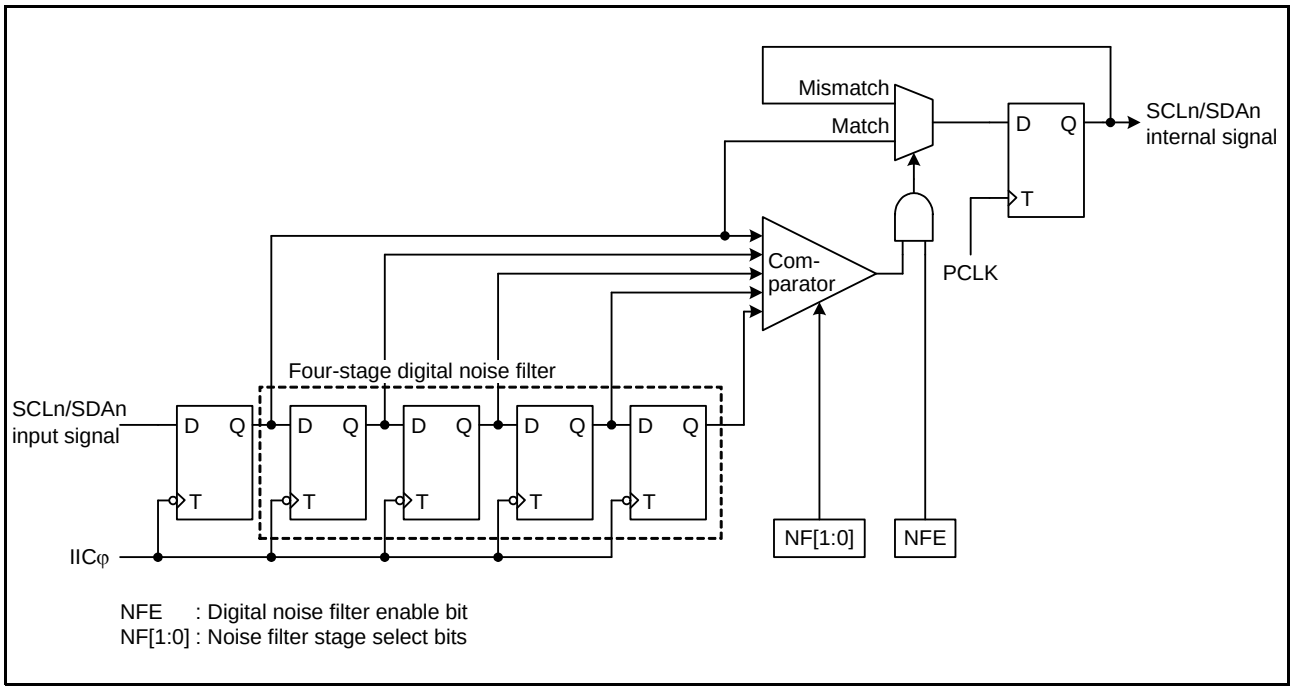


Figure 32.22 Block Diagram of the Digital Noise Filter

32.7 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.

32.7.1 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 32.23 to Figure 32.25 show the AASy flag set timing in three cases.

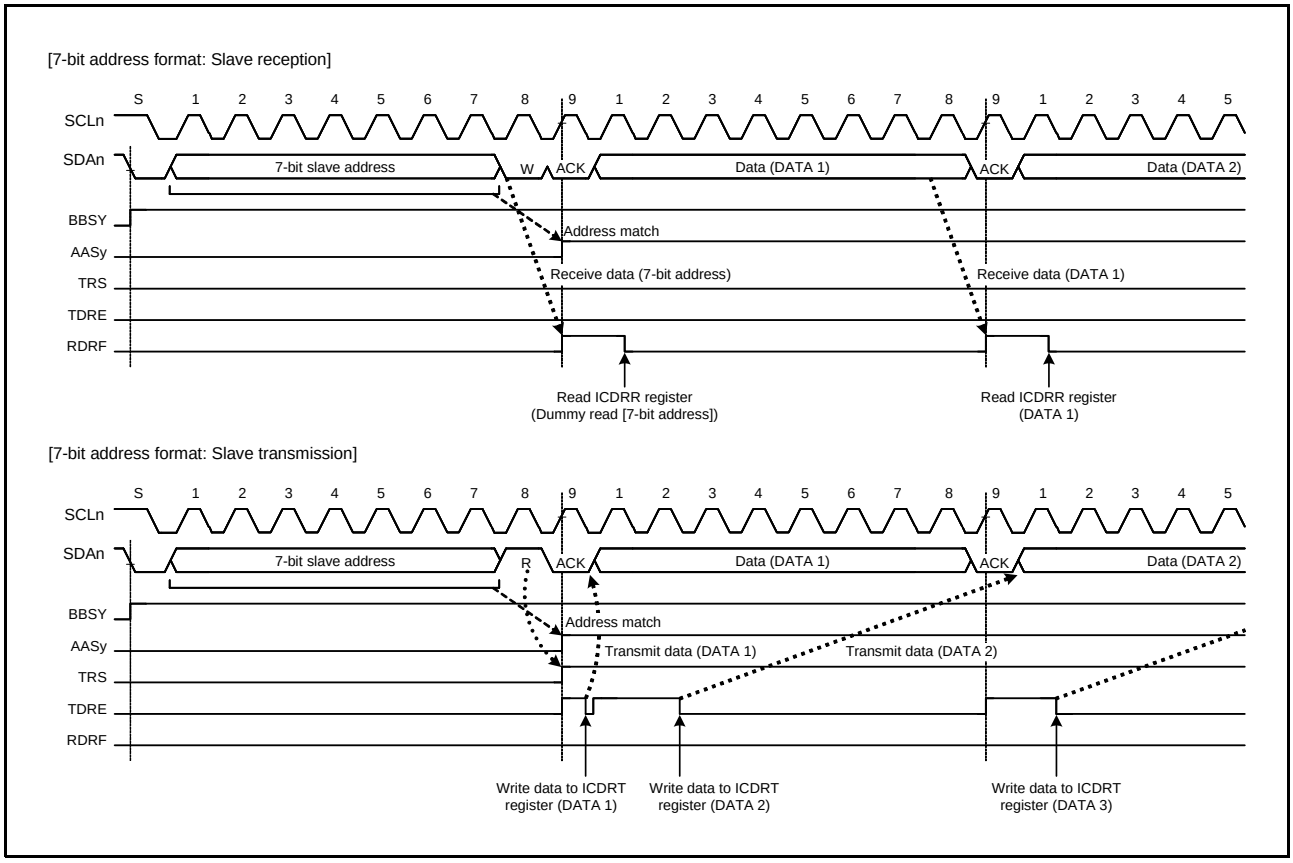


Figure 32.23 AASy Flag Set Timing with 7-Bit Address Format Selected

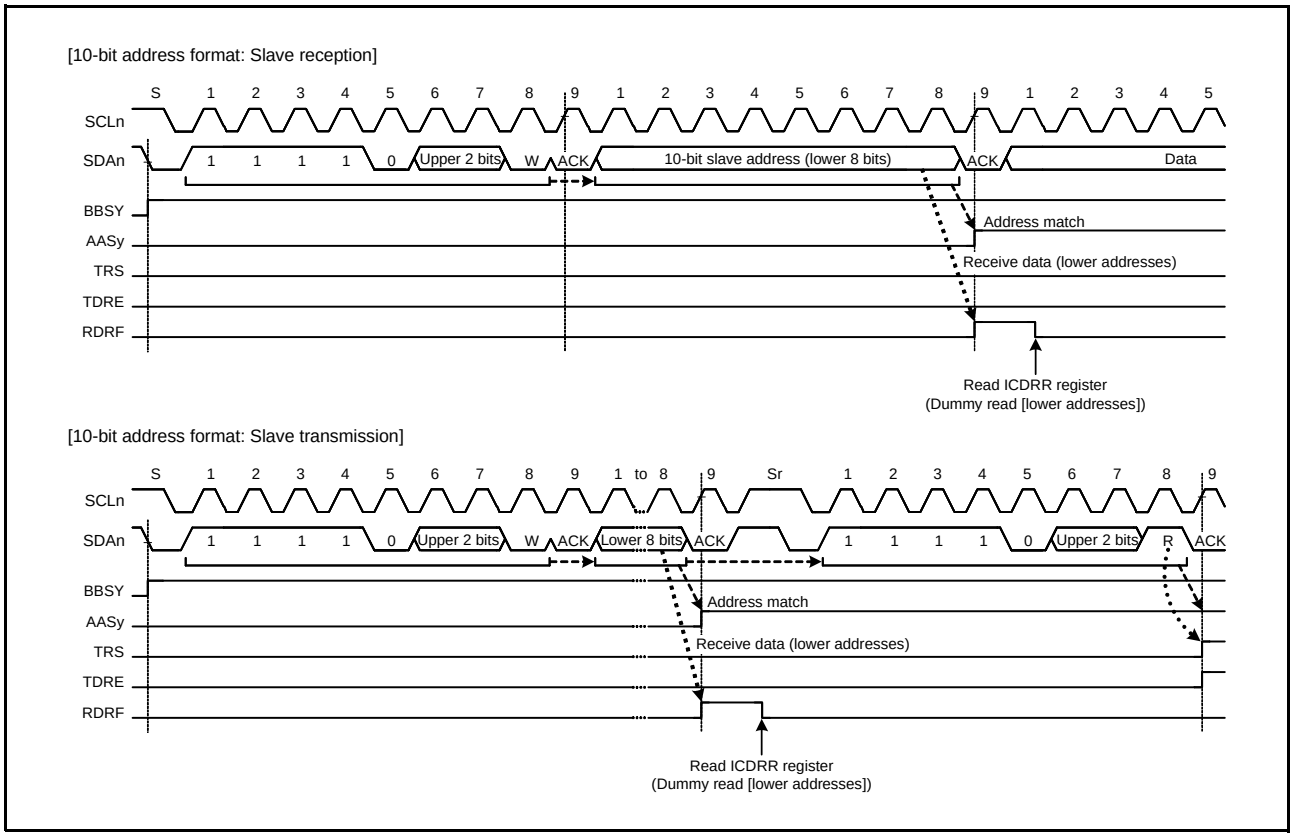


Figure 32.24 AASy Flag Set Timing with 10-Bit Address Format Selected

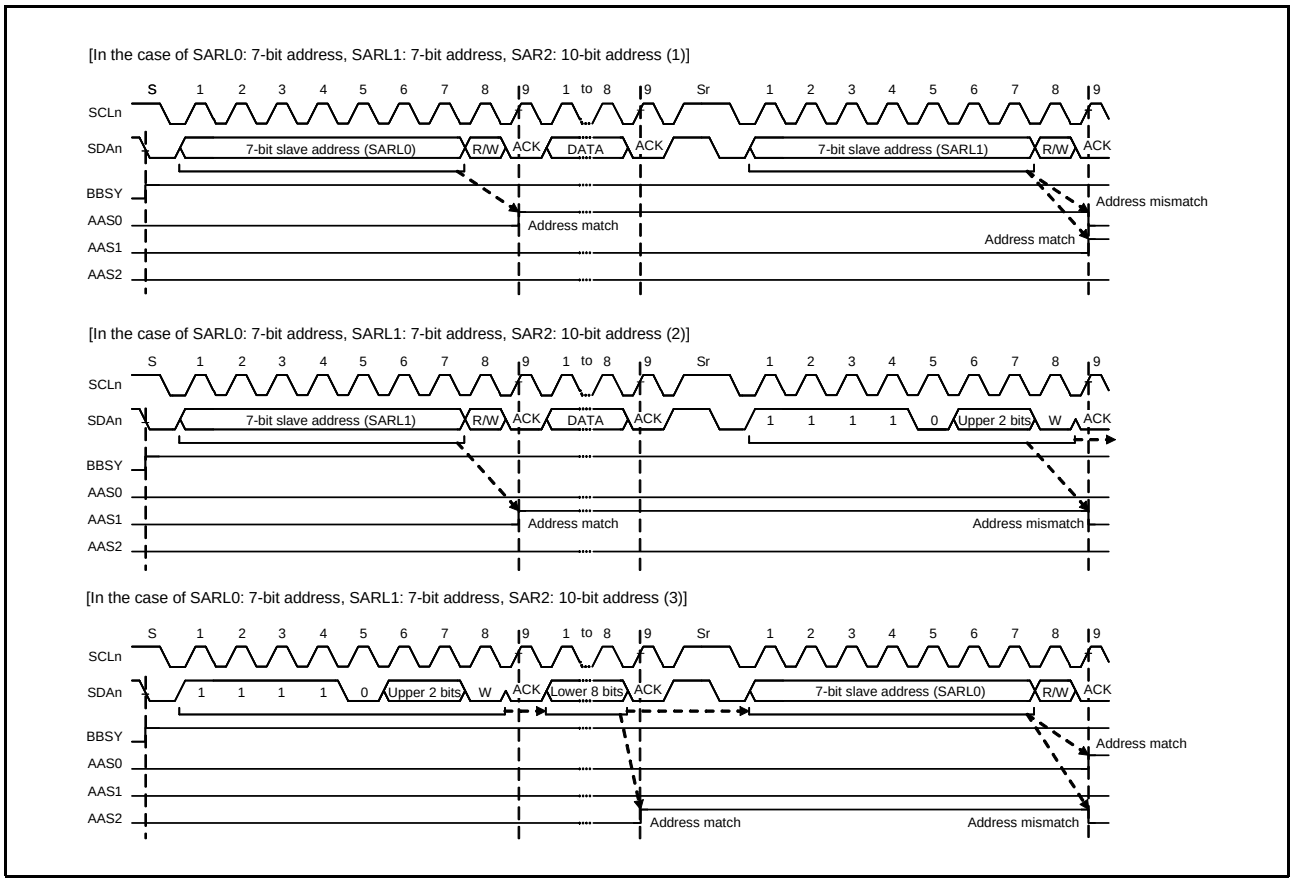


Figure 32.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

32.7.2 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.

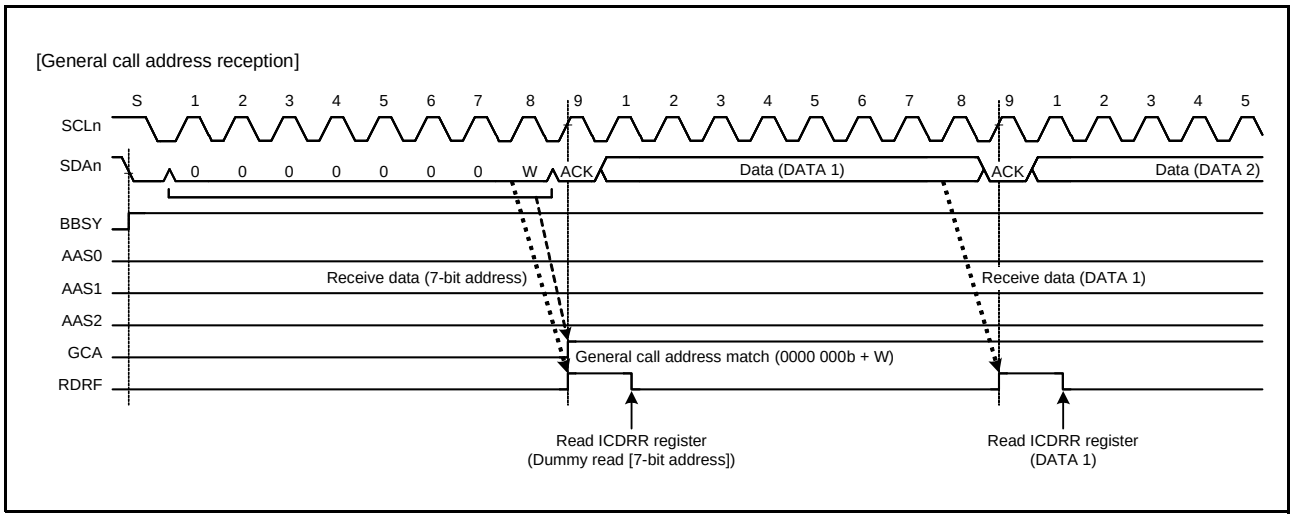


Figure 32.26 Timing of GCA Flag Setting during Reception of General Call Address

32.7.3 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.

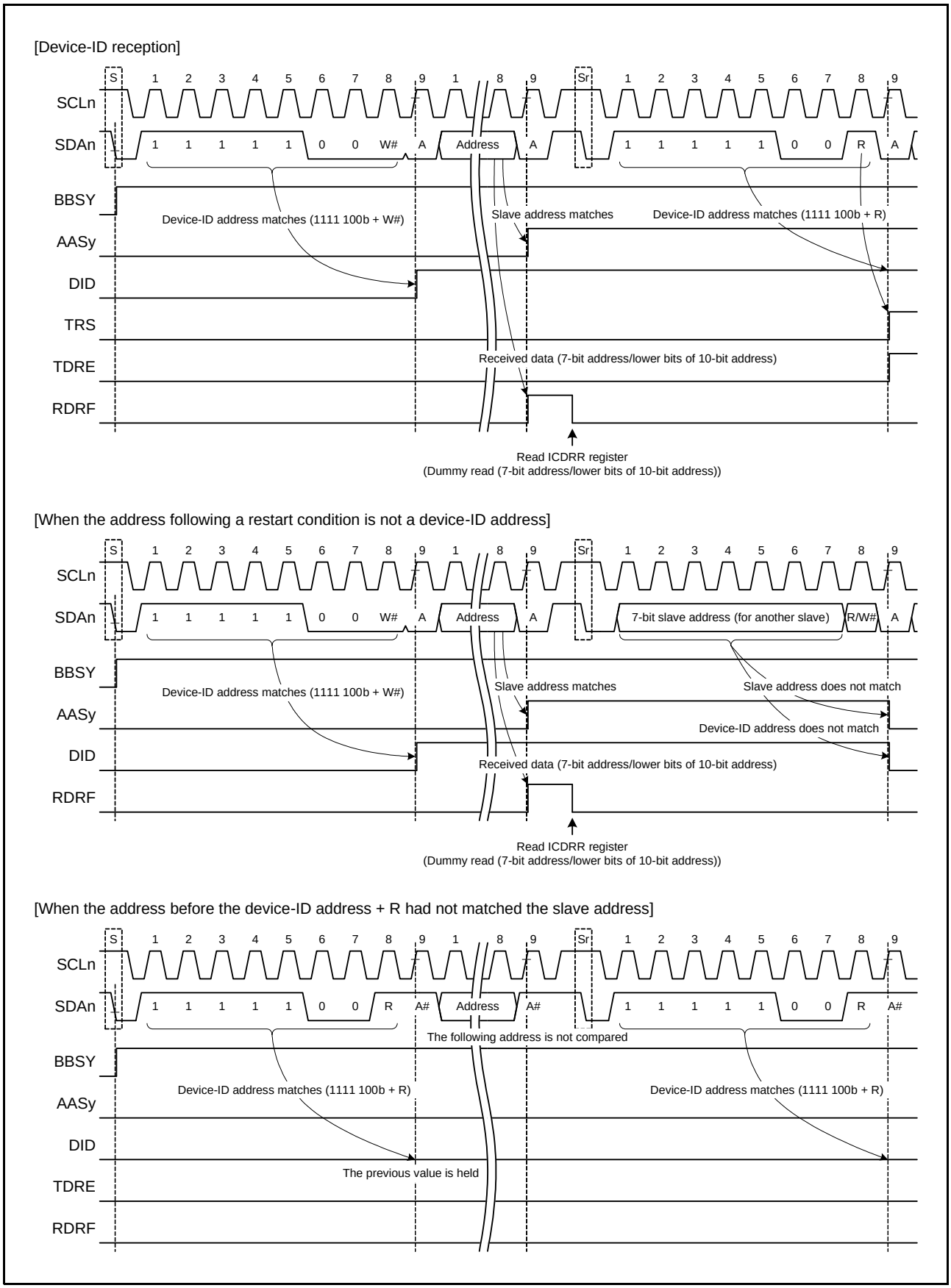


Figure 32.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

32.7.4 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.

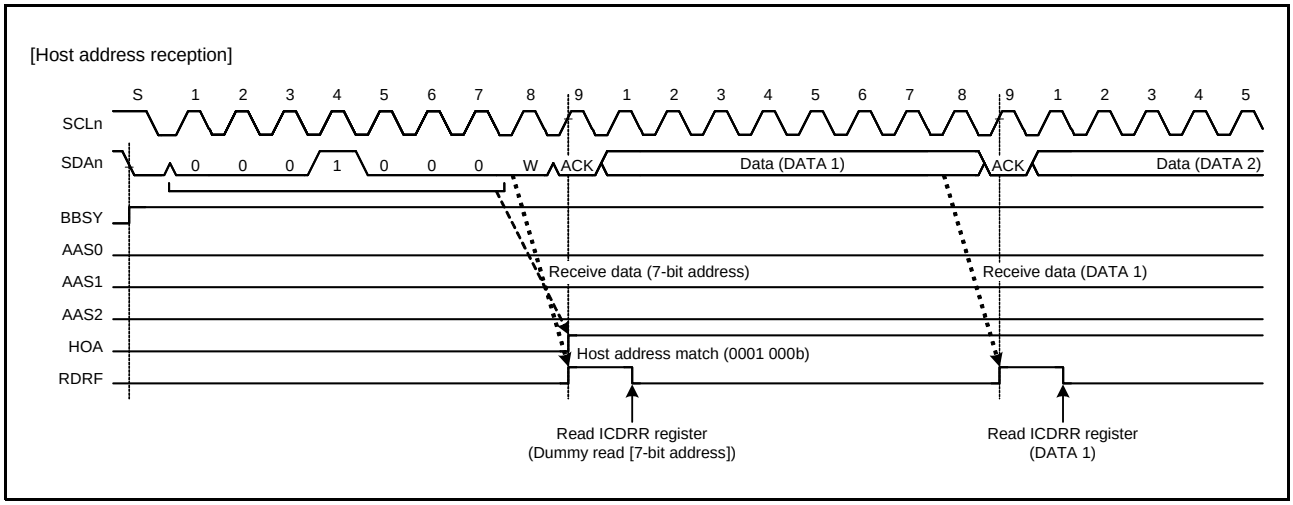


Figure 32.28 HOA Flag Set Timing during Reception of Host Address

32.8 Automatic Low-Hold Function for SCL

32.8.1 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

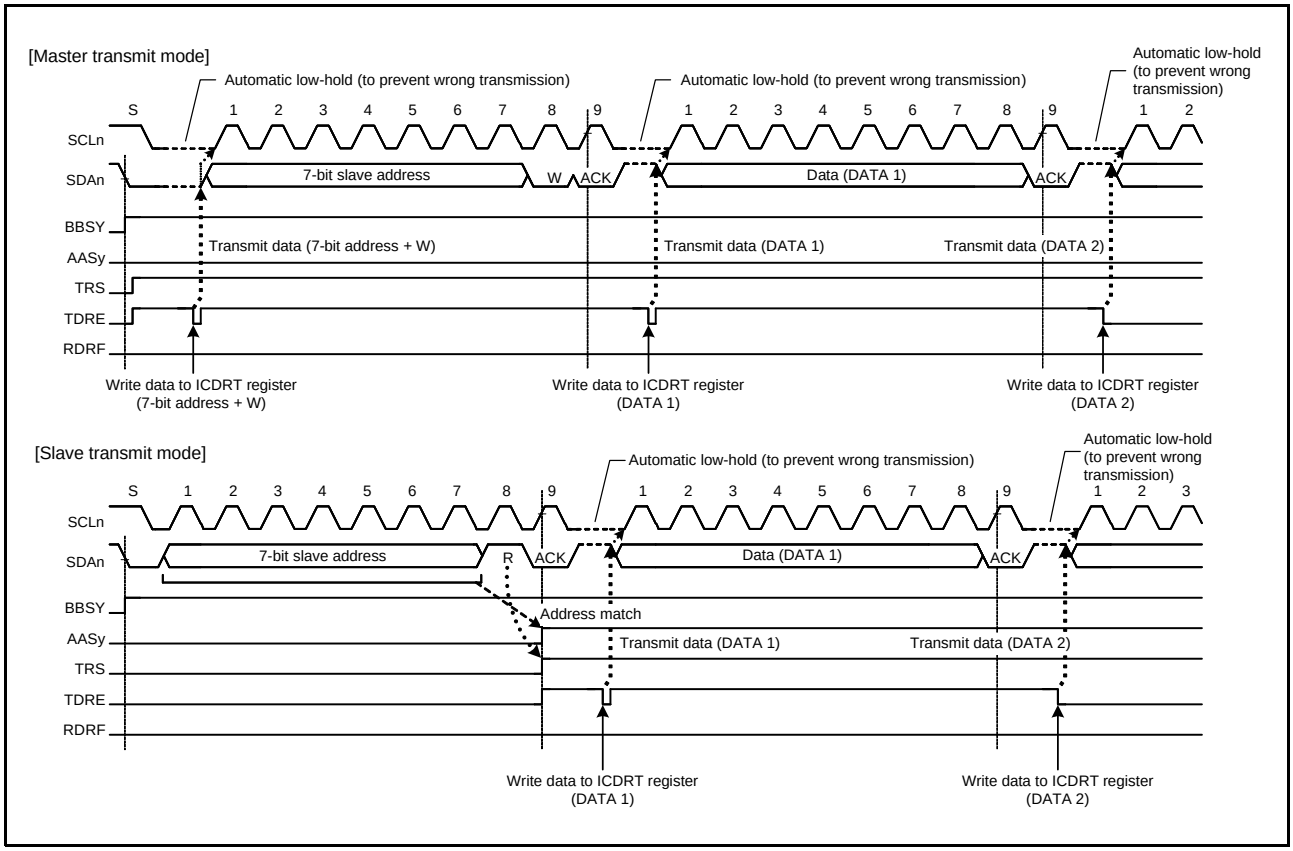


Figure 32.29 Automatic Low-Hold Operation in Transmit Mode

32.8.2 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.

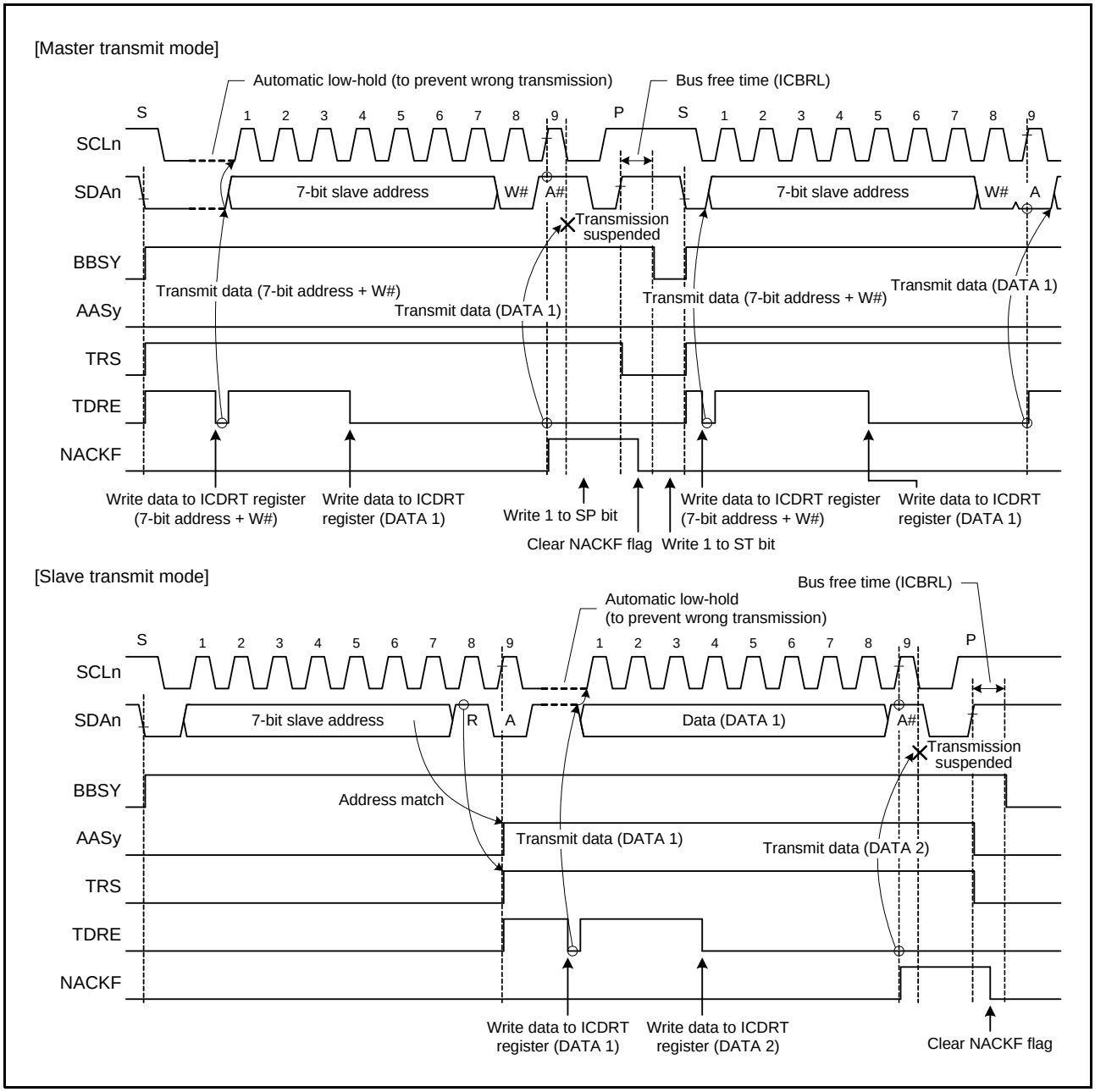


Figure 32.30 Suspension of Data Transmission When NACK is Received (NACKE = 1)

32.8.3 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.

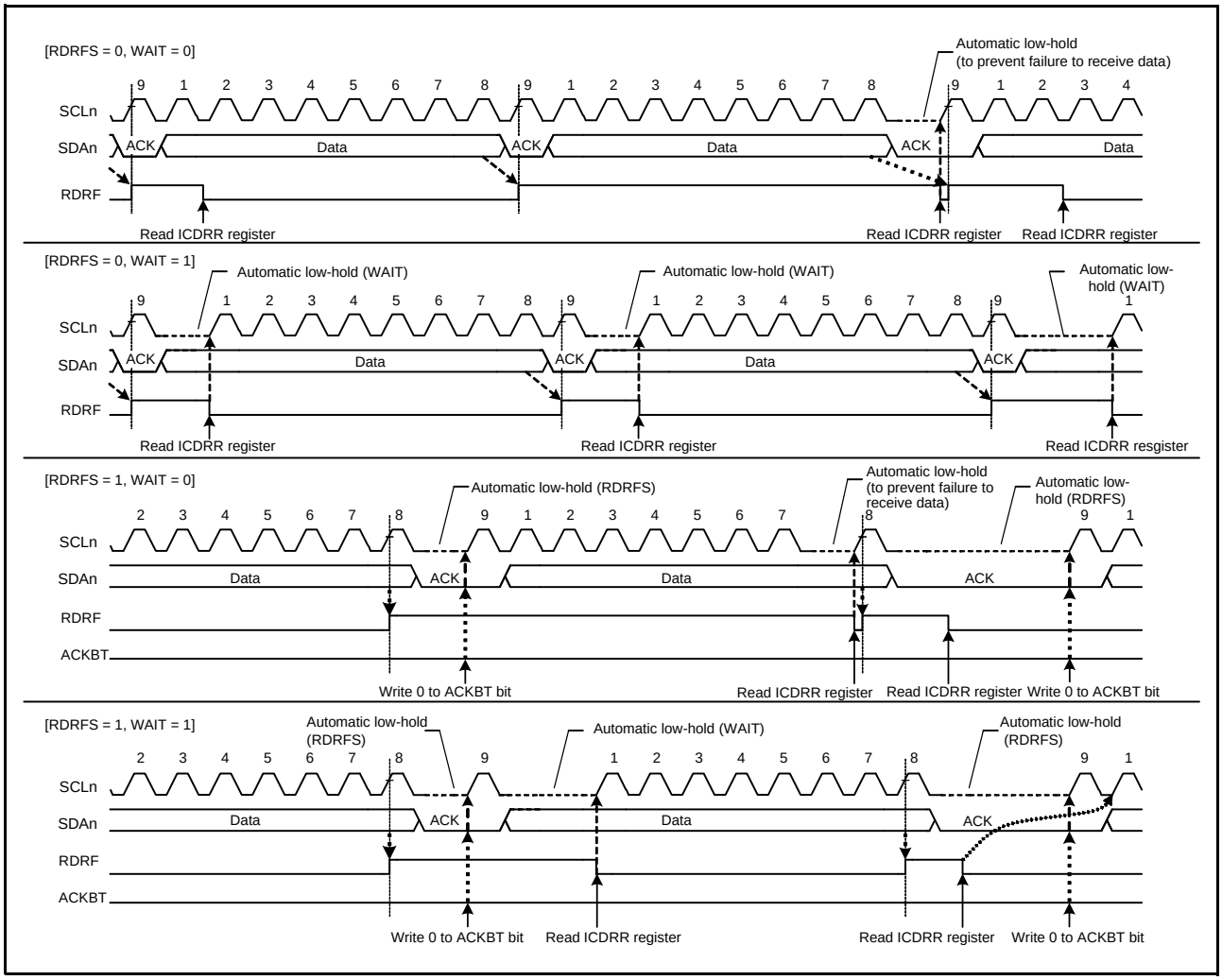


Figure 32.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

32.9 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.

32.9.1 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)

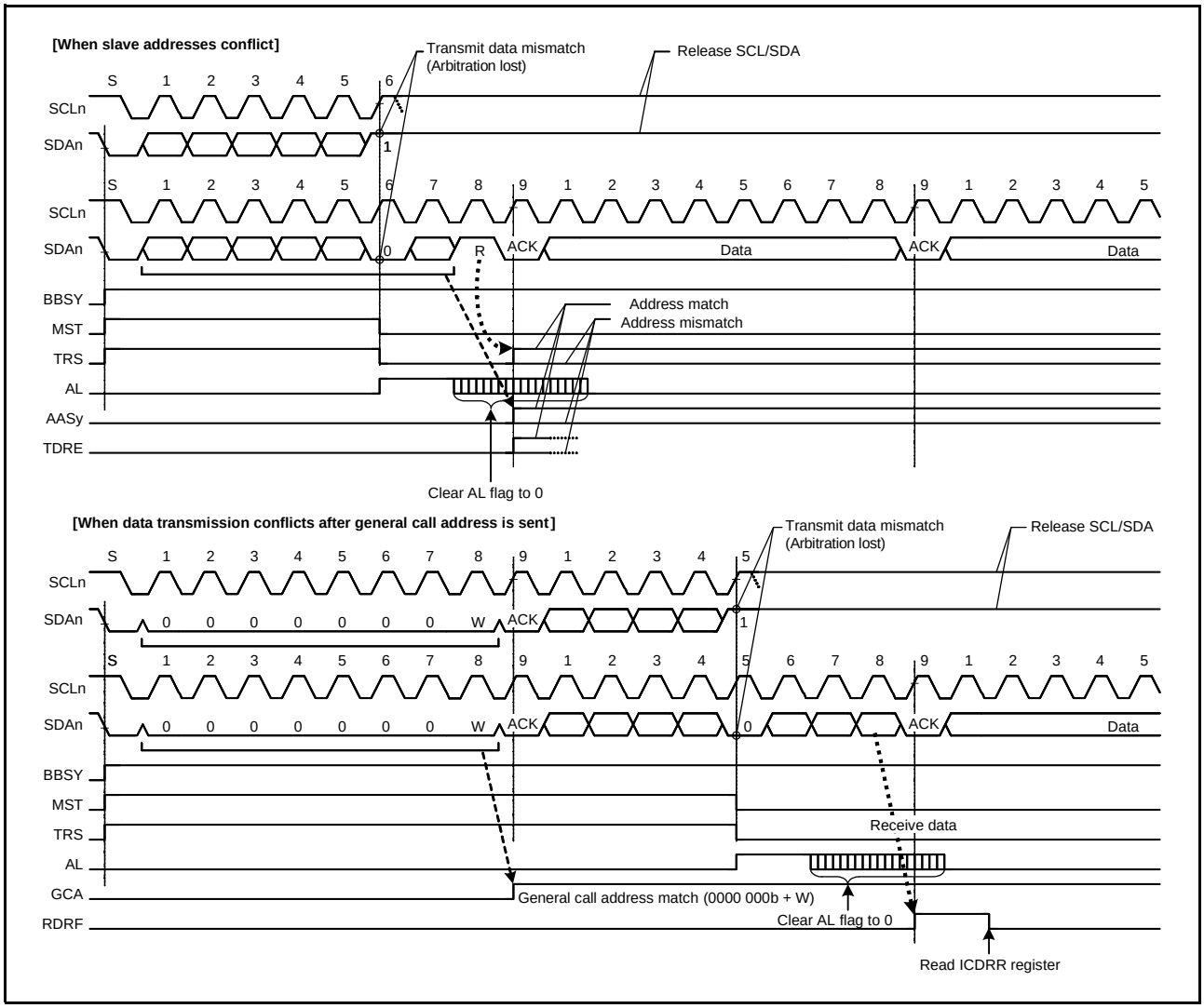


Figure 32.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

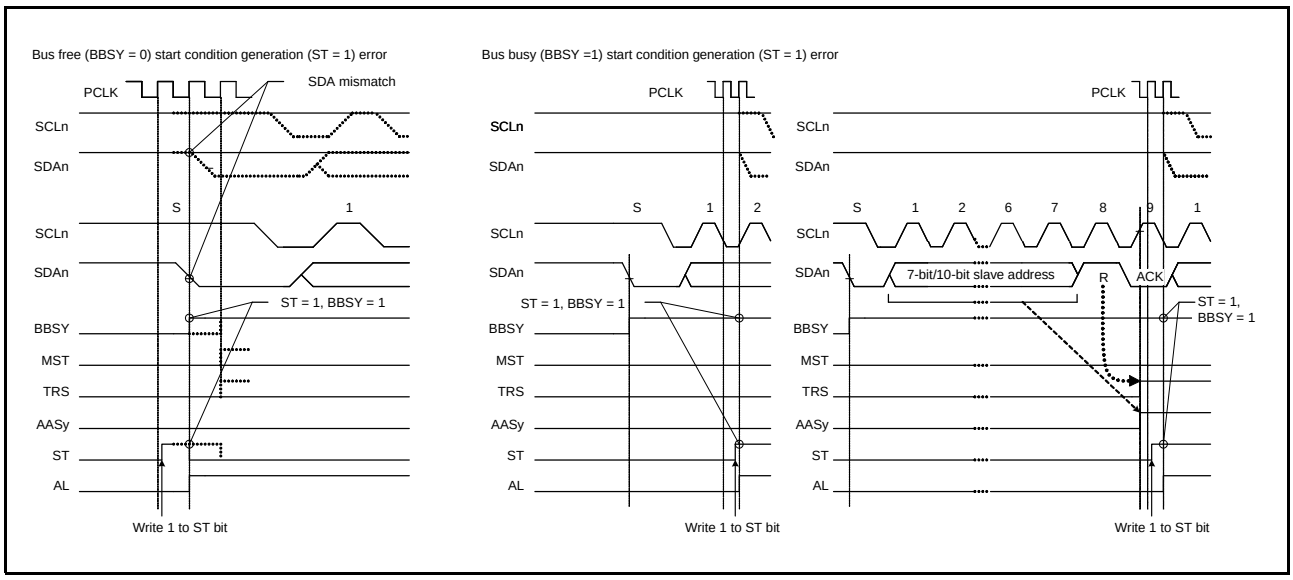


Figure 32.33 Arbitration-Lost When a Start Condition is Generated (MALE = 1)

32.9.2 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 32.34 shows an example of NACK transmission arbitration-lost detection.

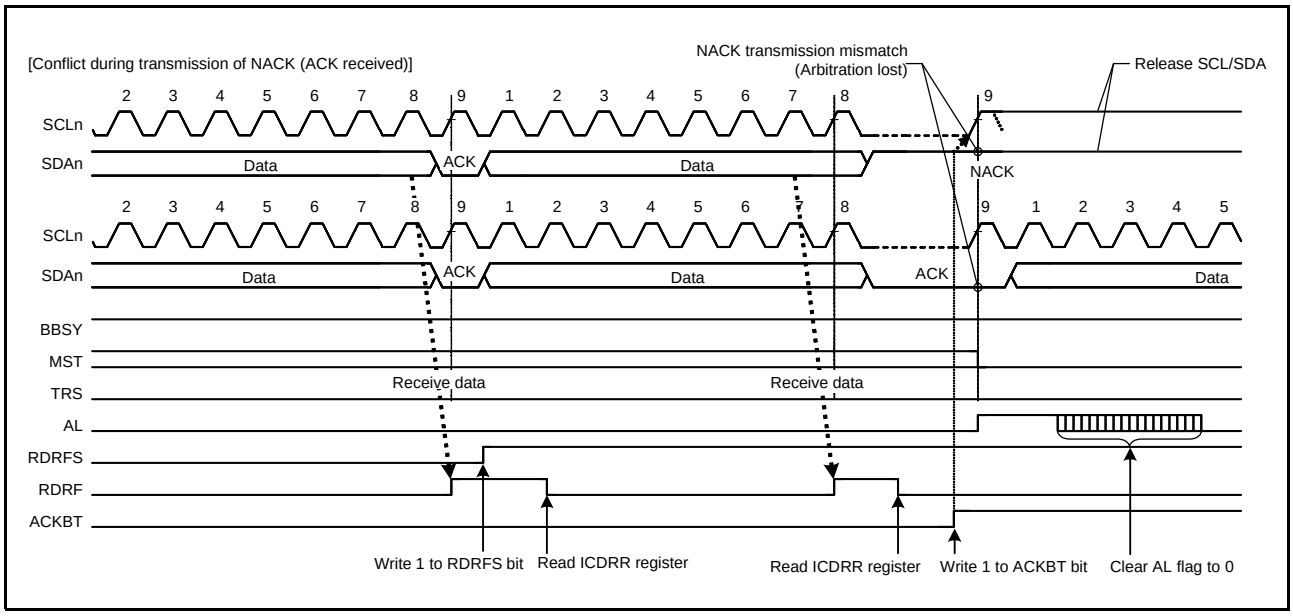


Figure 32.34 Example of NACK transmission arbitration-lost Detection (NALE = 1)

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)

32.9.3 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)

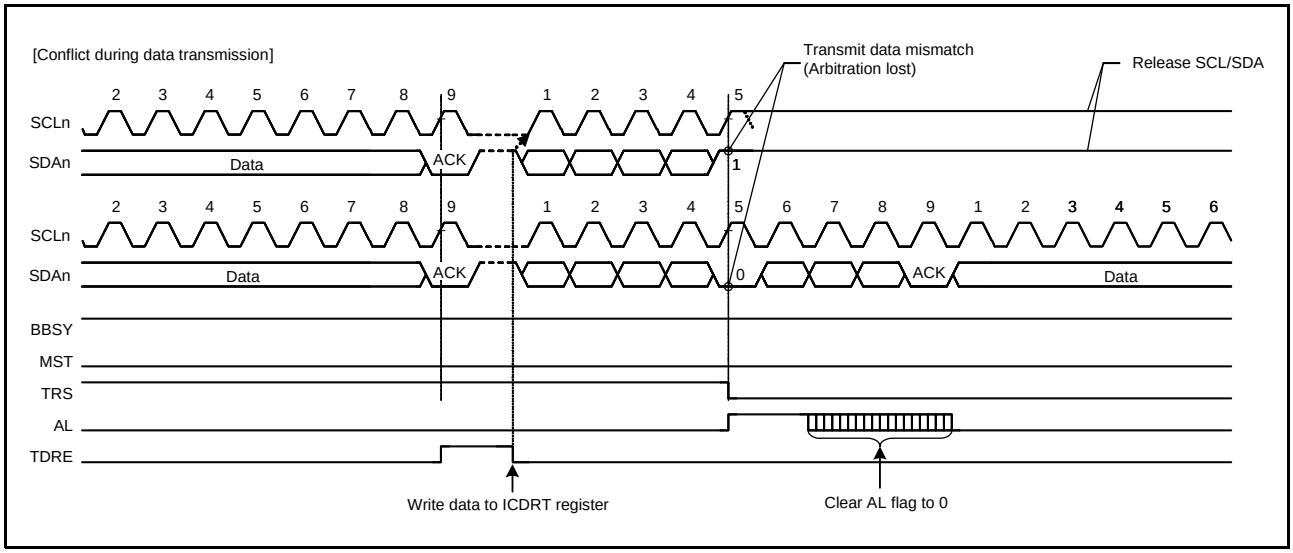


Figure 32.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

32.10 Start Condition/Restart Condition/Stop Condition Generating Function

32.10.1 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.

32.10.2 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.

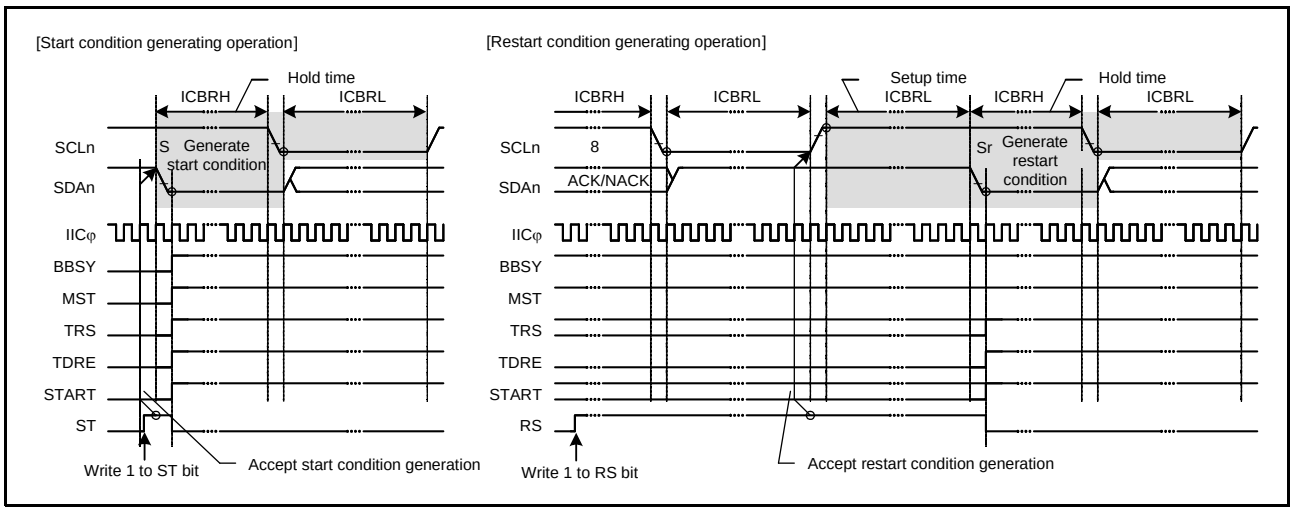


Figure 32.36 Start Condition/Restart Condition Generation Timing (ST and RS Bits)

32.10.3 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).

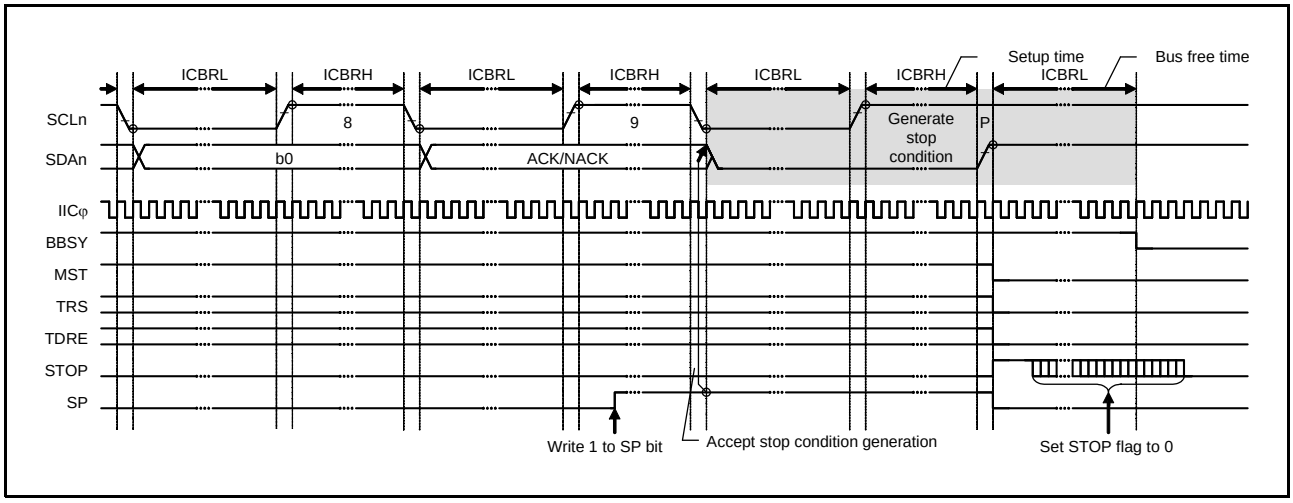


Figure 32.37 Stop Condition Generation Timing (SP Bit)

32.11 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.

32.11.1 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.

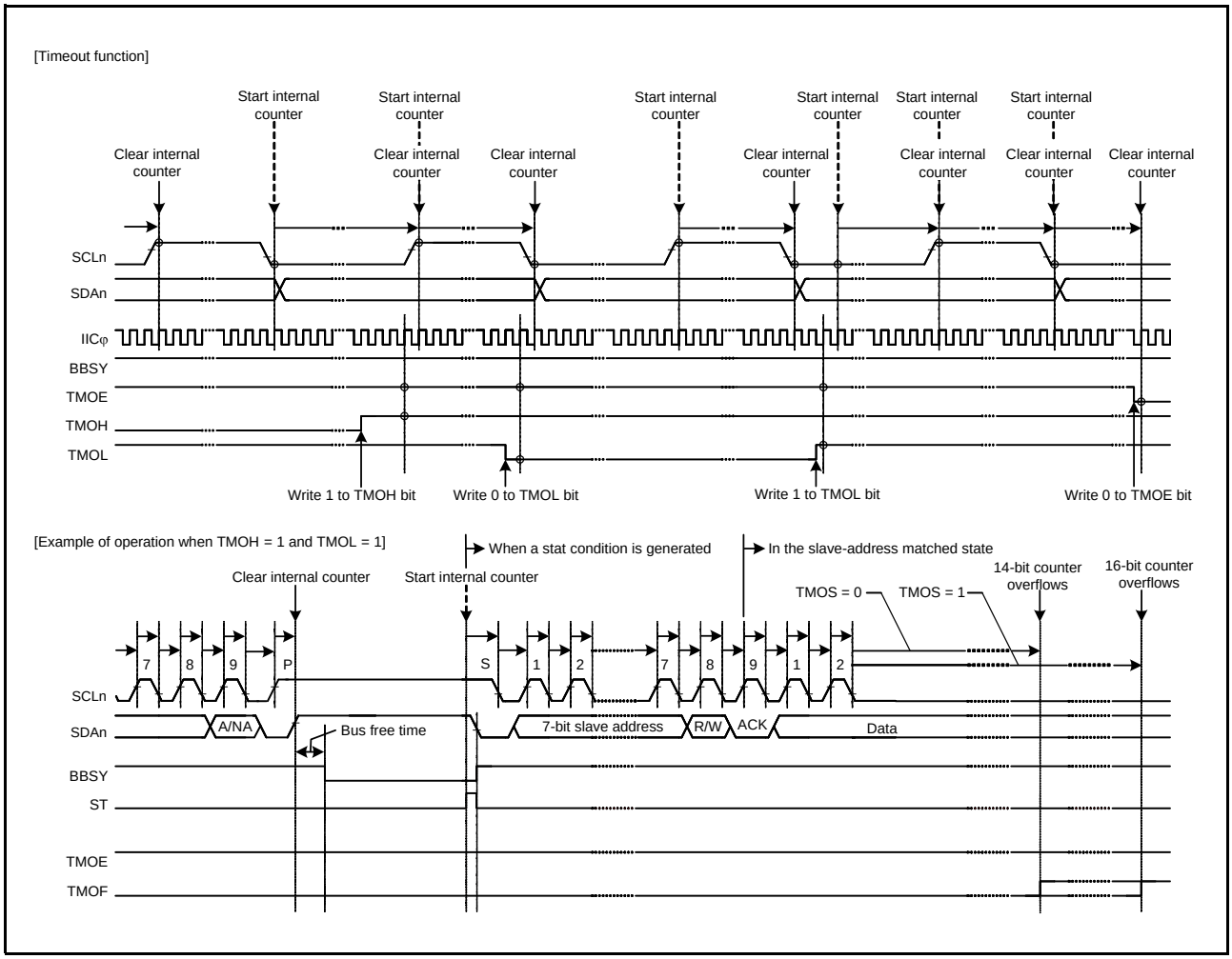


Figure 32.38 Timeout Function

32.11.2 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 32.39 shows the operation timing of the additional SCL output function (CLO bit).

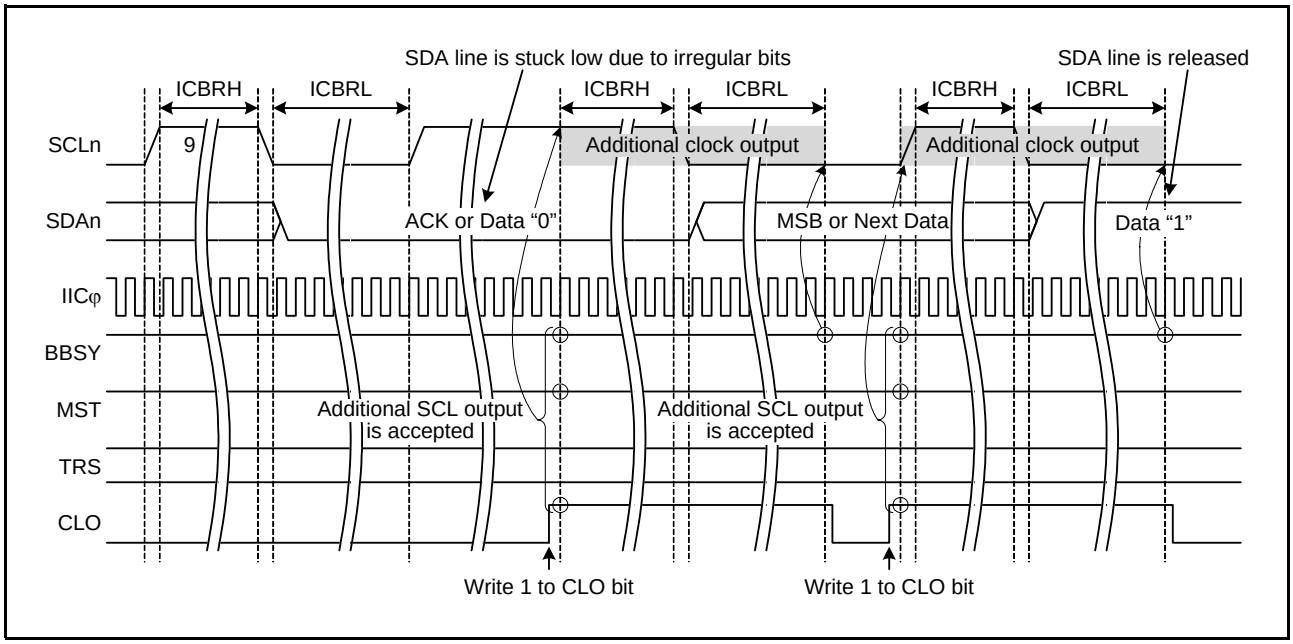


Figure 32.39 Additional SCL Output Function (CLO Bit)

32.11.3 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 32.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.

32.12 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.

32.12.1 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).

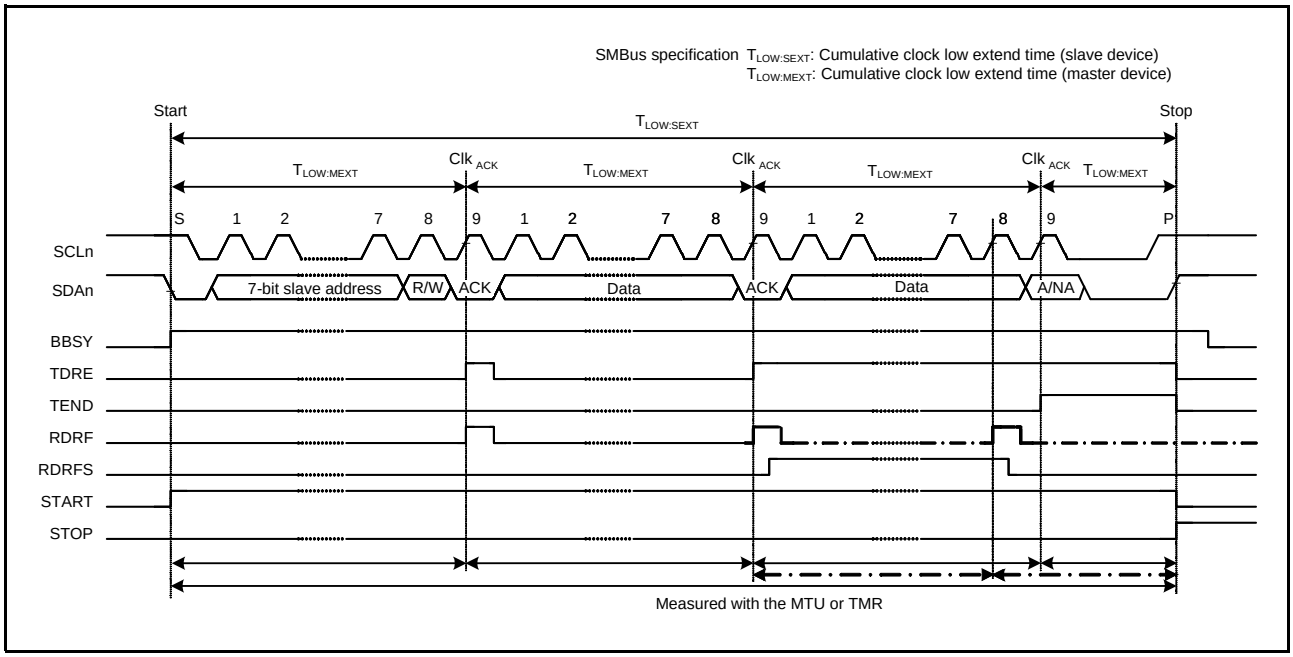


Figure 32.40 SMBus Timeout Measurement

32.12.2 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 35, 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.

32.12.3 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.

32.13 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 32.6 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 32.6 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.

32.13.1 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.

32.14 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 32.7 lists the reset states of registers and functions when a reset is applied or a condition is detected.

Table 32.7 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.

32.15 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

32.15.1 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 32.6.

32.16 Usage Notes

32.16.1 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.

32.16.2 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.

33. CAN FD Module (CANFD-Lite)

33.1 Overview

This MCU implements one channel of the CAN FD (Controller Area Network with Flexible Data Rate) module that complies with the ISO 11898-1:2015 Standard.

Table 33.1 lists the specifications of the CAN FD module, and Figure 33.1 shows a block diagram of the CAN FD module.

Table 33.1 CAN FD Module Specifications

Item	Description
Protocol	ISO 11898-1:2015 compliant
Data transfer rate	Arbitration phase: up to 1 Mbps Data phase: up to 8 Mbps*1
Operating frequency*2	Register block: up to 60 MHz (PCLKB) Message buffer RAM: up to 120 MHz (PCLKA)
Operating clock for data link layer (DLL clock)	Up to 60 MHz (either CANFDMCLK or CANFDCLK can be selected)
Frame types	Classic CAN (CAN 2.0) • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID) • Remote frame in base format (11-bit ID) • Remote frame in extended format (29-bit ID) CAN FD*1 • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID)
Data length	Classic CAN: 0 to 8 bytes CAN FD: 0 to 8, 12, 16, 20, 24, 32, 48, or 64 bytes*1
Message buffers	• 32 receive message buffers • Four transmit message buffers • One transmit queue Messages can be automatically transfered to the transmit queue.
FIFOs	Variable size FIFO buffers • Two receive FIFOs • One common FIFO that can be configured as a receive FIFO or transmit FIFO
Automatic transmission interval adjustment	Available when the common FIFO is configured as a transmit FIFO The interval between messages sent from the FIFO can be adjusted.
Acceptance filter	Filterable in the following fields: • IDE bit (base format, extended format, or both) • ID field • RTR bit (data frame or remote frame) (only for Classic CAN) • DLC field (data length) Protection function when the payload size is exceeded Acceptance filter list (AFL) entries can be updated during communication.
Software support	Label information is automatically added to received messages
Timer	Transmission and reception timestamp function
Power down function	Module start/stop function for each CAN node (CH_SLEEP and GL_SLEEP mode) Transition to module stop state is possible.
RAM	RAM with ECC protection

Note 1. This is only available for products that support the CAN FD protocol.
Note 2. The frequency ratio of PCLKA and PCLKB should be 2 : 1. Also, the frequency of PCLKB should be equal to or higher than that of the DLL clock.

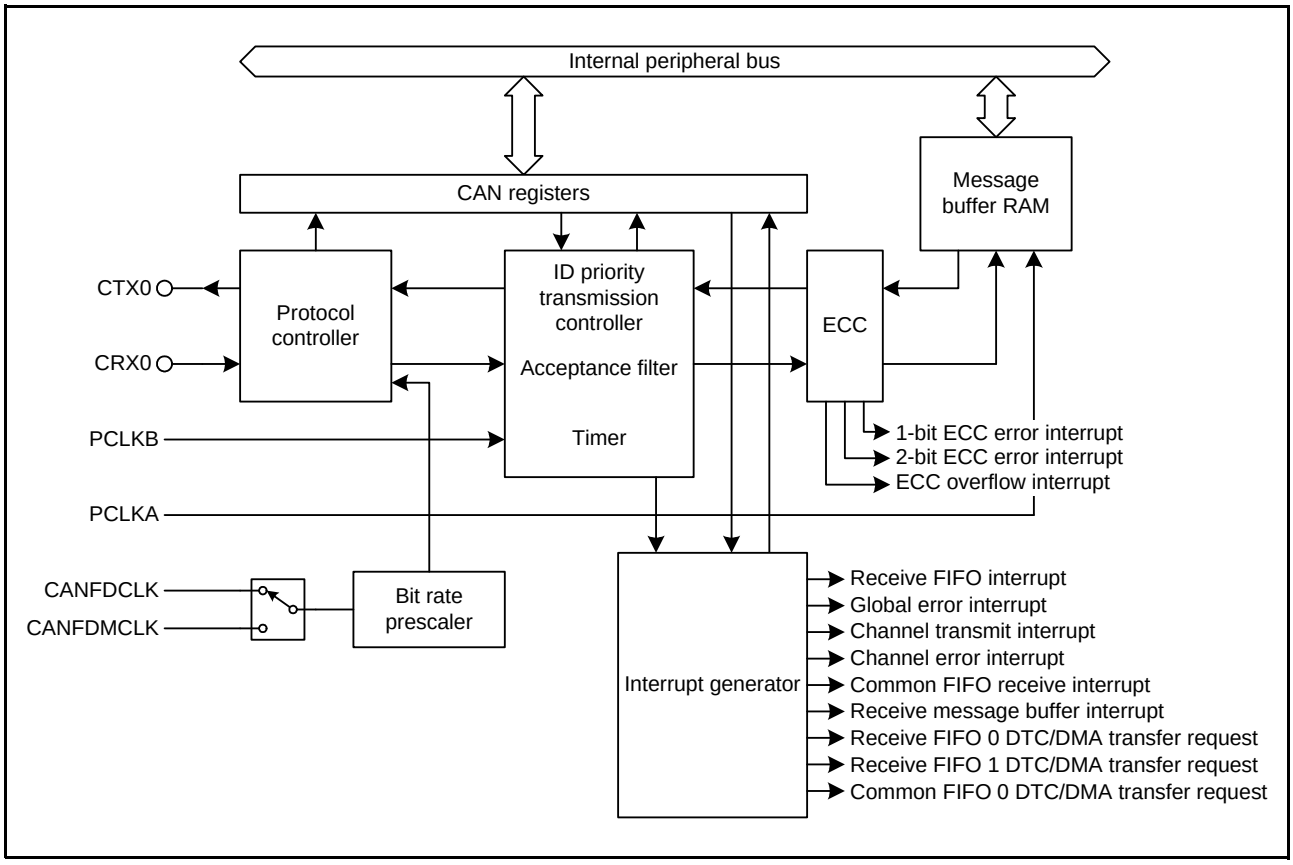


Figure 33.1 CAN FD Module Block Diagram

- CRX0, CTX0
  I/O pins of the CAN FD module
- Protocol controller
  Handles CAN FD protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling, etc.
- Message buffer RAM
  Used for message buffers or FIFOs for transmitting and receiving messages. Each message has an individual identifier, data length code, a data field, a message pointer for upper layer application, and a timestamp.
- Acceptance filter
  Performs filtering of received messages. The entries set in the acceptance filter list are used for the filtering process.
- Timers
  Two timers: one used for the receive timestamp function and the other for adjusting the message transmission interval from the transmit FIFO.

Table 33.2 lists the pin configuration of the CAN FD module.

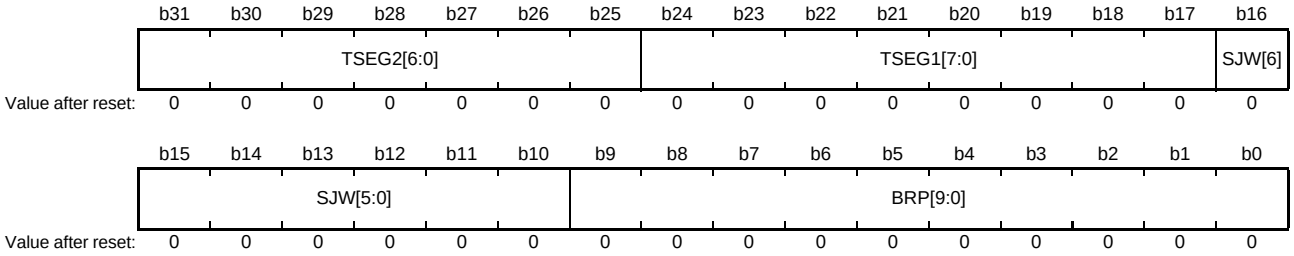
Table 33.2 CAN FD Module Pin Configuration

Pin Name	I/O	Function
CRX0	Input	Receive data input
CTX0	Output	Transmit data output

33.2 Register Descriptions

33.2.1 Nominal Bit Rate Configuration Register (NBCR)

Address(es): CANFD0.NBCR 000A 8000h



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	BRP[9:0]	Bit Rate Prescaler Setting	000h: No division 001h: Divided by 2 : : 3FEh: Divided by 1023 3FFh: Divided by 1024	R/W
b16 to b10	SJW[6:0]	Resynchronization Jump Width Control	00h: 1 Tq 01h: 2 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W
b24 to b17	TSEG1[7:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : FEh: 255 Tq FFh: 256 Tq	R/W
b31 to b25	TSEG2[6:0]	Time Segment 2 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W

This register is used to set the nominal bit rate during transmission and reception.
The value cannot be changed in CH_OPERATION or CH_SLEEP mode. Rewrite this register in CH_RESET or CH_HALT mode.
For details on the setting values, refer to section 33.4.1.2, Bit Timing.

BRP[9:0] Bits (Bit Rate Prescaler Setting)

These bits are used to define a period of 1 Tq (Time Quantum) that is the basis for CAN communication. Set the division ratio for the operating clock of the data link layer (DLL clock) selected by the GCFG.DLLCS bit. If the set value is n, the bit rate prescaler divides the DLL clock by n + 1.

SJW[6:0] Bits (Resynchronization Jump Width Control)

These bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 128 Tq can be set. Set a value less than or equal to that of the TSEG2[6:0] bits.

TSEG1[7:0] Bits (Time Segment 1 Control)

These bits are used to specify the total value (TSEG1) of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a Tq value. A value from 2 to 256 Tq can be set.

TSEG2[6:0] Bits (Time Segment 2 Control)

These bits are used to specify the value (TSEG2) of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 128 Tq can be set. Set a value less than that of the TSEG1[7:0] bits.

33.2.2 Channel Control Register (CHCR)

Address(es): CANFD0.CHCR 000A 8004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ROME	BFT	—	—	—	CTMS[1:0]		CTME	EDM	BOM[1:0]		—	TDCVIE	SCOVIE	ECOVIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Channel Mode Control	b1 b0 0 0: Requests transition to CH_OPERATION mode 0 1: Requests transition to CH_RESET mode 1 0: Requests transition to CH_HALT mode 1 1: Keep current mode	R/W
b2	SLPRQ	CH_SLEEP Mode Request	0: Requests release from CH_SLEEP mode 1: Requests transition to CH_SLEEP mode	R/W
b3	RTBO	Forced Recovery from Bus-Off*1	0: Forced recovery from bus-off state is disabled. 1: Forced recovery from bus-off state is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BEIE	Bus Error Interrupt Enable*2	0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.	R/W
b9	EWIE	Error Warning Interrupt Enable*2	0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.	R/W
b10	EPIE	Error Passive Interrupt Enable*2	0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.	R/W
b11	BOEIE	Bus-Off Entry Interrupt Enable*2	0: Bus-off entry interrupt is disabled. 1: Bus-off entry interrupt is enabled.	R/W
b12	BORIE	Bus-Off Recovery Interrupt Enable*2	0: Bus-off recovery interrupt is disabled. 1: Bus-off recovery interrupt is enabled.	R/W
b13	OLIE	Overload Interrupt Enable*2	0: Overload interrupt is disabled. 1: Overload interrupt is enabled.	R/W
b14	BLIE	Bus Lock Interrupt Enable*2	0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.	R/W
b15	ALIE	Arbitration Lost Interrupt Enable*2	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b16	TAIE	Transmission Abort Interrupt Enable *2	0: Transmission abort interrupt is disabled. 1: Transmission abort interrupt is enabled.	R/W
b17	ECOVIE	Error Occurrence Counter Overflow Interrupt Enable*2	0: Error occurrence counter overflow interrupt is disabled. 1: Error occurrence counter overflow interrupt is enabled.	R/W
b18	SCOVIE	Success Occurrence Counter Overflow Interrupt Enable*2	0: Success occurrence counter overflow interrupt is disabled. 1: Success occurrence counter overflow interrupt is enabled.	R/W
b19	TDCVIE	Transceiver Delay Compensation Violation Interrupt Enable*2, *3	0: Transceiver delay compensation violation interrupt is disabled. 1: Transceiver delay compensation violation interrupt is enabled.	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22, b21	BOM[1:0]	Bus-Off Recovery Mode Select*2	b22 b21 0 0: Normal mode (ISO 11898-1 compliant) 0 1: Automatically enters CH_HALT mode at bus-off entry 1 0: Automatically enters CH_HALT mode at bus-off end 1 1: Enters CH_HALT mode by software (during bus-off recovery)	R/W
b23	EDM	Error Display Mode Select*4	0: Only the first error detected is indicated. 1: All detected errors are indicated.	R/W
b24	CTME	Channel Test Mode Enable	0: Channel test mode is disabled. 1: Channel test mode is enabled.	R/W
b26, b25	CTMS[1:0]	Channel Test Mode Select*5	b26 b25 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loop back mode) 1 1: Self-test mode 1 (internal loop back mode)	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	BFT	Bit Flip Test*5	0: First bit of received data stream is not inverted. 1: First bit of received data stream is inverted.	R/W
b31	ROME	Restricted Operation Mode Enable *3, *5	0: Limited operation mode is disabled. 1: Limited operation mode is enabled.	R/W

Note 1. Set this bit in CH_OPERATION mode.
Note 2. Rewrite these bits in CH_RESET mode.
Note 3. Do not set this bit to 1 in Classic Only mode.
Note 4. Set this bit in CH_RESET or CH_HALT mode.
Note 5. Set these bits in CH_HALT mode.

This register controls the modes of the channel. It is also used to enable interrupt generation when an error is detected on the CAN bus and to set the test mode.

MDC[1:0] Bits (Channel Mode Control)

The MDC[1:0] bits are used to specify the mode of the CAN channel. For the CAN mode transition, refer to section 33.3.2, Channel Modes.
In CH_SLEEP mode, the value of these bits cannot be changed. When the CANFD module is in GL_HALT mode, these bits can only be set to 10b (CH_HALT mode) or 01b (CH_RESET mode).
These bits automatically become 10b when the mode is changed to CH_HALT mode by the setting of the CHCR.BOM[1:0] bits.
When writing to these bits and transition to CH_HALT mode (at the entry to bus-off state when the CHCR.BOM[1:0] bits are 01b, and at the end of bus-off state when the CHCR.BOM[1:0] bits are 10b) occur at the same time, priority is given to writing from the CPU. The value of these bits are automatically updated only if the above event occurs when these bits are 00b (CH_OPERATION mode).

SLPRQ Bit (CH_SLEEP Mode Request)

This bit is used to control the transition to CH_SLEEP mode and the return from CH_SLEEP mode.
Setting this bit to 1 in CH_RESET mode requests the channel to transition to CH_SLEEP mode. Setting this bit to 0 in CH_SLEEP mode requests the channel to transition to CH_RESET mode. These bits cannot be changed in other modes.

RTBO Bit (Forced Recovery from Bus-Off)

This bit is used to forcibly recover from the bus-off state. Use this bit only when the CHCR.BOM[1:0] bits are 00b.
This bit is automatically set to 0. The read value is always 0.
When this bit is set to 1 during bus-off state, the channel transitions from bus-off state to integrating state within 1 bit time. Also the CHSR.REC[7:0] and TEC[7:0] bits are set to 00h and the CHSR.BOST flag is set to 0. Other registers and bits do not change. In this case, even if the bus-off recovery interrupt is enabled, the bus-off recovery interrupt does not

generated.
If this bit is set to 1 except in the bus-off state, nothing occurs.

BEIE Bit (Bus Error Interrupt Enable)

When the CHESR.BEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EWIE Bit (Error Warning Interrupt Enable)

When the CHESR.EWDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EPIE Bit (Error Passive Interrupt Enable)

When the CHESR.EPDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the CHESR.BOEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the CHESR.BORDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

OLIE Bit (Overload Interrupt Enable)

When the CHESR.OLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BLIE Bit (Bus Lock Interrupt Enable)

When the CHESR.BLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

ALIE Bit (Arbitration Lost Interrupt Enable)

When the CHESR.ALDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TAIE Bit (Transmission Abort Interrupt Enable)

When the transmission from the transmit message buffer of the channel is successfully aborted while this bit is 1, a channel transmit interrupt request is generated.

ECOVIE Bit (Error Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.ECOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

SCOVIE Bit (Success Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.SCOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TDCVIE Bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the FDSTS.TDCFVF flag is set to 1 while this bit is 1, a channel error interrupt request is generated. Do not set this bit to 1 in Classic only mode.

BOM[1:0] Bits (Bus-Off Recovery Mode Select)

These bits control the recovery timing from the bus-off state.

EDM Bit (Error Display Mode Select)

This bit controls the indication mode of the error flags (b14 to b8) in the CHESR register.
When this bit is 0, only the flag corresponding to the first error detected becomes 1. If multiple errors are detected at the same time, all corresponding flags become 1. The other flags do not become 1 until all the flags in b14 to b8 have been

cleared.
When this bit is 1, the error flags are updated each time an error is detected.

CTME Bit (Channel Test Mode Enable)

This bit is used to enable the channel test mode.
This bit cannot be changed in CH_SLEEP mode. Change this bit in CH_HALT mode. When the channel transitions to CH_RESET mode, this bit becomes 0.

CTMS[1:0] Bits (Channel Test Mode Select)

These bits are used to select the test mode.
These bits cannot be changed in CH_SLEEP or CH_RESET mode.When the channel transitions to CH_RESET mode, these bits become 00b.

BFT Bit (Bit Flip Test)

This bit is used to check the CRC generator in the protocol controller.
Setting this bit to 1 inverts the first bit of the message data stream (ID bit) so that the result of the internally generated CRC does not match the received CRC value.
Note that as a result of bit inversion, a stuff error may be detected instead of a CRC error. Refer to the bit stuffing rule when using this function.
The CRC value generated internally can be checked in the following bits.

- The CHESR.CRC15[14:0] bits (for Classical CAN frames)
- The FDCRC.CRC21[20:0] bits (for CAN FD frames)

When using the BFT bit, it is required for other CAN nodes to send the reference message.

Note: Since the transmit and receive modes share the same CRC generator, there is no need to test individually CRC errors in transmit mode.

Bit flip test mode is enabled when both the BFT and CTME bits are 1 and the CTMS[1:0] bits are 00b (basic test mode).
Using this function on the transmit node causes a bit error or arbitration lost.
When the channel transitions to CH_RESET mode, this bit becomes 0.

ROME Bit (Restricted Operation Mode Enable)

When both this bit and the CTME bit are 1, the limited operation mode is enabled. Use this mode only in basic test mode (CTMS[1:0] bits = 00b). Also, do not set this bit to 1 in Classic only mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

33.2.3 Channel Status Register (CHSR)

Address(es): CANFD0.CHSR 000A 8008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEC[7:0]								REC[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
	—	—	—	—	—	—	—	RESI	CRDY	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	CH_RESET Status Flag	0: Not in CH_RESET mode 1: In CH_RESET mode	R
b1	HLTST	CH_HALT Status Flag	0: Not in CH_HALT mode 1: In CH_HALT mode	R
b2	SLPST	CH_SLEEP Status Flag	0: Not in CH_SLEEP mode 1: In CH_SLEEP mode	R
b3	EPST	Error Passive Status Flag	0: Not in error passive state 1: In error passive state	R
b4	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b5	TRMST	Transmit Status Flag	0: Channel is not transmitting 1: Transmission in progress	R
b6	RECST	Receive Status Flag	0: Channel is not receiving 1: Reception in progress	R
b7	CRDY	Communication Ready Flag	0: Channel is not ready for communication 1: Channel is ready for communication	R
b8	RESI	Receive ESI Flag*1	0: No message with ESI flag set to 1 was received. 1: At least 1 message with ESI flag set to 1 was received.	R/(W) *2
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	REC[7:0]	Reception Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
b31 to b24	TEC[7:0]	Transmission Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the mode, error and transmission or reception status of the CAN channel together with its reception and transmission error count values.

RSTST Flag (CH_RESET Status Flag)

The RSTST flag indicates whether the CAN channel is in CH_RESET mode.
This flag is automatically set to 1 when the CAN channel enters CH_RESET mode, and is automatically set to 0 when the CAN channel exits CH_RESET mode. When the mode is changed from CH_RESET mode to CH_SLEEP mode, the RSTST flag remains 1.

HLTST Flag (CH_HALT Status Flag)

The HLTST flag indicates whether the CAN channel is in CH_HALT mode.

This flag is automatically set to 1 when the CAN channel enters CH_HALT mode, and is automatically set to 0 when the CAN channel exits CH_HALT mode.

SLPST Flag (CH_SLEEP Status Flag)

The SLPST flag indicates whether the CAN channel is in CH_SLEEP mode.
This flag is automatically set to 1 when the CAN channel enters CH_SLEEP mode, and is automatically set to 0 when the CAN channel exits CH_SLEEP mode.

EPST Flag (Error Passive Status Flag)

The EPST flag indicates whether the CAN channel has entered the error passive state.
This flag is automatically set to 1 when the value of the REC[7:0] or TEC[7:0] bits exceeds 127.
This flag is automatically set to 0 when the CAN channel exits the error passive state or enters CH_RESET mode.

BOST Flag (Bus-Off Status Flag)

The BOST flag indicates whether the CAN channel has entered the bus-off state.
This flag is automatically set to 1 when the value of the TEC[7:0] bits exceeds 255 and the CAN channel is in the bus-off state.
This flag is automatically set to 0 when the CAN channel exits bus-off state.

TRMST Flag (Transmit Status Flag)

The TRMST flag indicates whether the CAN channel is transmitting a message.
This flag is automatically set to 1 when the CAN channel is operating as a transmitter node or is in the bus-off state, and is automatically set to 0 when the CAN channel is in the idle state or starts operating as a receiver node.

RECST Flag (Receive Status Flag)

The RECST flag indicates whether the CAN channel is receiving a message.
This flag is automatically set to 1 when the CAN channel is operating as a receiver node, and is automatically set to 0 when the CAN channel is in the idle state or starts operating as a transmitter node.

CRDY Flag (Communication Ready Flag)

The CRDY flag indicates whether the CAN channel is ready for communication.
This flag is automatically set to 1 when the CAN channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting CH_RESET or CH_HALT mode.
This flag is automatically set to 0 when the CAN channel is in CH_RESET or CH_HALT mode.

Note: This flag is 1 in the bus-off state.

RESI Flag (Receive ESI Flag)

The RESI flag is set to 1 when the ESI bit is sampled recessively for a received message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is cleared by writing 0 to it. This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only this flag to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

REC[7:0] Bits (Reception Error Count)

The REC[7:0] bits indicate the value of the reception error counter.
The value in bus-off state is indeterminate.
These bits are automatically set to 00h when the CANFD module enters GL_RESET or the CAN channel is in

CH_RESET mode.

TEC[7:0] Bits (Transmission Error Count)

The TEC[7:0] bits indicate the value of the transmission error counter.

These bits are automatically set to 00h when CANFD module is in GL_RESET or CAN channel is in CH_RESET mode.

33.2.4 Channel Error Status Register (CHESR)

Address(es): CANFD0.CHESR 000A 800Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CRC15[14:0]														
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
	—	ADEDF	B0EDF	B1EDF	CEDF	AEDF	FEDF	SEDF	ALDF	BLDF	OLDF	BORDF	BOEDF	EPDF	EWDF	BEDF
Value after 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	BEDF	Bus Error Detect Flag*1	0: Bus error is not detected 1: Bus error is detected	R/(W) *2
b1	EWDF	Error Warning Detect Flag*1	0: Error warning is not detected 1: Error warning is detected	R/(W) *2
b2	EPDF	Error Passive Detect Flag*1	0: Error passive is not detected 1: Error passive is detected	R/(W) *2
b3	BOEDF	Bus-Off Entry Detect Flag*1	0: Bus-off entry is not detected 1: Bus-off entry is detected	R/(W) *2
b4	BORDF	Bus-Off Recovery Detect Flag*1	0: Bus-off recovery is not detected 1: Bus-off recovery is detected	R/(W) *2
b5	OLDF	Overload Detect Flag*1	0: Overload is not detected 1: Overload is detected	R/(W) *2
b6	BLDF	Bus Lock Detect Flag*1	0: Bus lock is not detected 1: Bus lock is detected	R/(W) *2
b7	ALDF	Arbitration Lost Detect Flag*1	0: Arbitration lost is not detected 1: Arbitration lost is detected	R/(W) *2
b8	SEDF	Stuff Error Detect Flag*1	0: Stuff error is not detected 1: Stuff error is detected	R/(W) *2
b9	FEDF	Form Error Detect Flag*1	0: Form error is not detected 1: Form error is detected	R/(W) *2
b10	AEDF	Acknowledge Error Detect Flag*1	0: Acknowledge error is not detected 1: Acknowledge error is detected	R/(W) *2
b11	CEDF	CRC Error Detect Flag*1	0: CRC error is not detected 1: CRC error is detected	R/(W) *2
b12	B1EDF	Bit 1 Error Detect Flag*1	0: Bit 1 error is not detected 1: Bit 1 error is detected	R/(W) *2
b13	B0EDF	Bit 0 Error Detect Flag*1	0: Bit 0 error is not detected 1: Bit 0 error is detected	R/(W) *2
b14	ADEDF	ACK Delimiter Error Detect Flag*1	0: Acknowledge delimiter error is not detected 1: Acknowledge delimiter error is detected	R/(W) *2
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30-b16	CRC15[14:0]	CRC_15 Test	These bits show the CRC_15 value calculated for the CAN2.0 CAN frame.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of various error conditions detectable regardless of the interrupt enable/disable setting of the Channel Control Register (CHCR). It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for the conditions under which each error occurs.

Only a single bit can be cleared at a time. Do not use the bit clear instruction to clear the flag. Set only the flag to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

BEDF Flag (Bus Error Detect Flag)

The BEDF flag indicates a detection of an error state, flagged by b14 to b8 in this register.
This flag is automatically set to 1 when a bus error is detected, and is automatically set to 0 when the CAN channel is in CH_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

EWDF Flag (Error Warning Detect Flag)

The EWDF flag indicates whether an error warning condition has been detected for the CAN channel.
This flag is automatically set to 1 when the value of either the CHSR.TEC[7:0] or REC[7:0] bits exceeds 95.
The setting of this flag only occurs when the value of the TEC[7:0] or REC[7:0] bits initially exceeds 95. Therefore, if the TEC[7:0] or REC[7:0] bits remains > 95 and the EWDF flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 96 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 95 to a value > 95.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the CAN channel is in CH_RESET mode.

EPDF Flag (Error Passive Detect Flag)

The EPDF flag indicates a detection of a CAN channel error passive state.
This flag is automatically set to 1 when the CAN error state becomes error passive state.
The setting of this flag only occurs when the value of either the CHSR.TEC[7:0] or REC[7:0] bits initially exceeds 127. Therefore, if the value of either the TEC[7:0] or REC[7:0] bits remains > 127 and the flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 128 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 127 to a value > 127.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the CAN channel is in CH_RESET mode.

BOEDF Flag (Bus-Off Entry Detect Flag)

The BOEDF flag indicates a detection of a CAN channel bus-off entry state.
This flag is automatically set to 1 when the CAN error state enters the bus-off state.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.

BORDF Flag (Bus-Off Recovery Detect Flag)

The BORDF flag indicates a detection of a CAN channel bus-off recovery state.
This flag is automatically set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CHCR.BOM[1:0] bits are 00b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 10b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 11b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs.

The flag is not set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CH_RESET mode is requested
- When setting the CHCR.RTBO bit to 1 (the CAN channel returns to error active)
- When the CHCR.BOM[1:0] bits are 01b

- When the CHCR.BOM[1:0] bits are 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set to 1.

OLDF Flag (Overload Detect Flag)

The OLDF flag indicates a detection of a CAN channel overload state.
This flag is automatically set to 1 when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.

BLDF Flag (Bus Lock Detect Flag)

The BLDF flag indicates a detection of a CAN channel bus lock condition.
This flag is automatically set to 1 when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.

ALDF Flag (Arbitration Lost Detect Flag)

The ALDF flag indicates a detection of a CAN channel bus arbitration lost condition.
The flag is automatically set to 1 when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.

SEDF Flag (Stuff Error Detect Flag)

The SEDF flag indicates a detection of a CAN stuff error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a stuff error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

FEDF Flag (Form Error Detect Flag)

The FEDF flag indicates a detection of a CAN form error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

AEDF Flag (Acknowledge Error Detect Flag)

The AEDF flag indicates a detection of a CAN acknowledge error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.

This flag is automatically set to 1 when an acknowledge error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CEDF Flag (CRC Error Detect Flag)

The CEDF flag indicates a detection of a CAN CRC error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a CRC error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B1EDF Flag (Bit 1 Error Detect Flag)

The B1EDF flag indicates a detection of a recessive bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B0EDF Flag (Bit 0 Error Detect Flag)

The B0EDF flag indicates a detection of a dominant bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

ADEDF Flag (ACK Delimiter Error Detect Flag)

The ADEDF flag indicates a detection of an acknowledge delimiter bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected during the acknowledge delimiter state of frame transmission. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CRC15[14:0] Bits (CRC_15 Test)

The CRC15[14:0] bits indicate the calculated CRC_15 value when the CHCR.CTME bit is 1 (channel test mode enabled). If the CHCR.CTME bit is 0, then these bits are always read as 0000h.
The CRC_15 value which is read from these bits show the CAN2.0 CRC value calculated by the CAN channel logic.
The value of the CRC15[14:0] bits is updated in the first bit of the CRC field of the Classical CAN frame.
These bits are automatically set to 0000h when the CAN channel is in CH_RESET mode.

33.2.5 Data Bit Rate Configuration Register (DBCR)

Address(es): CANFD0.DBCR 000A 8100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SJW[3:0]				—	—	—	—	TSEG2[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
	—	—	—	TSEG1[4:0]					BRP[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	BRP[7:0]	Bit Rate Prescaler Setting	00h: No division 01h: Divided by 2 : : FEh: Divided by 255 FFh: Divided by 256	R/W
b12 to b8	TSEG1[4:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : 1Eh: 31 Tq 1Fh: 32 Tq	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	TSEG2[3:0]	Time Segment 2 Control	0h: Setting prohibited 1h: 2 Tq : : Eh: 15 Tq Fh: 16 Tq	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	SJW[3:0]	Resynchronization Jump Width Control	0h: 1 Tq 1h: 2 Tq : : Fh: 16 Tq	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register configures the transmission/reception data bit rate parameters of the channel.
In Classic only mode, there is no need to configure this register.

BRP[7:0] Bits (Bit Rate Prescaler Setting)

The BRP[7:0] bits define the number of the DLL clock contained in 1 Tq (Time Quantum).

TSEG1[4:0] Bits (Time Segment 1 Control)

The TSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 Tq can be set.
The TSEG1[4:0] bits are also used to set the propagation time segment.
Do not write any other value to these bits. Refer to section 33.4.1.2, Bit Timing for more details.

TSEG2[3:0] Bits (Time Segment 2 Control)

The TSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A

value from 2 to 16 Tq can be set.
Do not write any other value to these bits.

SJW[3:0] Bits (Resynchronization Jump Width Control)

The SJW[3:0] bits set the resynchronization jump width. A value from 1 to 16 Tq can be set.

33.2.6 CAN FD Configuration Register (FDCFG)

Address(es): CANFD0.FDCFG 000A 8104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TESI	TDCE	SSPC	—	—	—	—	—	ECC[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	ECC[2:0]	Error Occurrence Counter Configuration*2	b2 b0 0 0 0: All CAN transmitter or receiver frames 0 0 1: All CAN transmitter frames 0 1 0: All CAN receiver frames 0 1 1: Setting prohibited 1 0 0: Only transmitter or receiver CAN FD data-phase 1 0 1: Only transmitter CAN FD data-phase 1 1 0: Only receiver CAN FD data-phase 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	SSPC	Secondary Sample Point Configuration*2	0: Measured + offset 1: Offset-only	R/W
b9	TDCE	Transceiver Delay Compensation Enable*2	0: Transceiver delay compensation is disabled 1: Transceiver delay compensation is enabled	R/W
b10	TESI	Transmit ESI Configuration*2	0: The ESI flag in the transmission frame reflects the error status of the node itself. 1: The ESI flag in the transmission frame reflects the ESI bit in the message buffer if the node is not error passive, and the error status of the node itself If the node is error passive.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TDCO[7:0]	Transceiver Delay Compensation Offset*2	Sets the offset value for the transceiver delay compensation	R/W
b27 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	FDOE	FD Only Mode Enable*3	0: FD only mode is disabled 1: FD only mode is enabled	R/W
b29	REFE	Receive Edge Filter Enable*3	0: Reception edge filter is disabled 1: Reception edge filter is enabled	R/W
b30	CLOE	Classic Only Mode Enable*3, *4	0: Classic only mode is disabled 1: Classic only mode is enabled	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The value after reset is 0 for products that support the CAN FD protocol, and 1 for products that support only CAN 2.0 protocol.
Note 2. Set these bits in CH_RESET or CH_HALT mode.
Note 3. Set these bits in CH_RESET mode.
Note 4. This bit can only be written for products that support the CAN FD protocol. For products that support only CAN 2.0 protocol, this bit is reserved and fixed to 1.

This register configures which communication direction (transmitter/receiver) errors are counted.

ECC[2:0] Bits (Error Occurrence Counter Configuration)

The ECC[2:0] bits select which type of CAN frame the protocol errors should be counted for.

SSPC Bit (Secondary Sample Point Configuration)

The SSPC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN channel. If the bit is 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.
Do not set this bit to 1 in Classic only mode.

TDCE Bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN channel.
Do not set this bit to 1 in Classic only mode.

TESI Bit (Transmit ESI Configuration)

The TESI bit selects whether to reflect the error status of the node itself or the value of the ESI bit in the message buffer (CFB0.HF2.ESI bit or TMBn.HF2.ESI bit) in the ESI flag of the transmission message.
Do not set this bit to 1 in Classic only mode.

TDCO[7:0] Bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the offset of the secondary sample point. How this value is used, depends on the SSPC setting. If the SSPC bit is 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in the TDCO[7:0] bits, rounded down to the nearest integer number of Tq. Otherwise, the result is equal to the value in the TDCO[7:0] bits. Refer to section 33.4.1.5, Transceiver Delay Compensation for details.
The actual offset value is interpreted as TDCO[7:0] + 1. For example, if 4 is set in TDCO[7:0], the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.
Do not set to these bits in Classic only mode.

FDOE Bit (FD Only Mode Enable)

The FDOE bit enables the transmission and reception of CAN FD frames only. If enabled, communication in Classical CAN frame format is disabled. The value of the FDF bit in the message buffer (CFB0.HF2.FDF bit or TMBn.HF2.FDF bit) is arbitrary because transmission of Classical CAN frames is not possible.
If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and responds with error frames. If a Classical CAN frame is configured for transmission, the FDF bit is transmitted recessive, therefore a CAN FD frame is transmitted. If the data length code (DLC) is configured to be 9 bytes or more, the remaining data bytes are padded with CCh.
Do not set the FDOE and CLOE bits to 1 simultaneously.

REFE Bit (Receive Edge Filter Enable)

The REFE bit enables the reception edge filter during the integrating state. When this bit is 1, two or more consecutive dominant Tq are required to detect an edge for hard synchronization.
Do not set this bit to 1 in Classic only mode.

CLOE Bit (Classic Only Mode Enable)

The CLOE bit enables the Classic only mode. If this bit is set to 1, the protocol controller can only transmit Classical CAN frames and responds to CAN FD frames with a form error or CRC error.
Do not set the CLOE and FDOE bits to 1 simultaneously.

Table 33.3 Operation Mode Configuration

CLOE bit	FDOE bit	Operation mode
0	0	CAN FD mode
0	1	FD only mode
1	0	Classic only mode
1	1	Setting prohibited

33.2.7 CAN FD Control Register (FDCTR)

Address(es): CANFD0.FDCTR 000A 8108h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCCL	ECCL
Value after 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	ECCL	Error Occurrence Counter Clear	When 1 is written to this bit, the error occurrence counter is cleared. This bit is read as 0.	R/W
b1	SCCL	Success Occurrence Counter Clear	When 1 is written to this bit, the success occurrence counter is cleared. This bit is read as 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the error and success occurrence counters.

ECCL Bit (Error Occurrence Counter Clear)

The ECCL bit is used to clear the error occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel is in CH_RESET mode.

SCCL Bit (Success Occurrence Counter Clear)

The SCCL bit is used to clear the success occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel is in CH_RESET mode.

33.2.8 CAN FD Status Register (FDSTS)

Address(es): CANFD0.FDSTS 000A 810Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SC[7:0]								EC[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
	TDCV	—	—	—	—	—	SCOV	ECOV	TDCR[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	TDCR[7:0]	Transceiver Delay Compensation Result	Indicates the transceiver delay compensation result when the transceiver delay has been measured	R
b8	ECOV	Error Occurrence Counter Overflow Flag*1	0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/(W) *2
b9	SCOV	Success Occurrence Counter Overflow Flag*1	0: Success occurrence counter has not overflowed 1: Success occurrence counter has overflowed	R/(W) *2
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TDCV	Transceiver Delay Compensation Violation Flag*1	0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/(W) *2
b23 to b16	EC[7:0]	Error Occurrence Counter	These bits show the error occurrence counter value.	R
b31 to b24	SC[7:0]	Success Occurrence Counter	These bits show the success occurrence counter value.	R

Note 1. Set this flag to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] Bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.
The measured delay is represented by the number of the DLL clock cycles. The result depends on the configuration of the FDCFG.SSPC bit and the offset value in the FDCFG.TDCO[7:0] bits. Refer to section 33.4.1.5, Transceiver Delay Compensation for details.
The TDCR[7:0] bits are updated at the falling edge between the FDF bit and res bit when the FDCFG.SSPC bit is 0 and the FDCFG.TDCE bit is 1 (transceiver delay compensation is enabled).
These bits are automatically set to 0 when the CAN channel is in CH_RESET mode.

ECOV Flag (Error Occurrence Counter Overflow Flag)

The ECOV flag indicates whether the CAN channel error occurrence counter has overflowed.
This flag is set to 1 if a CAN bus error specified in the FDCFG.ECC[2:0] bits is detected when the EC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

SCOV Flag (Success Occurrence Counter Overflow Flag)

The SCOV flag indicates whether the CAN channel success occurrence counter has overflowed.

This flag is set to 1 if a successful message reception or successful message transmission occurs when the SC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

TDCV Flag (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.
The transceiver delay has some variations depending on the physical parameters such as temperature. The TDCR[7:0] bits are updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCV flag captures this violation.
This flag is set to 1 when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times – 2 DLL clock) and the internal bit overruns.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

EC[7:0] Bits (Error Occurrence Counter)

The EC[7:0] bits are used together with the SC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
This higher error rate can be detected depending on the configuration of the FDCFG.ECC[2:0] bits.
The EC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to the FDCTR.ECCL bit.
These bits are updated when an error occurs, according to the configuration of the FDCFG.ECC[2:0] bits. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel is in CH_RESET mode.

SC[7:0] Bits (Success Occurrence Counter)

The SC[7:0] bits are used together with the EC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
The SC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to the FDCTR.SCCL bit.
These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel is in CH_RESET mode.

Note: In Loopback mode, the counter is incremented twice.

33.2.9 CAN FD CRC Register (FDCRC)

Address(es): CANFD0.FDCRC 000A 8110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SBC[3:0]				—	—	—	CRC21[20:16]				
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
	CRC21[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
b20 to b0	CRC21[20:0]	CRC_21 Test	These bits show the CRC_17 value or CRC_21 value calculated for the CAN FD frame.	R
b23 to b21	—	Reserved	These bits are read as 0.	R
b27 to b24	SBC[3:0]	Stuff Bit Counter	These bits shows the stuff bit count (Mod 8) for the CAN FD frame.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

This register holds the CRC value calculated for the CAN FD frame.

CRC21[20:0] Bits (CRC_21 Test)

The calculated CRC_17 value or CRC_21 value can be read from this bits when the CHCR.CTME bit is 1 (channel test mode).
When the CHCR.CTME bit is 0, the CRC21[20:0] bits are read as 000000h.
The CRC21[20:0] bits is updated in the first bit of the CRC field of the CAN FD frame.
When the CRC_17 field is used, the CRC21[20:17] bits are read as 0.
These bits are automatically set to 000000h when the CAN channel is in CH_RESET mode.

SBC[3:0] Bits (Stuff Bit Counter)

The SBC[3:0] bits contain the stuff count value of the CAN FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Graycoded) for a CAN FD frame when the CHCR.CTME bit is enabled in SBC[3:1]. SBC[0] is the parity bit.
When the CHCR.CTME bit is 0, the SBC[3:0] bits are always read as 0000b.
The value of the SBC[3:0] bits is updated in the first bit of CRC field of the CAN FD frame. These bits are automatically set to 0000b when the CAN channel is in CH_RESET mode.

33.2.10 Global Configuration Register (GCFG)

Address(es): CANFD.GCFG 000A 8014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ITP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSCS	TSP[3:0]				—	—	OMRC	DLLCS	MME	DRE	DCE	TPRI
Value after 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	TPRI	Transmission Priority Setting	0: ID priority 1: Message buffer number priority	R/W
b1	DCE	DLC Check Enable	0: DLC check is disabled 1: DLC check is enabled	R/W
b2	DRE	DLC Replacement Enable	0: DLC replacement is disabled 1: DLC replacement is enabled	R/W
b3	MME	Mirror Mode Enable	0: Mirror mode is disabled 1: Mirror mode is enabled	R/W
b4	DLLCS	DLL Clock Select	0: CANFDCLK 1: CANFDMCLK	R/W
b5	OMRC	Payload-Overflowed Message Reception Configuration	0: The message is discarded. 1: The message payload is cut to fit the specified message size	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TSP[3:0]	Timestamp Counter Prescaler Setting	b11 b8 0 0 0 0: No division 0 0 0 1: Divide-by-2 (= 2^1) 0 0 1 0: Divide-by-4 (= 2^2) 0 0 1 1: Divide-by-8 (= 2^3) : : 1 1 0 1: Divide-by-8192 (= 2^{13}) 1 1 1 0: Divide-by-16384 (= 2^{14}) 1 1 1 1: Divide-by-32768 (= 2^{15})	R/W
b12	TSCS	Timestamp Counter Source Select	0: Count source for timestamp counter is peripheral module clock (PCLKB) 1: Count source for timestamp counter is bit time clock	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	ITP[15:0]	Interval Timer Prescaler Setting	FIFO interval timer prescaler value. Set the divided value for the peripheral module clock (PCLKB)	R/W

This register is used to select the transmission priority to be used for all the transmit message buffers and the clock source for the CAN protocol engine. The GCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI Bit (Transmission Priority Setting)

The TPRI bit selects the transmission priority for the CAN channel.

This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode. Do not select message buffer number priority when using the transmit queue.

DCE Bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

DRE Bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE bit is 1, the CANFD module stores the configured value (AFLn.PTR0.DLC[3:0]) of the DLC in the destination receive message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination receive message buffer or FIFO buffer is unchanged.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

MME Bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

DLLCS Bit (DLL Clock Select)

The DLLCS bit selects the clock source for CAN communication.This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only when CANFD module is in GL_RESET mode.

OMRC Bit (Payload-Overflowed Message Reception Configuration)

The OMRC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size (RMCR.PLS[2:0], RFCRn.PLS[2:0], and CFCR0.PLS[2:0]). The received message payload is always compared with the available message payload size in the message buffer.
This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only when CANFD module is in GL_RESET mode.
When this bit is set to 1 and payload overflow occurs, the DLC value is stored in the receive message buffer or FIFO buffer unchanged.

TSP[3:0] Bits (Timestamp Counter Prescaler Setting)

The value configured in the TSP[3:0] bits defines the period of the count source used for the timestamp counter.
These bits cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

TSCS Bit (Timestamp Counter Source Select)

The TSCS bit allows the selection of the count source for the timestamp counter.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.
Additionally, do not set this bit to 1 when CAN FD communication is used.

Note: The bit time clock varies depending on the nominal bit rate and data bit rate configuration.

ITP[15:0] Bits (Interval Timer Prescaler Setting)

The ITP[15:0] bits allow the definition of a reference clock for the FIFO interval timer count source. When these bits are 0000h, the timer is disabled.
These bits cannot be written in GL_SLEEP mode. Write to these bits only when CANFD module is in GL_RESET mode.

33.2.11 Global Control Register (GCR)

Address(es): CANFD.GCR 000A 8018h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	POIE	THLIE	MLIE	DEIE	—	—	—	—	—	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Global Mode Control	b1 b0 0 0: Request transition to GL_OPERATION mode 0 1: Request transition to GL_RESET mode 1 0: Request transition to GL_HALT mode 1 1: Keep current value	R/W
b2	SLPRQ	GL_SLEEP Mode Request	0: Request to release GL_SLEEP 1: Request transition to GL_SLEEP	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DEIE	DLC Error Interrupt Enable	0: DLC error interrupt is disabled 1: DLC error interrupt is enabled	R/W
b9	MLIE	Message Lost Interrupt Enable	0: Message lost interrupt is disabled 1: Message lost interrupt is enabled	R/W
b10	THLIE	Transmission History Entry Lost Interrupt Enable	0: Transmission history entry lost interrupt is disabled 1: Transmission history entry lost interrupt is enabled	R/W
b11	POIE	Payload Overflow Interrupt Enable	0: Payload overflow interrupt is disabled 1: Payload overflow interrupt is enabled	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TSCR	Timestamp Counter Reset	When 1 is written to this bit, the timestamp counter is reset. This bit is read as 0.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the Global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

MDC[1:0] Bits (Global Mode Control)

The MDC[1:0] bits are used to specify the modes of the CANFD module. For the mode transition of the CANFD module, refer to section 33.3.1, Global Modes.

These bits cannot be written in GL_SLEEP mode.

To request the CANFD module to transition to GL_SLEEP mode, first set these bits to 01b to enter GL_RESET mode, then set the GCR.SLPRQ bit to 1.

SLPRQ Bit (GL_SLEEP Mode Request)

The SLPRQ bit is used to control the transition to GL_SLEEP mode and the return from GL_SLEEP mode.

When this bit is set to 1, the transition to CH_SLEEP mode is also requested for CAN channel.

This bit can only be written when the CANFD module is in GL_RESET or GL_SLEEP mode.

DEIE Bit (DLC Error Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

This bit cannot be written in GL_SLEEP mode.

MLIE Bit (Message Lost Interrupt Enable)

When the MLIE bit is 1, an interrupt is generated if a message lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

THLIE Bit (Transmission History Entry Lost Interrupt Enable)

When the THLIE bit is 1, an interrupt is generated if a transmission history entry lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

POIE Bit (Payload Overflow Interrupt Enable)

When the POIE bit is 1, an interrupt is generated when a message payload overflow condition occurs.
This bit cannot be written in GL_SLEEP mode.

TSCR Bit (Timestamp Counter Reset)

When the TSCR bit is 1, the Timestamp Counter Register (TSCR) is reset to 00000000h.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit in GL_RESET mode.
This bit is automatically set to 0.

33.2.12 Global Status Register (GSR)

Address(es): CANFD.GSR 000A 801Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	RAMST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	GL_RESET Status Flag	0: Not in GL_RESET mode 1: In GL_RESET mode	R
b1	HLTST	GL_HALT Status Flag	0: Not in GL_HALT mode 1: In GL_HALT mode	R
b2	SLPST	GL_SLEEP Status Flag	0: Not in GL_SLEEP mode 1: In GL_SLEEP mode	R
b3	RAMST	RAM Initialization Status Flag	0: RAM initialization is completed 1: RAM initialization is in progress	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register indicates the global status of the CANFD module.

RSTST Flag (GL_RESET Status Flag)

The RSTST flag indicates the status of the CANFD module in GL_RESET mode.
This flag is automatically set to 1 when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains 1. This flag is automatically set to 0 when the CANFD module enters GL_HALT or GL_OPERATON mode.

HLTST Flag (GL_HALT Status Flag)

The HLTST flag indicates the status of the CANFD module in GL_HALT mode.
This flag is automatically set to 1 when the CANFD module enters GL_HALT mode. This flag is automatically set to 0 when the CANFD module exits the GL_HALT mode.

SLPST Flag (GL_SLEEP Status Flag)

The SLPST flag indicates the status of the CANFD module in GL_SLEEP mode.
This flag is automatically set to 1 when the CANFD module enters GL_SLEEP mode. This flag is automatically set to 0 when the CANFD module exits the GL_SLEEP mode.

RAMST Flag (RAM Initialization Status Flag)

The RAMST flag indicates the status of the CANFD module's RAM initialization.
This flag is automatically set to 1 when the CANFD module enters GL_SLEEP mode after release from MCU reset. This flag is automatically set to 0 when the CANFD module completed RAM initialization.

33.2.13 Global Error Status Register (GESR)

Address(es): CANFD.GESR 000A 8020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PODF	THLDF	MLDF	DEDF
Value after 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	DEDF	DLC Error Detect Flag	0: DLC error is not detected 1: DLC error is detected	R/(W) *1
b1	MLDF	Message Lost Detect Flag	0: Message lost error is not detected 1: Message lost error is detected	R
b2	THLDF	Transmission History Entry Lost Detect Flag	0: Transmission history entry lost is not detected 1: Transmission history entry lost is detected	R
b3	PODF	Payload Overflow Detect Flag	0: Payload overflow is not detected 1: Payload overflow is detected	R/(W) *1
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EEDF0	Channel 0 ECC Error Detect Flag	0: ECC error is not detected during transmission scan 1: ECC error is detected during transmission scan	R/(W) *1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the detection of global errors.

DEDF Flag (DLC Error Detect Flag)

The DEDF flag indicates the error status of the DLC.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 1 when a DLC error is detected in a received frame.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The flag is cleared by writing 0 to it. This flag is automatically set to 0 in GL_RESET mode.

MLDF Flag (Message Lost Detect Flag)

The MLDF flag indicates the status of the message lost error.
This flag is automatically set to 1 when a FIFO message lost error is detected.
This flag is automatically set to 0 when:

- All FIFO message lost flags (RFSRn.LOST, CFSR0.LOST) are cleared
- The CANFD module is in GL_RESET mode.

THLDF Flag (Transmission History Entry Lost Detect Flag)

The THLDF flag indicates the status of the transmission history entry lost error.
This flag is automatically set to 1 when a transmission history entry lost error is detected.
This flag is automatically set to 0 when:

- The transmission history lost flag (THSR.LOST) is cleared
- The CANFD module is in GL_RESET mode.

PODF Flag (Payload Overflow Detect Flag)

The PODF flag is automatically set to 1 when a message payload overflow is detected on at least one channel.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 0 in GL_RESET mode.

EEDF0 Flag (Channel 0 ECC Error Detect Flag)

The EEDF0 flag specifies whether an ECC error has occurred.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This bit is automatically set to 0 in GL_RESET mode.

33.2.14 Transmit Interrupt Status Register (TISR)

Address(es): CANFD.TISR 000A 80A4h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after 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	TSIF0	Channel 0 Transmission Successful Interrupt Flag	0: Channel 0 transmission successful interrupt is not generated. 1: Channel 0 transmission successful interrupt is generated.	R
b1	TAIF0	Channel 0 Transmission Abort Interrupt Flag	0: Channel 0 transmission abort interrupt is not generated. 1: Channel 0 transmission abort interrupt is generated.	R
b2	TQIF0	Channel 0 Transmit Queue Interrupt Flag	0: Channel 0 transmit queue interrupt is not generated. 1: Channel 0 transmit queue interrupt is generated.	R
b3	CFTIF0	Channel 0 Common FIFO Transmission Interrupt Flag	0: Channel 0 common FIFO transmission interrupt is not generated. 1: Channel 0 common FIFO transmission interrupt is generated.	R
b4	THIF0	Channel 0 Transmission History Interrupt Flag	0: Channel 0 transmission history interrupt is not generated. 1: Channel 0 transmission history interrupt is generated.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

This register indicates the detection of transmit specific interrupts.

TSIF0 Flag (Channel 0 Transmission Successful Interrupt Flag)

The TSIF0 flag becomes 1 if transmission from the transmit message buffer n on channel 0 is successful when the transmit message buffer n interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TMIER0.TMIEn bits = 0)
- When the Transmission Result flags (TMSRn.TXRF[1:0]) are cleared
- When in GL_RESET or CH_RESET mode.

TAIF0 Flag (Channel 0 Transmission Abort Interrupt Flag)

The TAIF0 flag becomes 1 if transmission from the transmit message buffer n on channel 0 is aborted when the transmission abort interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CHCR.TAIE bit = 0)
- When the Transmission Result flags (TMSRn.TXRF[1:0]) are cleared
- When in GL_RESET or CH_RESET mode.

TQIF0 Flag (Channel 0 Transmit Queue Interrupt Flag)

The TQIF0 flag becomes 1 if the Transmit Queue Interrupt flag of channel 0 is set to 1 when the transmit queue interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQIE bit = 0)
- When the Transmission Result flags (TQSR0.TQIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFTIF0 Flag (Channel 0 Common FIFO Transmission Interrupt Flag)

The CFTIF0 flag becomes 1 if the Common FIFO Transmit Interrupt flag (CFSR0.CFTIF) of channel 0 is set to 1 when the common FIFO transmit interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (CFCR0.CFTIE bit = 0)
- When the Common Transmit FIFO Interrupt flag (CFSR0.CFTIF) is cleared
- When in GL_RESET or CH_RESET mode.

THIF0 Flag (Channel 0 Transmission History Interrupt Flag)

The THIF0 flag becomes 1 if the Transmission History Interrupt flag (THSR.THIF) of channel 0 is set to 1 when the transmission history interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (THCR.THIE bit = 0)
- When the Transmission History Interrupt flag (THSR.THIF) is cleared
- When in GL_RESET or CH_RESET mode.

33.2.15 Timestamp Counter Register (TSCR)

Address(es): CANFD.TSCR 000A 8024h

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

This register stores the timestamp based on the selected configuration.
The timestamp value is stored in the TSCR register based on the configuration of the GCFG.TSCS bit and GCFG.TSP[3:0] bits. The accuracy of the timestamp counter cannot be guaranteed when transitioning to GL_HALT state.
This register is automatically set to 00000000h in GL_RESET mode.

33.2.16 Acceptance Filter List Control Register (AFCR)

Address(es): CANFD.AFCR 000A 8028h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	AFLWE	—	—	—	—	—	—	—	PAGE
Value after 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	PAGE	Access Page Setting	Select the page number of the Acceptance Filter List	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	AFLWE	AFL Write Enable	0: Acceptance Filter List data access is disabled 1: Acceptance Filter List data access is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to select the Acceptance Filter List page for reading or writing entries into the Acceptance Filter List.

PAGE Bit (Access Page Setting)

The PAGE bit is used to select the page number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.
Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.
This bit cannot be written in GL_SLEEP mode.

AFLWE Bit (AFL Write Enable)

The AFLWE bit prevents write access to the Acceptance Filter List by setting this bit to 0 after configuration of the Acceptance Filter List.
Data can be read from the Acceptance Filter List independent of the status of this bit.
This bit cannot be written in GL_SLEEP mode.
Setting this bit to 1 enables write access for the Acceptance Filter List.

33.2.17 Acceptance Filter List Configuration Register (AFCFG)

Address(es): CANFD.AFCFG 000A 802Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	RN0[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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after 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. The write value should be 0.	R/W
b21 to b16	RN0[5:0]	Channel 0 Number of Rules Setting	Number of rules in the Acceptance Filter List	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to define the number of rules for entry in the Acceptance Filter List.
The maximum number of available entries in the Acceptance Filter List is 32.

RN0[5:0] Bits (Channel 0 Number of Rules Setting)

The RN0[5:0] bits define the number of rules in the Acceptance Filter List. This bit can set 32 rules or less.
These bits can only be written in GL_RESET mode.

33.2.18 Acceptance Filter List n ID Register (AFLn.IDR) (n = 0 to 15)

Address(es): CANFD.AFL0.IDR 000A 8120h, CANFD.AFL1.IDR 000A 8130h, CANFD.AFL2.IDR 000A 8140h, CANFD.AFL3.IDR 000A 8150h, CANFD.AFL4.IDR 000A 8160h, CANFD.AFL5.IDR 000A 8170h, CANFD.AFL6.IDR 000A 8180h, CANFD.AFL7.IDR 000A 8190h, CANFD.AFL8.IDR 000A 81A0h, CANFD.AFL9.IDR 000A 81B0h, CANFD.AFL10.IDR 000A 81C0h, CANFD.AFL11.IDR 000A 81D0h, CANFD.AFL12.IDR 000A 81E0h, CANFD.AFL13.IDR 000A 81F0h, CANFD.AFL14.IDR 000A 8200h, CANFD.AFL15.IDR 000A 8210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	LPC	ID[28:16]												
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
	ID[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
b28 to b0	ID[28:0]	ID Field	ID part of the Acceptance Filter List entry	R/W
b29	LPC	Loopback Configuration	0: Message with reception attribute 1: Message with transmission attribute	R/W
b30	RTR	RTR	0: Data frame 1: Remote frame	R/W
b31	IDE	IDE	0: Standard ID 1: Extended ID	R/W

This register is used to configure the ID field in the rule entry of the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

ID[28:0] Bits (ID Field)

The ID[28:0] bits indicate the CAN identifier (ID) field of each entry in the Acceptance Filter List.
The acceptance filter process compares this field against the ID of a received message. For alignment of these bits in base and extended formats, refer to section 33.2.60, Identifier Bits Alignment.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

LPC Bit (Loopback Configuration)

The LPC bit is used to select whether entry in the Acceptance Filter List gets the reception or transmission attribute. This attribute determines the validity of the entry in Mirror mode, Loopback mode, and during standard (non-loopback) reception. Refer to section 33.5.8, Loopback Modes for detailed description of the validity of the Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and reception/transmission attribute.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

RTR Bit (RTR)

The RTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received message.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

IDE Bit (IDE)

The IDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the

received message.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

33.2.19 Acceptance Filter List n Mask Register (AFLn.MASK) (n = 0 to 15)

Address(es): CANFD.AFL0.MASK 000A 8124h, CANFD.AFL1.MASK 000A 8134h, CANFD.AFL2.MASK 000A 8144h, CANFD.AFL3.MASK 000A 8154h, CANFD.AFL4.MASK 000A 8164h, CANFD.AFL5.MASK 000A 8174h, CANFD.AFL6.MASK 000A 8184h, CANFD.AFL7.MASK 000A 8194h, CANFD.AFL8.MASK 000A 81A4h, CANFD.AFL9.MASK 000A 81B4h, CANFD.AFL10.MASK 000A 81C4h, CANFD.AFL11.MASK 000A 81D4h, CANFD.AFL12.MASK 000A 81E4h, CANFD.AFL13.MASK 000A 81F4h, CANFD.AFL14.MASK 000A 8204h, CANFD.AFL15.MASK 000A 8214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDEM	RTRM	IFL1	IDM[28:16]												
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
	IDM[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
b28 to b0	IDM[28:0]	ID Mask Field	0: Corresponding ID bit is not used for ID matching 1: Corresponding ID bit is used for ID matching	R/W
b29	IFL1	Information Label 1	Configure an information label 1 to be attached to a received message	R/W
b30	RTRM	RTR Mask	0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
b31	IDEM	IDE Mask	0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

This register is used to configure the mask field of each rule for entries in the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

IDM[28:0] Bits (ID Mask Field)

The IDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Acceptance Filter List entry.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

IFL1 Bit (Information Label 1)

The IFL1 bit is the upper bit of a 2-bit information label to be attached to a received message accepted by the associated entry in the Acceptance Filter List.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.
This bit is stored in the upper bit of the Information Label Field (RMBn.HF2.IFL[1], RFBn.HF2.IFL[1], CFB0.HF2.IFL[1]) of the storage location of a received message.

RTRM Bit (RTR Mask)

The RTRM bit is the RTR mask bit for each entry in the Acceptance Filter List.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

IDEM Bit (IDE Mask)

The IDEM bit is the IDE mask bit for each entry in the Acceptance Filter List.
When the IDE mask bit is 0, the ID comparison depends on the IDE bit of the received message.

- If the IDE bit of the received message is 0, the standard ID comparison takes place.
- If the IDE bit of the received message is 1, the extended ID comparison takes place.

Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

33.2.20 Acceptance Filter List n Pointer 0 Register (AFLn.PTR0) (n = 0 to 15)

Address(es): CANFD.AFL0.PTR0 000A 8128h, CANFD.AFL1.PTR0 000A 8138h, CANFD.AFL2.PTR0 000A 8148h, CANFD.AFL3.PTR0 000A 8158h, CANFD.AFL4.PTR0 000A 8168h, CANFD.AFL5.PTR0 000A 8178h, CANFD.AFL6.PTR0 000A 8188h, CANFD.AFL7.PTR0 000A 8198h, CANFD.AFL8.PTR0 000A 81A8h, CANFD.AFL9.PTR0 000A 81B8h, CANFD.AFL10.PTR0 000A 81C8h, CANFD.AFL11.PTR0 000A 81D8h, CANFD.AFL12.PTR0 000A 81E8h, CANFD.AFL13.PTR0 000A 81F8h, CANFD.AFL14.PTR0 000A 8208h, CANFD.AFL15.PTR0 000A 8218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMBE	—	—	DMB[4:0]					IFL0	—	—	—	DLC[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	DLC[3:0]	DLC Field	Configure a minimum DLC value of received message	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IFL0	Information Label 0	Configure an information label 0 to be attached to a received message	R/W
b12 to b8	DMB[4:0]	Destination Message Buffer Setting	Configure the receive message buffer number for storage of received messages	R/W
b14 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DMBE	Destination Message Buffer Setting Enable	0: The Destination Message Buffer Setting bit is disabled 1: The Destination Message Buffer Setting bit is enabled	R/W
b31 to b16	PTR[15:0]	Pointer	Configure a 16-bit pointer to be attached to a received message	R/W

This register is used to configure the data length code (DLC), software pointer, and destination message buffer for each rule entry in the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

DLC[3:0] Bits (DLC Field)

The DLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Acceptance Filter List (automatic DLC filter function).
DLC filter process is only passed if the DLC value of the message accepted by an entry in the Acceptance Filter List is equal to or higher than the DLC value configured for this associated Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0000b.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

IFL0 Bit (Information Label 0)

The IFL0 bit is the lower bit of a 2-bit information label that can be attached to a received message accepted by the Acceptance Filter List entry.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.
This bit is stored in the lower bit of the Information Label Field (RMBn.HF2.IFL[0], RFBn.HF2.IFL[0], CFB0.HF2.IFL[0]) of the storage location of a received message.

DMB[4:0] Bits (Destination Message Buffer Setting)

The DMB[4:0] bits allow the configuration of a message buffer as the destination for a received message that passes the acceptance check of the Acceptance Filter List entry. Configure the destination message buffer number.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.
Write the RMCR.NMB[5:0] bits to configure the number of receive message buffers. The value to be entered in the DMB[4:0] bits should only be between 00000b and 'NMB[5:0] – 1'.
If RMCR.NMB[5:0] is 000000b, set the DMB[4:0] bits to 00000b.

DMBE Bit (Destination Message Buffer Setting Enable)

The DMBE bit allows the enabling or disabling of the receive message buffer as the destination message buffer for a received message that passes the acceptance check of the Acceptance Filter List entry.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the Acceptance Filter List entry. The pointer is added during message storage in the message buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

33.2.21 Acceptance Filter List n Pointer 1 Register (AFLn.PTR1) (n = 0 to 15)

Address(es): CANFD.AFL0.PTR1 000A 812Ch, CANFD.AFL1.PTR1 000A 813Ch, CANFD.AFL2.PTR1 000A 814Ch, CANFD.AFL3.PTR1 000A 815Ch, CANFD.AFL4.PTR1 000A 816Ch, CANFD.AFL5.PTR1 000A 817Ch, CANFD.AFL6.PTR1 000A 818Ch, CANFD.AFL7.PTR1 000A 819Ch, CANFD.AFL8.PTR1 000A 81ACh, CANFD.AFL9.PTR1 000A 81BCh, CANFD.AFL10.PTR1 000A 81CCh, CANFD.AFL11.PTR1 000A 81DCh, CANFD.AFL12.PTR1 000A 81ECh, CANFD.AFL13.PTR1 000A 81FCh, CANFD.AFL14.PTR1 000A 820Ch, CANFD.AFL15.PTR1 000A 821Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CF0E	—	—	—	—	—	—	RF1E	RF0E
Value after 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	RF0E	Receive FIFO 0 Destination Enable	0: Do not select receive FIFO 0 as the message storage destination 1: Selects receive FIFO 0 as the message storage destination	R/W
b1	RF1E	Receive FIFO 1 Destination Enable	0: Do not select receive FIFO 1 as the message storage destination 1: Selects receive FIFO 1 as the message storage destination	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CF0E	Common FIFO 0 Destination Enable	0: Do not select common FIFO 0 as the message storage destination 1: Selects common FIFO 0 as the message storage destination	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the destination FIFO buffer for each rule entry in the Acceptance Filter List.
Up to two storage destinations can be specified for received messages. Two FIFO buffers, or one FIFO buffer and one receive message buffer are valid.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.
Write to this register only when the CAN channel is in CH_RESET or CH_HALT mode.

RF0E Bit (Receive FIFO 0 Destination Enable)

The RF0E bit allows the configuration of the receive FIFO 0 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

RF1E Bit (Receive FIFO 1 Destination Enable)

The RF1E bit allows the configuration of the receive FIFO 1 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

CF0E Bit (Common FIFO 0 Destination Enable)

The CF0E bit allows the configuration of the common FIFO 0 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.
The common FIFO 0 must be configured as the receive FIFO.

33.2.22 Receive Message Buffer Configuration Register (RMCR)

Address(es): CANFD.RMCR 000A 8030h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PLS[2:0]			—	—	NMB[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	NMB[5:0]	Number of Message Buffer Setting	Configure the number of receive message buffers	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	PLS[2:0]	Payload Size Setting	b10 b8 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the total number of receive message buffers allocated to a channel.

NMB[5:0] Bits (Number of Message Buffer Setting)

The NMB[5:0] bits are used to configure the number of receive message buffers.
These bits can only be written in GL_RESET mode.
Set a value between 0 and 32. 0 indicates that no receive message buffer is allocated.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits are used to configure the message buffer payload size.
These bits can only be written in GL_RESET mode.

33.2.23 Receive Message Buffer New Data Register (RMNDR)

Address(es): CANFD.RMNDR 000A 8034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NDF[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
	NDF[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	NDF[31:0]	New Data Flag	0: New data is not stored in corresponding receive message buffer 1: New data is stored in corresponding receive message buffer	R/(W) *1

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register specifies the new data storage status of the receive message buffers.
The bit number of this register corresponds to the buffer number of the received message buffer.

NDF[31:0] Flags (New Data Flag)

The NDF[31:0] flags indicate that the new data is stored in the corresponding receive message buffer. The NDF[0] flag corresponds to receive message buffer 0.
These flags are automatically set to 1 when storage of new messages are in the corresponding receive message buffer.
When RMCR.PLS[2:0] = 000b (up to 8 bytes payload), the duration of message storage is 6 PCLKB cycles.
When RMCR.PLS[2:0] > 000b, the duration of message storage is 6 PCLKB cycles + 1 for each 4 bytes (up to 20 PCLKB cycles for 64 bytes).
Do not write to these flags when the CANFD module is in GL_RESET or GL_SLEEP mode.
These flags are automatically set to 0 when the CANFD module is in GL_RESET mode.
These flags cannot be cleared when message storage in the corresponding receive message buffer is in progress.
Do not use the bit clear instruction to clear the flags. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

33.2.24 Receive FIFO n Configuration Register (RFCRn) (n = 0, 1)

Address(es): CANFD.RFCR0 000A 803Ch, CANFD.RFCR1 000A 8040h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFITH[2:0]			RFIM	—	FDS[2:0]			—	PLS[2:0]			—	—	RFIE	RFE
Value after 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	RFE	Receive FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	RFIE	Receive FIFO Interrupt Enable	0: FIFO interrupt generation is disabled 1: FIFO interrupt generation is enabled	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	FDS[2:0]	FIFO Depth Setting	b10 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages Settings other than above are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	RFIM	Receive FIFO Interrupt Mode Setting	0: Interrupt is generated when the number of messages stored in the receive FIFO reaches the value of the RFITH[2:0] bits from a value smaller than the RFITH[2:0] bits 1: Interrupt is generated at the end of every received message storage	R/W
b15 to b13	RFITH[2:0]	Receive FIFO Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt generated when FIFO is 1/8 full 0 0 1: Interrupt generated when FIFO is 1/4 full 0 1 0: Interrupt generated when FIFO is 3/8 full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8 full 1 0 1: Interrupt generated when FIFO is 3/4 full 1 1 0: Interrupt generated when FIFO is 7/8 full 1 1 1: Interrupt generated when FIFO is full	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

These registers are used to configure and control the two receive FIFOs.

RFE Bit (Receive FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the receive FIFO is cleared to empty.
This bit can only be written in GL_HALT or GL_OPERATION mode.
This bit can be set to 1 only when the FIFO depth is 4 to 48 (001b ≤ FDS[2:0] ≤ 101b).
Set the RFE bit to 1 with a separate write access to the RFCRn register, after all the other flags in the RFCRn register are set.
This bit is automatically set to 0 when the CANFD module is in GL_RESET mode.

RFIE Bit (Receive FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.
This bit cannot be written in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM.
This is the maximum number of bytes which can be received by this FIFO.
These bits can only be written in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits select the depth of the FIFO in units of the number of messages. If the FIFO depth is set to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

RFIM Bit (Receive FIFO Interrupt Mode Setting)

The RFIM bit selects the interrupt generation condition for the FIFO.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

RFITH[2:0] Bits (Receive FIFO Interrupt Threshold Setting)

The RFITH[2:0] bits select the counter value of the FIFO for generation of receive FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the RFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 33.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

33.2.25 Receive FIFO n Status Register (RFSRn) (n = 0, 1)

Address(es): CANFD.RFSR0 000A 8044h, CANFD.RFSR1 000A 8048h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	—	RFIF	LOST	FULL	EMPTY
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	EMPTY	Receive FIFO Empty Flag	0: Message in receive FIFO 1: No message in receive FIFO (empty)	R
b1	FULL	Receive FIFO Full Flag	0: Receive FIFO not full 1: Receive FIFO full	R
b2	LOST	Message Lost Flag	0: Receive FIFO message lost has not occurred 1: Receive FIFO message lost has occurred	R/(W) *1
b3	RFIF	Receive FIFO Interrupt Flag	0: Receive FIFO interrupt condition is not satisfied 1: Receive FIFO interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Receive FIFO Fill Level	Indicate the number of messages stored in receive FIFO	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

These registers show the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Receive FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The FLVL[5:0] flags are 00000b
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFD module is in GL_RESET mode.

The EMPTY flag is automatically set to 0 when the first message is stored in the receive FIFO.

FULL Flag (Receive FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO buffer matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO buffer is less than the configured FIFO depth
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFD module is in GL_RESET mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage when the FIFO buffer is already full. This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

The flag is set to 0:

- By writing 0 to it

- When the CANFD module is in GL_RESET mode.

Write to the LOST flag only when CANFD module is in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

RFIF Flag (Receive FIFO Interrupt Flag)

The RFIF flag is automatically set to 1 when the configured interrupt condition is satisfied. This flag is not automatically cleared when the receive FIFO is disabled.
The flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

Write to this flag only when the CANFD module is in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

FLVL[5:0] Bits (Receive FIFO Fill Level)

The FLVL[5:0] bits indicate the number of messages stored in the receive FIFO that can be read by the CPU. These bits are automatically set to 000000b when the FIFO is disabled and when the CANFD module is in GL_RESET mode.

33.2.26 Receive FIFO n Pointer Control Register (RFPCRn) (n = 0, 1)

Address(es): CANFD.RFPCR0 000A 804Ch, CANFD.RFPCR1 000A 8050h

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

This register can be used to increment the read pointer of the corresponding receive FIFOs.
When the value 000000FFh is written to this register, the pointer of the corresponding receive FIFO is moved to the next FIFO entry. Write to this register only when the corresponding receive FIFO is enabled and not empty.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
Do not write to this register when DTC/DMA transfer is enabled (DTCR.RFDTEn bit = 1).

33.2.27 Common FIFO 0 Configuration Register (CFCR0)

Address(es): CANFD.CFCR0 000A 8054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TINT[7:0]								FDS[2:0]			—	—	—	LTM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFITH[2:0]			CFIM	ITR	ITCS	—	MODE	—	PLS[2:0]			—	CFTIE	CFRIE	CFE
Value after 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	CFE	Common FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	CFRIE	Common FIFO Receive Interrupt Enable	0: FIFO receive interrupt generation is disabled 1: FIFO receive interrupt generation is enabled	R/W
b2	CFTIE	Common FIFO Transmit Interrupt Enable	0: FIFO transmit interrupt generation is disabled 1: FIFO transmit interrupt generation is enabled	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	MODE	Operation Mode Setting	0: Receive FIFO mode 1: Transmit FIFO mode	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	ITCS	Interval Timer Count Source Select	0: Reference clock (× 1 / × 10 period) 1: Bit time clock	R/W
b11	ITR	Interval Timer Resolution Select	0: Reference clock period × 1 1: Reference clock period × 10	R/W
b12	CFIM	Common FIFO Interrupt Mode Setting	Receive FIFO mode: 0: Reception interrupt generated when the number of messages stored in the common FIFO reaches CFITH[2:0] value from a lower value 1: Reception interrupt generated at the end of every received message storage Transmit FIFO mode: 0: Transmission interrupt generated when common FIFO transmits the last message successfully 1: Transmission interrupt generated for every successfully transmitted message	R/W
b15 to b13	CFITH[2:0]	Common FIFO Receive Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt is generated when FIFO is 1/8 full 0 0 1: Interrupt is generated when FIFO is 1/4 full 0 1 0: Interrupt is generated when FIFO is 3/8 full 0 1 1: Interrupt is generated when FIFO is 1/2 full 1 0 0: Interrupt is generated when FIFO is 5/8 full 1 0 1: Interrupt is generated when FIFO is 3/4 full 1 1 0: Interrupt is generated when FIFO is 7/8 full 1 1 1: Interrupt is generated when FIFO is full	R/W

Bit	Symbol	Bit Name	Description	R/W
b17 to b16	LTM[1:0]	Linked Transmit Message Buffer Select	Transmission scan link position of the corresponding channel	R/W
b20 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b21	FDS[2:0]	FIFO Depth Setting	b23 b21 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages Settings other than the above are prohibited.	R/W
b31 to b24	TINT[7:0]	Transmission Interval Setting	Delay the start of transmission from the FIFO if configured in transmit mode, delay is a multiple of basic interval timer clock source unit	R/W

CFE Bit (Common FIFO Enable)

Setting the CFE bit to 1 enables the FIFO. The FIFO is disabled when this bit is set to 0.
This bit can also be used, by clearing it, to abort transmission from common FIFO when configured in transmit FIFO mode, or to stop reception into the common FIFO in receive FIFO mode.
This bit can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
This bit can only be set to 1 when the FIFO depth is between 4 and 48 (001b ≤ FDS[2:0] ≤ 101b).
Set the CFE bit to 1 with a separate write access to the CFCR0 register, after setting all the other bits in this register.
This bit is automatically set to 0 when the CANFD module is in GL_RESET mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit is automatically set to 0 when the CAN channel is in CH_RESET mode.

CFRIE Bit (Common FIFO Receive Interrupt Enable)

The CFRIE bit enables or disables generation of common FIFO receive interrupt. If this flag is 1, common FIFO receive interrupt is generated when the common FIFO receive interrupt flag is set to 1 after reception of a frame in the corresponding FIFO buffer.
Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFTIE Bit (Common FIFO Transmit Interrupt Enable)

The CFTIE bit enables or disables generation of common FIFO transmit interrupt. If this flag is 1, common FIFO transmit interrupt is generated when the common FIFO transmit interrupt flag is set to 1 after transmission of a frame from the corresponding FIFO buffer.
Do not write to this bit when the CANFD module is in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.
For details, refer to section 33.6, FIFO Buffers and Message Buffer Configuration.
These bits can only be written in GL_RESET mode.

MODE Bit (Operation Mode Setting)

The MODE bit is used to select the operation mode of the common FIFO. When the MCU reset is applied, all the common FIFO are configured in receive FIFO mode.
This bit cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to this bit only when the CANFD module is in GL_RESET mode.

ITCS Bit (Interval Timer Count Source Select)

The ITCS bit is used to select the count source for the transmission interval timer.
Do not write to this bit when the CANFD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.
Do not write 1 to this bit when CAN FD communication is used.

Note: The bit time clock can vary depending on the nominal bit rate and data bit rate configuration.

ITR Bit (Interval Timer Resolution Select)

The ITR bit is used to select the resolution of the reference clock for the transmission interval timer.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit when the CFE bit is set to 1.

CFIM Bit (Common FIFO Interrupt Mode Setting)

The CFIM bit is used to select the condition for the generation of common FIFO interrupts.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only when the CANFD module is in GL_RESET mode.

CFITH[2:0] Bits (Common FIFO Receive Interrupt Threshold Setting)

The CFITH[2:0] bits are used to select the message counter value for the generation of common FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the CFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 33.6.2.1, FIFO Buffers Configuration.
Write to these bits only when the CANFD module is in GL_RESET mode.

LTM[1:0] Bits (Linked Transmit Message Buffer Select)

The LTM[1:0] bits are used to select the number of the transmit message buffer that links common FIFO configured in transmit FIFO mode, for transmission scanning.
These bits cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to these bits only when the CANFD module is in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits are used to select the FIFO depth in units of the number of messages. If the FIFO depth is configured to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

TINT[7:0] Bits (Transmission Interval Setting)

The TINT[7:0] bits are used to select the delay in the start of transmission for all messages transmitted from the common FIFO configured in transmit FIFO mode. The delay is an integral multiple of the count source cycle of the interval timer (reference clock period × 1, reference clock period × 10, or bit-time clock period).
These bits cannot be written in GL_SLEEP mode.
Do not write to these bits when the CFE bit is set to 1.
When the GCFG.ITP[15:0] bits are set to 0000h, set the TINT[7:0] bits to 00h.

33.2.28 Common FIFO 0 Status Register (CFSR0)

Address(es): CANFD.CFSR0 000A 8058h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	CFTIF	CFRIF	LOST	FULL	EMPTY
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	EMPTY	Common FIFO Empty Flag	0: Message in common FIFO 1: No message in common FIFO (empty)	R
b1	FULL	Common FIFO Full Flag	0: Common FIFO is not full 1: Common FIFO is full	R
b2	LOST	Message Lost Flag	0: No message lost in common FIFO 1: Common FIFO message lost	R/(W) *1
b3	CFRIF	Common FIFO Receive Interrupt Flag	0: Common FIFO receive interrupt condition is not satisfied 1: Common FIFO receive interrupt condition is satisfied	R/(W) *1
b4	CFTIF	Common FIFO Transmit Interrupt Flag	0: Common FIFO transmit interrupt condition is not satisfied 1: Common FIFO transmit interrupt condition is satisfied	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Common FIFO Fill Level	Indicate the number of messages stored in common FIFO	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Common FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The CPU has read all messages from the FIFO buffer in receive FIFO mode
- All messages have been transmitted from the FIFO buffer in transmit FIFO mode
- The common FIFO is disabled by setting the CFCR0.CFE bit to 0
- The CANFD module is in GL_RESET mode
- The CAN channel transits to CH_RESET mode in transmit FIFO mode.

The EMPTY flag is automatically set to 0 when:

- The first reception message is stored in the FIFO buffer in receive FIFO mode
- The first message to be transmitted is stored in the FIFO buffer in transmit FIFO mode.

FULL Flag (Common FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO is less than the configured FIFO depth
- The common FIFO is disabled by setting the CFCR0.CFE bit to 0
- The CANFD module is in GL_RESET mode
- The CAN channel transits to CH_RESET mode in transmit FIFO mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage of a new message when FIFO buffer is already full in receive FIFO mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
The LOST flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the CAN channel transits to CH_RESET mode in transmit FIFO mode.

CFRIF Flag (Common FIFO Receive Interrupt Flag)

The CFRIF flag is set to 1 when the configured interrupt condition is satisfied in receive FIFO mode.
The CFRIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFRIF flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

CFTIF Flag (Common FIFO Transmit Interrupt Flag)

The CFTIF flag is set to 1 when the configured interrupt condition is satisfied in transmit FIFO mode.
The CFTIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFTIF flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the CAN channel is in CH_RESET mode.

FLVL[5:0] Bits (Common FIFO Fill Level)

The FLVL[5:0] bits indicate the following:

- The number of messages stored by the CPU and waiting to be transmitted in the transmit FIFO mode

- The number of CPU-readable messages stored by CANFD in the receive FIFO mode.

The FLVL[5:0] bits are automatically set to 000000b when:

- The FIFO is disabled
- The CANFD module is in GL_RESET mode
- The CAN channel transits to CH_RESET mode in the transmit FIFO mode.

33.2.29 Common FIFO 0 Pointer Control Register (CFPCR0)

Address(es): CANFD.CFPCR0 000A 805Ch

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

This register can be used to increment the read or write pointer of the corresponding common FIFO.
When the value 000000FFh is written into this register, the read pointer of the corresponding common FIFO in receive FIFO mode, or the write pointer of the corresponding common FIFO in transmit FIFO mode moves to the next FIFO entry.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
Write to this register only when:

- The common FIFO is enabled and is not empty in receive FIFO mode
- The common FIFO is enabled and is not full in transmit FIFO mode.

Do not write to the CFPCR0 register when DTC/DMA transfer is enabled (DTCR.CFDTE0 bit = 1).

33.2.30 FIFO Empty Status Register (FESR)

Address(es): CANFD.FESR 000A 8060h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFEMP0	—	—	—	—	—	—	RFEMP1	RFEMP0
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	RFEMP0	Receive FIFO 0 Empty Flag	0: Message in corresponding receive FIFO 0 1: No message in corresponding receive FIFO 0 (empty)	R
b1	RFEMP1	Receive FIFO 1 Empty Flag	0: Message in corresponding receive FIFO 1 1: No message in corresponding receive FIFO 1 (empty)	R
b7 to b2	—	Reserved	These bits are read as 0	R
b8	CFEMP0	Common FIFO 0 Empty Flag	0: Message in corresponding common FIFO 0 1: No message in corresponding common FIFO 0 (empty)	R
b31 to b9	—	Reserved	These bits are read as 0	R

This register shows the status of the empty flags of the FIFO buffers.

RFEMP0 Flag (Receive FIFO 0 Empty Flag)

The RFEMP0 flag is set to 1 when the RFSR0.EMPTY flag is set to 1, and is set to 0 when the RFSR0.EMPTY flag is set to 0.

The RFEMP0 flag is set to 1 when the CANFD module is in GL_RESET mode.

RFEMP1 Flag (Receive FIFO 1 Empty Flag)

The RFEMP1 flag is set to 1 when the RFSR1.EMPTY flag is set to 1, and is set to 0 when the RFSR1.EMPTY flag is set to 0.

The RFEMP1 flag is set to 1 when the CANFD module is in GL_RESET mode.

CFEMP0 Flag (Common FIFO 0 Empty Flag)

The CFEMP0 flag is set to 1 when the CFSR0.EMPTY flag is set to 1, and is set to 0 when the CFSR0.EMPTY flag is set to 0.

The CFEMP0 flag is set to 1 when the CANFD module is in GL_RESET mode.

33.2.31 FIFO Full Status Register (FFSR)

Address(es): CANFD.FFSR 000A 8064h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFFUL0	—	—	—	—	—	—	RFFUL1	RFFUL0
Value after 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	RFFUL0	Receive FIFO 0 Full Flag	0: Corresponding receive FIFO 0 is not full 1: Corresponding receive FIFO 0 is full	R
b1	RFFUL1	Receive FIFO 1 Full Flag	0: Corresponding receive FIFO 1 is not full 1: Corresponding receive FIFO 1 is full	R
b7 to b2	—	Reserved	These bits are read as 0	R
b8	CFFUL0	Common FIFO 0 Full Flag	0: Common FIFO 0 is not full 1: Common FIFO 0 is full	R
b31 to b9	—	Reserved	These bits are read as 0	R

This register shows the status of the full flags of the FIFO buffers.

RFFUL0 Flag (Receive FIFO 0 Full Flag)

The RFFUL0 flag is set to 1 when the RFSR0.FULL flag is set to 1, and is set to 0 when the RFSR0.FULL flag is set to 0.
The RFFUL0 flag is set to 0 when CANFD module is in GL_RESET mode.

RFFUL1 Flag (Receive FIFO 1 Full Flag)

The RFFUL1 flag is set to 1 when the RFSR1.FULL flag is set to 1, and is set to 0 when the RFSR1.FULL flag is set to 0.
The RFFUL1 flag is set to 0 when CANFD module is in GL_RESET mode.

CFFUL0 Flag (Common FIFO 0 Full Flag)

The CFFUL0 flag is set to 1 when the CFSR0.FULL flag is set to 1 is set to 0 when the CFSR0.FULL flag is set to 0.
The CFFUL0 flag is set to 0 when the CANFD module is in GL_RESET mode.

33.2.32 FIFO Message Lost Status Register (FMLSR)

Address(es): CANFD.FMLSR 000A 8068h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFML0	—	—	—	—	—	—	RFML1	RFML0
Value after 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	RFML0	Receive FIFO 0 Message Lost Flag	0: Message lost has not occurred in the receive FIFO 0 1: Message lost has occurred in the receive FIFO 0	R
b1	RFML1	Receive FIFO 1 Message Lost Flag	0: Message lost has not occurred in the receive FIFO 1 1: Message lost has occurred in the receive FIFO 1	R
b7 to b2	—	Reserved	These bits are read as 0	R
b8	CFML0	Common FIFO 0 Message Lost Flag	0: Message lost has not occurred in the common FIFO 0 1: Message lost has occurred in the common FIFO 0	R
b31 to b9	—	Reserved	These bits are read as 0	R

This register shows the status of the message lost flags of the FIFO buffers.

RFML0 Flag (Receive FIFO 0 Message Lost Flag)

The RFML0 flag is set to 1 when the RFSR0.LOST flag is set to 1, and is set to 0 when the RFSR0.LOST flag is set to 0. The RFML0 flag is cleared when the CANFD module is in GL_RESET mode.

RFML1 Flag (Receive FIFO 1 Message Lost Flag)

The RFML1 flag is set to 1 when the RFSR1.LOST flag is set to 1, and is set to 0 when the RFSR1.LOST flag is set to 0. The RFML1 flag is cleared when the CANFD module is in GL_RESET mode.

CFML0 Flag (Common FIFO 0 Message Lost Flag)

The CFML0 flag is set to 1 when the CFSR0.LOST flag is set to 1, and is set to 0 when the CFSR0.LOST flag is set to 0. The CFML0 flag is cleared when the CANFD module is in GL_RESET mode.

33.2.33 Receive FIFO Interrupt Status Register (RFISR)

Address(es): CANFD.RFISR 000A 806Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFIF1	RFIF0
Value after 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	RFIF0	Receive FIFO0 Interrupt Flag	0: Receive FIFO 0 interrupt condition is not satisfied 1: Receive FIFO 0 interrupt condition is satisfied	R
b1	RFIF1	Receive FIFO1 Interrupt Flag	0: Receive FIFO 1 interrupt condition is not satisfied 1: Receive FIFO 1 interrupt condition is satisfied	R
b31 to b2	—	Reserved	These bits are read as 0	R

This register shows the status of the interrupt flags of the receive FIFOs.

RFIF0 Flag (Receive FIFO0 Interrupt Flag)

The RFIF0 flag is set to 1 when the RFSR0.RFIF flag is set to 1, and is set to 0 when the RFSR0.RFIF flag is set to 0. The RFIF0 flag is cleared when the CANFD module is in GL_RESET mode.

RFIF1 Flag (Receive FIFO1 Interrupt Flag)

The RFIF1 flag is set to 1 when the RFSR1.RFIF flag is set to 1, and is set to 0 when the RFSR1.RFIF flag is set to 0. The RFIF1 flag is cleared when the CANFD module is in GL_RESET mode.

33.2.34 DMA Transfer Control Register (DTCR)

Address(es): CANFD.DTCR 000A 80C8h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFDTE0	—	—	—	—	—	—	RFDTE1	RFDTE0
Value after 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	RFDTE0	Receive FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b1	RFDTE1	Receive FIFO 1 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CFDTE0	Common FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the start and stop of DTC/DMA transfer operation.

RFDTE0 Bit (Receive FIFO 0 DMA Transfer Enable)

The RFDTE0 bit enables or disables DTC/DMA transfer request for receive FIFO 0.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFD module is in GL_RESET mode.

RFDTE1 Bit (Receive FIFO 1 DMA Transfer Enable)

The RFDTE1 bit enables or disables DTC/DMA transfer request for receive FIFO 1.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFD module is in GL_RESET mode.

CFDTE0 Bit (Common FIFO 0 DMA Transfer Enable)

The CFDTE0 bit enables or disables DTC/DMA transfer request for common FIFO.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
Do not enable DTC/DMA transfer for the common FIFO that is configured as transmit FIFO.
This bit is set to 0 when the CANFD module is in GL_RESET mode.

33.2.35 DMA Transfer Status Register (DTSR)

Address(es): CANFD.DTSR 000A 80CCh

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFDTS0	—	—	—	—	—	—	RFDTS1	RFDTS0
Value after 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	RFDTS0	Receive FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b1	RFDTS1	Receive FIFO 1 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	CFDTS0	Common FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b31 to b9	—	Reserved	These bits are read as 0.	R

This register shows the status of the DTC/DMA transfer.

RFDTS0 Flag (Receive FIFO 0 DMA Transfer Status Flag)

The RFDTS0 flag is automatically set to 1 when the DTCR.RFDTE0 bit is set to 1 and the receive FIFO 0 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTE0 bit is set to 0 or the receive FIFO 0 is empty.

When the DTCR.RFDTE0 bit is set to 0 while DTC/DMA transfer for the receive FIFO 0 is in progress, the RFDTS0 flag becomes 0 when the DTC/DMA transfer is completed.

This flag is set to 0 when the CANFD module is in GL_RESET mode.

RFDTS1 Flag (Receive FIFO 1 DMA Transfer Status Flag)

The RFDTS1 flag is automatically set to 1 when the DTCR.RFDTE1 bit is set to 1 and the receive FIFO 1 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTE1 bit is set to 0 or the receive FIFO 1 is empty.

When the DTCR.RFDTE1 bit is set to 0 while DTC/DMA transfer for the receive FIFO 1 is in progress, the RFDTS1 flag becomes 0 when the DTC/DMA transfer is completed.

This flag is set to 0 when the CANFD module is in GL_RESET mode.

CFDTS0 Flag (Common FIFO 0 DMA Transfer Status Flag)

The CFDTS0 flag is automatically set to 1 when the DTCR.CFDTE0 bit is set to 1 and the common FIFO 0 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.CFDTE0 bit is set to 0 or the common FIFO 0 is empty.

When the DTCR.CFDTE0 bit is set to 0 while DTC/DMA transfer for the common FIFO 0 is in progress, the CFDTS0 flag becomes 0 when the DTC/DMA transfer is complete.

This flag is set to 0 when the CANFD module is in GL_RESET mode.

33.2.36 Transmit Message Buffer n Control Register (TMCRn) (n = 0 to 3)

Address(es): CANFD.TMCR0 000A 8070h, CANFD.TMCR1 000A 8071h, CANFD.TMCR2 000A 8072h, CANFD.TMCR3 000A 8073h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ONESHOT	TARQ	TXRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQ	Transmission Request	0: Message transmission is not requested 1: Message transmission is requested	R/W
b1	TARQ	Transmission Abort Request	0: Message transmission abort is not requested 1: Message transmission abort is requested	R/W
b2	ONESHOT	One-shot Transmission Enable	0: Transmission in one-shot mode is disabled 1: Transmission in one-shot mode is enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmit message buffer functions.

TXRQ Bit (Transmission Request)

When the TXRQ bit is set to 1, the CANFD module logic requests transmitting the message stored in the corresponding message buffer.
This bit can only be written in CH_HALT or CH_OPERATION mode.
This bit cannot be set to 1 if the corresponding transmit message buffer is linked to the common FIFO configured in transmit FIFO or is a part of the transmit queue.
This bit can be set to 1 only when the TMSRn.TXRF[1:0] flags are set to 00b.
This bit cannot be directly cleared by a CPU write access. The TXRQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort, requested by the TARQ bit
- When a CAN bus error or arbitration lost is detected if the ONESHOT bit is set to 1
- When the CANFD module is in GL_RESET or the CAN channel is in CH_RESET mode.

TARQ Bit (Transmission Abort Request)

When the TARQ bit is set to 1, the CANFD module logic requests aborting the transmission of the frame stored in the corresponding message buffer.
In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame will be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.
However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (pin for reception) before it starts transmission from the selected message buffer.
Write to the TARQ bit only when the CAN channel is in CH_HALT or CH_OPERATION mode. This bit can be set to 1 only when the TXRQ bit is set to 1.
This bit cannot be set to 0 by a CPU write access. This bit is set to 0 if writing 1 by CPU and clearing by the CAN channel occur at the same time.
The TARQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort
- When a CAN bus error or arbitration lost is detected

- When the CANFD module is in GL_RESET or the CAN channel enters CH_RESET mode.

ONESHOT Bit (One-shot Transmission Enable)

When the ONESHOT bit is set to 1, the CANFD module logic request transmitting the message only once.
If the transmission is successful, the TMSRn.TXRF[1:0] flags are set to 10b or 11b. Otherwise, the transmission is automatically aborted and TMSRn.TXRF[1:0] flags are set to 01b due to a bus error or an arbitration lost detection.
The ONESHOT bit remains 1 if the transmission has completed successfully or aborted due to a CAN bus error or an arbitration lost detection.
Write to this bit only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Set this bit at the same time as the TXRQ bit. Clear this bit with a write access.
If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.
The ONESHOT bit is automatically set to 0 when the CANFD module is in GL_RESET or the CAN channel is in CH_RESET mode.

33.2.37 Transmit Message Buffer n Status Register (TMSRn) (n = 0 to 3)

Address(es): CANFD.TMSR0 000A 8074h, CANFD.TMSR1 000A 8075h, CANFD.TMSR2 000A 8076h, CANFD.TMSR3 000A 8077h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TARQS	TXRQS	TXRF[1:0]		TXSF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXSF	Transmission Status Flag	0: No transmission is in progress 1: Transmission is in progress	R
b2, b1	TXRF[1:0]	Transmission Result Flag	b2 b1 0 0: No result (transmission is not requested or in progress) 0 1: Transmission was aborted. 1 0: Transmission was successful and transmission abort is not requested. 1 1: Transmission was successful and transmission abort is requested.	R/W
b3	TXRQS	Transmission Request Status Flag	0: Transmission is not requested 1: Transmission is requested	R
b4	TARQS	Transmission Abort Request Status Flag	0: Transmission abort is not requested 1: Transmission abort is requested	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register shows the status of the transmission and transmission abort for the transmit message buffers.

TXSF Flag (Transmission Status Flag)

The TXSF flag is automatically set to 1 at the start of the transmission from the corresponding transmit message buffer. This flag is automatically set to 0 when:

- Transmission stops
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

TXRF[1:0] Flags (Transmission Result Flag)

The TXRF[1:0] flags show the transmission result for the transmit message buffer.
Write to these flags only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Each bit of these flags is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The TXRF[1:0] flags are automatically set to 00b when the CANFD module is in GL_RESET or the CAN channel is in CH_RESET mode.

TXRQS Flag (Transmission Request Status Flag)

The TXRQS flag reflects the value of the TMCRn.TXRQ bit.
The TXRQS flag is set to 1 when the TMCRn.TXRQ bit is set to 1. This flag is set to 0 when the TMCRn.TXRQ bit is set to 0.

TARQS Flag (Transmission Abort Request Status Flag)

The TARQS flag reflects the value of the TMCRn.TARQ bit.
The TARQS flag is set to 1 when the TMCRn.TARQ bit is set to 1. This flag is set to 0 when the TMCRn.TARQ bit is set to 0.

33.2.38 Transmit Message Buffer Transmission Request Status Register 0 (TMTRSR0)

Address(es): CANFD.TMTRSR0 000A 8078h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TXRQS3	TXRQS2	TXRQS1	TXRQS0
Value after 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	TXRQS0	Transmit Message Buffer 0 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 0. 1: Transmission is requested for the transmit message buffer 0	R
b1	TXRQS1	Transmit Message Buffer 1 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 1. 1: Transmission is requested for the transmit message buffer 1.	R
b2	TXRQS2	Transmit Message Buffer 2 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 2. 1: Transmission is requested for the transmit message buffer 2.	R
b3	TXRQS3	Transmit Message Buffer 3 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 3. 1: Transmission is requested for the transmit message buffer 3.	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the status of the transmission request in each transmit message buffer.

TXRQSn Flag (Transmit Message Buffer n Transmission Request Status Flag) (n = 0 to 3)

The TXRQSn flag indicates status of the TMCRn.TXRQ bit.

Each flag is set to 1 only when the TMCRn.TXRQ bit is set to 1 and the message buffer does not belong to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TXRQ bit is set to 0
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

33.2.39 Transmit Message Buffer Transmission Abort Request Status Register 0 (TMARSR0)

Address(es): CANFD.TMARSR0 000A 807Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TARQS3	TARQS2	TARQS1	TARQS0
Value after 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	TARQS0	Transmit Message Buffer 0 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 0 1: Transmission abort is requested for the transmit message buffer 0	R
b1	TARQS1	Transmit Message Buffer 1 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 1 1: Transmission abort is requested for the transmit message buffer 1	R
b2	TARQS2	Transmit Message Buffer 2 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 2 1: Transmission abort is requested for the transmit message buffer 2	R
b3	TARQS3	Transmit Message Buffer 3 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 3 1: Transmission abort is requested for the transmit message buffer 3	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the status of the transmission abort request in each transmit message buffer.

TARQSn Flag (Transmit Message Buffer n Transmission Abort Request Status Flag) (n = 0 to 3)

The TARQSn flag indicates status of the TMCRn.TARQ bit.

Each flag is set to 1 when the TMCRn.TARQ bit is set to 1 or the message buffer belongs to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TARQ bit is set to 0
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

33.2.40 Transmit Message Buffer Transmission Completion Status Register 0 (TMTCSR0)

Address(es): CANFD.TMTCSR0 000A 8080h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TXCF3	TXCF2	TXCF1	TXCF0
Value after 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	TXCF0	Transmit Message Buffer 0 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 0 1: Transmission is completed for the transmit message buffer 0	R
b1	TXCF1	Transmit Message Buffer 1 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 1 1: Transmission is completed for the transmit message buffer 1	R
b2	TXCF2	Transmit Message Buffer 2 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 2 1: Transmission is completed for the transmit message buffer 2	R
b3	TXCF3	Transmit Message Buffer 3 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 3 1: Transmission is completed for the transmit message buffer 3	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the transmission completion status for each transmit message buffer.

TXCFn Flag (Transmit Message Buffer n Transmission Completion Status) (n = 0 to 3)

The TXCFn flag indicates the transmission completion status for the transmit message buffer n.

Each flag is automatically set to 1 when the TMSRn.TXRF[1] bit is set to 1.

Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1] bit is set to 0
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

33.2.41 Transmit Message Buffer Transmission Abort Status Register 0 (TMTASR0)

Address(es): CANFD.TMTASR0 000A 8084h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TAF3	TAF2	TAF1	TAF0
Value after 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	TAF0	Transmit Message Buffer 0 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 0. 1: Transmission is aborted for the transmit message buffer 0.	R
b1	TAF1	Transmit Message Buffer 1 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 1. 1: Transmission is aborted for the transmit message buffer 1.	R
b2	TAF2	Transmit Message Buffer 2 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 2. 1: Transmission is aborted for the transmit message buffer 2.	R
b3	TAF3	Transmit Message Buffer 3 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 3. 1: Transmission is aborted for the transmit message buffer 3.	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the transmission abort status for each transmit message buffer.

TAFn Flag (Transmit Message Buffer n Transmission Abort Status Flag) (n = 0 to 3)

The TAFn flag indicates the transmission abort status for the transmit message buffer n.

Each flag is automatically set to 1 when the TMSRn.TXRF[1:0] flags are set to 01b.

Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1:0] flags are set to 00b
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

33.2.42 Transmit Message Buffer Interrupt Enable Register (TMIER0)

Address(es): CANFD.TMIER0 000A 8088h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMIE3	TMIE2	TMIE1	TMIE0
Value after 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	TMIE0	Transmit Message Buffer 0 Interrupt Enable	0: Interrupt for transmit message buffer 0 is disabled. 1: Interrupt for transmit message buffer 0 is enabled.	R/W
b1	TMIE1	Transmit Message Buffer 1 Interrupt Enable	0: Interrupt for transmit message buffer 1 is disabled. 1: Interrupt for transmit message buffer 1 is enabled.	R/W
b2	TMIE2	Transmit Message Buffer 2 Interrupt Enable	0: Interrupt for transmit message buffer 2 is disabled. 1: Interrupt for transmit message buffer 2 is enabled.	R/W
b3	TMIE3	Transmit Message Buffer 3 Interrupt Enable	0: Interrupt for transmit message buffer 3 is disabled. 1: Interrupt for transmit message buffer 3 is enabled.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the interrupt for each transmit message buffer.

TMIEn Bit (Transmit Message Buffer n Interrupt Enable) (n = 0 to 3)

If the TMIEn bit is set to 1, an interrupt is generated when transmission from the transmit message buffer n is completed successfully.

Refer to section 33.10, Interrupts and DTC/DMA Requests for the specification of the interrupt for transmit message buffer.

This bit cannot be written when the CANFD module is in GL_SLEEP mode.

Do not write to the TMIEn bit when:

- The CAN channel is in CH_SLEEP mode
- The transmit message buffer n is part of a transmit queue
- The transmit message buffer n is linked to a common FIFO by the CFCR0.LTM[1:0] bits.

33.2.43 Transmit Queue 0 Configuration Register (TQCR0)

Address(es): CANFD0.TQCR0 000A 808Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	QDS[1:0]		TQIM	—	TQIE	—	—	—	—	TQE
Value after 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	TQE	Transmit Queue Enable	0: Transmit queue is disabled. 1: Transmit queue is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	TQIE	Transmit Queue Interrupt Enable	0: Transmit queue transmission interrupt is disabled. 1: Transmit queue transmission interrupt is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TQIM	Transmit Queue Interrupt Mode Setting	0: When the last message is successfully transmitted 1: At every successful transmission	R/W
b9, b8	QDS[1:0]	Queue Depth Setting	b9 b8 0 0: 0 messages (disabled) 0 1: Setting prohibited 1 0: 3 messages (transmit message buffer 0 to 2) 1 1: 4 messages (transmit message buffer 0 to 3)	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the transmit queue 0.

TQE Bit (Transmit Queue Enable)

The TQE bit enables the transmit queue. When the TQE bit is set to 1, the transmit message buffer is used to configure the transmit queue.

The TQE bit cannot be set to 1 if the configured transmit queue depth is 0 (QDS[1:0] = 00b).

This bit cannot be written in GL_SLEEP mode.

Also, this bit cannot be written in CH_RESET or CH_SLEEP mode.

The TQE bit is automatically set to 0 when the CAN channel is in CH_RESET mode.

TQIE Bit (Transmit Queue Interrupt Enable)

When the TQIE bit is set to 1, an interrupt is generated based on the setting of the TQIM bit.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQIM Bit (Transmit Queue Interrupt Mode Setting)

The TQIM bit selects the interrupt generation condition for the transmit queue.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP, CH_HALT, or CH_OPERATION mode.

QDS[1:0] Bits (Queue Depth Setting)

The QDS[1:0] bits select the depth of the transmission queue.

When these bits are set to 10b, the message buffers 0 to 2 are used, and when these bits are set to 11b, the message buffers 0 to 3 are used.
These bits cannot be written in GL_SLEEP mode.
Also, these bits cannot be written in CH_HALT or CH_OPERATION mode.
Do not write to these bits when the CAN channel is in CH_SLEEP mode.

33.2.44 Transmit Queue 0 Status Register (TQSR0)

Address(es): CANFD0.TQSR0 000A 8090h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FLVL[2:0]			—	—	—	—	—	TQIF	FULL	EMPTY
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	EMPTY	Transmit Queue Empty Flag	0: There is at least one message in transmit queue 1: No message in transmit queue (empty)	R
b1	FULL	Transmit Queue Full Flag	0: Transmit queue is not full 1: Transmit queue is full	R
b2	TQIF	Transmit Queue Interrupt Flag	0: Transmit queue interrupt condition is not satisfied 1: Transmit queue interrupt condition is satisfied	R/(W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	FLVL[2:0]	Transmit Queue Fill Level	Indicates the number of messages in the transmit queue	R
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of the transmit queue 0.

EMPTY Flag (Transmit Queue Empty Flag)

This flag is automatically set to 1 when:

- The TQCR0.TQE bit is set to 0 (transmit queue is disabled)
- No messages are stored in the transmit queue
- The last message is transmitted from the transmit queue
- The CAN channel is in CH_RESET mode.

This flag is automatically set to 0 when the first message to be transmitted is stored in the transmit queue.

FULL Flag (Transmit Queue Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the transmit queue matches the configured transmit queue depth.

This flag is automatically set to 0 when:

- The number of messages stored in the transmit queue is less than the configured transmit queue depth
- The CAN channel is in CH_RESET mode.

TQIF Flag (Transmit Queue Interrupt Flag)

The TQIF flag is not cleared automatically if the transmit queue is disabled.

When stopping the transmit queue, this flag should be cleared, after setting the TQCR0.TQE bit to 0 and checking an empty state of transmit queue.

Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

This flag is automatically set to 1 when the configured interrupt condition is satisfied for the transmit queue.

This flag is set to 0:

- By writing 0 to it
- When the CAN channel is in CH_RESET mode.

This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag cannot be written when the CAN channel is in CH_SLEEP or CH_RESET mode.

FLVL[2:0] Bits (Transmit Queue Fill Level)

The FLVL[2:0] bits indicates the number of messages in the transmit queue.
These bits are automatically set to 000b when the CAN channel is in CH_RESET mode.

33.2.45 Transmit Queue 0 Pointer Control Register (TQPCR0)

Address(es): CANFD0.TQPCR0 000A 8094h

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

This register is used to increment the write pointer to the transmit queue.
When the value 000000FFh is written to this register, the write pointer to the transmit queue is updated and transmission of the existing message is requested.
The read value from this register is always 00000000h.
This register cannot be written when the CAN channel is in CH_SLEEP or CH_RESET mode.
Write to this register only when:

- The transmit queue is enabled and not full.

33.2.46 Transmission History Configuration Register (THCR)

Address(es): CANFD0.THCR 000A 8098h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THRC	THIM	THIE	—	—	—	—	—	—	—	THE
Value after 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	THE	Transmission History Enable	0: Transmission history buffer is disabled 1: Transmission history buffer is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	THIE	Transmission History Interrupt Enable	0: Transmission history interrupt is disabled 1: Transmission history interrupt is enabled	R/W
b9	THIM	Transmission History Interrupt Mode Setting	0: Interrupt is generated if transmission history level reaches 3/4 of the transmission history depth 1: Interrupt is generated for every successfully stored entry	R/W
b10	THRC	Transmission History Recording Condition Setting	0: Transmit FIFO + Transmit queue 1: Transmit message buffer + Transmit FIFO + Transmit queue	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmission history functions.

THE Bit (Transmission History Enable)

The THE bit enables the transmission history buffer when it is set to 1.
This bit cannot be written when the CAN channel is in CH_RESET or CH_SLEEP mode.
This bit is automatically set to 0 when the CAN channel is in CH_RESET mode.

THIE Bit (Transmission History Interrupt Enable)

The THIE bit enables the generation of the transmission history interrupt when it is set to 1.
This bit cannot be written when the CANFD module is in GL_SLEEP mode.

THIM Bit (Transmission History Interrupt Mode Setting)

The THIM bit selects the transmission history interrupt generation condition.
This bit cannot be written when the CANFD module is in GL_SLEEP mode.
Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

THRC Bit (Transmission History Recording Condition Setting)

The THRC bit selects the condition for storing an entry in the transmission history buffer after successful transmission.
This bit cannot be written when the CANFD module is in GL_SLEEP mode.
Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

33.2.47 Transmission History Status Register (THSR)

Address(es): CANFD0.THSR 000A 809Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FLVL[3:0]				—	—	—	—	THIF	LOST	FULL	EMPTY
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	EMPTY	Transmission History Empty Flag	0: There is a transmission history in the transmission history buffer 1: There is no transmission history in the transmission history buffer (empty)	R
b1	FULL	Transmission History Full Flag	0: Transmission history buffer is not full 1: Transmission history buffer is full	R
b2	LOST	Transmission History Lost Flag	0: Transmission history has not been lost 1: Transmission history has been lost	R/(W) *1
b3	THIF	Transmission History Interrupt Flag	0: Transmission history interrupt condition is not satisfied 1: Transmission history interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	FLVL[3:0]	Transmission History Fill Level	Number of transmission histories stored in transmission history buffer	R
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of data stored in the transmission history buffer.

EMPTY Flag (Transmission History Empty Flag)

This flag is automatically set to 0 when the first transmission history is stored to the transmission history buffer.
This flag is automatically set to 1 when:

- The CPU has read all the transmission histories from the transmission history buffer
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel is in CH_RESET mode.

FULL Flag (Transmission History Full Flag)

The FULL flag is automatically set to 1 when the number of transmission histories in the transmission history buffer is eight.
This flag is automatically set to 0 when:

- The number of transmission histories in the transmission history buffer is less than eight
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel is in CH_RESET mode.

LOST Flag (Transmission History Lost Flag)

The LOST flag is set to 1 when a new transmission history cannot be stored because the related transmission history buffer is already full.

Write to this flag only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel is in CH_RESET mode.

THIF Flag (Transmission History Interrupt Flag)

The THIF flag is set to 1 when the configured interrupt condition is satisfied.
Write to this flag only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel is in CH_RESET mode.

FLVL[3:0] Bits (Transmission History Fill Level)

The FLVL[3:0] bits show the number of transmission histories stored in the transmission history buffer.
These bits are automatically set to 0000b when the CAN channel is in CH_RESET mode.

33.2.48 Transmission History Access Register 0 (THACR0)

Address(es): CANFD0.THACR0 000A 8740h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BN[1:0]		BT[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	BT[2:0]	Transmitted Buffer Type	b2 b0 0 0 1: Transmit message buffer 0 1 0: Common FIFO 1 0 0: Transmit Queue	R
b4, b3	BN[1:0]	Transmitted Buffer Number	Number of the message buffer	R
b15 to b5	—	Reserved	These bits are read as 0	R
b31 to b16	TS[15:0]	Transmitted Timestamp	Transmit timestamp value	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

BT[2:0] Bits (Transmitted Buffer Type)

The BT[2:0] bits indicate which type of buffer the read history came from.

BN[1:0] Bits (Transmitted Buffer Number)

The BN[1:0] bits indicate which buffer number the read history came from.
For a common FIFO, these bits indicate the number of the linked transmit message buffer.

TS[15:0] Bits (Transmitted Timestamp)

The TS[15:0] bits indicate the timestamp used by the software driver.

33.2.49 Transmission History Access Register 1 (THACR1)

Address(es): CANFD0.THACR1 000A 8744h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTR[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	PTR[15:0]	Transmitted Pointer	Stores the value of the PTR[15:0] field attached to the transmit message	R
b17, b16	IFL[1:0]	Transmitted Information Label	Stores the value of the IFL[1:0] field attached to the transmit message	R
b31 to b18	—	Reserved	These bits are read as 0	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

PTR[15:0] Bits (Transmitted Pointer)

The PTR[15:0] bits store the value of the pointer field (TMBn.HF2.PTR[15:0] bits or CFB0.HF2.PTR[15:0] bits) attached to the transmit message.

IFL[1:0] Bits (Transmitted Information Label)

The IFL[1:0] bits store the value of the information label field (TMBn.HF2.IFL[1:0] bits or CFB0.HF2.IFL[1:0] bits) attached to the transmit message.

33.2.50 Transmission History Pointer Control Register (THPCR)

Address(es): CANFD0.THPCR 000A 80A0h

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

This register is used to increment the read pointer of the transmission history buffer.
When 000000FFh is written to this register, the read pointer of the transmission history buffer is moved to the next history.
The read value from this register is 00000000h.
This register can only be written in CH_HALT or CH_OPERATION mode.
Write 000000FFh to this register only when the transmission history buffer is enabled and not empty.

33.2.51 Global Reset Control Register (GRCR)

Address(es): CANFD.GRCR 000A 80D8h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after 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	SRST	Software Reset	0: Releases the software reset 1: Initiates the software reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control rewriting of the SRST bit. These bits are read as 00h.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SRST Bit (Software Reset)

When the SRST bit is set to 1, the CANFD module is in the same state as the MCU reset. When a reset is required, write 1 to this bit and then 0.

After releasing the software reset, the CANFD module is in GL_SLEEP mode.

After a software reset, the RAM initialization sequence is not performed. Initialize the RAM with software.

Similarly, if a software reset is performed during RAM initialization, the RAM is not initialized. Initialize the RAM with software.

KEY[7:0] Bits (Key Code)

When rewriting the value of the SRST bit, set these bits to C4h at the same time (write this register in 32 bits).

33.2.52 Global Test Mode Configuration Register (GTMCR)

Address(es): CANFD.GTMCR 000A 80A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	RTPS[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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after 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. The write value should be 0.	R/W
b19 to b16	RTPS[3:0]	RAM Test Page Select	Select the page of RAM to be tested	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to specify the page number of the RAM to be tested in RAM test mode.

RTPS[3:0] Bits (RAM Test Page Select)

The RTPS[3:0] bits select the RAM page number that the CPU reads and writes when the CANFD module is in RAM test mode. Specify the page number in the range of 0 to 9.

Refer to section 33.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

These bits cannot be written in GL_RESET or GL_SLEEP mode. Write to these bits only in GL_HALT mode.

These bits are automatically set to 0000b when the CAN channel is in GL_RESET mode.

33.2.53 Global Test Mode Enable Register (GTMER)

Address(es): CANFD.GTMER 000A 80ACh

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	—
Value after 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	RTME	RAM Test Mode Enable	0: RAM Test mode is disabled. 1: RAM Test mode is enabled.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the global test modes of the CANFD module.

RTME Bit (RAM Test Mode Enable)

When the RTME bit is set to 1, the CANFD module is in RAM test mode. Refer to section 33.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

This bit can only be set to 1 when the CANFD module is in GL_HALT mode. To exit the RAM test mode, set this bit to 0 in GL_HALT mode.

This bit is automatically set to 0 when the CANFD module is in GL_RESET mode.

33.2.54 Global CAN FD Configuration Register (GFDCFG)

Address(es): CANFD.GFDCFG 000A 80B0h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TSCPS[1:0]		—	—	—	—	—	—	—	PXEDIS
Value after 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	PXEDIS	Protocol Exception Event Detection Disable	0: Protocol exception event detection is enabled. 1: Protocol exception event detection is disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	TSCPS[1:0]	Timestamp Capturing Point Select	b9 b8 0 0: Sample point of SOF (start of frame) 0 1: EOF (end of frame) when frame is taken valid 1 0: Sample point of SOF (Classical CAN frame), or sample point of res bit following the FDF bit (CAN FD frame) 1 1: Setting prohibited	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PXEDIS Bit (Protocol Exception Event Detection Disable)

The PXEDIS bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is set to 1, protocol exception event detection is disabled, and an error frame is transmitted if a protocol exception (a recessive reserved bit following the FDF bit) is detected.

This bit can only be written in GL_RESET mode.

TSCPS[1:0] Bits (Timestamp Capturing Point Select)

The TSCPS[1:0] bits selects the capture point of the timestamps for transmitted and received messages.

When the TSCPS[1:0] bits are 10b, the timestamp is captured at the reserved bit following the FDF bit in CAN FD frames and at the SOF in Classical CAN frames.

These bits can only be written in GL_RESET mode.

33.2.55 Global Test Mode Lock Key Register (GTMLKR)

Address(es): CANFD.GTMLKR 000A 80B8h

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

This register is used to unlock the protection for RAM test mode. Refer to section 33.9.2, Global Test Modes for the specification of the lock key.
To put the CANFD module into RAM test mode, two unlock keys must be written to this register in consecutive bus cycles.
The read value from this register is always 00000000h.
This register cannot be written when the CANFD module is in GL_SLEEP or GL_RESET mode.
Do not write to this register when the CANFD module is in GL_OPERATION mode.

33.2.56 RAM Test Page Access Register k (RTPARk) (k = 0 to 63)

Address(es): CANFD.RTPAR0 000A 8280h to CANFD.RTPAR63 000A 837Ch

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

This register can be read or written when the CANFD module is in RAM test mode.
This register can only be written in GL_HALT mode when RAM test mode is enabled.

33.2.57 Acceptance Filter List Ignore Entry Setting Register (AFIGSR)

Address(es): CANFD.AFIGSR 000A 80C0h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IGES[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	IGES[4:0]	Ignore Entry Select	Set the rule number to be ignored during acceptance filtering.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGES[4:0] Bits (Ignore Entry Select)

The IGES[4:0] bits are used to specify the rule number to update when updating the acceptance filter.

Write to these bits only when the AFIGER.IGEE bit is 0.
These bits cannot be written in GL_SLEEP mode.

33.2.58 Acceptance Filter List Ignore Entry Enable Register (AFIGER)

Address(es): CANFD.AFIGER 000A 80C4h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	IGEE
Value after 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	IGEE	Ignore Entry Enable	0: The value of the AFIGSR.IGES[4:0] bits is invalid. 1: The value of the AFIGSR.IGES[4:0] bits is valid.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Controls the validity of rewriting the IGEE bit	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGEE Bit (Ignore Entry Enable)

When the IGEE bit is set to 1, the entry selected by the AFIGSR.IGES[4:0] bits is ignored.

This bit is automatically set to 0 when the CANFD module is in GL_RESET mode.

KEY[7:0] Bits (Key Code)

Writing to the IGEE bit is enabled when C4h is written to the KEY[7:0] bits. The value read from these bits is always 00h.

Write the IGEE bit and the KEY [7:0] bits at the same time.

33.2.59 Receive Message Buffer Interrupt Enable Register (RMIER)

Address(es): CANFD.RMIER 000A 8038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RMIE31	RMIE30	RMIE29	RMIE28	RMIE27	RMIE26	RMIE25	RMIE24	RMIE23	RMIE22	RMIE21	RMIE20	RMIE19	RMIE18	RMIE17	RMIE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMIE15	RMIE14	RMIE13	RMIE12	RMIE11	RMIE10	RMIE9	RMIE8	RMIE7	RMIE6	RMIE5	RMIE4	RMIE3	RMIE2	RMIE1	RMIE0
Value after 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	RMIE0	Receive Message Buffer 0 Interrupt Enable	0: Interrupt for receive message buffer 0 is disabled. 1: Interrupt for receive message buffer 0 is enabled.	R/W
b1	RMIE1	Receive Message Buffer 1 Interrupt Enable	0: Interrupt for receive message buffer 1 is disabled. 1: Interrupt for receive message buffer 1 is enabled.	R/W
b2	RMIE2	Receive Message Buffer 2 Interrupt Enable	0: Interrupt for receive message buffer 2 is disabled. 1: Interrupt for receive message buffer 2 is enabled.	R/W
b3	RMIE3	Receive Message Buffer 3 Interrupt Enable	0: Interrupt for receive message buffer 3 is disabled. 1: Interrupt for receive message buffer 3 is enabled.	R/W
b4	RMIE4	Receive Message Buffer 4 Interrupt Enable	0: Interrupt for receive message buffer 4 is disabled. 1: Interrupt for receive message buffer 4 is enabled.	R/W
b5	RMIE5	Receive Message Buffer 5 Interrupt Enable	0: Interrupt for receive message buffer 5 is disabled. 1: Interrupt for receive message buffer 5 is enabled.	R/W
b6	RMIE6	Receive Message Buffer 6 Interrupt Enable	0: Interrupt for receive message buffer 6 is disabled. 1: Interrupt for receive message buffer 6 is enabled.	R/W
b7	RMIE7	Receive Message Buffer 7 Interrupt Enable	0: Interrupt for receive message buffer 7 is disabled. 1: Interrupt for receive message buffer 7 is enabled.	R/W
b8	RMIE8	Receive Message Buffer 8 Interrupt Enable	0: Interrupt for receive message buffer 8 is disabled. 1: Interrupt for receive message buffer 8 is enabled.	R/W
b9	RMIE9	Receive Message Buffer 9 Interrupt Enable	0: Interrupt for receive message buffer 9 is disabled. 1: Interrupt for receive message buffer 9 is enabled.	R/W
b10	RMIE10	Receive Message Buffer 10 Interrupt Enable	0: Interrupt for receive message buffer 10 is disabled. 1: Interrupt for receive message buffer 10 is enabled.	R/W
b11	RMIE11	Receive Message Buffer 11 Interrupt Enable	0: Interrupt for receive message buffer 11 is disabled. 1: Interrupt for receive message buffer 11 is enabled.	R/W
b12	RMIE12	Receive Message Buffer 12 Interrupt Enable	0: Interrupt for receive message buffer 12 is disabled. 1: Interrupt for receive message buffer 12 is enabled.	R/W
b13	RMIE13	Receive Message Buffer 13 Interrupt Enable	0: Interrupt for receive message buffer 13 is disabled. 1: Interrupt for receive message buffer 13 is enabled.	R/W
b14	RMIE14	Receive Message Buffer 14 Interrupt Enable	0: Interrupt for receive message buffer 14 is disabled. 1: Interrupt for receive message buffer 14 is enabled.	R/W
b15	RMIE15	Receive Message Buffer 15 Interrupt Enable	0: Interrupt for receive message buffer 15 is disabled. 1: Interrupt for receive message buffer 15 is enabled.	R/W
b16	RMIE16	Receive Message Buffer 16 Interrupt Enable	0: Interrupt for receive message buffer 16 is disabled. 1: Interrupt for receive message buffer 16 is enabled.	R/W
b17	RMIE17	Receive Message Buffer 17 Interrupt Enable	0: Interrupt for receive message buffer 17 is disabled. 1: Interrupt for receive message buffer 17 is enabled.	R/W
b18	RMIE18	Receive Message Buffer 18 Interrupt Enable	0: Interrupt for receive message buffer 18 is disabled. 1: Interrupt for receive message buffer 18 is enabled.	R/W
b19	RMIE19	Receive Message Buffer 19 Interrupt Enable	0: Interrupt for receive message buffer 19 is disabled. 1: Interrupt for receive message buffer 19 is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	RMIE20	Receive Message Buffer 20 Interrupt Enable	0: Interrupt for receive message buffer 20 is disabled. 1: Interrupt for receive message buffer 20 is enabled.	R/W
b21	RMIE21	Receive Message Buffer 21 Interrupt Enable	0: Interrupt for receive message buffer 21 is disabled. 1: Interrupt for receive message buffer 21 is enabled.	R/W
b22	RMIE22	Receive Message Buffer 22 Interrupt Enable	0: Interrupt for receive message buffer 22 is disabled. 1: Interrupt for receive message buffer 22 is enabled.	R/W
b23	RMIE23	Receive Message Buffer 23 Interrupt Enable	0: Interrupt for receive message buffer 23 is disabled. 1: Interrupt for receive message buffer 23 is enabled.	R/W
b24	RMIE24	Receive Message Buffer 24 Interrupt Enable	0: Interrupt for receive message buffer 24 is disabled. 1: Interrupt for receive message buffer 24 is enabled.	R/W
b25	RMIE25	Receive Message Buffer 25 Interrupt Enable	0: Interrupt for receive message buffer 25 is disabled. 1: Interrupt for receive message buffer 25 is enabled.	R/W
b26	RMIE26	Receive Message Buffer 26 Interrupt Enable	0: Interrupt for receive message buffer 26 is disabled. 1: Interrupt for receive message buffer 26 is enabled.	R/W
b27	RMIE27	Receive Message Buffer 27 Interrupt Enable	0: Interrupt for receive message buffer 27 is disabled. 1: Interrupt for receive message buffer 27 is enabled.	R/W
b28	RMIE28	Receive Message Buffer 28 Interrupt Enable	0: Interrupt for receive message buffer 28 is disabled. 1: Interrupt for receive message buffer 28 is enabled.	R/W
b29	RMIE29	Receive Message Buffer 29 Interrupt Enable	0: Interrupt for receive message buffer 29 is disabled. 1: Interrupt for receive message buffer 29 is enabled.	R/W
b30	RMIE30	Receive Message Buffer 30 Interrupt Enable	0: Interrupt for receive message buffer 30 is disabled. 1: Interrupt for receive message buffer 30 is enabled.	R/W
b31	RMIE31	Receive Message Buffer 31 Interrupt Enable	0: Interrupt for receive message buffer 31 is disabled. 1: Interrupt for receive message buffer 31 is enabled.	R/W

This register enables or disables the interrupt for each receive message buffer.

RMIEn Bit (Receive Message Buffer n Interrupt Enable) (n = 0 to 31)

If the RMIEn bit is set to 1, an interrupt is generated when reception to the receive message buffer n is completed successfully.

Refer to section 33.10, Interrupts and DTC/DMA Requests for the specification of the receive message buffer interrupt.

This bit cannot be written when the CANFD module is in GL_SLEEP mode.

33.2.60 Identifier Bits Alignment

Base format (11-bit length identifier): ID-28 to ID-18 are located in b10 to b0, and b28 to b11 are 0.
Extended format (29-bit length identifier): ID-28 to ID-0 are located in b28 to b0

Table 33.4 Standard Identifier (11-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18

Table 33.5 Extended Identifier (29-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1	RTR	—	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18	ID-17	ID-16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0

33.2.61 Message Buffer Structure

The message buffer configuration consists of four types of message buffer:

- Receive message buffer (RMBn)
- Receive FIFO (RFBn)
- Common FIFO (CFB0)
- Transmit message buffer (TMBn)

n is a message buffer number whose range changes depending on the type of message buffer.
Refer to Figure 33.33 for an overview of this configuration. For more information on the number and types of message buffers, refer to section 33.6, FIFO Buffers and Message Buffer Configuration.

33.2.61.1 Start Addresses

The start address of each message buffer is calculated using the number of related message buffer components.
The start addresses for each register in the message buffer are listed in Table 33.6.

Table 33.6 Start Address for Each Register of the Message Buffer Component

Message Buffer	Symbol	n	Register	p	Start Address
Receive Message Buffer	RMBn	0 to 7	HF0	—	000A 8920h + n × 4Ch
			HF1	—	000A 8924h + n × 4Ch
			HF2	—	000A 8928h + n × 4Ch
			DFp	0 to 15	000A 892Ch + n × 4Ch + p × 4
	RMBn	8 to 15	HF0	—	000A 8D20h + (n – 8) × 4Ch
			HF1	—	000A 8D24h + (n – 8) × 4Ch
			HF2	—	000A 8D28h + (n – 8) × 4Ch
			DFp	0 to 15	000A 8D2Ch + (n – 8) × 4Ch + p × 4
	RMBn	16 to 23	HF0	—	000A 9120h + (n – 16) × 4Ch
			HF1	—	000A 9124h + (n – 16) × 4Ch
			HF2	—	000A 9128h + (n – 16) × 4Ch
			DFp	0 to 15	000A 912Ch + (n – 16) × 4Ch + p × 4
	RMBn	24 to 31	HF0	—	000A 9520h + (n – 24) × 4Ch
			HF1	—	000A 9524h + (n – 24) × 4Ch
			HF2	—	000A 9528h + (n – 24) × 4Ch
			DFp	0 to 15	000A 952Ch + (n – 24) × 4Ch + p × 4
Receive FIFO	RFBn	0, 1	HF0	—	000A 8520h + n × 4Ch
			HF1	—	000A 8524h + n × 4Ch
			HF2	—	000A 8528h + n × 4Ch
			DFp	0 to 15	000A 852Ch + n × 4Ch + p × 4
Common FIFO	CFB0	0	HF0	—	000A 85B8h
			HF1	—	000A 85BCh
			HF2	—	000A 85C0h
			DFp	0 to 15	000A 85C4h + p × 4
Transmit FIFO Buffer	TMBn	0 to 3	HF0	—	000A 8604h + n × 4Ch
			HF1	—	000A 8608h + n × 4Ch
			HF2	—	000A 860Ch + n × 4Ch
			DFp	0 to 15	000A 8610h + n × 4Ch + p × 4

33.2.61.2 Receive Message Buffer n (RMBn) (n = 0 to 31)

The total number of the receive message buffer (RMB) is 32, as shown in Figure 33.33.
The receive message buffer consists of the following registers:

- RMBn.HF0
- RMBn.HF1
- RMBn.HF2
- RMBn.DF0 to RMBn.DF15

Table 33.7 shows the structure of this buffer.

Table 33.7 Structure of Receive Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	RMBn.HF0	Receive Message Buffer n Header Field 0	IDE, RTR, ID
+04h	RMBn.HF1	Receive Message Buffer n Header Field 1	DLC, timestamp
+08h	RMBn.HF2	Receive Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RMBn.DF0	Receive Message Buffer n Data Field 0	DATA0 to DATA3
+10h	RMBn.DF1	Receive Message Buffer n Data Field 1	DATA4 to DATA7
+14h	RMBn.DF2	Receive Message Buffer n Data Field 2	DATA8 to DATA11
+18h	RMBn.DF3	Receive Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	RMBn.DF4	Receive Message Buffer n Data Field 4	DATA16 to DATA19
+20h	RMBn.DF5	Receive Message Buffer n Data Field 5	DATA20 to DATA23
+24h	RMBn.DF6	Receive Message Buffer n Data Field 6	DATA24 to DATA27
+28h	RMBn.DF7	Receive Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	RMBn.DF8	Receive Message Buffer n Data Field 8	DATA32 to DATA35
+30h	RMBn.DF9	Receive Message Buffer n Data Field 9	DATA36 to DATA39
+34h	RMBn.DF10	Receive Message Buffer n Data Field 10	DATA40 to DATA43
+38h	RMBn.DF11	Receive Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	RMBn.DF12	Receive Message Buffer n Data Field 12	DATA48 to DATA51
+40h	RMBn.DF13	Receive Message Buffer n Data Field 13	DATA52 to DATA55
+44h	RMBn.DF14	Receive Message Buffer n Data Field 14	DATA56 to DATA59
+48h	RMBn.DF15	Receive Message Buffer n Data Field 15	DATA60 to DATA63

33.2.61.3 Receive Message Buffer n Header Field 0 (RMBn.HF0) (n = 0 to 31)

Address(es): CANFD.RMB0.HF0 000A 8920h, CANFD.RMB1.HF0 000A 896Ch, CANFD.RMB2.HF0 000A 89B8h, CANFD.RMB3.HF0 000A 8A04h, CANFD.RMB4.HF0 000A 8A50h, CANFD.RMB5.HF0 000A 8A9Ch, CANFD.RMB6.HF0 000A 8AE8h, CANFD.RMB7.HF0 000A 8B34h, CANFD.RMB8.HF0 000A 8D20h, CANFD.RMB9.HF0 000A 8D6Ch, CANFD.RMB10.HF0 000A 8DB8h, CANFD.RMB11.HF0 000A 8E04h, CANFD.RMB12.HF0 000A 8E50h, CANFD.RMB13.HF0 000A 8E9Ch, CANFD.RMB14.HF0 000A 8EE8h, CANFD.RMB15.HF0 000A 8F34h, CANFD.RMB16.HF0 000A 9120h, CANFD.RMB17.HF0 000A 916Ch, CANFD.RMB18.HF0 000A 91B8h, CANFD.RMB19.HF0 000A 9204h, CANFD.RMB20.HF0 000A 9250h, CANFD.RMB21.HF0 000A 929Ch, CANFD.RMB22.HF0 000A 92E8h, CANFD.RMB23.HF0 000A 9334h, CANFD.RMB24.HF0 000A 9520h, CANFD.RMB25.HF0 000A 956Ch, CANFD.RMB26.HF0 000A 95B8h, CANFD.RMB27.HF0 000A 9604h, CANFD.RMB28.HF0 000A 9650h, CANFD.RMB29.HF0 000A 969Ch, CANFD.RMB30.HF0 000A 96E8h, CANFD.RMB31.HF0 000A 9734h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
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
	ID[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
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive message buffer.
For alignment of these bits in base and extended format, refer to section 33.2.60, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.
The RTR bit indicates whether the receive message buffer stores a data frame or a remote frame.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.
The IDE bit indicates whether the message stored in the receive message buffer has a standard identifier or extended identifier.

33.2.61.4 Receive Message Buffer n Header Field 1 (RMBn.HF1) (n = 0 to 31)

Address(es): CANFD.RMB0.HF1 000A 8924h, CANFD.RMB1.HF1 000A 8970h, CANFD.RMB2.HF1 000A 89BCh, CANFD.RMB3.HF1 000A 8A08h, CANFD.RMB4.HF1 000A 8A54h, CANFD.RMB5.HF1 000A 8AA0h, CANFD.RMB6.HF1 000A 8AECh, CANFD.RMB7.HF1 000A 8B38h, CANFD.RMB8.HF1 000A 8D24h, CANFD.RMB9.HF1 000A 8D70h, CANFD.RMB10.HF1 000A 8DBCh, CANFD.RMB11.HF1 000A 8E08h, CANFD.RMB12.HF1 000A 8E54h, CANFD.RMB13.HF1 000A 8EA0h, CANFD.RMB14.HF1 000A 8EECh, CANFD.RMB15.HF1 000A 8F38h, CANFD.RMB16.HF1 000A 9124h, CANFD.RMB17.HF1 000A 9170h, CANFD.RMB18.HF1 000A 91BCh, CANFD.RMB19.HF1 000A 9208h, CANFD.RMB20.HF1 000A 9254h, CANFD.RMB21.HF1 000A 92A0h, CANFD.RMB22.HF1 000A 92ECh, CANFD.RMB23.HF1 000A 9338h, CANFD.RMB24.HF1 000A 9524h, CANFD.RMB25.HF1 000A 9570h, CANFD.RMB26.HF1 000A 95BCh, CANFD.RMB27.HF1 000A 9608h, CANFD.RMB28.HF1 000A 9654h, CANFD.RMB29.HF1 000A 96A0h, CANFD.RMB30.HF1 000A 96ECh, CANFD.RMB31.HF1 000A 9738h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[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
	TS[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	TS[15:0]	Timestamp	Timestamp value stored for the message in the receive message buffer	R
b27 to b16	—	Reserved	These bits are read as 0	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame.	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RMCR.PLS[2:0] bits.

33.2.61.5 Receive Message Buffer n Header Field 2 (RMBn.HF2) (n = 0 to 31)

Address(es): CANFD.RMB0.HF2 000A 8928h, CANFD.RMB1.HF2 000A 8974h, CANFD.RMB2.HF2 000A 89C0h, CANFD.RMB3.HF2 000A 8A0Ch, CANFD.RMB4.HF2 000A 8A58h, CANFD.RMB5.HF2 000A 8AA4h, CANFD.RMB6.HF2 000A 8AF0h, CANFD.RMB7.HF2 000A 8B3Ch, CANFD.RMB8.HF2 000A 8D28h, CANFD.RMB9.HF2 000A 8D74h, CANFD.RMB10.HF2 000A 8DC0h, CANFD.RMB11.HF2 000A 8E0Ch, CANFD.RMB12.HF2 000A 8E58h, CANFD.RMB13.HF2 000A 8EA4h, CANFD.RMB14.HF2 000A 8EF0h, CANFD.RMB15.HF2 000A 8F3Ch, CANFD.RMB16.HF2 000A 9128h, CANFD.RMB17.HF2 000A 9174h, CANFD.RMB18.HF2 000A 91C0h, CANFD.RMB19.HF2 000A 920Ch, CANFD.RMB20.HF2 000A 9258h, CANFD.RMB21.HF2 000A 92A4h, CANFD.RMB22.HF2 000A 92F0h, CANFD.RMB23.HF2 000A 933Ch, CANFD.RMB24.HF2 000A 9528h, CANFD.RMB25.HF2 000A 9574h, CANFD.RMB26.HF2 000A 95C0h, CANFD.RMB27.HF2 000A 960Ch, CANFD.RMB28.HF2 000A 9658h, CANFD.RMB29.HF2 000A 96A4h, CANFD.RMB30.HF2 000A 96F0h, CANFD.RMB31.HF2 000A 973Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after 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	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0	R
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0	R
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

33.2.61.6 Receive Message Buffer n Data Field p (RMBn.DFp) (n = 0 to 31; p = 0 to 15)

Address(es): CANFD.RMB0.DF0 000A 892Ch to CANFD.RMB0.DF15 000A 8968h,
CANFD.RMB1.DF0 000A 8978h to CANFD.RMB1.DF15 000A 89B4h,
CANFD.RMB2.DF0 000A 89C4h to CANFD.RMB2.DF15 000A 8A00h,
CANFD.RMB3.DF0 000A 8A10h to CANFD.RMB3.DF15 000A 8A4Ch,
CANFD.RMB4.DF0 000A 8A5Ch to CANFD.RMB4.DF15 000A 8A98h,
CANFD.RMB5.DF0 000A 8AA8h to CANFD.RMB5.DF15 000A 8AE4h,
CANFD.RMB6.DF0 000A 8AF4h to CANFD.RMB6.DF15 000A 8B30h,
CANFD.RMB7.DF0 000A 8B40h to CANFD.RMB7.DF15 000A 8B7Ch,
CANFD.RMB8.DF0 000A 8D2Ch to CANFD.RMB8.DF15 000A 8D68h,
CANFD.RMB9.DF0 000A 8D78h to CANFD.RMB9.DF15 000A 8DB4h,
CANFD.RMB10.DF0 000A 8DC4h to CANFD.RMB10.DF15 000A 8E00h,
CANFD.RMB11.DF0 000A 8E10h to CANFD.RMB11.DF15 000A 8E4Ch,
CANFD.RMB12.DF0 000A 8E5Ch to CANFD.RMB12.DF15 000A 8E98h,
CANFD.RMB13.DF0 000A 8EA8h to CANFD.RMB13.DF15 000A 8EE4h,
CANFD.RMB14.DF0 000A 8EF4h to CANFD.RMB14.DF15 000A 8F30h,
CANFD.RMB15.DF0 000A 8F40h to CANFD.RMB15.DF15 000A 8F7Ch,
CANFD.RMB16.DF0 000A 912Ch to CANFD.RMB16.DF15 000A 9168h,
CANFD.RMB17.DF0 000A 9178h to CANFD.RMB17.DF15 000A 91B4h,
CANFD.RMB18.DF0 000A 91C4h to CANFD.RMB18.DF15 000A 9200h,
CANFD.RMB19.DF0 000A 9210h to CANFD.RMB19.DF15 000A 924Ch,
CANFD.RMB20.DF0 000A 925Ch to CANFD.RMB20.DF15 000A 9298h,
CANFD.RMB21.DF0 000A 92A8h to CANFD.RMB21.DF15 000A 92E4h,
CANFD.RMB22.DF0 000A 92F4h to CANFD.RMB22.DF15 000A 9330h,
CANFD.RMB23.DF0 000A 9340h to CANFD.RMB23.DF15 000A 937Ch,
CANFD.RMB24.DF0 000A 952Ch to CANFD.RMB24.DF15 000A 9568h,
CANFD.RMB25.DF0 000A 9578h to CANFD.RMB25.DF15 000A 95B4h,
CANFD.RMB26.DF0 000A 95C4h to CANFD.RMB26.DF15 000A 9600h,
CANFD.RMB27.DF0 000A 9610h to CANFD.RMB27.DF15 000A 964Ch,
CANFD.RMB28.DF0 000A 965Ch to CANFD.RMB28.DF15 000A 9698h,
CANFD.RMB29.DF0 000A 96A8h to CANFD.RMB29.DF15 000A 96E4h,
CANFD.RMB30.DF0 000A 96F4h to CANFD.RMB30.DF15 000A 9730h,
CANFD.RMB31.DF0 000A 9740h to CANFD.RMB31.DF15 000A 977Ch

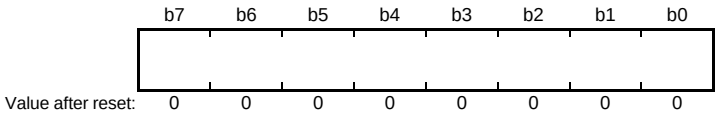
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	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

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message. Unused data bytes are filled with 00h.

33.2.61.7 Receive Message Buffer n Data Register k (RMBn.DATAk) (n = 0 to 31; k = 0 to 63)

Address(es): CANFD.RMB0.DATA0 000A 892Ch to CANFD.RMB0.DATA63 000A 896Bh,
CANFD.RMB1.DATA0 000A 8978h to CANFD.RMB1.DATA63 000A 89B7h,
CANFD.RMB2.DATA0 000A 89C4h to CANFD.RMB2.DATA63 000A 8A03h,
CANFD.RMB3.DATA0 000A 8A10h to CANFD.RMB3.DATA63 000A 8A4Fh,
CANFD.RMB4.DATA0 000A 8A5Ch to CANFD.RMB4.DATA63 000A 8A9Bh,
CANFD.RMB5.DATA0 000A 8AA8h to CANFD.RMB5.DATA63 000A 8AE7h,
CANFD.RMB6.DATA0 000A 8AF4h to CANFD.RMB6.DATA63 000A 8B33h,
CANFD.RMB7.DATA0 000A 8B40h to CANFD.RMB7.DATA63 000A 8B7Fh,
CANFD.RMB8.DATA0 000A 8D2Ch to CANFD.RMB8.DATA63 000A 8D6Bh,
CANFD.RMB9.DATA0 000A 8D78h to CANFD.RMB9.DATA63 000A 8DB7h,
CANFD.RMB10.DATA0 000A 8DC4h to CANFD.RMB10.DATA63 000A 8E03h,
CANFD.RMB11.DATA0 000A 8E10h to CANFD.RMB11.DATA63 000A 8E4Fh,
CANFD.RMB12.DATA0 000A 8E5Ch to CANFD.RMB12.DATA63 000A 8E9Bh,
CANFD.RMB13.DATA0 000A 8EA8h to CANFD.RMB13.DATA63 000A 8EE7h,
CANFD.RMB14.DATA0 000A 8EF4h to CANFD.RMB14.DATA63 000A 8F33h,
CANFD.RMB15.DATA0 000A 8F40h to CANFD.RMB15.DATA63 000A 8F7Fh,
CANFD.RMB16.DATA0 000A 912Ch to CANFD.RMB16.DATA63 000A 916Bh,
CANFD.RMB17.DATA0 000A 9178h to CANFD.RMB17.DATA63 000A 91B7h,
CANFD.RMB18.DATA0 000A 91C4h to CANFD.RMB18.DATA63 000A 9203h,
CANFD.RMB19.DATA0 000A 9210h to CANFD.RMB19.DATA63 000A 924Fh,
CANFD.RMB20.DATA0 000A 925Ch to CANFD.RMB20.DATA63 000A 929Bh,
CANFD.RMB21.DATA0 000A 92A8h to CANFD.RMB21.DATA63 000A 92E7h,
CANFD.RMB22.DATA0 000A 92F4h to CANFD.RMB22.DATA63 000A 9333h,
CANFD.RMB23.DATA0 000A 9340h to CANFD.RMB23.DATA63 000A 937Fh,
CANFD.RMB24.DATA0 000A 952Ch to CANFD.RMB24.DATA63 000A 956Bh,
CANFD.RMB25.DATA0 000A 9578h to CANFD.RMB25.DATA63 000A 95B7h,
CANFD.RMB26.DATA0 000A 95C4h to CANFD.RMB26.DATA63 000A 9603h,
CANFD.RMB27.DATA0 000A 9610h to CANFD.RMB27.DATA63 000A 964Fh,
CANFD.RMB28.DATA0 000A 965Ch to CANFD.RMB28.DATA63 000A 969Bh,
CANFD.RMB29.DATA0 000A 96A8h to CANFD.RMB29.DATA63 000A 96E7h,
CANFD.RMB30.DATA0 000A 96F4h to CANFD.RMB30.DATA63 000A 9733h,
CANFD.RMB31.DATA0 000A 9740h to CANFD.RMB31.DATA63 000A 977Fh



These registers are read-only registers that store the data bytes of the received message.
Unused data bytes are filled with 00h.

33.2.61.8 Receive FIFO n (RFBn) (n = 0, 1)

The total number of the receive FIFO (RFB) is two, as shown in Figure 33.33.
The receive FIFO consists of the following registers:

- RFBn.HF0
- RFBn.HF1
- RFBn.HF2
- RFBn.DF0 to RFBn.DF15

Table 33.8 shows the structure of this buffer.

Table 33.8 Structure of Receive FIFO

Address Offset	Symbol	Register Name	Contents
+00h	RFBn.HF0	Receive FIFO n Header Field 0	IDE, RTR, ID
+04h	RFBn.HF1	Receive FIFO n Header Field 1	DLC, timestamp
+08h	RFBn.HF2	Receive FIFO n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RFBn.DF0	Receive FIFO n Data Field 0	DATA0 to DATA3
+10h	RFBn.DF1	Receive FIFO n Data Field 1	DATA4 to DATA7
+14h	RFBn.DF2	Receive FIFO n Data Field 2	DATA8 to DATA11
+18h	RFBn.DF3	Receive FIFO n Data Field 3	DATA12 to DATA15
+1Ch	RFBn.DF4	Receive FIFO n Data Field 4	DATA16 to DATA19
+20h	RFBn.DF5	Receive FIFO n Data Field 5	DATA20 to DATA23
+24h	RFBn.DF6	Receive FIFO n Data Field 6	DATA24 to DATA27
+28h	RFBn.DF7	Receive FIFO n Data Field 7	DATA28 to DATA31
+2Ch	RFBn.DF8	Receive FIFO n Data Field 8	DATA32 to DATA35
+30h	RFBn.DF9	Receive FIFO n Data Field 9	DATA36 to DATA39
+34h	RFBn.DF10	Receive FIFO n Data Field 10	DATA40 to DATA43
+38h	RFBn.DF11	Receive FIFO n Data Field 11	DATA44 to DATA47
+3Ch	RFBn.DF12	Receive FIFO n Data Field 12	DATA48 to DATA51
+40h	RFBn.DF13	Receive FIFO n Data Field 13	DATA52 to DATA55
+44h	RFBn.DF14	Receive FIFO n Data Field 14	DATA56 to DATA59
+48h	RFBn.DF15	Receive FIFO n Data Field 15	DATA60 to DATA63

33.2.61.9 Receive FIFO n Header Filed 0 (RFBn.HF0) (n = 0, 1)

Address(es): CANFD.RFB0.HF0 000A 8520h, CANFD.RFB1.HF0 000A 856Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
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
	ID[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
b28-b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive FIFO.
For alignment of these bits in base and extended format, refer to section 33.2.60, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.

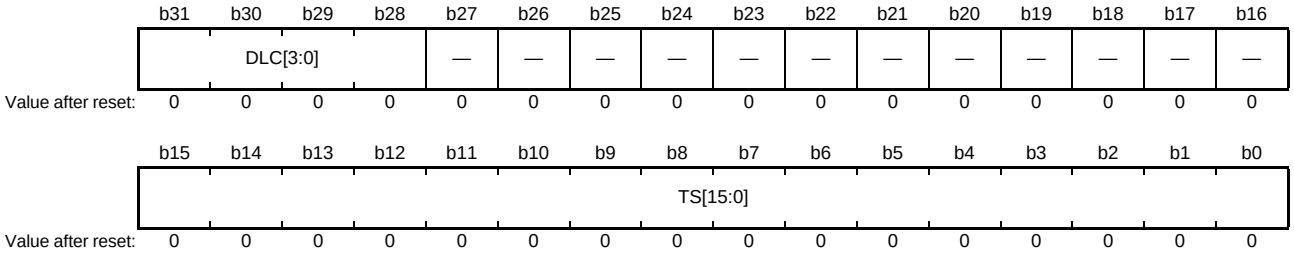
Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.

33.2.61.10 Receive FIFO n Header Filed 1 (RFBn.HF1) (n = 0, 1)

Address(es): CANFD.RFB0.HF1 000A 8524h, CANFD.RFB1.HF1 000A 8570h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame	R
b27 to b16	—	Reserved	These bits are read as 0	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RFCRn.PLS[2:0] bits.

33.2.61.11 Receive FIFO n Header Field 2 (RFBn.HF2) (n = 0, 1)

Address(es): CANFD.RFB0.HF2 000A 8528h, CANFD.RFB1.HF2 000A 8574h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after 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	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0	R
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0	R
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

33.2.61.12 Receive FIFO n Data Field p (RFBn.DFp) (n = 0, 1; p = 0 to 15)

Address(es): CANFD.RFB0.DF0 000A 852Ch to CANFD.RFB0.DF15 000A 8568h,
CANFD.RFB1.DF0 000A 8578h to CANFD.RFB1.DF15 000A 85B4h

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

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message.
Unused data bytes are filled with 00h.

33.2.61.13 Receive FIFO n Data Register k (RFBn.DATAk) (n = 0, 1; k = 0 to 63)

Address(es): CANFD.RFB0.DATA0 000A 852Ch to CANFD.RFB0.DATA63 000A 856Bh,
CANFD.RFB1.DATA0 000A 8578h to CANFD.RFB1.DATA63 000A 85B7h

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

These registers are read-only registers that store the data bytes of the received message.
Unused data bytes are filled with 00h.

33.2.61.14 Common FIFO 0 (CFB0)

The total number of the common FIFO (CFB) is one, as shown in Figure 33.33.
The common FIFO consists of the following registers:

- CFB0.HF0
- CFB0.HF1
- CFB0.HF2
- CFB0.DF0 to CFB0.DF15

Table 33.9 shows the structure of this buffer.

Table 33.9 Structure of Common FIFO

Address Offset	Symbol	Register Name	Contents
+00h	CFB0.HF0	Common FIFO 0 Header Field 0	IDE, RTR, ID
+04h	CFB0.HF1	Common FIFO 0 Header Field 1	DLC, timestamp
+08h	CFB0.HF2	Common FIFO 0 Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	CFB0.DF0	Common FIFO 0 Data Field 0	DATA0 to DATA3
+10h	CFB0.DF1	Common FIFO 0 Data Field 1	DATA4 to DATA7
+14h	CFB0.DF2	Common FIFO 0 Data Field 2	DATA8 to DATA11
+18h	CFB0.DF3	Common FIFO 0 Data Field 3	DATA12 to DATA15
+1Ch	CFB0.DF4	Common FIFO 0 Data Field 4	DATA16 to DATA19
+20h	CFB0.DF5	Common FIFO 0 Data Field 5	DATA20 to DATA23
+24h	CFB0.DF6	Common FIFO 0 Data Field 6	DATA24 to DATA27
+28h	CFB0.DF7	Common FIFO 0 Data Field 7	DATA28 to DATA31
+2Ch	CFB0.DF8	Common FIFO 0 Data Field 8	DATA32 to DATA35
+30h	CFB0.DF9	Common FIFO 0 Data Field 9	DATA36 to DATA39
+34h	CFB0.DF10	Common FIFO 0 Data Field 10	DATA40 to DATA43
+38h	CFB0.DF11	Common FIFO 0 Data Field 11	DATA44 to DATA47
+3Ch	CFB0.DF12	Common FIFO 0 Data Field 12	DATA48 to DATA51
+40h	CFB0.DF13	Common FIFO 0 Data Field 13	DATA52 to DATA55
+44h	CFB0.DF14	Common FIFO 0 Data Field 14	DATA56 to DATA59
+48h	CFB0.DF15	Common FIFO 0 Data Field 15	DATA60 to DATA63

33.2.61.15 Common FIFO 0 Header Field 0 (CFB0.HF0)

Address(es): CANFD.CFB0.HF0 000A 85B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	THENT	ID[28:16]												
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
	ID[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
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	Receive FIFO mode: Reserved. This bit is read as 0. Transmit FIFO mode: 0: Entry is not stored to the Transmission History after successful transmission. 1: Entry is stored to the Transmission History after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

In receive FIFO mode, this register is a read-only register for reading the ID field, IDE bit, and RTR bit of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the ID field, IDE bit, and RTR bit of the message to be transmitted at the end of the FIFO buffer.

ID[28:0] Bits (Identifier)

In receive FIFO mode, the ID[28:0] bits store the standard or extended identifier field of the received message.
In transmit FIFO mode, the ID[28:0] bits are used to set the value of the standard or extended identifier field of the message to be transmitted.
For alignment of these bits in base and extended format, refer to section 33.2.60, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

This bit is valid only in transmit FIFO mode.
The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message.

RTR Bit (Remote Transmission Request)

In receive FIFO mode, the RTR bit stores the value of the RTR bit of the received message.
In transmit FIFO mode, the RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When a CAN FD frame was received (receive mode), this bit reflects the value of the RRS bit. When transmitting a CAN FD frame (CFB0.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

In receive FIFO mode, the IDE bit stores the value of the IDE bit of the received message.
In transmit FIFO mode, the IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

33.2.61.16 Common FIFO 0 Header Field 1 (CFB0.HF1)

Address(es): CANFD.CFB0.HF1 000A 85BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[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
	TS[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	TS[15:0]	Timestamp	Timestamp value of the received CAN frame (in receive FIFO mode)	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame	R/W

In receive FIFO mode, this register is a read-only register for reading the data length code (DLC) and timestamp of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the data length code (DLC) of the message to be transmitted at the end of the FIFO buffer.

TS[15:0] Bits (Timestamp)

These bits are valid only in receive FIFO mode.
The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

In receive FIFO mode, the DLC[3:0] bits store the number of data bytes of the received message.
In transmit FIFO mode, The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the CFCR0.PLS[2:0] bits.

33.2.61.17 Common FIFO 0 Header Field 2 (CFB0.HF2)

Address(es): CANFD.CFB0.HF2 000A 85C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after 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	ESI	Error State Indicator	0: CAN FD frame received or to transmit by error active node 1: CAN FD frame received or to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame received or to transmit with no bit rate switch 1: CAN FD frame received or to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame received or to transmit 1: CAN FD frame received or to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter or sets the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter or sets the pointer to be stored in the Transmission History	R/W

In receive FIFO mode, this register is a read-only register for reading the FDF bit, BRS bit, and ESI flag of the received message, and information label and pointer for the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history at the end of the FIFO buffer.

ESI Bit (Error State Indicator)

In receive FIFO mode, the ESI bit stores the value of the ESI flag of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted. If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

In receive FIFO mode, the BRS bit stores the value of the BRS bit of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

In receive FIFO mode, the FDF bit stores the value of the FDF bit of the received CAN FD frame.
In transmit FIFO mode, the FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

In receive FIFO mode, the IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

In receive FIFO mode, the PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

33.2.61.18 Common FIFO 0 Data Field p (CFB0.DFp) (p = 0 to 15)

Address(es): CANFD.CFB0.DF0 000A 85C4h to CANFD.CFB0.DF15 000A 8600h

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

In receive FIFO mode, these registers are read-only registers for reading the data byte (p × 4 + 3) to data byte (p × 4) of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.

33.2.61.19 Common FIFO 0 Data Register k (CFB0.DATAk) (k = 0 to 63)

Address(es): CANFD.CFB0.DATA0 000A 85C4h to CANFD.CFB0.DATA63 000A 8603h

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

In receive FIFO mode, these registers are read-only registers for reading the data bytes of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing the data bytes of the message to be transmitted.

33.2.61.20 Transmit Message Buffer n (TMBn) (n = 0 to 3)

The total number of transmit message buffer (TMB) is four, as shown in Figure 33.33.
The transmit message buffer consists of the following registers:

- TMBn.HF0
- TMBn.HF1
- TMBn.HF2
- TMBn.DF0 to TMBn.DF15

Table 33.10 shows the structure of this buffer.

Table 33.10 Structure of Transmit Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	TMBn.HF0	Transmit Message Buffer n Header Field 0	IDE, RTR, ID
+04h	TMBn.HF1	Transmit Message Buffer n Header Field 1	DLC
+08h	TMBn.HF2	Transmit Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	TMBn.DF0	Transmit Message Buffer n Data Field 0	DATA0 to DATA3
+10h	TMBn.DF1	Transmit Message Buffer n Data Field 1	DATA4 to DATA7
+14h	TMBn.DF2	Transmit Message Buffer n Data Field 2	DATA8 to DATA11
+18h	TMBn.DF3	Transmit Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	TMBn.DF4	Transmit Message Buffer n Data Field 4	DATA16 to DATA19
+20h	TMBn.DF5	Transmit Message Buffer n Data Field 5	DATA20 to DATA23
+24h	TMBn.DF6	Transmit Message Buffer n Data Field 6	DATA24 to DATA27
+28h	TMBn.DF7	Transmit Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	TMBn.DF8	Transmit Message Buffer n Data Field 8	DATA32 to DATA35
+30h	TMBn.DF9	Transmit Message Buffer n Data Field 9	DATA36 to DATA39
+34h	TMBn.DF10	Transmit Message Buffer n Data Field 10	DATA40 to DATA43
+38h	TMBn.DF11	Transmit Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	TMBn.DF12	Transmit Message Buffer n Data Field 12	DATA48 to DATA51
+40h	TMBn.DF13	Transmit Message Buffer n Data Field 13	DATA52 to DATA55
+44h	TMBn.DF14	Transmit Message Buffer n Data Field 14	DATA56 to DATA59
+48h	TMBn.DF15	Transmit Message Buffer n Data Field 15	DATA60 to DATA63

33.2.61.21 Transmit Message Buffer n Header Field 0 (TMBn.HF0) (n = 0 to 3)

Address(es): CANFD.TMB0.HF0 000A 8604h, CANFD.TMB1.HF0 000A 8650h, CANFD.TMB2.HF0 000A 869Ch, CANFD.TMB3.HF0 000A 86E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	THENT	ID[28:16]												
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
	ID[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
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	0: Entry is not stored in transmission history after successful transmission. 1: Entry is stored in transmission history after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

This register is used to store the ID field, IDE bit, and RTR bit of the message to be transmitted, and to specify whether to store it in the transmission history.
Do not write to this register when the CAN channel is in CH_SLEEP mode.

ID[28:0] Bits (Identifier)

The ID[28:0] bits are used to set the value of the standard or extended identifier field of the message stored in the transmit message buffer.
For alignment of these bits in base and extended format, refer to section 33.2.60, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message stored in the transmit message buffer.

RTR Bit (Remote Transmission Request)

The RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When transmitting a CAN FD frame (TMBn.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

The IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

33.2.61.22 Transmit Message Buffer n Header Field 1 (TMBn.HF1) (n = 0 to 3)

Address(es): CANFD.TMB0.HF1 000A 8608h, CANFD.TMB1.HF1 000A 8654h, CANFD.TMB2.HF1 000A 86A0h, CANFD.TMB3.HF1 000A 86ECh

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

Bit	Symbol	Bit Name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes to be transmitted in a CAN frame.	R/W

This register is used to store the data length code (DLC) fields of the message to be transmitted.
Do not write to this register when the CAN channel is in CH_SLEEP mode.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted when the corresponding TMBn.HF0.RTR bit is set to 0.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

33.2.61.23 Transmit Message Buffer n Header Field 2 (TMBn.HF2) (n = 0 to 3)

Address(es): CANFD.TMB0.HF2 000A 860Ch, CANFD.TMB1.HF2 000A 8658h, CANFD.TMB2.HF2 000A 86A4h, CANFD.TMB3.HF2 000A 86F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after 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	ESI	Error State Indicator	0: CAN FD frame to transmit by error active node 1: CAN FD frame to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame to transmit with no bit rate switch 1: CAN FD frame to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame to transmit 1: CAN FD frame to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	A field that sets the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	A field that sets the pointer to be stored in the Transmission History	R/W

This register is used to store the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history.
Do not write to this register when the CAN channel is in CH_SLEEP mode.

ESI Bit (Error State Indicator)

The ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted.
If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

The BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

The FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

33.2.61.24 Transmit Message Buffer n Data Field p (TMBn.DFp) (n = 0 to 3; p = 0 to 15)

Address(es): CANFD.TMB0.DF0 000A 8610h to CANFD.TMB0.DF15 000A 864Ch,
CANFD.TMB1.DF0 000A 865Ch to CANFD.TMB1.DF15 000A 8698h,
CANFD.TMB2.DF0 000A 86A8h to CANFD.TMB2.DF15 000A 86E4h,
CANFD.TMB3.DF0 000A 86F4h to CANFD.TMB3.DF15 000A 8730h

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

These registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.
Do not write to these registers when the CAN channel is in CH_SLEEP mode.

33.2.61.25 Transmit Message Buffer n Data Register k (TMBn.DATAk) (n = 0 to 3; k = 0 to 63)

Address(es): CANFD.TMB0.DATA0 000A 8610h to CANFD.TMB0.DATA63 000A 864Fh,
CANFD.TMB1.DATA0 000A 865Ch to CANFD.TMB1.DATA63 000A 869Bh,
CANFD.TMB2.DATA0 000A 86A8h to CANFD.TMB2.DATA63 000A 86E7h,
CANFD.TMB3.DATA0 000A 86F4h to CANFD.TMB3.DATA63 000A 8733h

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

These registers are read/write registers for storing the data bytes of the message to be transmitted.
Do not write to these registers when the CAN channel is in CH_SLEEP mode.

33.2.62 ECC Control/Status Register (ECCSR)

Address(es): CANFD.ECCSR 000E D000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EC2EAS	EC1EAS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECEDWC[1:0]		—	—	ECOVF	EC2EC	EC1EC	—	—	ECEDE	EC1ECD	EC2EIE	EC1EIE	EC2EF	EC1EF	ECEF
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	ECEF	ECC Error Flag	0: These is no ECC error in the last read RAM data. 1: There is an ECC error in the last read RAM data.	R
b1	EC1EF	1-Bit ECC Error Detection Flag	0: 1-bit ECC error is not detected. 1: 1-bit ECC error is detected.	R
b2	EC2EF	2-Bit ECC Error Detection Flag	0: 2-bit ECC error is not detected. 1: 2-bit ECC error is detected.	R
b3	EC1EIE	1-Bit ECC Error Detection Interrupt Enable	0: 1-bit ECC error detection interrupt is disabled. 1: 1-bit ECC error detection interrupt is enabled.	R/W
b4	EC2EIE	2-Bit ECC Error Detection Interrupt Enable	0: 2-bit ECC error detection interrupt is disabled. 1: 2-bit ECC error detection interrupt is enabled.	R/W
b5	EC1ECD	1-Bit ECC Error Correction Disable	0: At 1-bit ECC error detection, the error correction is executed. 1: At 1-bit ECC error detection, the error correction is not executed.	R/W
b6	ECEDE	ECC Error Detection Enable	0: ECC error detection is disabled. 1: ECC error detection is enabled.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	EC1EC	EC1EF Flag Clear	Writing 1 to this bit clears the EC1EF flag. Writing 0 is ignored. This bit is read as 0.	R/W
b10	EC2EC	EC2EF Flag Clear	Writing 1 to this bit clears the EC2EF flag. Writing 0 is ignored. This bit is read as 0.	R/W
b11	ECOVF	ECC Overflow Detection Flag	0: The ECEAR register overflow has not occurred. 1: The ECEAR register overflow has occurred.	R
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECEDWC[1:0]	ECEDE Bit Write Control	Enables or disables write access to the ECEDE bit.	R/W
b16	EC1EAS	1-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 1-bit ECC error occurred is stored in the ECEAR register.	R
b17	EC2EAS	2-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 2-bit ECC error occurred is stored in the ECEAR register.	R
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ECEF Flag (ECC Error Flag)

The ECEF flag indicates whether there is an ECC error in the last read RAM data. This flag is updated each time the RAM is read.

If the ECEDE bit is set to 1 without initializing the RAM, the value of this flag has no meaning.

[Setting condition]

- If there is an ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Under the condition that there is no ECC error in read RAM data
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

EC1EF Flag (1-Bit ECC Error Detection Flag)

The EC1EF flag indicates that the 1-bit ECC error is detected in the RAM read data.
When the 1-bit ECC error interrupt is enabled and this flag is set to 1, the 1-bit ECC error interrupt (EC1EI) is generated.
When the 1-bit ECC error is detected again under the condition that this flag is set to 1, the interrupt is not generated.

[Setting condition]

- If there is a 1-bit ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If writing 1 to the EC1EC bit and detecting a 1-bit ECC error occur at the same time, the EC1EF flag becomes 0.

EC2EF Flag (2-Bit ECC Error Detection Flag)

The EC2EF flag indicates that the 2-bit ECC error is detected in the RAM read data.
When the 2-bit ECC error interrupt is enabled and this flag is set to 1, the 2-bit ECC error interrupt (EC2EI) is generated.
When the 2-bit ECC error is detected again under the condition that this flag is set to 1, the interrupt is not generated.

[Setting condition]

- If there is a 2-bit ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If writing 1 to the EC2EC bit and detecting a 2-bit ECC error occur at the same time, the EC2EF flag becomes 0.

EC1EIE Bit (1-Bit ECC Error Detection Interrupt Enable)

The EC1EIE bit enables or disables a 1-bit ECC error detection interrupt.
If the EC1EF flag becomes 1 when this bit is 1, a 1-bit ECC error detection interrupt (EC1EI) is generated.

EC2EIE Bit (2-Bit ECC Error Detection Interrupt Enable)

The EC2EIE bit enables or disables a 2-bit ECC error detection interrupt.
If the EC2EF flag becomes 1 when this bit is 1, a 2-bit ECC error detection interrupt (EC2EI) is generated.

EC1ECD Bit (1-Bit ECC Error Correction Disable)

The EC1ECD bit enables or disables to correct the 1-bit ECC error when the ECEDE bit is set to 1 (ECC error detection is enabled). When this bit is set to 1, the RAM output data is not corrected even if a 1-bit ECC error is detected.

ECEDE Bit (ECC Error Detection Enable)

Setting the ECEDE bit to 1 enables ECC error detection.
Writing to this bit is valid only when the ECEDWC[1:0] bits are set to 01b.

EC1EC Bit (EC1EF Flag Clear)

The EC1EC bit is used to clear the EC1EF flag.
The EC1EF flag is cleared by writing 1 to this bit while the EC1EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If the EC1EF flag is cleared by the EC1EC bit and the setting factor of the EC1EF flag occurs at the same time, the

EC1EF flag becomes 0.

EC2EC Bit (EC2EF Flag Clear)

The EC2EC bit is used to clear the EC2EF flag.
The EC2EF flag is cleared by writing 1 to this bit while the EC2EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If the EC2EF flag is cleared by the EC2EC bit and the setting factor of the EC2EF flag occurs at the same time, the EC2EF flag becomes 0.

ECOVF Flag (ECC Overflow Detection Flag)

If a new ECC error is detected and the address is overwritten when the address is already stored in the ECEAR register, the ECOVF flag becomes 1 and an ECC overflow interrupt (ECOVI) is generated.
The ECC overflow interrupt is generated again when this flag is set to 1 and new ECC error is detected.

[Setting condition]

- When new error address is captured under the condition that error address is already captured in the ECEAR register.

[Clearing condition]

- Writing 1 to the EC1EC bit
- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the ECOVF flag is cleared by the EC1EC or EC2EC bit and the setting condition of the ECOVF flag occurs at the same time, the ECOVF flag becomes 0.

ECEDWC[1:0] Bits (ECEDE Bit Write Control)

The ECEDWC[1:0] bits are used to enable or disable write access to the ECEDE bit. The read value is always 00b.
When the value of these bits is 01b, it is possible to have write access to the ECEDE bit. If these bits are other than 01b, write access to the ECEDE bit is ignored and the value does not change.

EC1EAS Flag (1-Bit ECC Error Detected Address Stored Flag)

The EC1EAS flag indicates that the address where the 1-bit ECC error occurred is stored in the ECEAR register when the ECEDE bit is set to 1 (ECC error detection is enabled).
When 1-bit ECC error is detected while the 2-bit ECC error address has already been captured in the ECEAR register, the ECEAR register is not updated and this flag is not updated.

[Setting condition]

- When a 1-bit ECC error is detected under the condition that the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the EC1EAS flag is cleared by the EC1EC bit and the setting factor of the EC1EAS flag occurs at the same time, the EC1EAS flag becomes 0.

EC2EAS Flag (2-Bit ECC Error Detected Address Stored Flag)

The EC2EAS flag indicates that the address where the 2-bit ECC error occurred is stored in the ECEAR register when the ECEDE bit is set to 1 (ECC error detection is enabled).
When 2-bit ECC error is detected while the 1-bit ECC error address has already been captured in the ECEAR register, the ECEAR register is updated and this flag becomes 1.

[Setting condition]

- When a 2-bit ECC error is detected under the condition that the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the EC2EAS flag is cleared by the EC2EC bit and the setting factor of the EC2EAS flag occurs at the same time, the EC2EAS flag becomes 0.

33.2.63 ECC Test Mode Register (ECTMR)

Address(es): CANFD.ECTMR 000E D004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECTMWC[1:0]		—	—	—	—	—	—	ECTME	—	—	—	—	—	ECDIS	—
Value after 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
b1	ECDIS	ECC Decoder Input Select	0: Input RAM output data to data input of decode circuit 1: Select the ECTDR register to data input of decode circuit	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ECTME	ECC Test Mode Enable	0: Access to the ECDIS bit and the ECTDR register is disabled 1: Access to the ECDIS bit and the ECTDR register is enabled	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECTMWC[1:0]	ECTME Bit Write Control	Enable or disable write access to the ECTME bit.	R/W

ECDIS Bit (ECC Decoder Input Select)

The ECDIS bit selects which of the data value read from RAM and the value of the ECTDR register is used as input data to the ECC decoder.

The write access to this bit is enabled when the ECTME bit is set to 1. It is also possible to set them at the same time.

This bit is cleared by setting the ECTME bit to 0.

ECTME Bit (ECC Test Mode Enable)

The ECTME bit is used to enable or disable the access to the ECDIS bit and the ECTDR register.

This bit can be written only when the ECTMWC[1:0] bits is set to 10b.

ECTMWC[1:0] Bits (ECTME Bit Write Control)

The ECTMWC[1:0] bits are used to enable or disable the write access to the ECTME bit. The read value is always 00b.

When the value of these bits is 10b, it is possible to have write access to the ECTME bit. If these bits are other than 10b, write access to the ECTME bit is ignored and the value does not change.

33.2.64 ECC Decoder Test Data Register (ECTDR)

Address(es): CANFD.ECTDR 000E D00Ch

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

This register is used to set the data for testing the ECC decode.
When the ECTMR.ECTME bit is set to 1, this register can be read and written.
When the ECTMR.ECTME bit is set to 0, the value of this register becomes 00000000h.
When the ECTMR.ECDIS bit is set to 1, the value set in this register is used as the input data of the ECC decoder instead of the data read from RAM.

33.2.65 ECC Error Address Register (ECEAR)

Address(es): CANFD.ECEAR 000E D010h

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

This register holds the address where the ECC error occurred.

If an ECC error is detected when the ECCSR.ECEDE bit is 1 (ECC error detection is enabled), b12 to b2 of the RAM address are stored in b10 to b0 of this register.
If the same error occurred again, this register is not updated.
If a 2-bit ECC error is detected while the address where the 1-bit ECC error occurred is already stored, the ECEAR register is overwritten with the new address and the ECCSR.EC2EAS flag is set to 1.
If a 1-bit ECC error is detected while the address where the 2-bit ECC error occurred is already stored, the ECEAR register is not updated and the ECCSR.EC1EAS flag is not updated.

33.3 Operating Mode

The operating modes of the CANFD module can be classified into two groups:

- Global modes
- Channel modes.

33.3.1 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes.
The Global modes of the CANFD module are:

- GL_SLEEP
- GL_RESET
- GL_HALT
- GL_OPERATION.

Figure 33.2 shows the possible transitions between the Global modes.

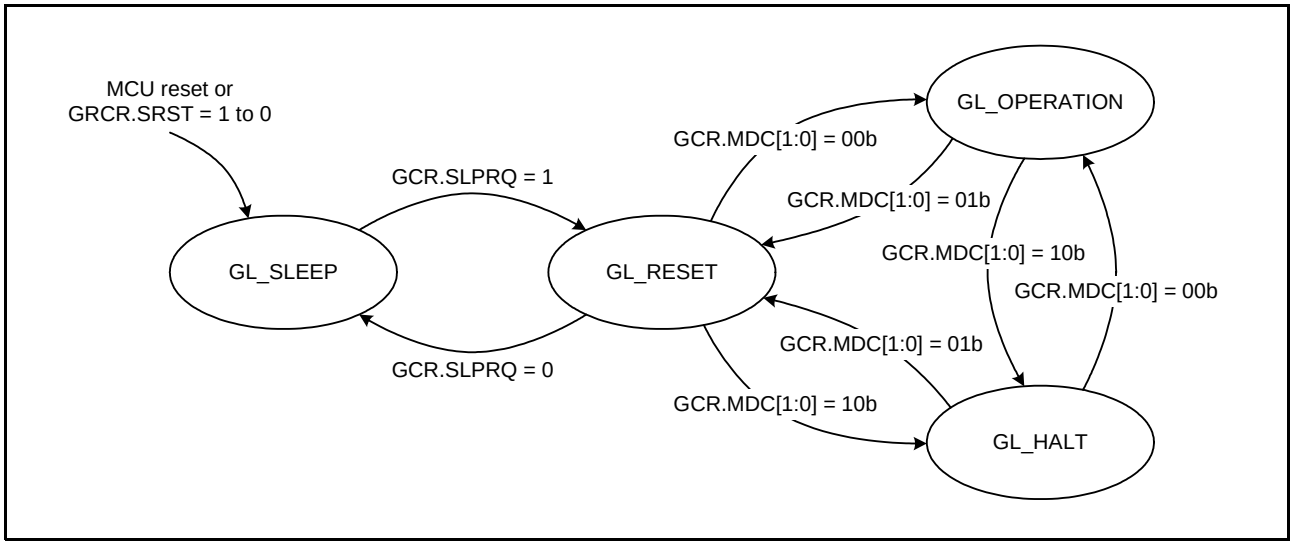


Figure 33.2 Transition between CANFD Global Modes

Change in the Global mode can affect the Channel mode. For details, refer to section 33.3.3, Global Mode and Channel Mode Transition Interactions.

33.3.1.1 GL_SLEEP Mode

The CANFD module automatically enters GL_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CANFD module also enters the GL_SLEEP mode when the GCR.SLPRQ bit is set to 1 while it is in GL_RESET mode. The SLPRQ bit cannot be set to 1 in GL_HALT or GL_OPERATION mode.
Setting the GCR.SLPRQ bit to 1 sets the CHCR.SLPRQ bit to 1 and forces CAN channel into the CH_SLEEP mode.
GL_SLEEP mode is used for power saving purpose. When CANFD module is in GL_SLEEP mode, only the clock for writing to the SLPRQ bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.
Read access from all registers is still possible and all register values are preserved.
After setting the GCR.SLPRQ bit to 1, it is necessary to confirm the GSR.SLPST flag that the GL_SLEEP status has been updated, indicating successful transition to GL_SLEEP mode before the GCR.SLPRQ bit can be cleared again.

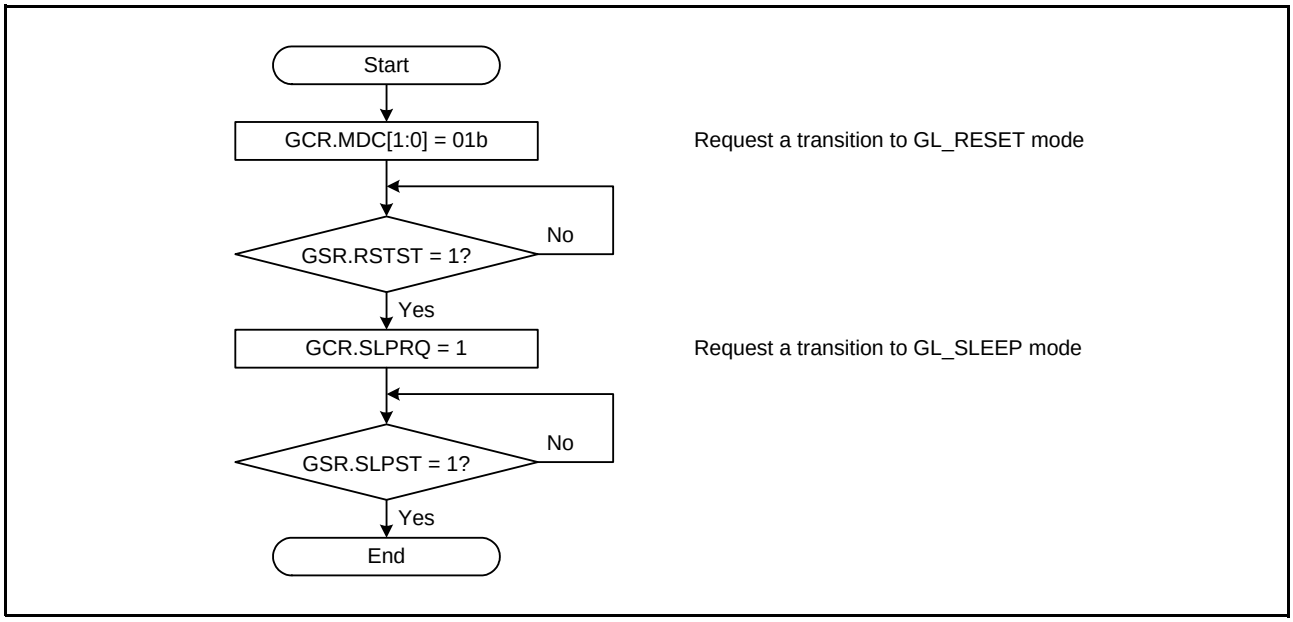


Figure 33.3 Procedure for Entering GL_SLEEP Mode

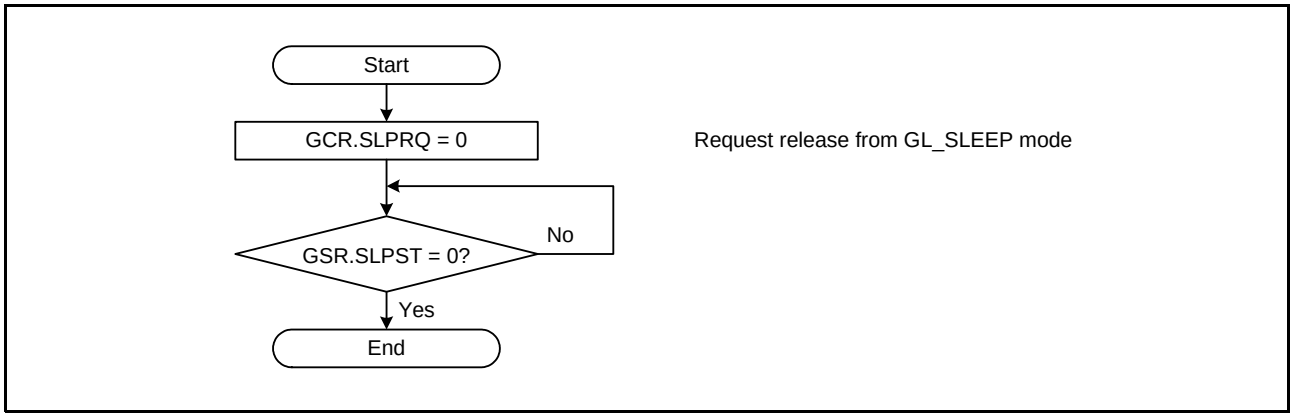


Figure 33.4 Procedure for Exiting GL_SLEEP Mode

33.3.1.2 GL_RESET Mode

The CANFD module enters GL_RESET mode in the following ways:

- The GCR.MDC[1:0] bits are set to 01b while the CANFD module is in GL_HALT or GL_OPERATION mode
- The GCR.SLPRQ bit is set to 0 while the CANFD module is in GL_SLEEP mode.

In GL_RESET mode, all CANFD module functions are suspended and all status and flag registers are initialized. Additionally all FIFOs and all transmit queues are disabled and transmission control bits are cleared.
In this mode, configuration registers other than the GTMCR register and interrupt enable registers are not initialized, so the CANFD module can be configured.
Setting the Global mode to GL_RESET by setting the GCR.MDC[1:0] bits to 01b sets the CHCR.MDC[1:0] bits to 01b and forces the channel into the CH_RESET mode.
When the channel is already in CH_RESET or CH_SLEEP mode, this automatic transition is not performed.
After setting the GCR.MDC[1:0] bits to 01b (GL_RESET mode), it is necessary to confirm that the GSR.RSTST flag has been updated, indicating successful transition to GL_RESET mode before the GCR.MDC[1:0] bits can be changed again.

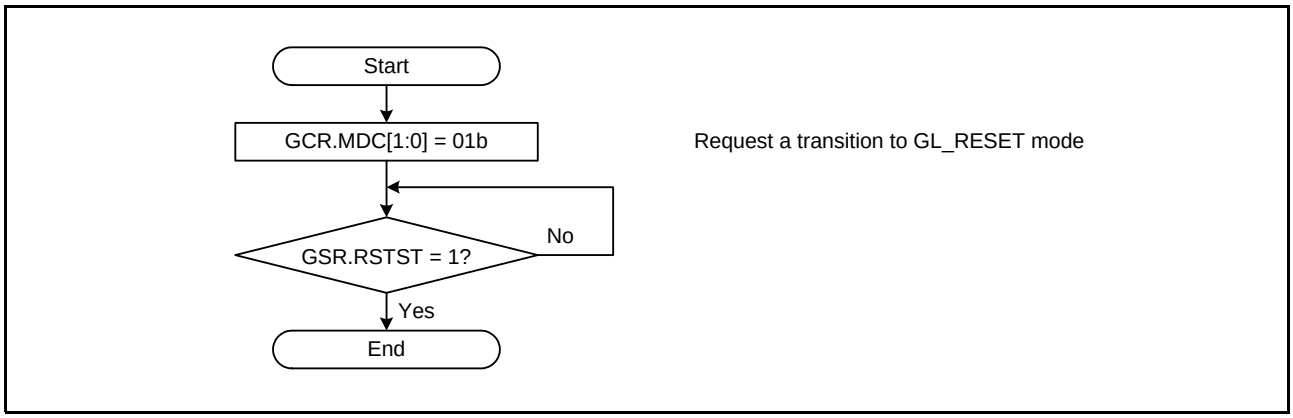


Figure 33.5 Procedure for Entering GL_RESET Mode

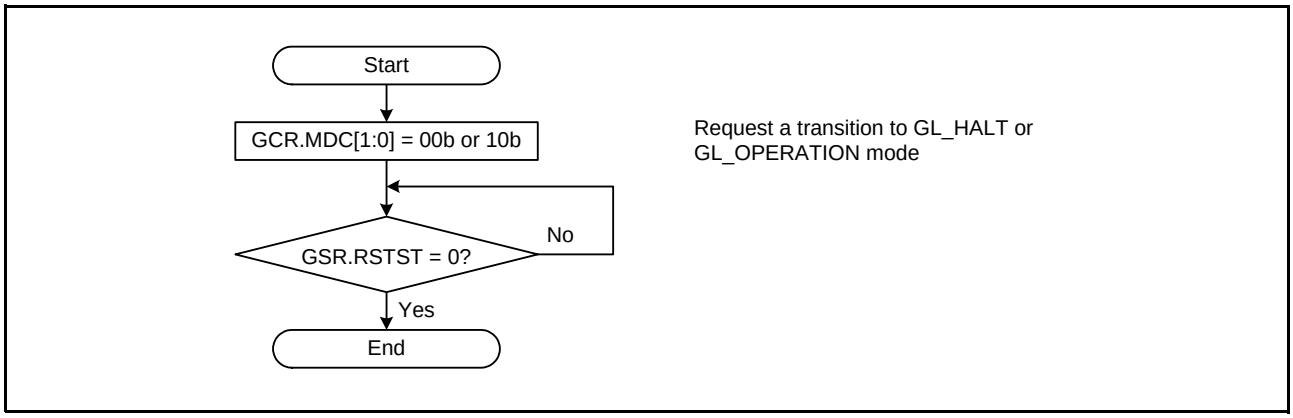


Figure 33.6 Procedure for Exiting GL_RESET Mode

33.3.1.3 GL_HALT Mode

The CANFD module enters GL_HALT mode in the following ways:

- Set the GCR.MDC[1:0] bits to 10b while the CANFD module is in GL_RESET mode:
  – the channels is in either CH_RESET or CH_SLEEP mode and remains in this mode.
- Set the GCR.MDC[1:0] bits to 10b while the CANFD module is in GL_OPERATION mode:
  – CAN channel in CH_RESET, CH_HALT, or CH_SLEEP mode remains in this mode
  – CAN channel in CH_OPERATION mode transits to CH_HALT mode
  – the GSR.HLTST flag is set to 1 when CAN channel exits CH_OPERATION mode.

If a transmission or reception is in progress for a CAN channel, the channel enters CH_HALT mode after waiting for completion of the communication.
Similarly, if a CAN channel is in bus-off, the channel may not enter CH_HALT mode until the bus-off recovery sequence is completed, depending on the channel configuration.
In GL_HALT mode, all communications are suspended and the status and flag registers do not change (only when a channel is in the bus-off, its CHSR.REC[7:0] and TEC[7:0] bits are set to 00h).
Additionally, the GTMCR register and GTMER register are not initialized in this mode. The GL_HALT mode is used to configure Global Test Modes.
Setting the Global mode to GL_HALT by setting the GCR.MDC[1:0] bits to 10b sets the CHCR.MDC[1:0] bits to 10b for the channels that are in CH_OPERATION mode and forces these channels into the CH_HALT mode.
For channels that are already in CH_RESET, CH_HALT, or CH_SLEEP mode, this automatic transition is not performed.
Therefore, the GL_HALT mode request can be used to shut down all CAN channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).
After setting the GCR.MDC[1:0] bits to 10b (GL_HALT mode), it is necessary to confirm that the GSR.HLTST flag has been updated to indicate a successful transition to GL_HALT mode. Do not set any other registers until confirming the GSR.HLTST flag is set.

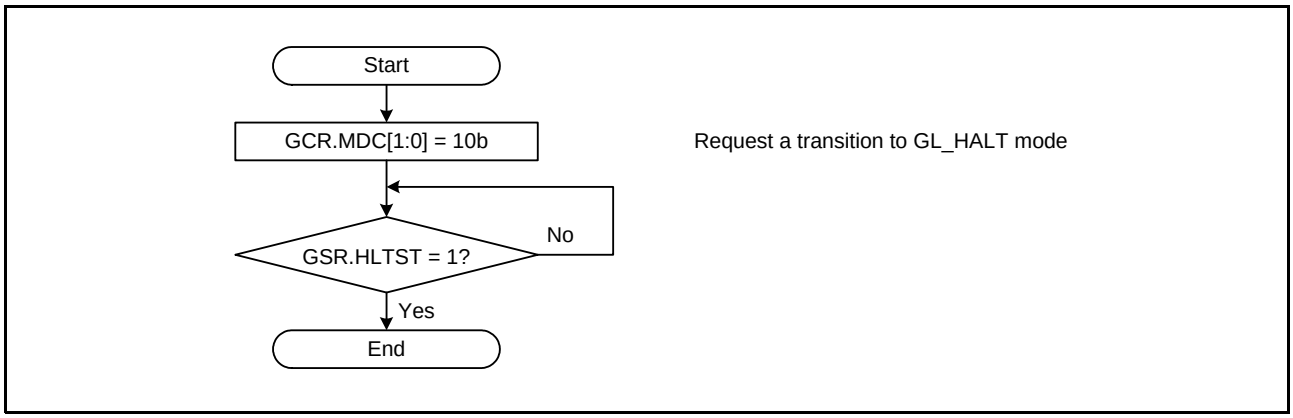


Figure 33.7 Procedure for Entering GL_HALT Mode

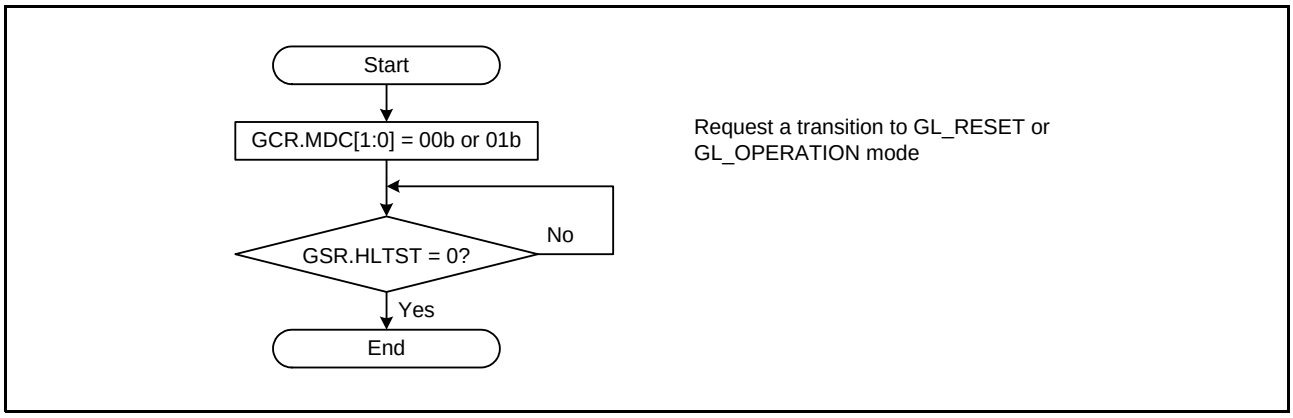


Figure 33.8 Procedure for Exiting GL_HALT Mode

33.3.1.4 GL_OPERATION Mode

The CANFD module enters this mode when the GCR.MDC[1:0] bits are set to 00b.
The CANFD channel can be set to CH_OPERATION mode and start CAN communication only when CANFD is in GL_OPERATION mode.
After setting the GCR.MDC[1:0] bits to 00b (GL_OPERATION mode), it is necessary to confirm that the GSR.RSTST flag and the GSR.HLTST flag have been set to 0 to indicate a successful transition to GL_OPERATION mode before the GCR.MDC[1:0] bits can be modified again.

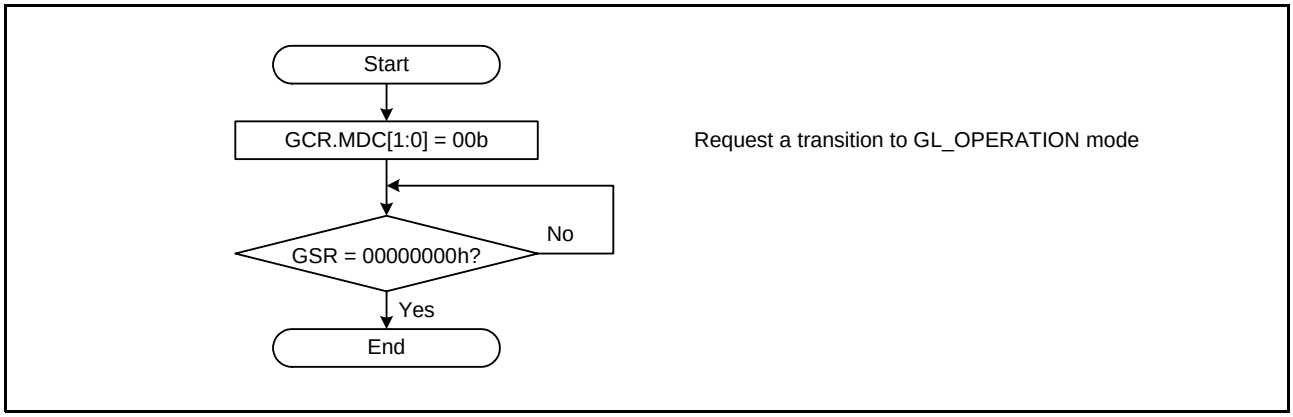


Figure 33.9 Procedure for Entering GL_OPERATION Mode

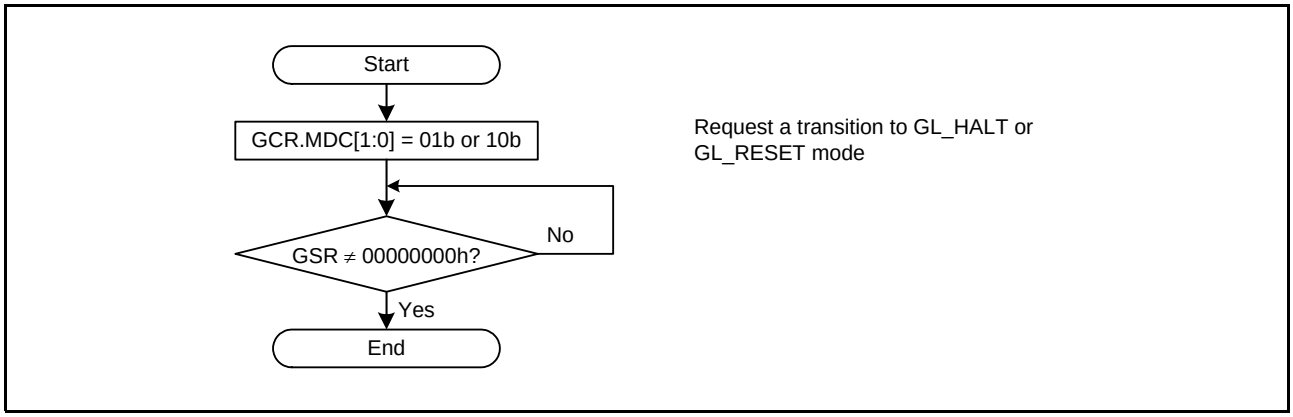


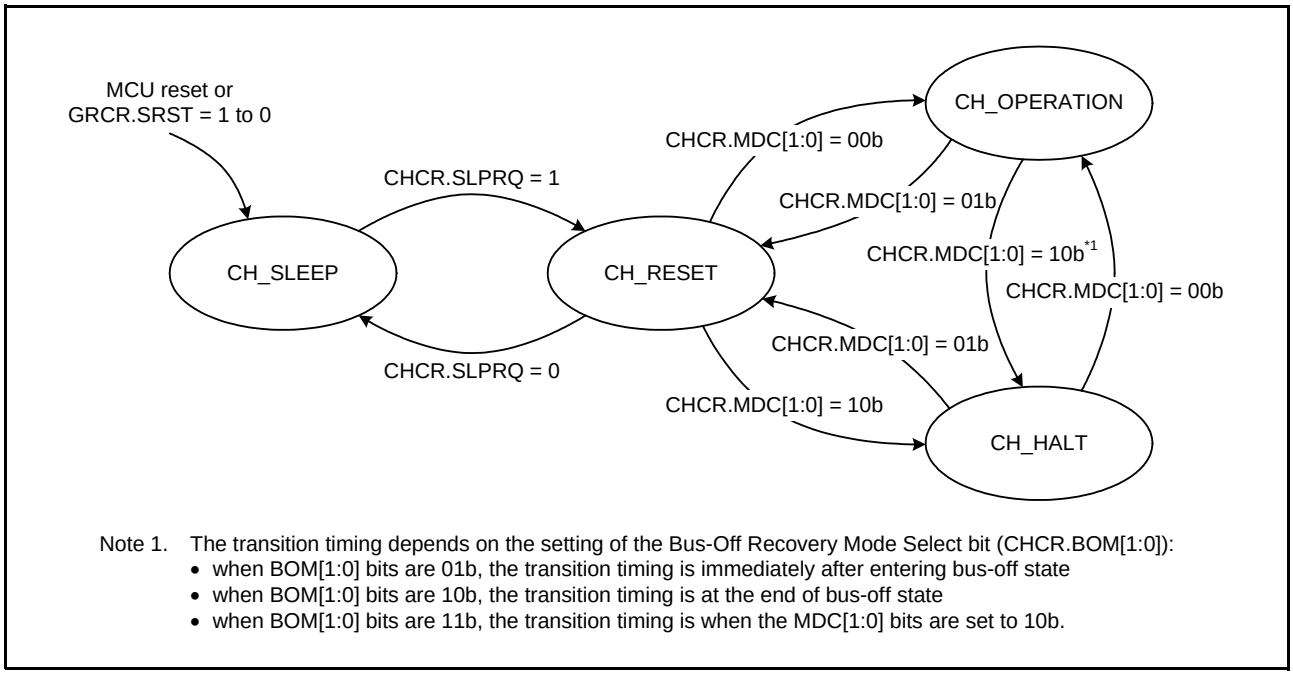
Figure 33.10 Procedure for Exiting GL_OPERATION Mode

33.3.2 Channel Modes

The Channel modes of the CANFD module are:

- CH_RESET
- CH_HALT
- CH_OPERATION
- CH_SLEEP.

Figure 33.11 shows the possible transitions between the channel modes.



Note 1. The transition timing depends on the setting of the Bus-Off Recovery Mode Select bit (CHCR.BOM[1:0]):
- when BOM[1:0] bits are 01b, the transition timing is immediately after entering bus-off state
- when BOM[1:0] bits are 10b, the transition timing is at the end of bus-off state
- when BOM[1:0] bits are 11b, the transition timing is when the MDC[1:0] bits are set to 10b.

Figure 33.11 Transition between CAN Channel Modes

33.3.2.1 CH_SLEEP Mode

The CAN channel of the CANFD module automatically enters CH_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CAN channel also enters CH_SLEEP mode when the CHCR.SLPRQ bit is set to 1 while the CAN channel is in CH_RESET mode. The CHCR.SLPRQ bit cannot be set to 1 in CH_HALT or CH_OPERATION mode.
Entering CH_SLEEP mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.
After setting the CHCR.SLPRQ bit to 1 and before setting it to 0 again, it is necessary to use the CHSR.SLPST flag to confirm that transition to CH_SLEEP mode was successful.
During CH_SLEEP mode, do not write to channel related registers. Read operation is still possible.

33.3.2.2 CH_RESET Mode

The CAN channel of CANFD module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 01b while the CAN channel is in CH_HALT or CH_OPERATION mode
- The CHCR.SLPRQ bit is set to 0 while the CAN channel is in CH_SLEEP mode
- The GCR.MDC[1:0] bits are set to 01b while the CAN channel is not in CH_SLEEP or CH_RESET mode.

In CH_RESET mode, all CAN channel status and flags are initialized.
In addition, all transmission-related control bits of the channel are cleared, and the transmit queue of the channel is also disabled.
In this mode, the configuration registers except for the bits related to the channel test mode are not initialized, so the CAN channel can be configured for communication.
After setting the CHCR.MDC[1:0] bits to 01b (CH_RESET mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag to confirm that transition to CH_RESET mode was successful.
Refer to Table 33.11 for the behavior of transitioning to CH_RESET mode while CAN communication is in progress.

33.3.2.3 CH_HALT Mode

The CAN channel of CANFD module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_RESET or CH_OPERATION mode
- The GCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_OPERATION mode.

In CH_HALT mode, all CAN communication of the channel is suspended, but all statuses and flags remain unchanged during CH_HALT mode (except for the bus-off state where the CHSR.REC[7:0] and TEC[7:0] bits for the channel are set to 00h).
In addition, in this mode, the bits related to the channel test mode are not initialized. Use CH_HALT mode to configure the channel test mode.
After setting the CHCR.MDC[1:0] bits to 10b (CH_HALT mode) and before modifying these bits again, it is necessary to use the CHSR.HLTST flag to confirm that transition to CH_HALT mode was successful.
Refer to Table 33.11 for the transition behavior to CH_HALT mode while CAN communication is in progress.

Table 33.11 Transition Behavior in CH_RESET Mode and CH_HALT Mode

Mode	State		
	Receiver	**Transmitter**	**Bus-Off**
CH_RESET mode (CHCR.MDC[1:0] = 01b)	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the bus-off recovery.
CH_HALT mode (CHCR.MDC[1:0] = 10b)	CAN channel transits to CH_HALT mode at the end of the ongoing reception or when an error occurs.*2	CAN channel transits to CH_HALT mode after completion of the ongoing transmission.	When the CHCR.BOM[1:0] bits are 00b, a CH_HALT mode request is accepted only after the completion of the full bus-off recovery sequence. When the CHCR.BOM[1:0] bits are 10b, the CAN channel transits automatically to CH_HALT mode after waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 01b, the CAN channel transits automatically to CH_HALT mode without waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 11b, the CAN channel transits to CH_HALT mode as soon as CH_HALT mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to CH_RESET mode is required only at the end of an ongoing communication, then CH_HALT mode can be requested first to prevent interruption of CAN communication by direct transition to CH_RESET mode. After the CAN channel enters CH_HALT mode, the CH_RESET mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to CH_RESET mode.

33.3.2.4 CH_OPERATION Mode (in Other than Bus-Off State)

The CH_OPERATION mode is activated by setting the CHCR.MDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering the CH_OPERATION mode, the CHSR.CRDY flag is set to 1 and the CAN channel:

- Enables the functions of the communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters.

At this point, the CAN channel can start transmission and reception of messages.
Within the CH_OPERATION mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (refer to Figure 33.12):

- Idle mode: The CAN channel is neither receiving nor transmitting
- Receive mode: The channel is receiving a message transmitted by another CAN node
- Transmit mode: The channel is transmitting a message
  (The channel may receive its own message simultaneously when Self Test mode is enabled.)
- Bus-off mode: The CAN channel is cut-off from CAN bus communication.

After setting the CHCR.MDC[1:0] bits to 00b (CH_OPERATION mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag and the CHSR.HLTST flag to confirm that transition to CH_OPERATION mode was successful.

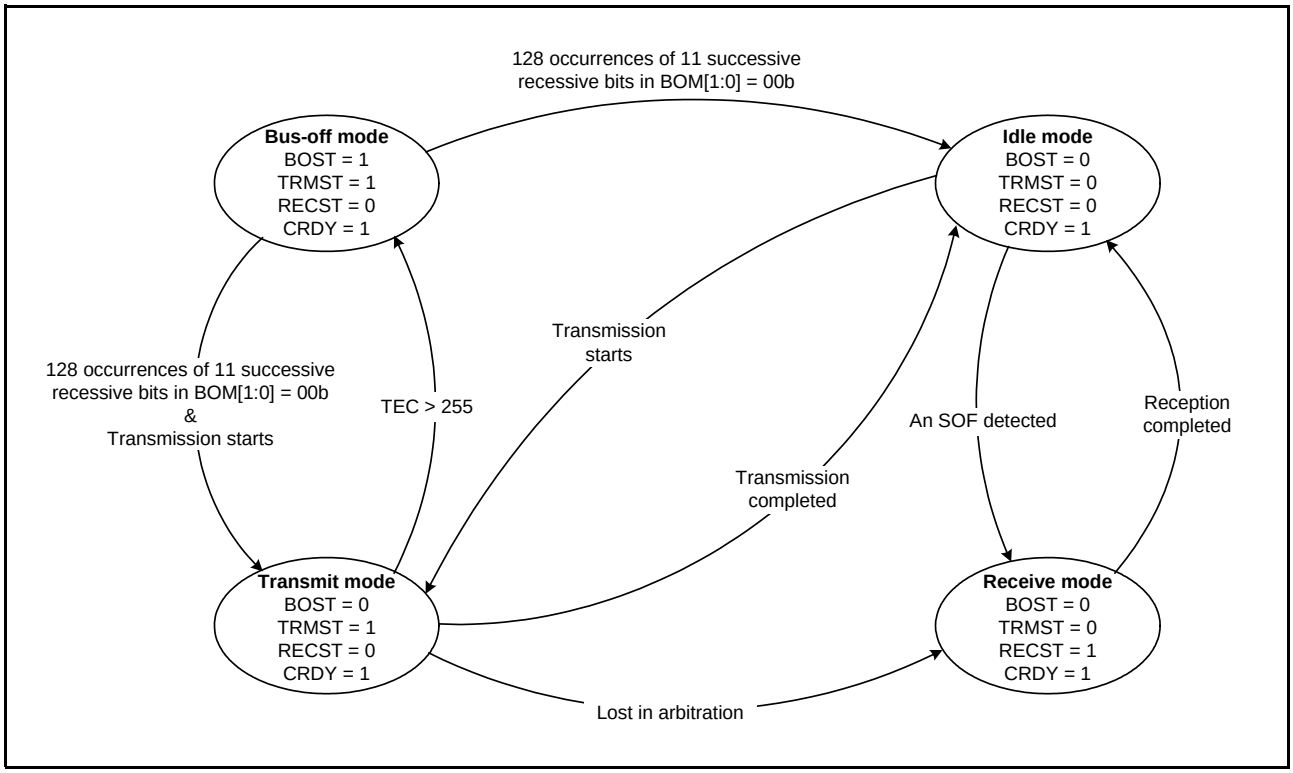


Figure 33.12 Sub-modes of CH_OPERATION Mode (only when CHCR.BOM[1:0] = 00b)

33.3.2.5 CH_OPERATION Mode (in Bus-Off State)

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CH_OPERATION mode from the bus-off state:

- CHCR.BOM[1:0] = 00b:
  Bus-off recovery is compliant to ISO 11898-1. The CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h, and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 01b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1 in this case.
- CHCR.BOM[1:0] = 10b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter CH_HALT mode when still in bus-off state if a request is made to enter CH_HALT mode. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1.
  Without setting CHCR.MDC[1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CHCR.BOM[1:0] = 00b.

If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no CH_HALT request has been generated during this period, then the CHESR.BORDF flag is set to 1.
When software writes to the CHCR.MDC[1:0] bits at the same time as the CAN channel enters CH_HALT mode (at the start of bus-off when CHCR.BOM[1:0] = 01b, or at the end of bus-off when CHCR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CHCR.MDC[1:0] bits to CH_HALT mode request is performed when the CHCR.MDC[1:0] bits value is previously 00b (CH_OPERATION mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting the CHCR.RTBO bit to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The CHESR.BORDF flag is not set to 1 in this case, and the CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h.
Before setting the CHCR.RTBO bit to 1, all pending transmissions from the transmit message buffers, transmit queues and/or common FIFO in transmit mode should be disabled.
The disable of the pending transmit message buffer, transmit queue or common FIFO must be confirmed by the corresponding acknowledge flags (TMSRn.TXRF[1:0] flags, TQSR0.EMPTY flag, and CFSR0.EMPTY flag).
The CHCR.RTBO bit should be used for bus-off recovery only when the CHCR.BOM[1:0] bits are set to 00b.
Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.
Table 33.12 lists the behavior of the bus-off entry detect flag (CHESR.BOEDF) and the bus-off recovery detect flag (CHESR.BORDF) according to the CHCR.BOM[1:0] bit settings.

Table 33.12 Behavior of Bus-off Entry and Recovery Flags

CHCR.BOM[1:0]	CHESR.BOEDF Flag	CHESR.BORDF Flag
00b	Becomes 1 on entry to bus-off state.	Becomes 1 on exit from bus-off state.
00b Set the CHCR.RTBO bit to 1		Becomes 1 only if normal bus-off recovery occurs before setting the CHCR.RTBO bit to 1.
01b		Does not become 1.
10b		Becomes 1 on exit from bus-off state.
11b		Becomes 1 only if normal bus-off recovery occurs before requesting a transition to CH_HALT mode.

To make efficient software, it is not necessary to wait for the bus-off recovery sequence to end.
It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 33.13.

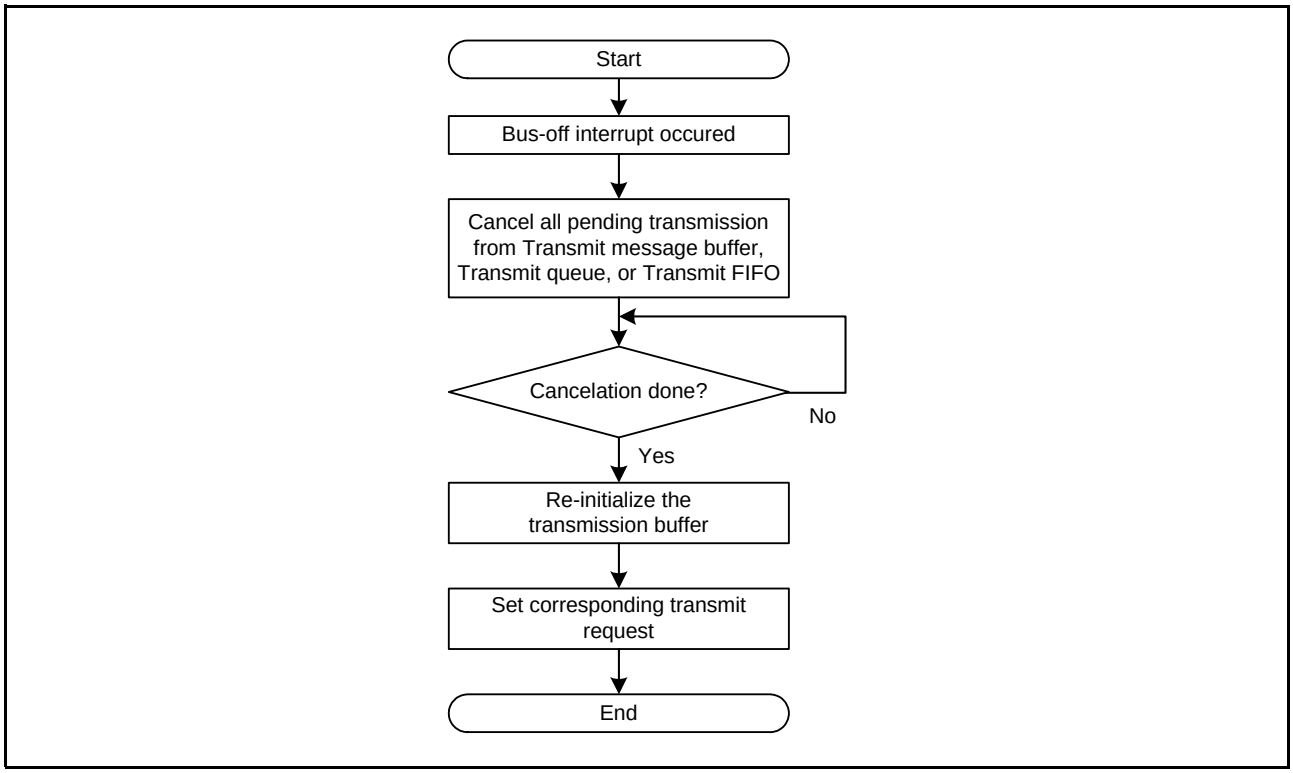


Figure 33.13 Transmission Re-Initialization During Bus-Off

33.3.3 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the CHCR.MDC[1:0] bits does not affect the GCR.MDC[1:0] bits.
- Changing the GCR.MDC[1:0] bits affects the channel mode control as described in Table 33.13.

Table 33.13 Interaction between Global Mode and Channel Mode Transition

Global Mode Change	Channel Mode before Changing Global Mode	Channel Mode after Changing Global Mode
GL_SLEEP → GL_RESET	CH_SLEEP	CH_SLEEP (no change)
GL_RESET → GL_SLEEP	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_SLEEP
GL_RESET → GL_HALT	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
GL_RESET → GL_OPERATION	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
GL_HALT → GL_RESET	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_RESET
GL_HALT → GL_OPERATION	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_HALT (no change)
GL_OPERATION → GL_RESET	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_RESET
	CH_OPERATION	CH_RESET
GL_OPERATION → GL_HALT	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_HALT (no change)
	CH_OPERATION	After communication ends, CH_HALT

33.3.3.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

Table 33.14 Maximum Transition Time for the Global Mode

From	To	Maximum Transition Time
GL_SLEEP	GL_RESET	3 × PCLKB*2
GL_RESET	GL_SLEEP	3 × PCLKB
GL_RESET	GL_HALT	10 × PCLKB
GL_RESET	GL_OPERATION	10 × PCLKB
GL_HALT	GL_RESET	2 bit times (1 Tq + 16 × PCLKB + 2 DLL clock cycles)
GL_HALT	GL_OPERATION	3 × PCLKB
GL_OPERATION	GL_RESET	2 bit times (1 Tq + 16 × PCLKB + 2 DLL clock cycles)
GL_OPERATION	GL_HALT	3 CAN frames (1 CAN frame + 3424 × PCLKB)*1 *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.
Note 2. Exit GL_SLEEP mode only when the GSR.RAMST flag is cleared.
Note 3. Tq, CAN frame and bit times are related to the individual channels. For the maximum transition time, the channel with the lowest bit rate must be used.

33.3.3.2 Timing of Channel Mode Change

Table 33.15 shows the transition time for the Channel mode changes.

Table 33.15 Maximum Transition Time for the Channel Mode

From	To	Maximum Transition Time
CH_SLEEP	CH_RESET	3 × PCLKB
CH_RESET	CH_SLEEP	3 × PCLKB
CH_RESET	CH_HALT	3 bit times (1 CAN bit + 2 Tq + 8 × PCLKB + 2 DLL clock cycles)
CH_RESET	CH_OPERATION	4 bit times (2 CAN bits + 1 TSEG1 + 12 × PCLKB + 2 DLL clock cycles)
CH_HALT	CH_RESET	2 bit times (1 Tq + 10 × PCLKB + 2 DLL clock cycles)
CH_HALT	CH_OPERATION	4 bit times (< 4 CAN bits)*3
CH_OPERATION	CH_RESET	2 bit times (1 Tq + 10 × PCLKB + 2 DLL clock cycles)
CH_OPERATION	CH_HALT	2 CAN frames (1 CAN frame + 13 CAN bits)*1 *2

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CHCR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.

Note 3. In general, if the bit rate prescaler value (NBCR.BRP[9:0]) is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the Tq clock, and new BRP value is captured when the counter reaches the value 0.

33.4 Initialization of CANFD Module

Before starting CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data bit rate)
- Bit Rate setting (nominal and data bit rate)
- CANFD setting
- Acceptance Filter setting (configuration of Acceptance Filter List)
- Receive FIFO and Transmit FIFO setting
- CAN operating mode setting.

33.4.1 Initialization of CAN Clock, Bit Timing and Bit Rate

33.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

1. Each segment setting

- SS = Fixed to 1 Tq
  TSEG1 = 2 Tq to 256 Tq (NBCR), 2 Tq to 32 Tq (DBCR)
  TSEG2 = 2 Tq to 128 Tq (NBCR), 2 Tq to 16 Tq (DBCR)
  SJW = 1 Tq to 128 Tq (NBCR), 1 Tq to 16 Tq (DBCR)
  SS + TSEG1 + TSEG2 = 8 Tq to 385 Tq (NBCR), 5 Tq to 49 Tq (DBCR)

2. Limitations on TSEG1, TSEG2 and SJW
   NBCR register: TSEG1 > TSEG2 ≥ SJW
   DBCR register: TSEG1 ≥ TSEG2 ≥ SJW

Table 33.16 shows an example of how to set the bit timing to achieve the required sample point settings.

Table 33.16 Bit Timing Examples

1 Bit	Set Value (Tq)				Sample Point (%)
	SS	TSEG1	TSEG2	SJW	
5 Tq	1	2	2	1	60.00
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
12 Tq	1	8	3	1	75.00
	1	9	2	1	83.33
15 Tq	1	10	4	1	73.33
	1	11	3	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
24 Tq	1	15	8	1	66.66
	1	16	7	1	70.83
50 Tq	1	39	10	4	80.00

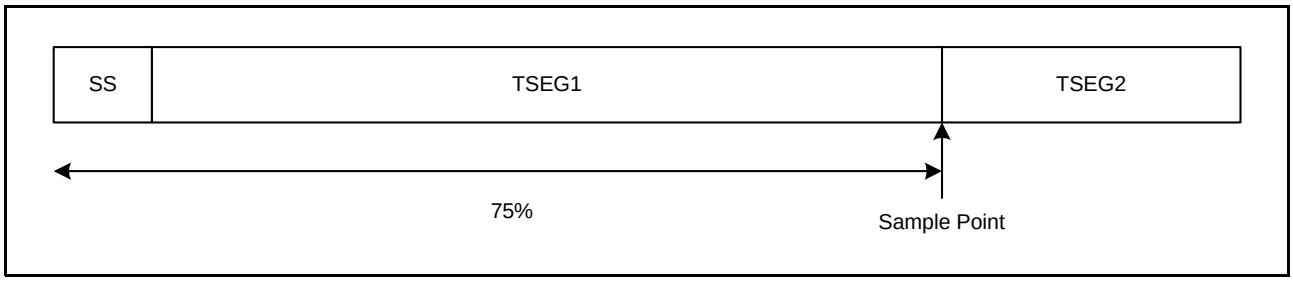


Figure 33.14 The Sample Point (in Case of 75%)

33.4.1.2 Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured using the NBCR and DBCR registers.

Figure 33.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (Tq), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the bit rate prescaler (nominal and data bit rate).

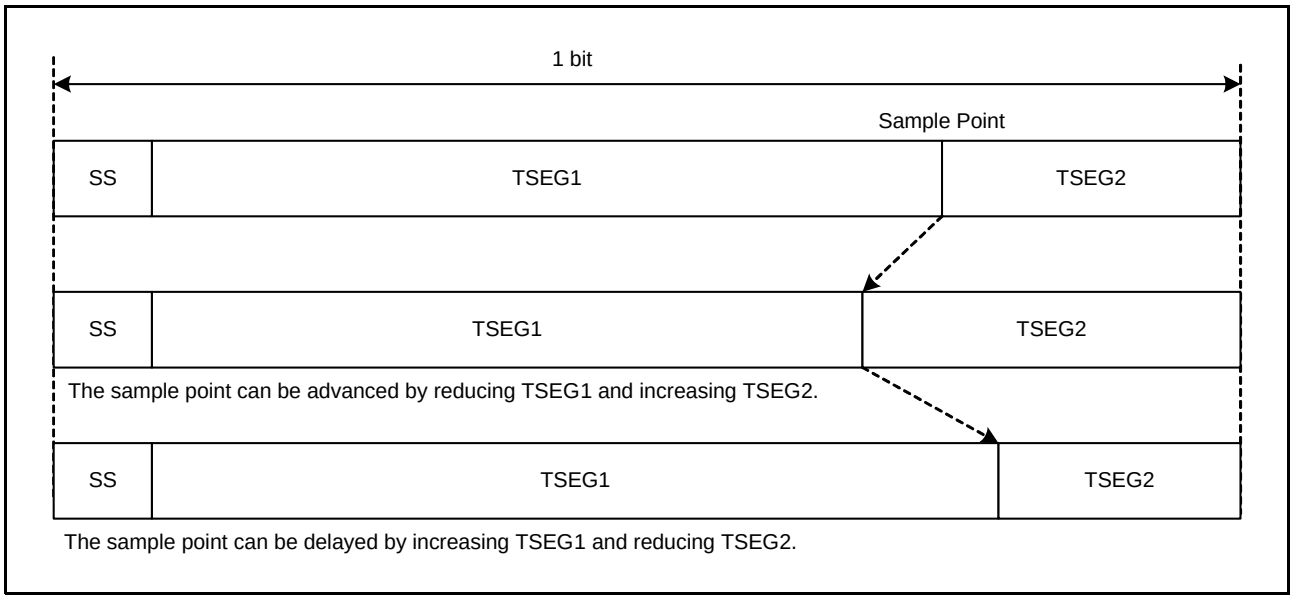


Figure 33.15 Segment Composition of A Bit and The Sample Point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in

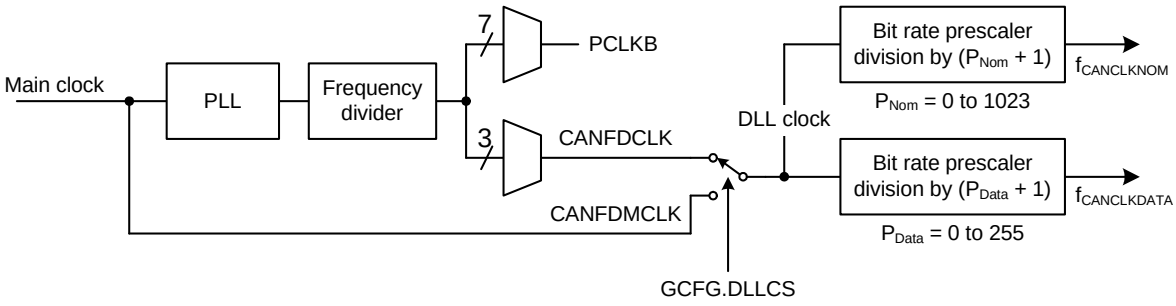
the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.

4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 33.15 shows an example of a typical sample point.

33.4.1.3 Bit Rate

The CAN communication clocks are generated by dividing the operating clock for data link layer (DLL clock). The DLL clock can be selected from either the internal clock (CANFDCLK) or the external clock (CANFDMCLK).
Figure 33.16 shows a block diagram of the circuit that generates the CAN communication clock.



Main clock:	External oscillator clock
CANFDCLK:	Clock equivalent to peripheral module clock
DLL clock:	Data Link Layer clock DLL clock ≤ PCLKB
PCLKB:	CANFD module system clock
P_{Nom}, P_{Data}	Value selected by the NBCR.BRP[9:0] bits and DBCR.BRP[7:0] bits
$f_{CANCLKNOM}$:	CAN nominal communication clock, f(CANCLKNOM) = (DLL clock) / (P_{Nom} + 1)
$f_{CANCLKDATA}$:	CAN data communication clock, f(CANCLKDATA) = (DLL clock) / (P_{Data} + 1)

Figure 33.16 Block Diagram of the Circuit that Generates the CAN Communication Clock

The bit rate is determined by the DLL clock frequency, the division value of the bit rate prescaler (P + 1), and the number of Tq per bit.

$$\text{Bit rate} = \frac{\text{DLL clock frequency}}{\text{Number of Tq per bit} \times (P + 1)} = \frac{\text{CAN communication clock frequency}}{\text{Number of Tq per bit}}$$

Table 33.17 lists examples of setting the nominal bit rate for Classical CAN frame.

Table 33.17 Nominal Bit Rate Setting Examples for Classical CAN Frame

Bit Rate	DLL Clock Frequency																	
	60 MHz		40 MHz		32 MHz		30 MHz		24 MHz		20 MHz		16 MHz		10 MHz		8 MHz*1	
	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1
1 Mbps	10 Tq 15 Tq	6 4	8 Tq 20 Tq	5 2	8 Tq 16 Tq	4 2	10 Tq 15 Tq	3 2	8 Tq 12 Tq 24 Tq	3 2 1	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1	10 Tq	1	8 Tq	1
500 kbps	10 Tq 15 Tq 20 Tq	12 8 6	8 Tq 20 Tq	10 4	8 Tq 16 Tq	8 4	10 Tq 15 Tq 20 Tq	6 4 3	8 Tq 12 Tq 24 Tq	6 4 2	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1
250 kbps	10 Tq 15 Tq 20 Tq	24 16 12	8 Tq 20 Tq	20 8	8 Tq 16 Tq	16 8	10 Tq 15 Tq 20 Tq	12 8 6	8 Tq 12 Tq 24 Tq	12 8 4	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2
125 kbps	10 Tq 15 Tq 20 Tq	48 32 24	8 Tq 20 Tq	40 16	8 Tq 16 Tq	32 16	10 Tq 15 Tq 20 Tq	24 16 12	8 Tq 12 Tq 24 Tq	24 16 8	10 Tq 20 Tq	16 8	8 Tq 16 Tq	16 8	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4
83.3 kbps	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	90 72 60 48 36 30	8 Tq 12 Tq 16 Tq 24 Tq	60 40 30 20	8 Tq 12 Tq 16 Tq 24 Tq	48 32 24 16	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	45 36 30 24 18 15	8 Tq 12 Tq 16 Tq 24 Tq	36 24 18 12	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	30 24 20 16 15 12 10	8 Tq 12 Tq 16 Tq 24 Tq	24 16 12 8	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	15 12 10 8 6 5	8 Tq	12
33.3 kbps	10 Tq 12 Tq 15 Tq 20 Tq	180 150 120 90	8 Tq 12 Tq 16 Tq 20 Tq 24 Tq	150 100 75 60 50	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	120 96 80 64 60 48 40	10 Tq 12 Tq 15 Tq 20 Tq	90 75 60 45	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	90 72 60 48 45 36 30	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	75 60 50 40 30 25	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	60 48 40 32 30 24 20	10 Tq 12 Tq 15 Tq 20 Tq	30 25 20 15	8 Tq	30

Note 1. Minimum frequency to achieve a nominal bit rate of 1 Mbps.

For optimum clock tolerance in networks using the CAN FD frame, the length of the time quantum should be the same in nominal bit time and in data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
In addition, do not set the DBCR.BRP[7:0] bits greater than 1 when using transceiver delay compensation.
Table 33.18 lists examples of setting the nominal and data bit rate for CAN FD frame.

Table 33.18 Nominal and Data Bit Rate Setting Examples for CAN FD Frame

Bit Rate		DLL Clock Frequency											
		60 MHz			40 MHz			30 MHz			20 MHz		
		No. of Tq			No. of Tq			No. of Tq			No. of Tq		
Nominal	Data	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1
1 Mbps	8 Mbps	—	—	—	40 Tq	5 Tq	1	—	—	—	—	—	—
1 Mbps	5 Mbps	60 Tq	12 Tq	1	40 Tq	8 Tq	1	30 Tq	6 Tq	1	—	—	—
1 Mbps	4 Mbps	60 Tq	15 Tq	1	40 Tq	10 Tq	1	—	—	—	20 Tq	5 Tq	1
1 Mbps	2 Mbps	60 Tq	30 Tq	1	40 Tq	20 Tq	1	30 Tq	15 Tq	1	20 Tq	10 Tq	1
500 kbps	2 Mbps	120 Tq	30 Tq	1	80 Tq	20 Tq	1	60 Tq	15 Tq	1	40 Tq	10 Tq	1

33.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate

Figure 33.17 shows the procedure for setting the bit timing and the bit rate.
These settings should be configured when the CAN channel is in CH_RESET mode.
The bit rate must be configured before moving to the channel communication state. Otherwise the mode does not switch correctly.

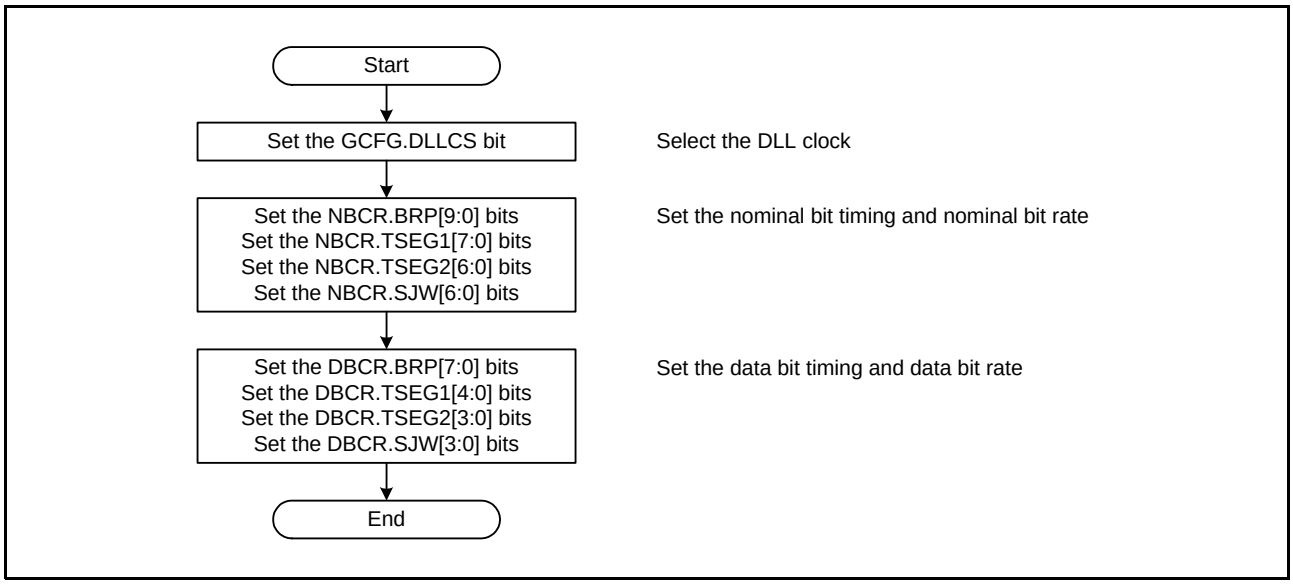


Figure 33.17 Procedure for Setting the Bit Timing and Bit Rate

33.4.1.5 Transceiver Delay Compensation

When a high bit rate is used such as 5 to 8 Mbps for the data phase, the transceiver delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN FD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.
There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN FD frames. This is derived from the Transceiver Delay Compensation Result bit (FDSTS.TDCR[7:0]) as shown in Figure 33.18.
The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

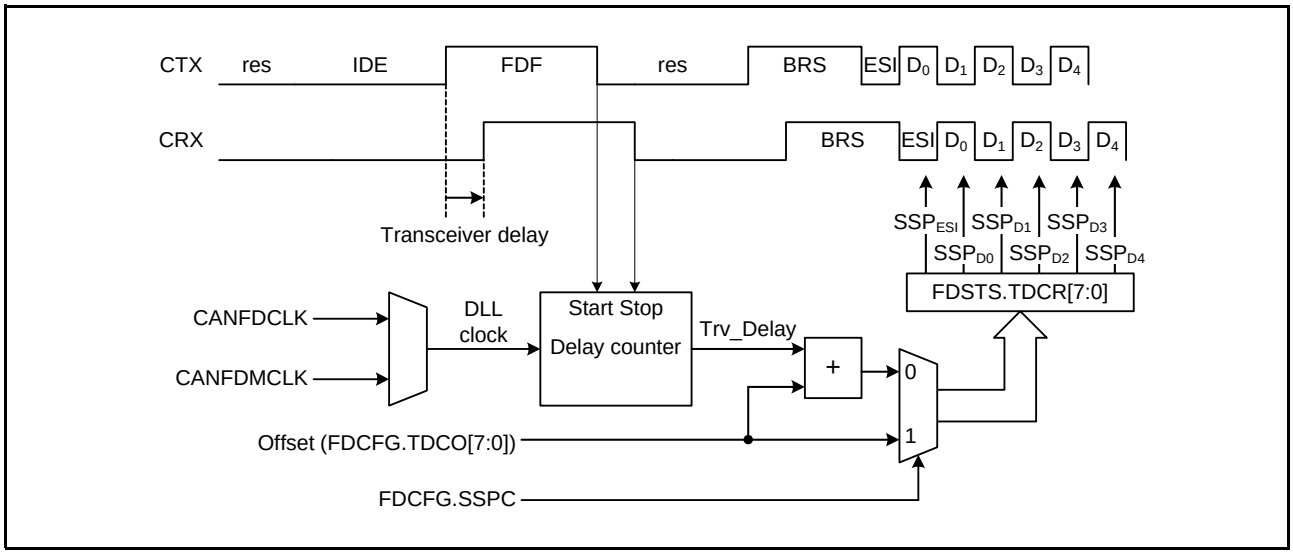


Figure 33.18 Transceiver Delay Compensation

The measured Trv_Delay is based on the number of DLL clock cycles. It counts up by one for each start clock until the dominant value can be observed at the CRX0 pin. Figure 33.19 shows the measurement example. Trv_Delay is counted up to 127 on each DLL clock.

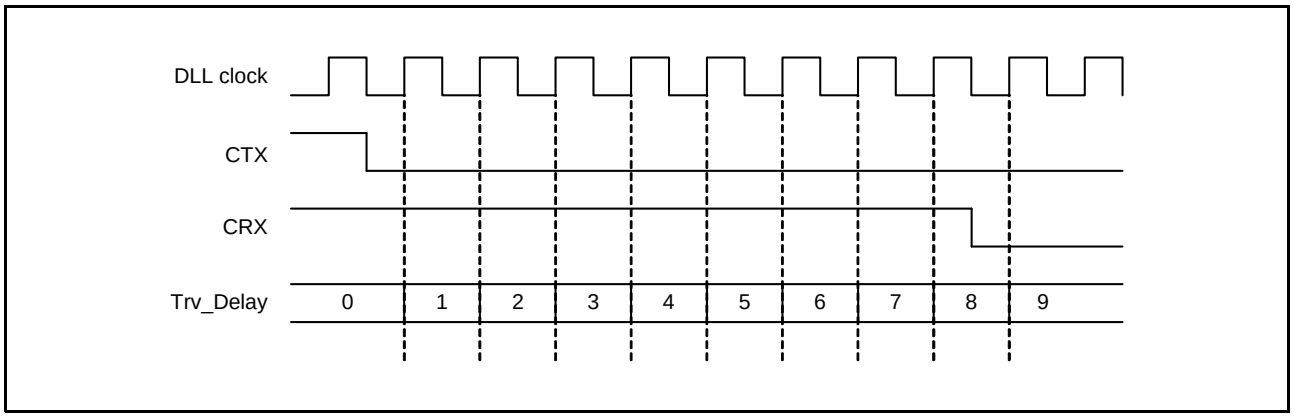


Figure 33.19 Trv_Delay Measurement Example

The SSP is calculated by taking the result from the FDSTS.TDCR[7:0] bits and rounding the value down to the nearest integer number of data Tq.
Figure 33.20 shows the positioning of the secondary sample point (SSP). When the FDCFG.SSPC bit is set to 0, the SSP is equal to the Trv_Delay (measured delay) + FDCFG.TDCO[7:0], rounded down to the nearest integer number of Tq. Normally, the TDCO[7:0] value has the magnitude of SS + TSEG1 in the data phase to position the SSP to a theoretical location of the sample point.
If the FDCFG.SSPC bit is set to 1, the SSP is defined by the FDCFG.TDCO[7:0] bits. If the DBCR.BRP[7:0] bits are greater than 00h, the value is also rounded down to the nearest integer number of Tq.

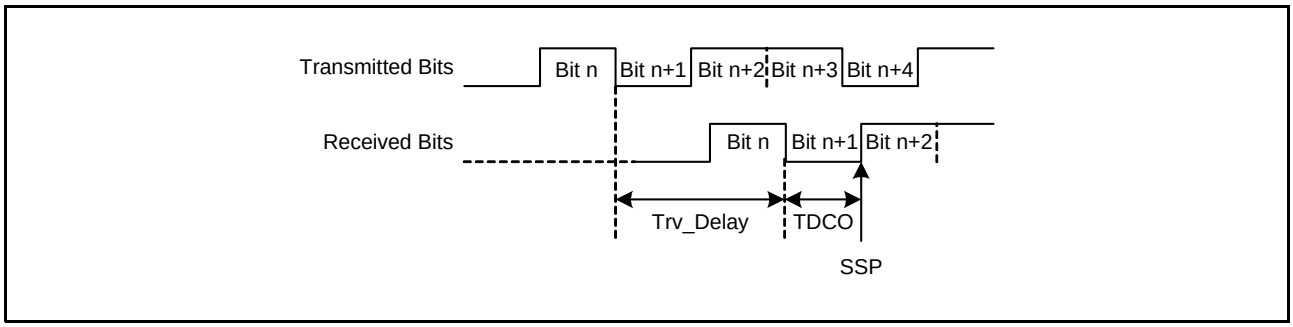


Figure 33.20 Positioning of the Secondary Sample Point (SSP)

The maximum delay time (Trv_Delay + TDCO[7:0]) which can be compensated by the CANFD module is (6 data bits – 2 DLL clock). The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.
If different values are used for the NBCR.BRP[9:0] bits and DBCR.BRP[7:0] bits, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 33.21.
The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
The bit rate can be changed by selecting different settings for the time segments. The nominal bit rate can be set from 8 to 385 Tqs and the data bit rate from 5 to 49 Tqs.

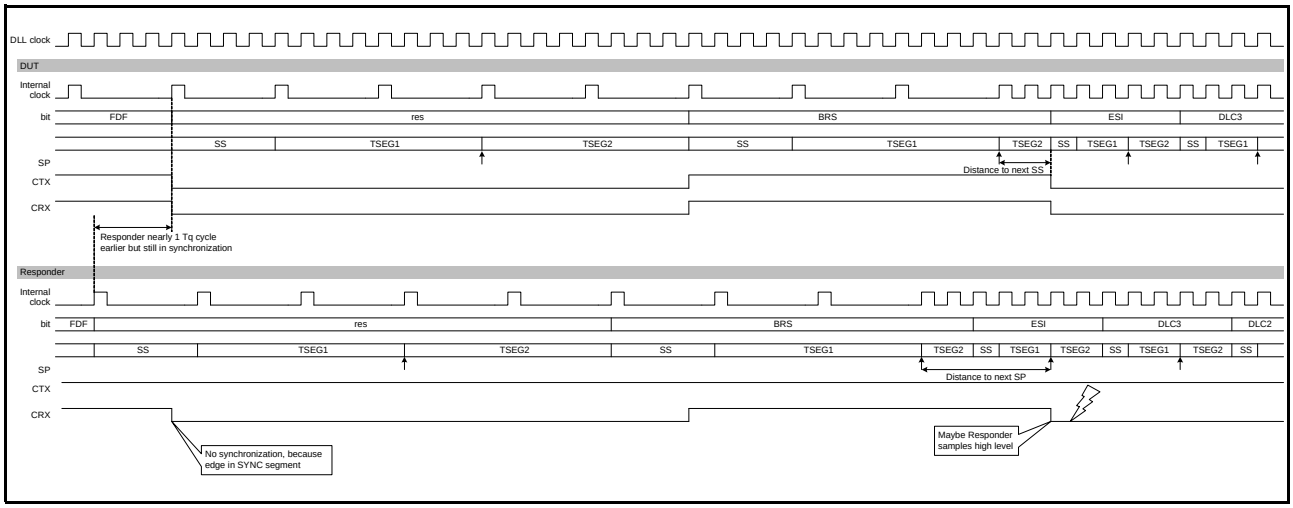


Figure 33.21 Loss of Synchronization Between Two CAN Nodes

The transceiver delay compensation measurement result is updated at the falling edge from FDF bit to res bit when configured accordingly (FDCFG.TDCE = 1, FDCFG.SSPC = 0).
Figure 33.22 shows the read flow to get the measured transceiver delay compensation result.

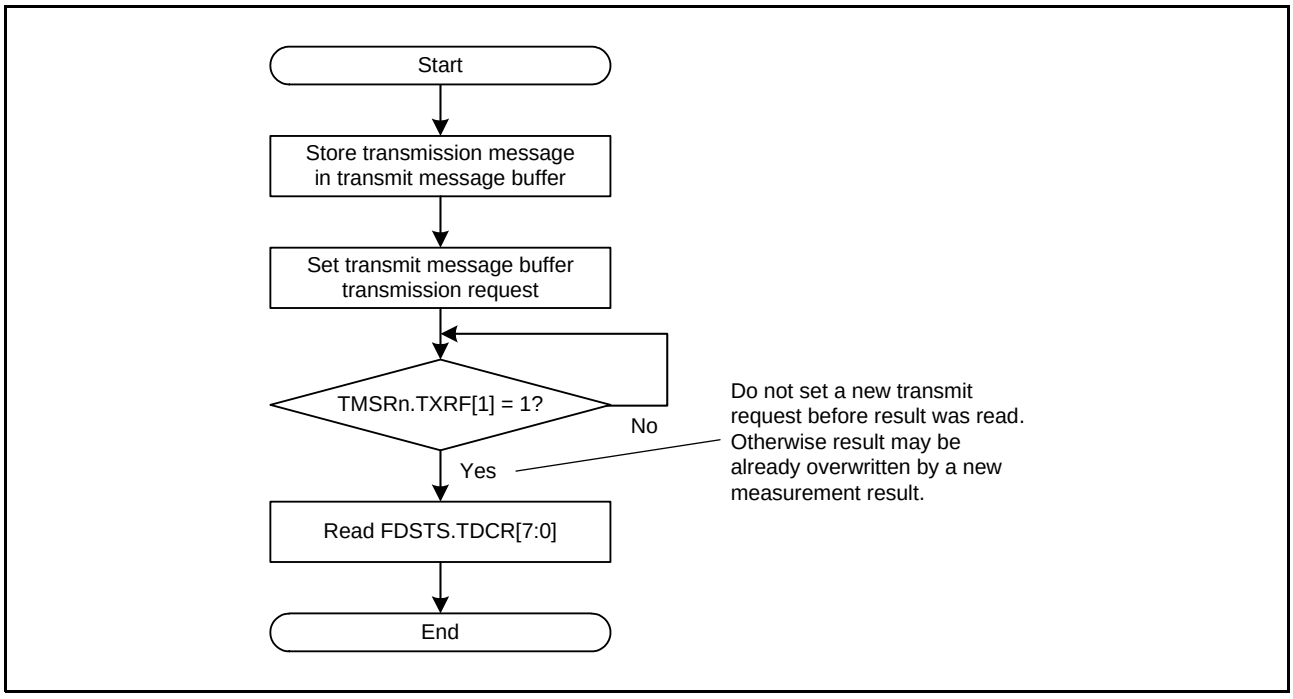


Figure 33.22 Transceiver Delay Compensation Result Read Flow

33.4.2 CANFD Module Configuration after a Reset

After release from a MCU reset or after setting and clearing the GRCR.SRST bit, the CANFD module enters GL_SLEEP mode automatically.

To configure the CANFD module settings, exit GL_SLEEP mode by setting the GCR.SLPRQ bit to 0.

After release from a MCU reset, the CANFD module starts RAM initialization. At this time, the GSR.RAMST flag is automatically set to 1 indicating that the CANFD module logic is initializing the RAM.

After RAM initialization is complete, this bit is automatically set to 0.

RAM initialization is necessary to prevent indefinite data in RAM from detecting the wrong ECC error after release from a MCU reset.

Do not access (read or write) other CANFD registers until the RAM initialization is complete and the GSR.RAMST flag is becomes 0.

Before going to communication mode, the Acceptance Filter List and message FIFO buffers must be configured. In addition, the CAN channel setting such as CAN bit timing must be configured. To make this configuration, the CAN channel must be released from CH_SLEEP mode and must be set for communication in CH_RESET mode (configuration mode).

Figure 33.23 shows the configuration procedure. For details about each step, refer to section 33.5, Filtering Using Acceptance Filter List (AFL), section 33.6, FIFO Buffers and Message Buffer Configuration, section 33.10, Interrupts and DTC/DMA Requests, and section 33.4.1.3, Bit Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting the GRCR.SRST bit to 1.

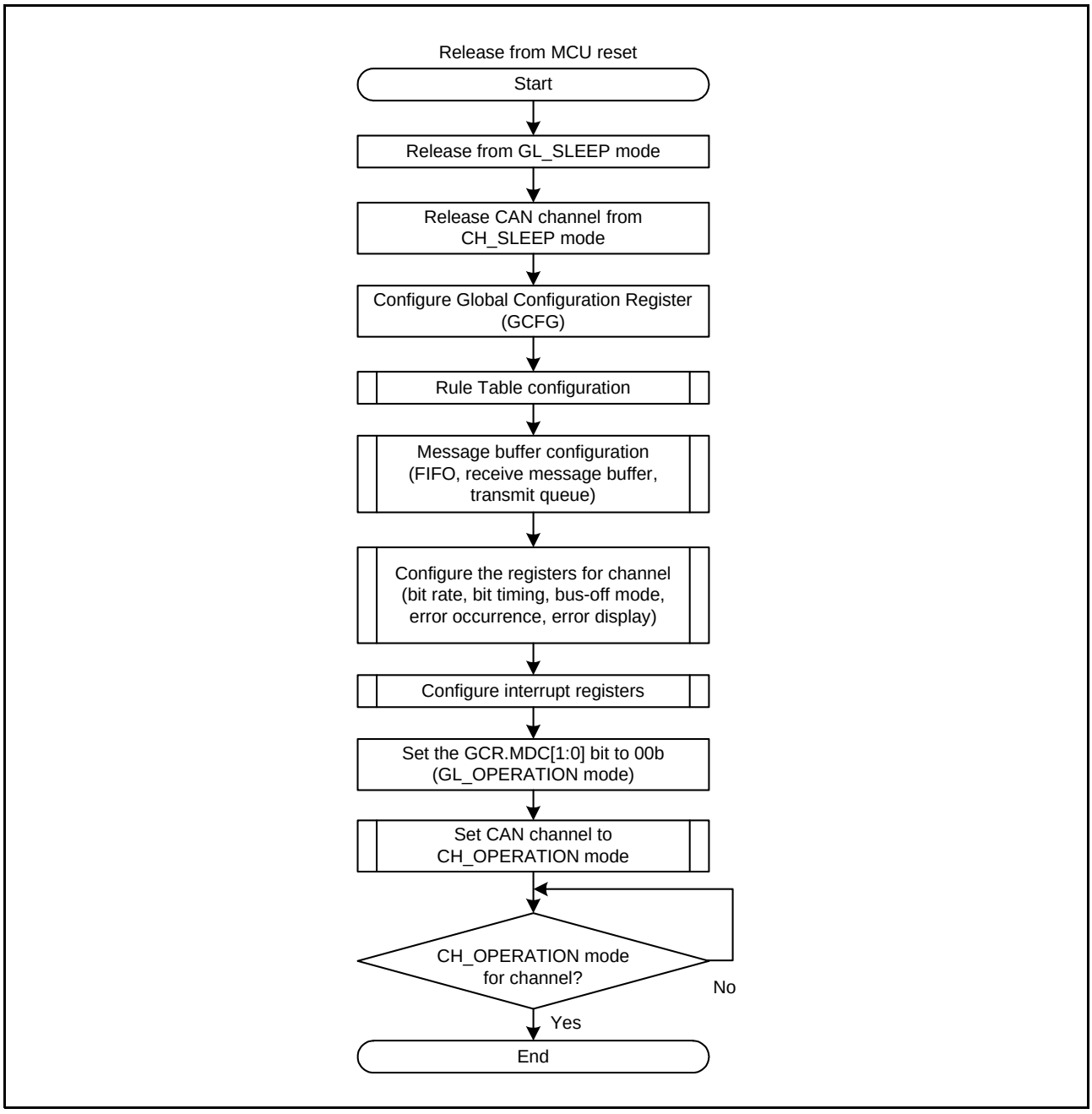


Figure 33.23 Configuration Procedure After A MCU Reset

33.5 Filtering Using Acceptance Filter List (AFL)

The CANFD module allows you to use the Acceptance Filter List (AFL) to filter message acceptance for channels. Each entry of the AFL defines a filter rule for received messages.
The followings are performed based on the AFL entries:

- Acceptance filtering based on the RTR value, IDE value, and ID value of received message
- DLC filtering based on DLC value of received message
- Payload overflow based on the GCFG.OMRC bit
- Storage of accepted messages in the specified message buffer/FIFO buffer
- Attaching a 16-bit pointer to the stored messages, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages.

The CANFD module allows a maximum of 32 AFL entries.

33.5.1 Acceptance Filtering Process

Acceptance filtering matches each AFL entry with the received message. Matching starts with the lowest AFL entry number.
The AFL search stops when the received message ID matches the specified ID/mask combination, or when the received message ID has been matched against all defined AFL entries. If no match occurs, then the received message is discarded. No notification is given to the application in this case.

33.5.2 DLC Filtering Process

DLC filtering is performed for each accepted message if DLC check is enabled (GCFG.DCE bit = 1). If the DLC value of the received message is equal to or higher than the DLC value specified for the AFL entry whose ID matches in the Acceptance Filtering process (the matching AFL entry), the DLC check is passed.
If DLC replacement (GCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0000b and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination receive message buffer (RMBn) or FIFO buffer. If DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are not stored in the RMBn/FIFO buffer. These excess data bytes are stored as 00h in the RMBn/FIFO buffer.
If DLC replacement is enabled and the DLC value specified in the matching AFL entry is 0000b (DLC filter function is disabled), the DLC value of the received message is stored in the RMBn/FIFO buffer.
If DLC replacement is disabled (GCFG.DRE bit = 0) and DLC check passes, the DLC value of the received message is stored in the RMBn/FIFO buffer. If the DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are also stored in the RMBn/FIFO buffer.
If DLC value of the received message is less than the DLC value specified in the matching AFL entry, DLC check fails. In this case, the received message is discarded and is not stored in anywhere.
Additionally, when the DLC check fails, the GESR.DEDF flag is set to 1. If interrupts are enabled, an error interrupt is also generated. If the DLC check fails, the DLC replacement setting has no effect.

33.5.3 Message Storage

If a received message has passed both acceptance filtering and DLC filtering, the message is stored in receive message buffers 0 to 31, receive FIFO 0, 1, or common FIFO 0 configured in receive FIFO mode.
This message storage target information is also defined in the AFL entry. Do not set a target at the AFL entry which is not configured.
Up to two message storage destinations can be specified. Do not specify more than 3 destinations.

33.5.4 Payload Overflow Process

There is a protection mechanism in case the received message contains data with a payload size that is longer than the size that can be stored in the storage (RMCR.PLS[2:0], RFCR0.PLS[2:0], RFCR1.PLS[2:0], or CFCR0.PLS[2:0]).
If GCFG.OMRC = 0 (message is discarded), the message with data byes that exceed the specified payload size are discarded and not stored. In this case, even if the FIFO is full, the corresponding FMLSR.RFML0, RFML1, or CFML0 flag is not set to 1.
If GCFG.OMRC = 1 (cut to specified size), only data bytes that exceed the specified payload size are discarded. In this case, if the FIFO is full, the corresponding FMLSR.RFML0, RFML1, or CFML0 flag is set to 1 (message lost has occurred).
Depending on the GCFG.DRE bit setting, either the DLC value of the received message or the DLC value specified in the AFL entry is stored.
Regardless of the GCFG.OMRC bit setting, the GESR.PODF flag is set to 1 if a payload overflow condition is detected.
The DLC filtering is performed before the payload overflow process. So for one reception frame, only one flag can be set to 1 at the same time with the GESR.DEDF flag or GESR.PODF flag.

33.5.5 Allocation of AFL Entries

The number of AFL entries (number of rules) can be configured using the AFCFG.RN0[5:0] bits (refer to Figure 33.24).

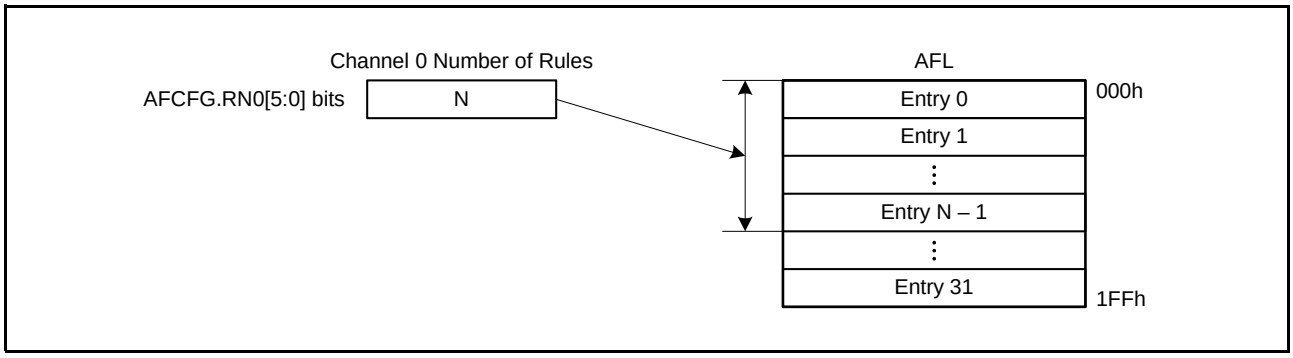


Figure 33.24 Configuration of AFL

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 32.
The CANFD module does not flag errors related to the configuration of the AFL.

33.5.6 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.
Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier field (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit matches this filed with the identifier field of the received message (29 bits of identifier field can be masked individually, refer to the description of Mask for Identifier field below).
- IDE bit:
  Acceptance filter unit matches this bit with the IDE bit of the received message and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, refer to the description of Mask for IDE bit below).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, refer to the description of Mask for RTR bit below).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier field (29 bits):
  Each bit in the identifier mask field can mask the corresponding identifier bit in the AFL entry (refer to Figure 33.25).
- Mask for IDE bit:
  If this Mask bit masks the IDE bit of an AFL entry, the AFL entry can accept messages in both Standard Identifier format and Extended Identifier format. For messages in Standard Identifier format, the Standard Identifier part of the AFL entry is compared, and for messages in Extended Identifier format, the Extended Identifier part of the AFL entry is compared.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of an AFL entry, the AFL entry can accept frame formats in both data frame and remote frame.
- Pointer (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added when storing a message in the message buffer area and can be used as a support function in the application.
  For example, the pointer information can be used to support the assignment of PDU IDs to the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a message accepted by the related AFL entry. This label is added when storing a message in the message buffer area and can be used as support function in the application.
- DLC field:
  If the DLC value of the received message is equal to or higher than the value set in this field, the DLC check is passed.
  If the DLC value of an AFL entry is set to 0000b, the DLC filtering for this entry is effectively disabled (all accepted messages pass DLC filtering).

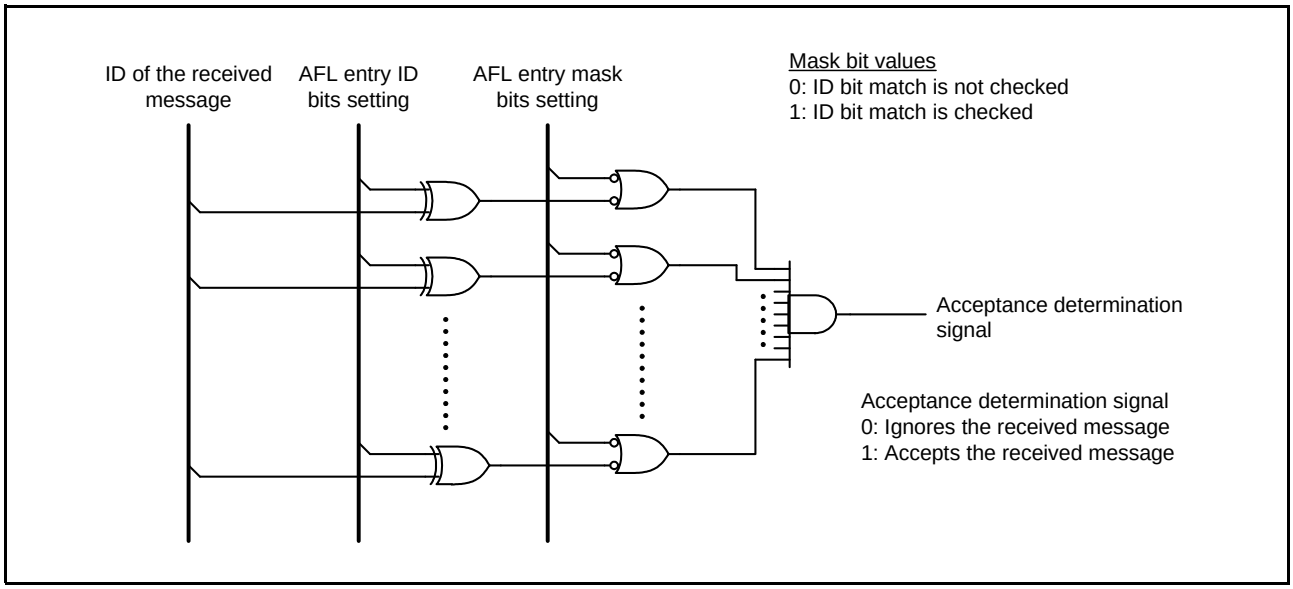


Figure 33.25 Acceptance Filter Function

Each AFL entry contains the following information for processing the received messages:

- Message buffer number of the receive message buffer used as the storage destination for received messages
- The Destination Message Buffer Setting Enable bit to specify the receive message buffer as the storage destination for received messages
- The FIFO Destination Enable bit to specify the FIFO as the storage destination for received messages.

There is no protection function for storing messages. Therefore, the FIFOFIFO Destination Enable bit must be set carefully.

33.5.7 Entering Entries in the AFL

One complete entry can be entered into AFL via the following registers:

- AFLn.IDR register: First part of the AFL entry
- AFLn.MASK register: Second part of the AFL entry
- AFLn.PTR0 register: Third part of the AFL entry
- AFLn.PTR1 register: Fourth part of the AFL entry.

These 16 sets of registers make up one page of AFL entries. There are 32 entries in the CANFD module, and all of these entries can be accessed by specifying the page with the AFCR.PAGE bit. The AFL should only be configured in CH_RESET or CH_HALT mode. Table 33.19 shows pages linked to the AFL entries.

Table 33.19 Pages Linked to the AFL Entries

Page	Linked AFL Entries
Page 0	Entry 0 to 15
Page 1	Entry 16 to 31

AFL access control is performed using the AFCR register (Figure 33.26). This register has the following bits:

- The PAGE bit for selecting the AFL page number
- The AFLWE bit that enables or disables the writing of data to prevent unnecessary write access to the AFL.

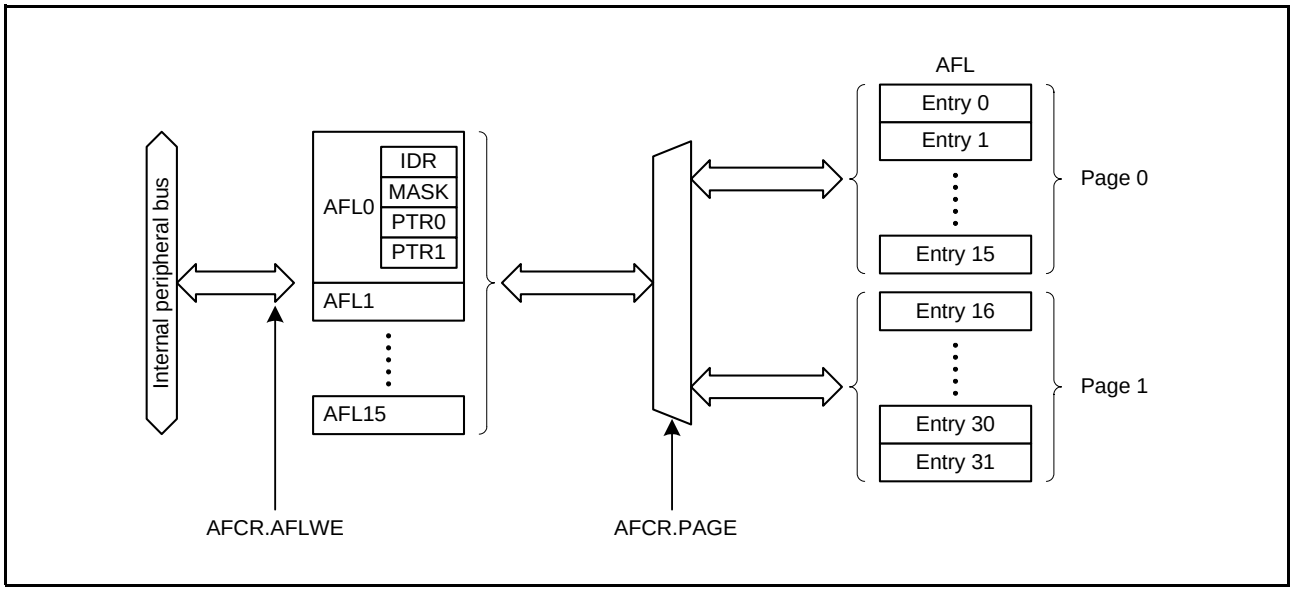


Figure 33.26 AFL Page Access

Follow the flow shown in Figure 33.27 to configure the AFL.
After entering all the entries, writing to the AFL must be disabled to prevent unnecessary write access to the AFL.
If the AFCR.AFLWE bit is set to 0, write protection is enabled during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION).
Even if the AFCR.AFLWE bit is set to 0, reading from AFL is possible during all Global modes (the consistency of AFL contents can be checked during execution).

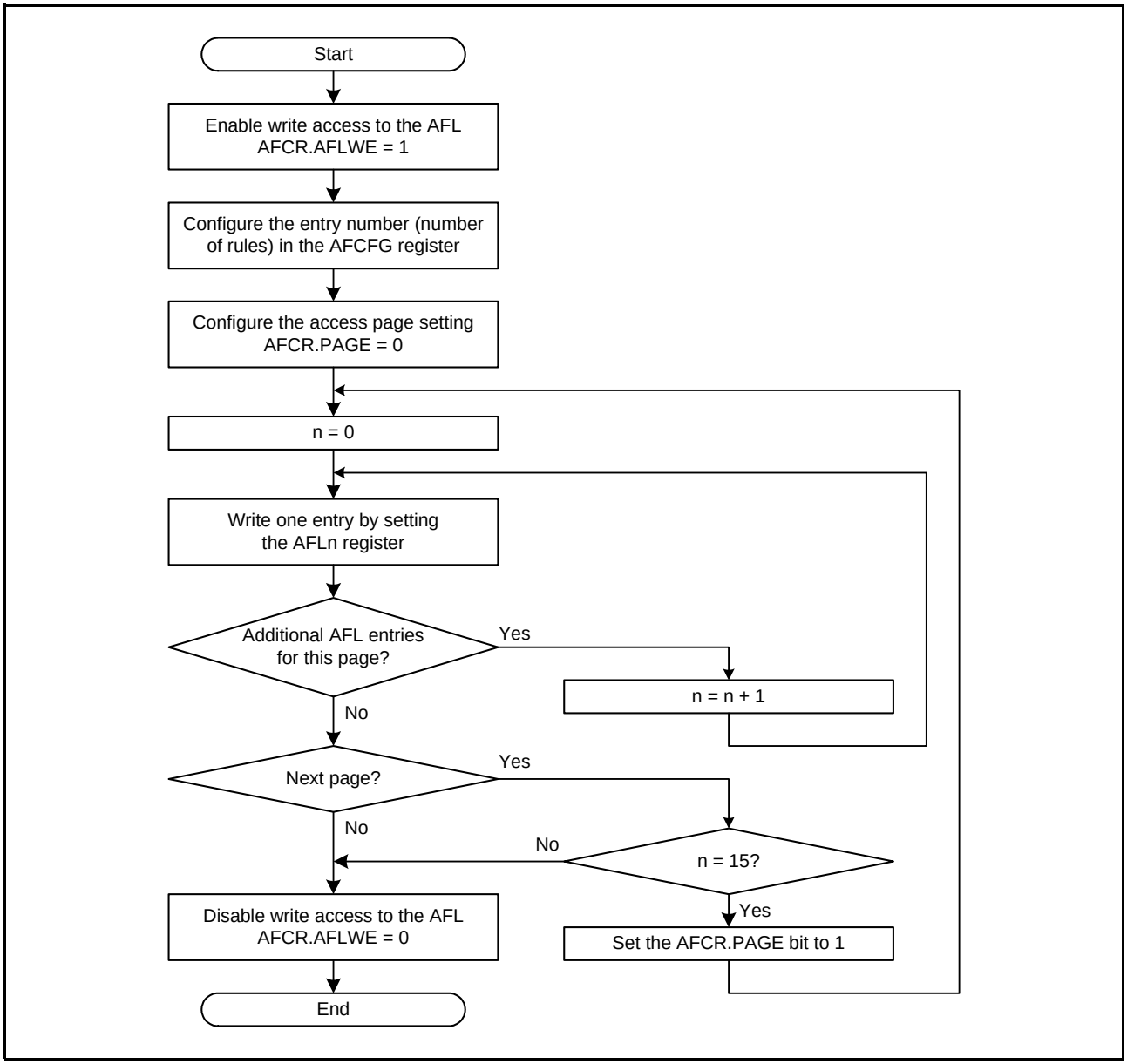


Figure 33.27 AFL Configuration Flow

33.5.8 Loopback Modes

AFL entries with the AFLn.IDR.LPC bit set to 1 are used only in loopback mode (self-test mode 0 or self-test mode 1) or mirror mode. If a message transmitted by another node on the CAN bus is received while in loopback mode, the AFL entry will not be used.
AFL entries with the AFLn.IDR.LPC bit set to 0 are used only for:

- Received messages transmitted by other nodes in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other nodes or the local node in loopback mode.

The mirror mode can be enabled with the GCFG.MME bit. If the message is successfully transmitted when the GCFG.MME bit is 1, the message is stored in the received message buffer or FIFO buffer if there is a matching entry in the AFL. The AFLn.IDR.LPC bit in the matching AFL entry must be set to 1 to store this frame.
If mirror mode and loopback mode are set at the same time, the loopback mode behavior is applied. Table 33.20 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 33.20 Behavior of Acceptance Filter Based on the Loopback Setting in AFL Entry

Mirror Mode (MME Bit)	Loopback Mode (Self-test Mode 0 or Self-test Mode 1)	Channel Mode	LPC Bit	AFL Entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

33.5.9 IDE Masking

The IDE bit set in the AFL entry with the AFLn.MASK.IDEM bit set to 0 is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is selected based on the received IDE bit.
The following lines show the examples.

- The ID and mask fields of an AFL entry x is configured as follows:
  – AFLx.IDR = C0553A20h → IDE = 1, RTR = 1, LPC = 0, ID[10:0] = 220h/ID[28:0] = 0553A20h
  – AFLx.MASK = 0000FFFFh → IDEM = 0. RTRM = 0, IDM[10:0] = 7FFh/IDM[28:0] = 0000FFFFh
- The comparison result for the four different received IDs with AFL entry x is described as follows:
  – If a frame with IDE = 0 and ID = 220h is received, this is considered as a match
  – If a frame with IDE = 0 and ID = 320h is received, this is not a match
  – If a frame with IDE = 1 and ID = 1FFF3A20h is received, this is considered as a match
  – If a frame with IDE = 1 and ID = 08803220h is received, this is not a match.

33.5.10 Updating AFL Entry during Communication

The AFL entry can be updated without disabling CAN communications.
Set the AFL entry number to update to the ignore entry select bits and set the ignore entry enable bit to 1.
The entry number specified here is ignored by AFL matching while the entry is being updated.
Figure 33.28 shows the update flow for an AFL entry.

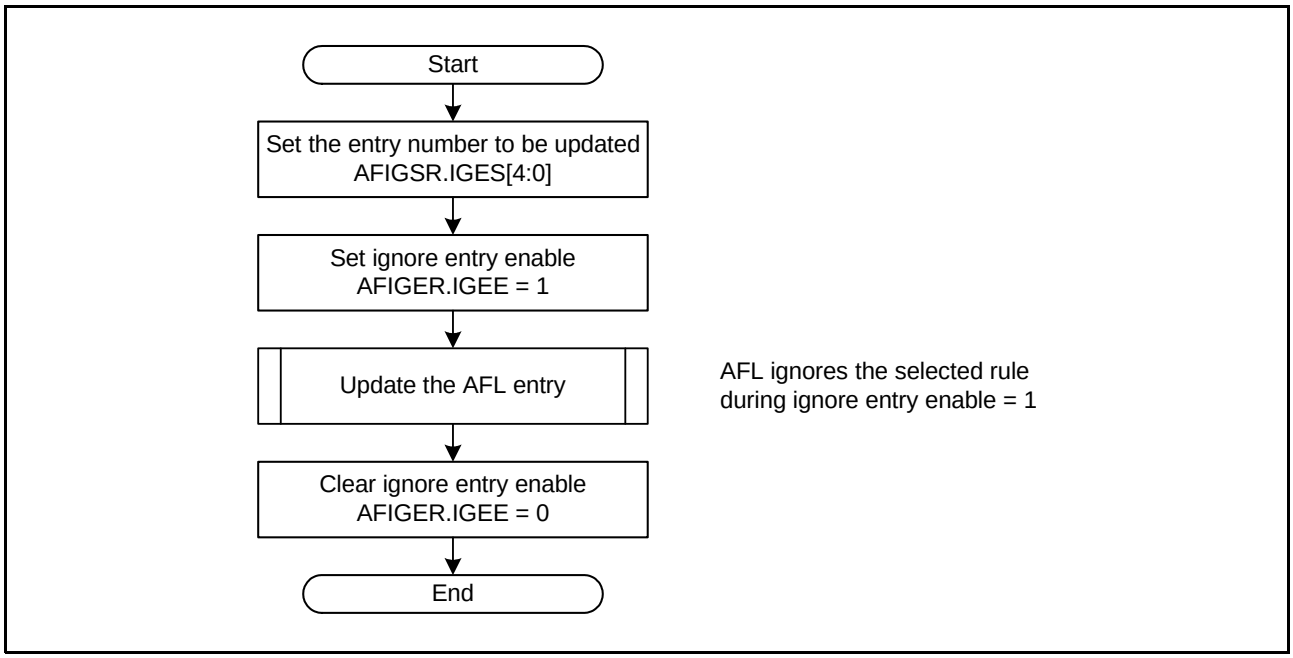


Figure 33.28 Update Flow for An AFL Entry

The method to update an AFL entry is as follows:

1. Set the entry number*1 to be updated to the AFIGSR.IGES[4:0] bit.
2. Set the value 0000C401h (key code and ignore entry enabled) to the AFIGER register.
3. Set the page number which includes the entry to be updated to the AFCR.PAGE bit. Set the AFCR.AFLWE bit to 1.
4. Set the new rule to the AFLn.IDR, AFLn.MASK, AFLn.PTR0, AFLn.PTR1 registers.
5. The AFCR.AFLWE bit is set to 0.
6. Set the value 0000C400h (key code and ignore entry disabled) to the AFIGER register.

Note 1. This entry number is not used for acceptance filtering between (2) and (5).

(1) Example 1: Deleting an Entry

The following describes how to delete entry 3 when the total number of entries is 6.

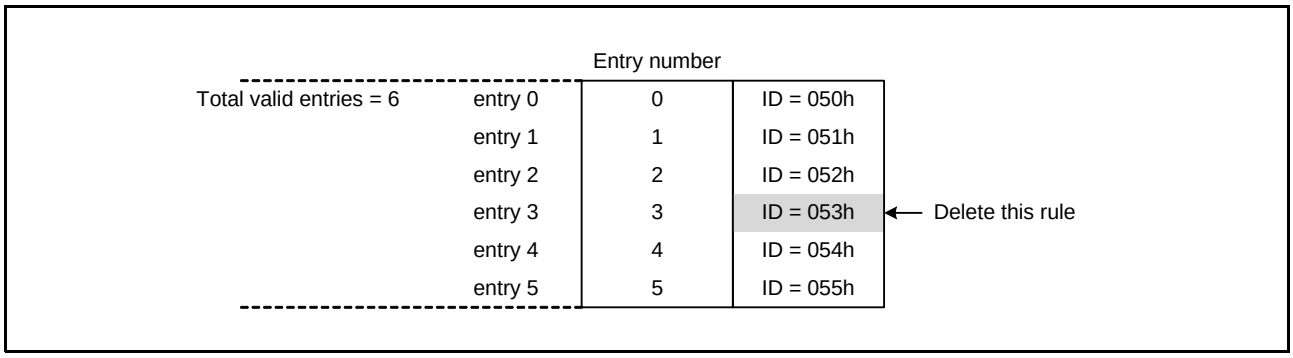


Figure 33.29 Example of Deleting an Entry (Before Deleting Entry 3)

[How to delete an entry]

(1) Set 00000003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the same rule as the previous rule by accessing the AFL3.IDR, AFL3.MASK, AFL3.PTR0, AFL3.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

Entry 3 is now deleted.

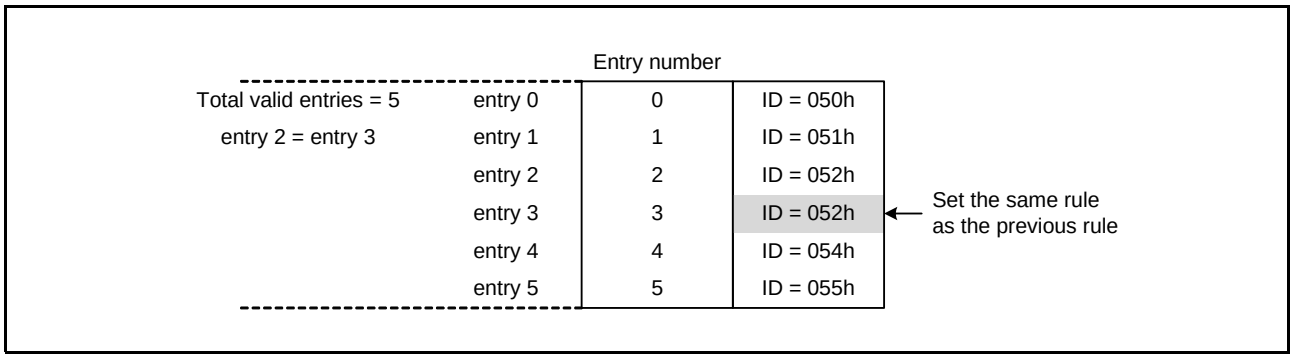


Figure 33.30 Example of Deleting an Entry (After Deleting Entry 3)

(2) Example 2: Adding an Entry (Update Unused Entry)

The following describes how to add a new entry to entry 3 when the total number of entries is 6.

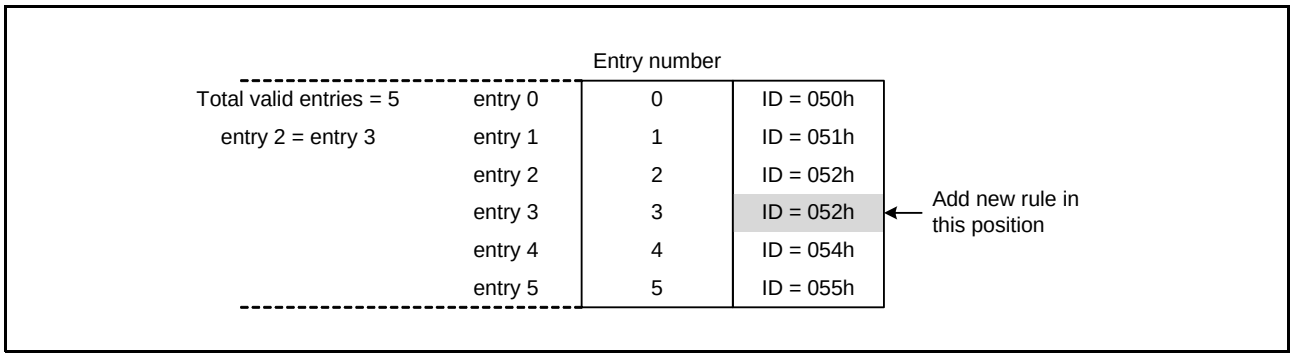


Figure 33.31 Example of Adding an Entry (Before Updating Entry 3)

[How to add an entry]

(1) Set 00000003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the new rule by accessing the AFL3.IDR, AFL3.MASK, AFL3.PTR0, AFL3.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

The new entry is now added.

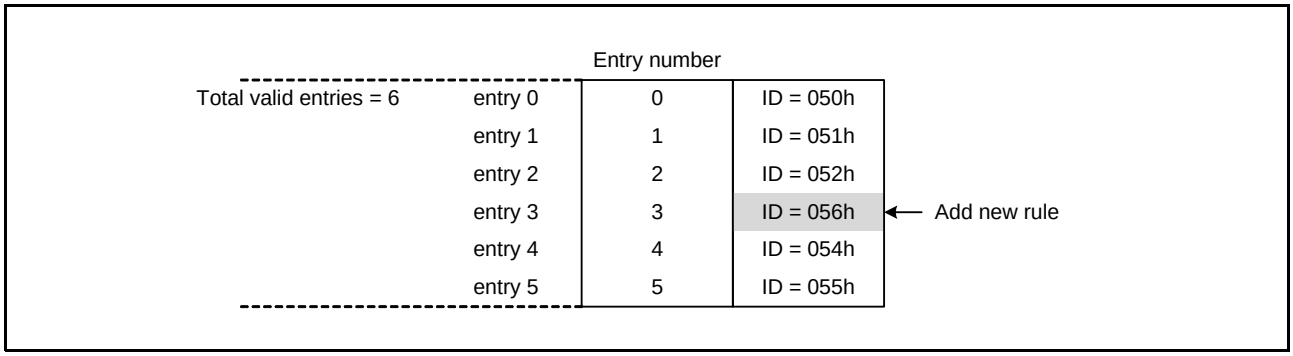


Figure 33.32 Example of Adding an Entry (After Updating Entry 3)

The acceptance filter can use entries in the range of value set in the AFCFG register and entries can be added/deleted within that range. Therefore, the AFCFG register should be set to the maximum number of entries to use.

33.6 FIFO Buffers and Message Buffer Configuration

This section describes the process for configuring the number of receive message buffers, the FIFO buffers, and the transmit message buffers in the CANFD module. The message buffers are mapped as shown in Figure 33.33.

The receive message buffers can be accessed with the RMBn register (n = 0 to 31).

The receive FIFOs can be accessed with the RFBn register (n = 0, 1).

The common FIFO can be accessed with the CFB0 register.

If the common FIFO is configured in transmit mode, only data can be written to the FIFO buffer by the CFB0 register.

If the common FIFO is configured in receive mode, only data can be read by the CFB0 register.

The transmit message buffers can be accessed with the TMBn register (n = 0 to 3).

If unused message buffer is read, it is read as undefined.

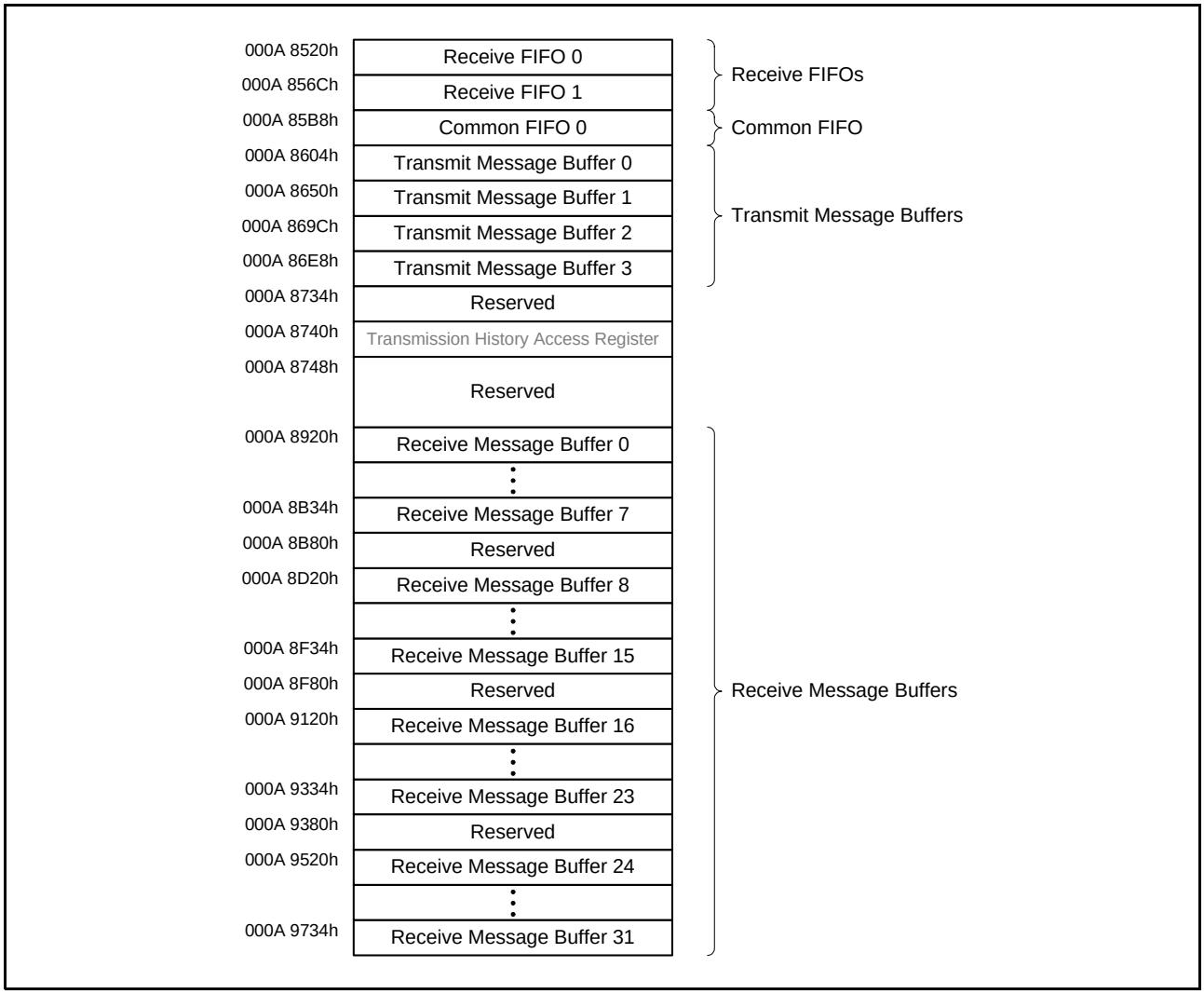


Figure 33.33 Message Buffer Configuration

33.6.1 Receive Message Buffers

In CANFD module, the received frames can be stored in receive message buffers based on the configuration of the AFL entries.
Additionally, the number of receive message buffers required by the system can be selected from 0 to 32.

33.6.1.1 Receive Message Buffer Configuration

The number of receive message buffers in CANFD module can be configured by writing to the RMCR.NMB[5:0] bits. Set the number of message buffers in the range 0 (no receive message buffers) to 32. Do not set a value larger than this.
The AFL entries for routing the received messages to receive message buffers must be configured to match the requirements of the system.
The AFL entries must also be configured properly, and an AFL entry for receive message buffers should not exceed the number of message buffers configured in the RMCR.NMB[5:0] bits.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The payload size of the receive message buffer can be configured with the RMCR.PLS[2:0] bits. The default size is 8 bytes and the maximum size is 64 bytes.
If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

33.6.2 FIFO Buffers

The CANFD module provides a FIFO buffers to store each receive/transmit frame.
There are two receive-only FIFOs, but common FIFO can be configured to store messages for transmission or reception.
These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- FIFO depth
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the transmit FIFO

If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

33.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.
The total number of FIFO buffers is three (two receive FIFOs + one common FIFO).

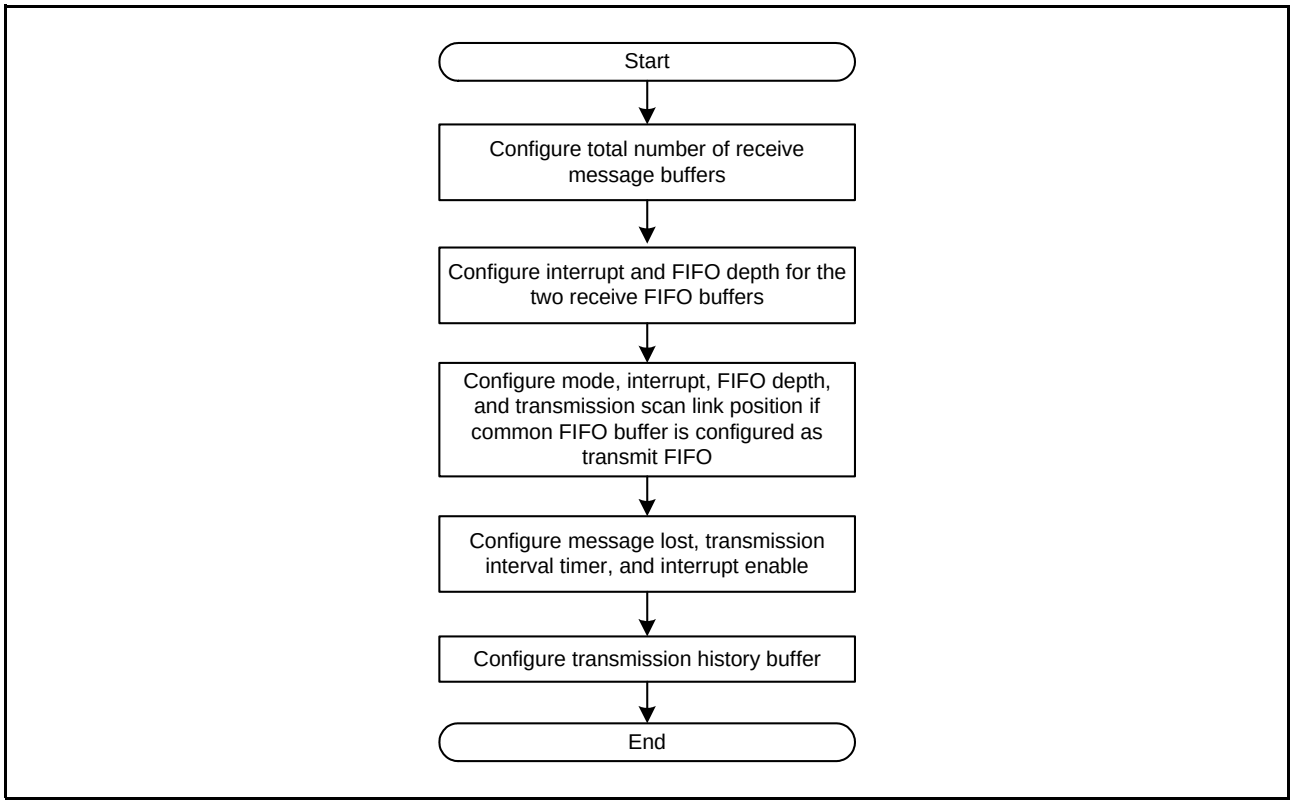


Figure 33.34 FIFO Buffer Configuration Flow in CANFD Module

As shown in Figure 33.34, the various FIFO buffers can be configured by writing to the Receive FIFO n Configuration Register and the Common FIFO 0 Configuration Register.

For the two receive FIFOs, the following parameters can be configured:

- Interrupts
- FIFO depth
- Payload size

For the common FIFO, the following parameters can be configured:

- Mode
- Interrupts
- FIFO depth
- Payload size
- Transmission scan link position

(1) FIFO mode configuration of common FIFO

The mode of the common FIFO can be configured by writing to the CFCR0.MODE bit. The modes that can be configured in the common FIFO are as follows:

- 00b: Receive FIFO mode (default mode after a MCU reset)
- 01b: Transmit FIFO mode.

Messages can only be read from the receive FIFOs and the common FIFO configured in receive FIFO mode. Messages are stored by the CANFD module in these FIFO buffers based on the AFL entries.
Messages can be read and written into the common FIFO configured in transmit FIFO mode.
The pointers can be incremented only when a new message is stored in the FIFO buffer and decremented only when a message is transmitted on the corresponding CAN channel by the CANFD module.
After a MCU reset, all the common FIFO is configured in receive FIFO mode by default. Only enable the FIFO buffers after configuring the common FIFO in the required modes.

(2) FIFO tansmit message buffer link configuration

When the common FIFO is configured as transmit FIFO, the FIFO buffer must be linked to a normal transmit message buffer to participate in the transmission scan of a CAN channel.
Do not write data into a transmit message buffer that is linked to a common FIFO. Also, the transmit message buffer linked to a common FIFO should not be a part of the transmit queue.
The link to the transmit message buffer of common FIFO can be configured by writing to the CFCR0.LTM[1:0] bits. The options available for linking the transmit message buffer are:

- 00b: Transmit message buffer 0
- 01b: Transmit message buffer 1
- 10b: Transmit message buffer 2
- 11b: Transmit message buffer 3.

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the RFCRn.FDS[2:0] bits and CFCR0.FDS[2:0] bits. The six available options for depth configuration are:

- 000b: 0 messages (FIFO buffer cannot be used)
- 001b: 4 messages
- 010b: 8 messages
- 011b: 16 messages
- 100b: 32 messages
- 101b: 48 messages.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 16 messages (1216 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFD module does not have the function to check the validity of the configuration.

Note: If the FIFO depth of the common FIFO is 4 messages or more (CFCR0.FDS[2:0] > 000b), the link between the common FIFO and the transmit message buffer is valid regardless of whether the FIFO is disabled or enabled. If the FIFO depth is 0 messages, the link between the common FIFO and transmit message buffer is invalid regardless of whether the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The payload size of each FIFO buffer can be configured by writing to the RFCRn.PLS[2:0] bits and CFCR0.PLS[2:0] bits. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 16 messages (1216 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFD module does not have the function to check the validity of the configuration.

(5) FIFO interrupt configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the RFCRn.RFIM bit and CFCR0.CFIM bit. The two available options are:

- RFIM/CFIM = 0:
  – Receive FIFO mode: An interrupt is generated when the FIFO fill level reaches RFCRn.RFITH[2:0] or CFCR0.CFITH[2:0] value
  – Transmit FIFO mode: An interrupt is generated when the FIFO transmits the last message successfully
- RFIM/CFIM = 1:
  – Receive FIFO mode: An interrupt is generated each time the received message is stored
  – Transmit FIFO mode: An interrupt is generated each time a message is successfully transmitted.

If the RFCRn.RFIM bit for the receive FIFO is 0, an interrupt is generated based on the setting of the RFCRn.RFITH[2:0] bits.
Similarly, if the CFCR0.CFIM bit for a common FIFO set to receive FIFO mode is 0, an interrupt is generated based on the setting of the CFCR0.CFITH[2:0] bits.
The eight options available to set the FIFO fill level for generating interrupts are:

- 000b: Interrupt generated when FIFO is 1/8 Full
- 001b: Interrupt generated when FIFO is 1/4 Full
- 010b: Interrupt generated when FIFO is 3/8 Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8 Full
- 101b: Interrupt generated when FIFO is 3/4 Full
- 110b: Interrupt generated when FIFO is 7/8 Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message fill level matches the set value.
However, as shown in Table 33.21, there are some restrictions on the RFITH[2:0] and CFITH[2:0] bit settings, depending on the FDS[2:0] bits (FIFO depth setting) of each register.

Table 33.21 FIFO Interrupt Threshold and FIFO Depth Settings

	RFITH[2:0]/CFITH[2:0]							
FDS[2:0]	**111b (full)**	**110b (7/8)**	**101b (3/4)**	**100b (5/8)**	**011b (1/2)**	**010b (3/8)**	**001b (1/4)**	**000b (1/8)**
000b (0 messages)	Invalid (FIFO cannot be enabled)							
001b (4 messages)	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited
010b (8 messages)	Allowed							
011b (16 messages)	Allowed							
100b (32 messages)	Allowed							
101b (48 messages)	Allowed							

33.6.2.2 FIFO Buffers Control

To enable the receive FIFO interrupt, set the RFIE bit in the RFCRn register (n = 0, 1) to 1.
To enable the common FIFO interrupt, set either the CFRIE or CFTIE bit in the CFCR0 register:
After configuration is complete, each FIFO can be enabled by setting the RFCRn.RFE bit and CFCR0.CFE bit to 1 to allow transmission and reception of messages.

33.7 Reception and Transmission

33.7.1 Reception

In the CANFD module, messages received on any of the channels, will be stored in receive message buffers or in receive FIFOs or common FIFO configured in receive FIFO mode depending upon the Acceptance Filter List entries:

- up to 32 receive message buffers can be configured
- two receive FIFOs available
- up to one common FIFO can be configured in receive mode

33.7.1.1 Message Storage in Receive Message Buffers

When a message is successfully received and stored in a receive message buffer, the corresponding NDR[n] flag in the RMNDR register is set to 1.
The stored message can be read from the corresponding receive message buffer.
If a new message is stored into a receive message buffer before the previous message in this message buffer is read, the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the receive message buffer. If such a loss of messages is not acceptable, store related messages by using receive FIFO.

Note: Unused data bytes are filled with 00h depending on the DLC value.

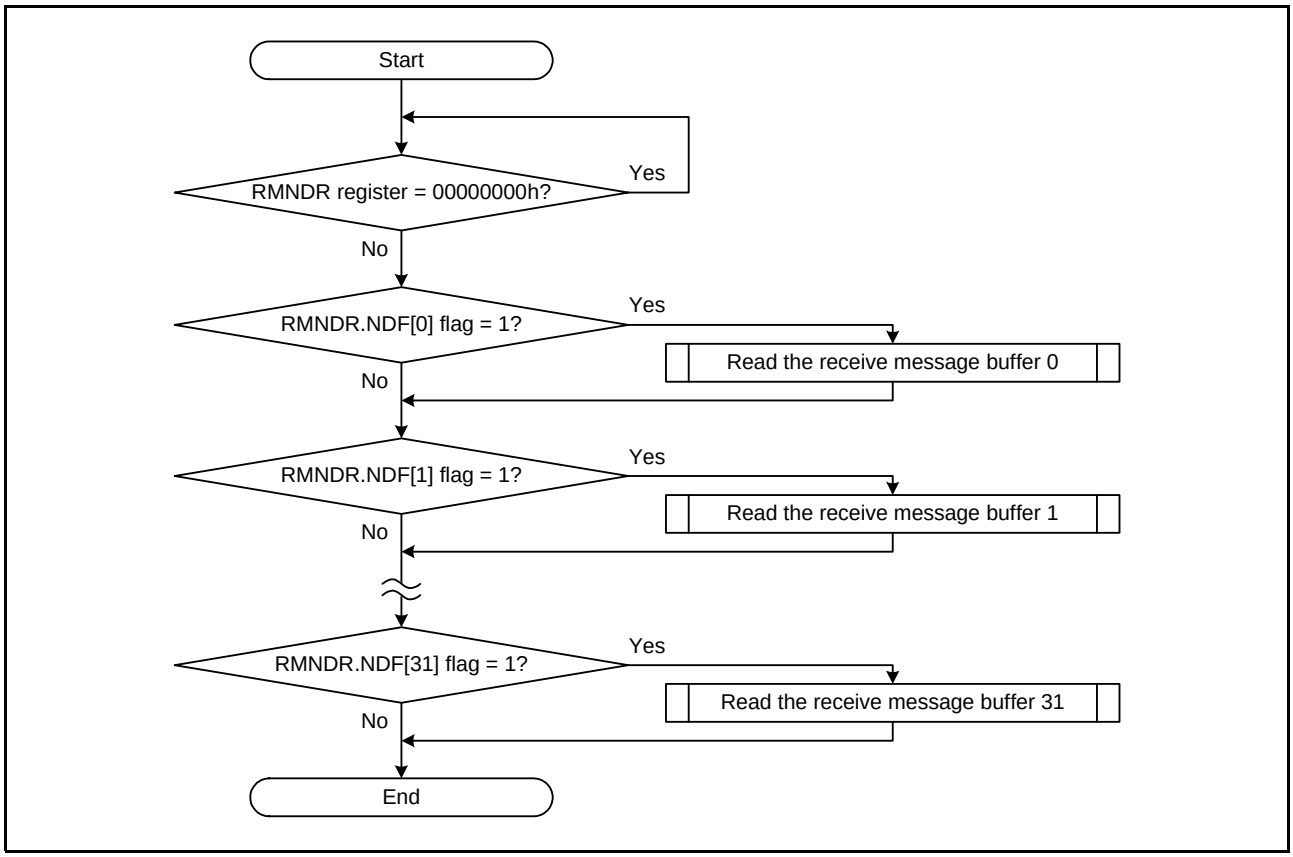


Figure 33.35 Receive Message Buffer Message Access Flow (Example for Polling Case)

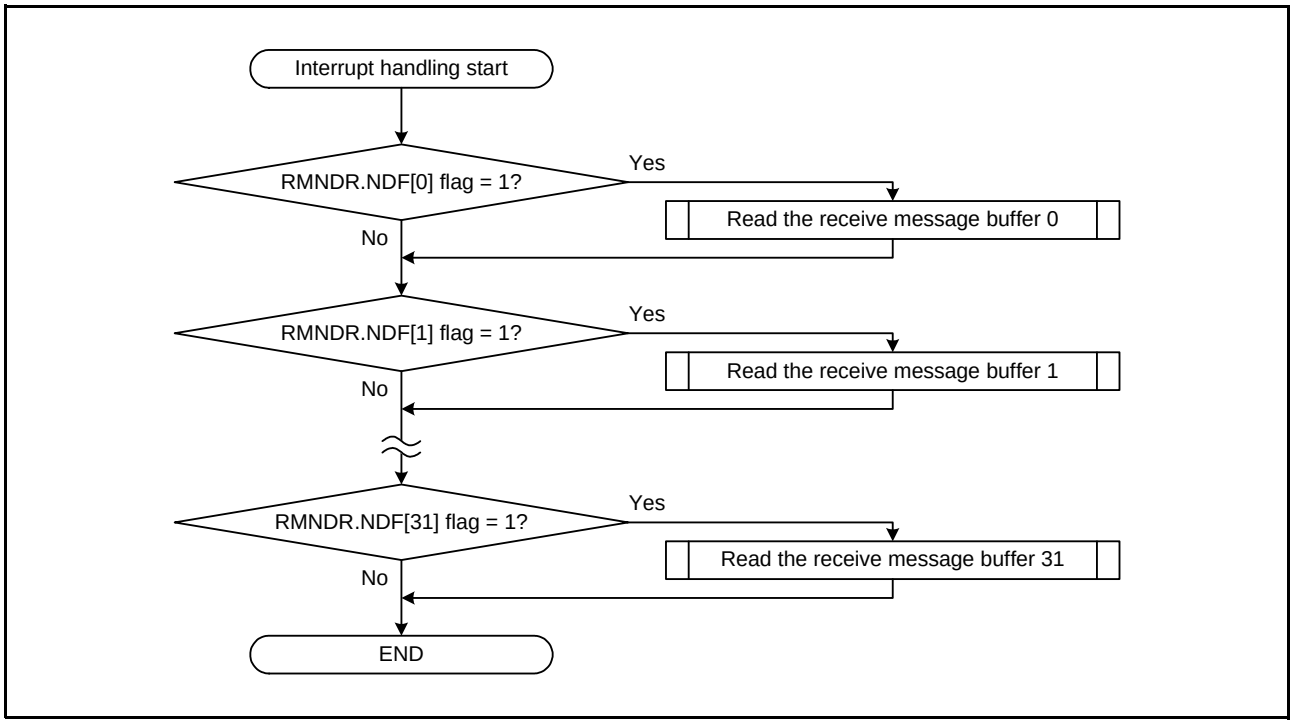


Figure 33.36 Receive Message Buffer Message Access Flow (Example for Interrupt Case)

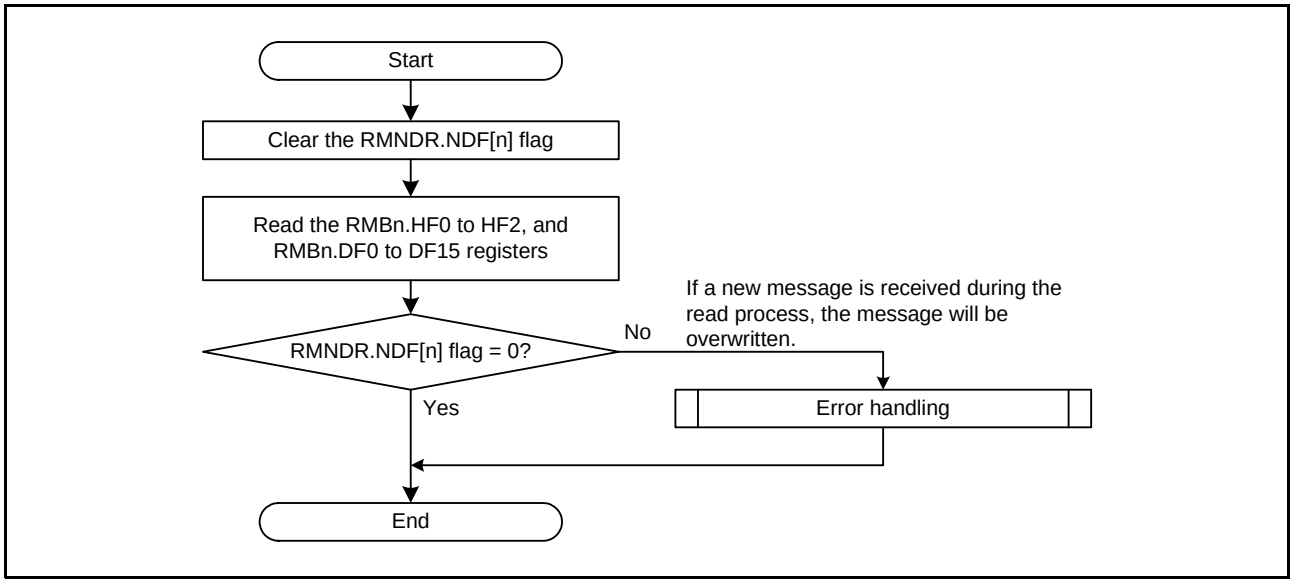


Figure 33.37 Receive Message Buffer n Read Flow

33.7.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to receive FIFOs or common FIFO configured in receive mode should be configured based on the requirements of the system.
The AFLn.PTR1.RF0E, RF1E, or CF0E bit in the matching AFL entry selects the FIFO buffers to which the related reception message will be stored.
When the received message is stored in one or more receive FIFOs or common FIFO configured in receive FIFO mode, the message counter value is incremented in the corresponding Receive FIFO n Status Register or Common FIFO 0 Status Register.
Depending upon the configuration of the FIFO buffers, an interrupt may also be generated. The message can be read from the corresponding FIFO access registers.

Note: Since many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, then the FIFO full flag is set.
When the value 000000FFh is written to the corresponding FIFO Pointer Control Register, then the message count is decremented by 1.

Only write 000000FFh to the FIFO Pointer Control Register after completely reading the message from the FIFO Access Register of the corresponding FIFO.
When all the messages stored in the FIFO are read, then the FIFO empty flag is set.
If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO message lost flag is set and the new message will be lost (messages already stored will not be overwritten).
To prevent message loss due to overrun, set an appropriate interrupt generation threshold and generate an interrupt before the FIFO becomes full.

The receive FIFOs and the common FIFO configured in receive FIFO mode can be disabled at any time by clearing the RFCRn.RFE bit or CFCR0.CFE bit.
When the RFCRn.RFE bit or CFCR0.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further messages can be stored into the FIFO.
When the receive FIFOs or common FIFO configured in receive FIFO mode is set to be read by DTC/DMA transfer, do not read the FIFO buffer on the CPU or write 000000FFh to the FIFO Pointer Control Register (RFPCR0, RFPCR1, or CFPCR0). The FIFO read pointer is automatically updated when read by DTC / DMA transfer.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, then the interrupt flag will not be cleared automatically. The interrupt flag should be cleared before disabling the FIFO.
Note: When the next frame is received before clearing the receive interrupt flag, the receive interrupt flag is not set again.

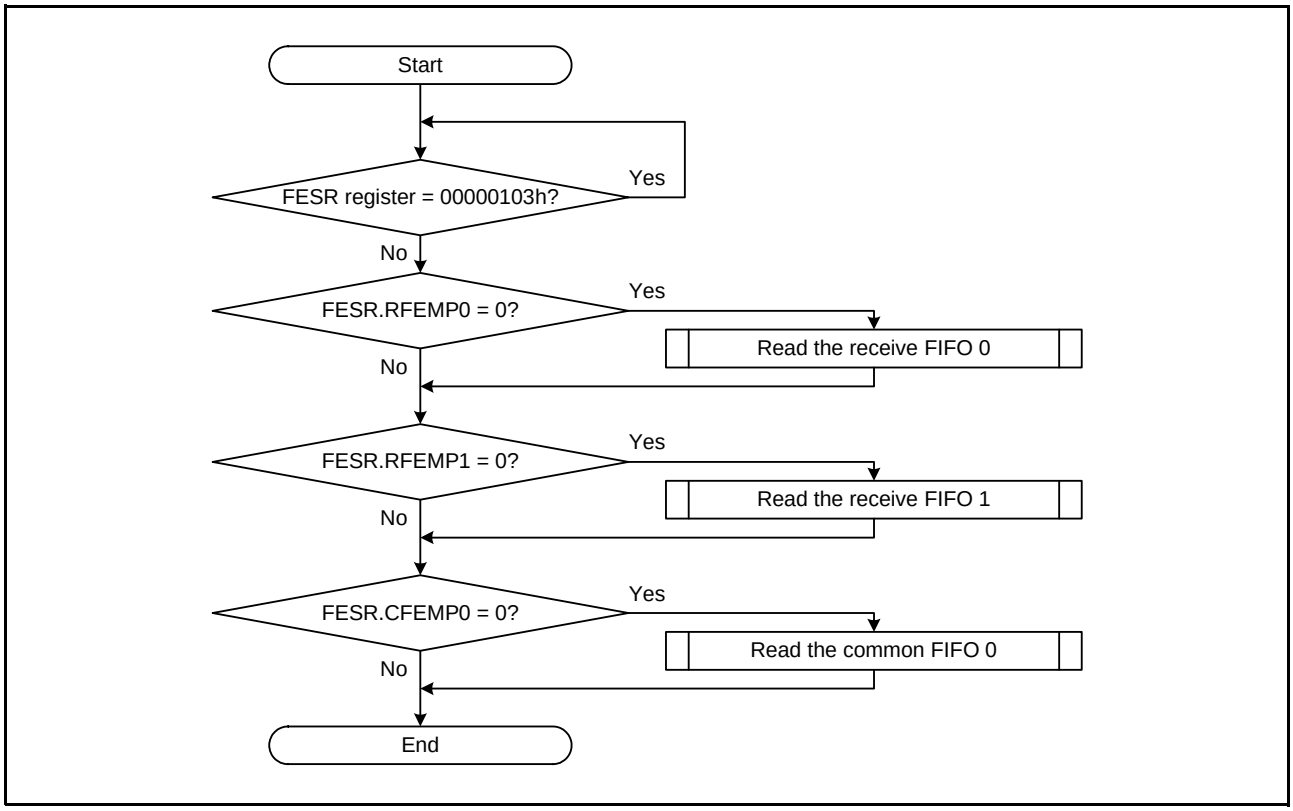


Figure 33.38 FIFO Buffer Message Access Flow (Example for Polling Case)

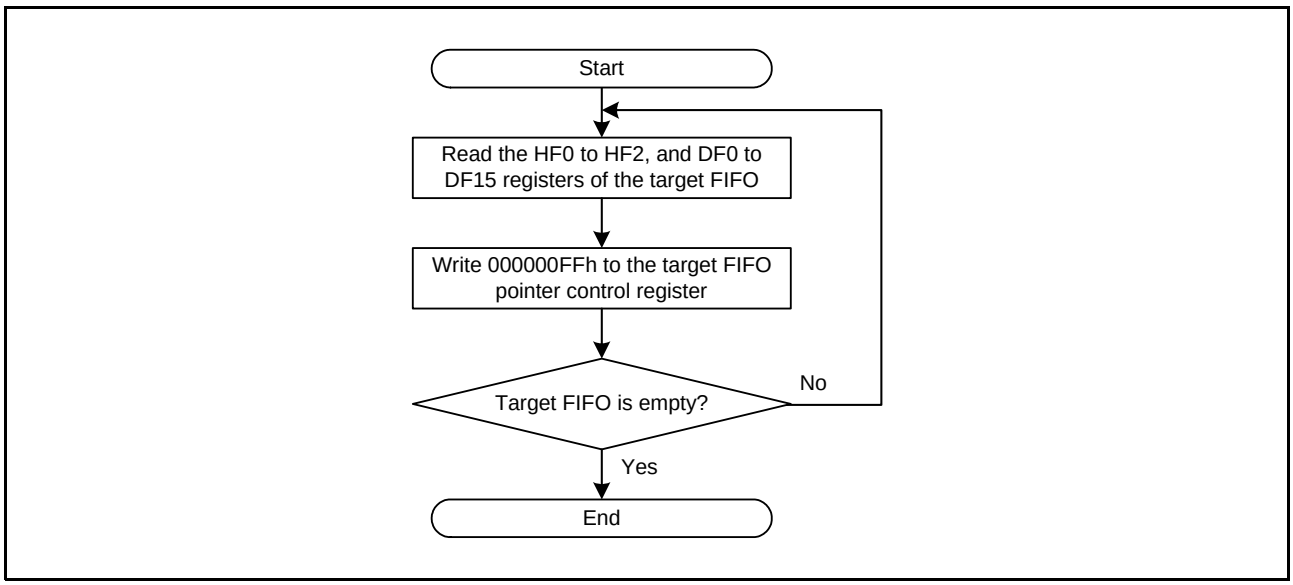


Figure 33.39 Receive FIFO Read Flow (Example for Polling Case)

If the interrupt flag is cleared after the FIFO read process is completed, the interrupt flag is not set even if the next frame has already been received. The FIFO read process must be performed and the interrupt flag must be cleared before the reception of the next frame is completed. If the process is not in time, make sure the FIFO is empty, clear the interrupt flag, and make sure the FIFO is empty again.

33.7.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of a received message or transmission time of a successful transmitted message. The timestamp counter value will be captured based on the GFDCFG.TSCPS[1:0] bit setting (sample point of SOF, EOF when the frame is taken valid, or sample point of res bit following the FDF bit in case of a CAN FD frame). For reception the timestamp counter value is stored together with the message ID and data into the target receive message buffer or receive FIFO.
For transmit message the timestamp counter value is stored as part of the transmission history entry.
The counter can be clocked from PCLKB or the bit timing clock of the CAN channel. The counter count source can be set with the GCFG.TSCS bit. If the GCFG.TSCS bit is 0, PCLKB is used. If it is 1, the bit time clock of CAN channel is used.
The count source of the timestamp counter can be divided by the coefficient defined by the GCFG.TSP bit (timestamp prescaler).
The timestamp counter can be reset to 0000h with the GCR.TSCR bit (timestamp counter reset).

33.7.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- Transmit queue transmission

The CANFD module has four transmit message buffers. These message buffers are only used for transmission and cannot be configured for reception.
Additionally transmission from transmit queue and/or common FIFO configured in transmit FIFO mode can be set in the following way (refer to Figure 33.33):

- Transmit queue
  3 or 4 transmit message buffers for one channel can be grouped to form a transmit queue with a common access window.
  Transmit message buffer 0 acts as an access window for transmit queue 0 (TXQ0).
- Common FIFO (transmit FIFO mode)
  The CANFD module has one common FIFO. Common FIFO configured in transmit FIFO mode is linked to any of the transmit message buffers 0 to 3.
  The linked transmit message buffer replaces the common FIFO. Do not access the TMCRn or TMSRn registers of the linked send message buffer.

Note: Common FIFO should not be linked to transmit message buffers that are already part of a transmit queue.

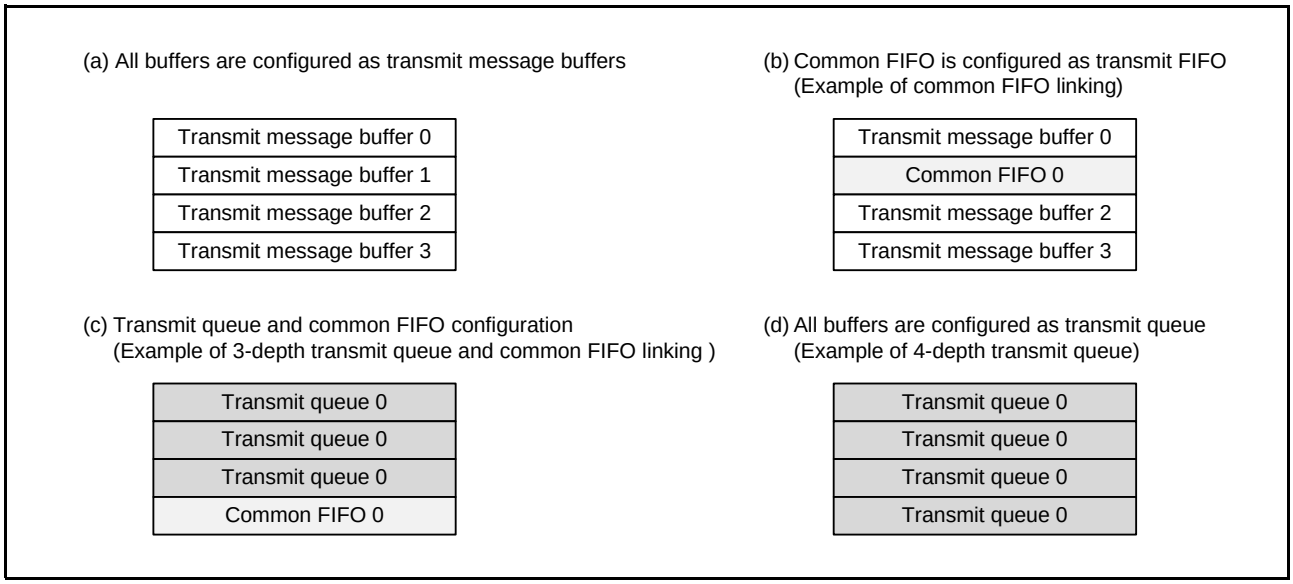


Figure 33.40 Channel Transmit Message Buffer Configuration

33.7.2.1 Transmission Priority

If two or more transmit message buffers of a channel are configured for transmission, the transmission priority in the CANFD module can be selected from the followings:

- CAN ID priority
- Message buffer number priority

The transmission priority is common for all message buffers. It can be configured via the GCFG.TPRI bit.
For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode.
Do not select message buffer number priority when using the transmit queue.
For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All transmit message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode and includes the transmit queue message buffers.
If the ID of two or more message buffers is the same, then the smaller message buffer number will have higher priority for transmission.

Note: For common FIFO configured in transmit FIFO mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.
If the message is being transmitted from the FIFO, then the next pending message within the same FIFO will be considered in the transmission arbitration.
In contrast to this, all transmit message buffers of a transmit queue will participate in internal transmission arbitration.

Figure 33.41 below shows the transmission configuration flow.

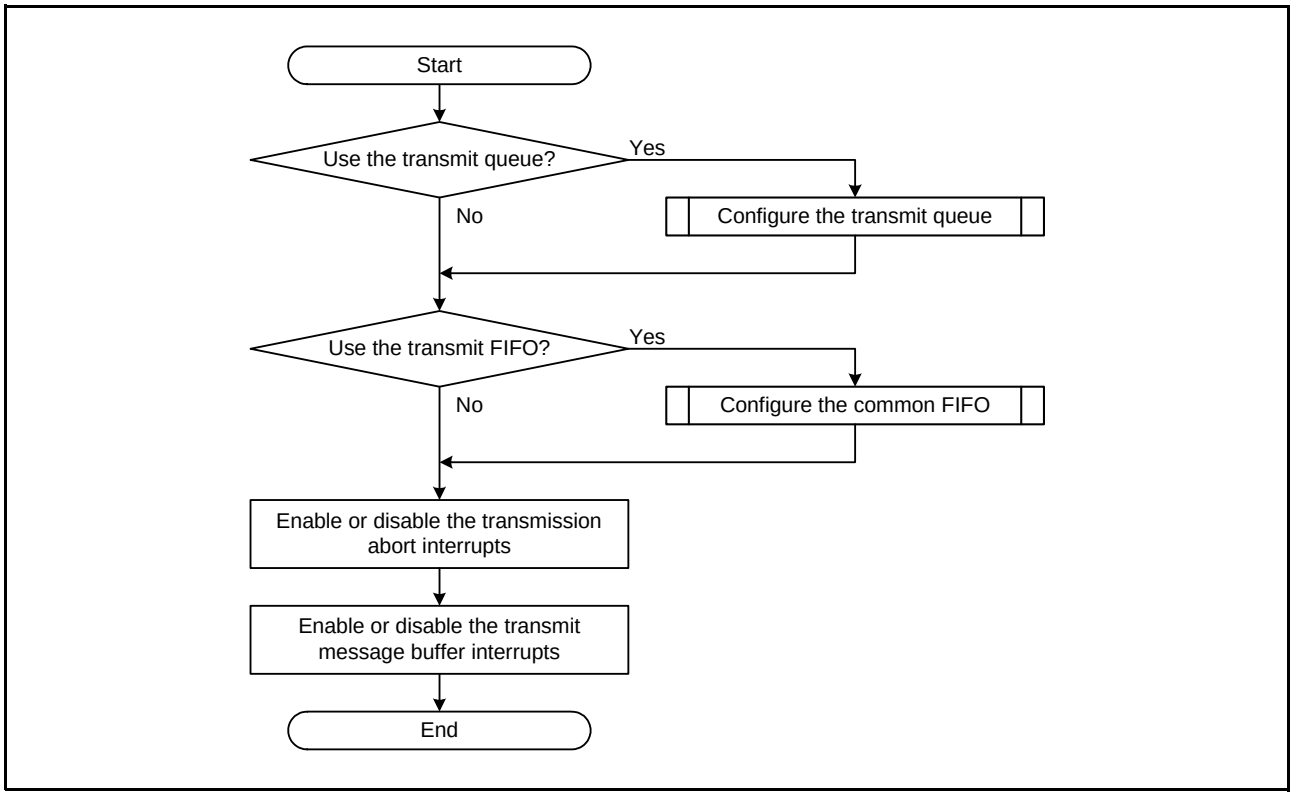


Figure 33.41 Transmission Configuration Flow

33.7.2.2 Message Transmission from Transmit Message Buffer

Each transmit message buffer has two modes of message transmission:

- Normal Transmission Mode
  When the message buffer is set to normal transmission mode, the data frames or remote frames set in that message buffer can be transmitted.
  Completion of normal transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when a normal transmission is successful.
  When arbitration is lost or an error occurs during transmission, the message transmission will be retried if no transmission abort request is set in the transmit message buffer.
  A new internal transmission arbitration is performed for all message buffers that have a transmission request.
- One-Shot Transmission Mode
  When the TMCRn.ONESHOT bit is set to 1, the message buffer is placed in one-shot transmission mode and attempts to transmit the message only once.
  Completion of one-shot transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when the one-shot transmission is successful.
  The TXRF[1:0] flags are set to 01b when arbitration is lost or an error occurs during transmission. In this case, message transmission will not be retried.

Figure 33.42 shows the transmission request procedure from transmit message buffer.

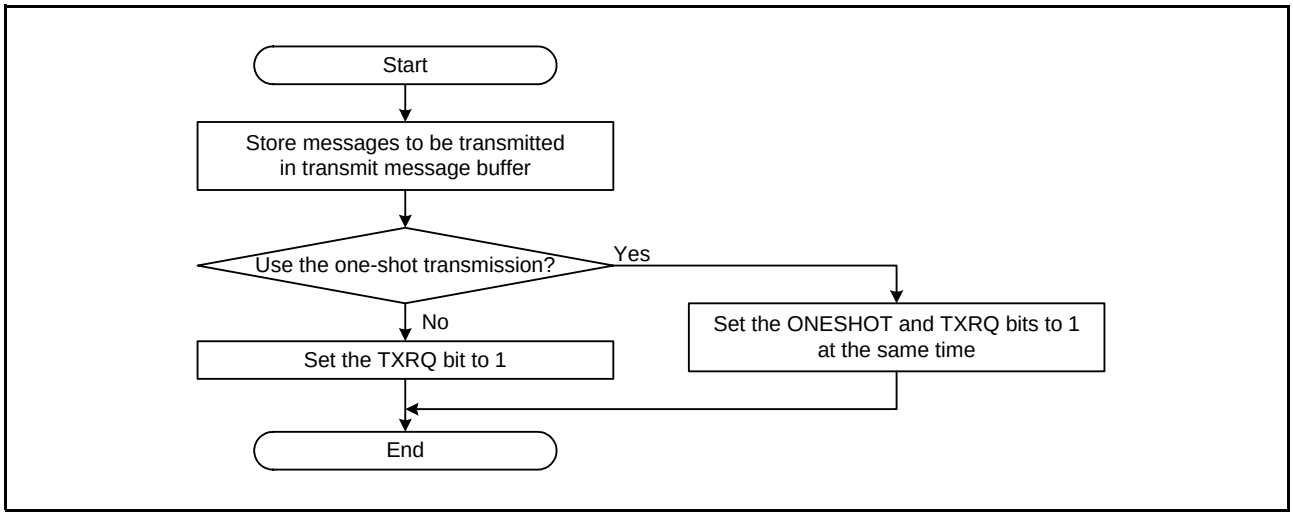


Figure 33.42 Transmission Request Procedure from Transmit Message Buffer

Table 33.22 shows configuration of the TMCRn register.

Table 33.22 Configuration of TMCRn Register

Transmission Request TXRQ Bit	Transmission Abort Request TARQ Bit	One-Shot Transmission Enable ONESHOT Bit	Message Buffer Status
0	0	0	Normal transmission is stopped.
0	0	1	One-shot transmission is stopped.
1	0	0	Data frames or remote frames are transmitted in normal transmission mode.
1	0	1	A data frame or a remote frame is transmitted in one-shot transmission mode.
1	1	0	Transmission abort is requested.
1	1	1	One-shot transmission abort is requested.

Figure 33.43 shows timings for successful transmission from two message buffers.

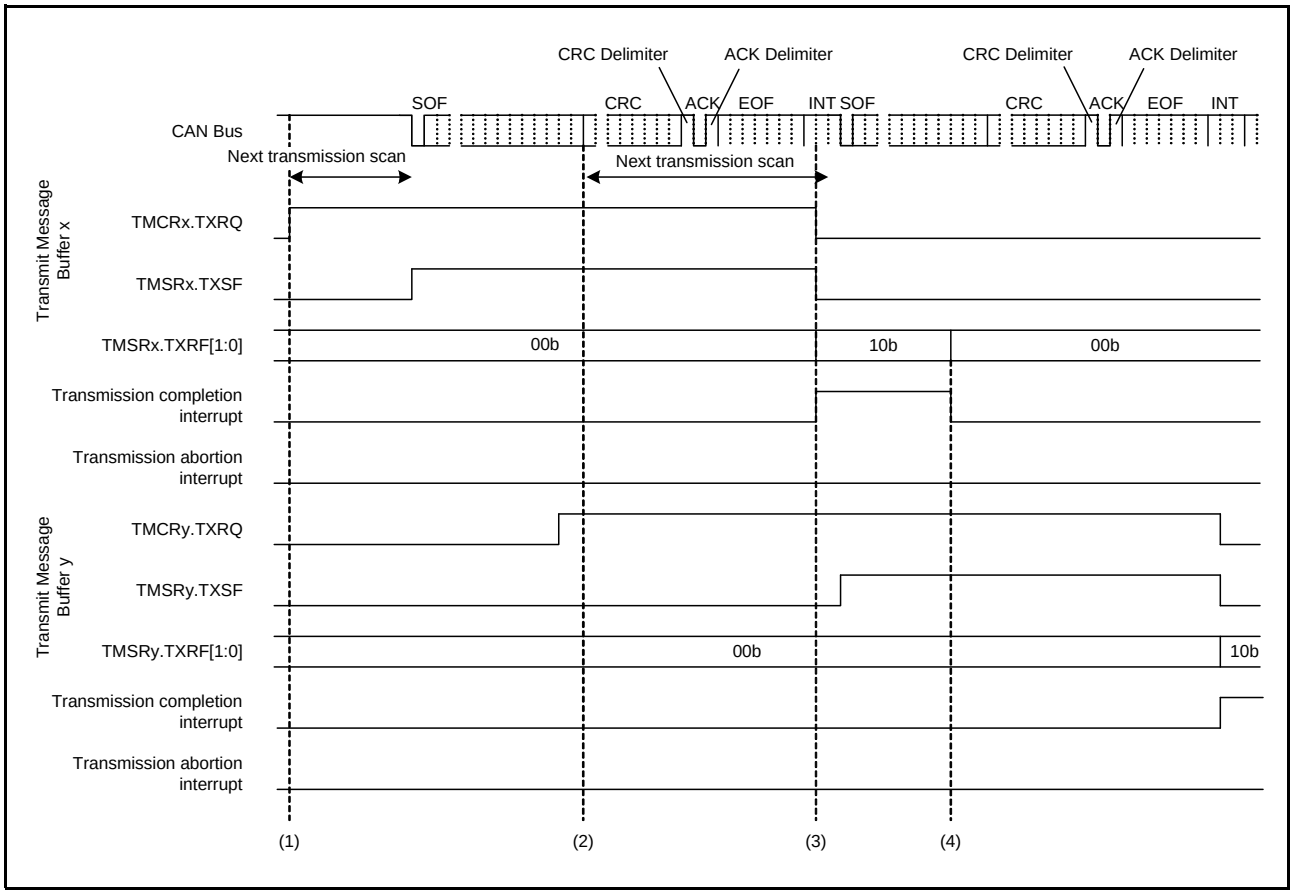


Figure 33.43 Timing of Request and Flag Bits for Successful Transmission

(1) If the TMCRx.TXRQ bit is set to 1 in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF bit is set to 1 (transmitting), and CAN channel starts the transmission*1.

(2) At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.

(3) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 10b and the TMSRx.TXSF flag and the TMCRx.TXRQ bit are cleared.
When the TMIER0.TMIEx bit is set to 1 (transmit message buffer interrupt enabled), the successful transmission interrupt request is generated. To clear the related interrupt line the TMSRx.TXRF[1:0] flags have to be cleared.

(4) Before starting the next transmission, clear the TMSRx.TXRF[1:0] flags. Load the next message in the transmit message buffer and set the TMCRx.TXRQ bit to 1 again.
The TMCRx.TXRQ bit cannot be set to 1 again before TMSRx.TXRF[1:0] flags are cleared.

Note: The setting point of the TMSRx.TXSF flag is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs either during the transmission or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

The Figure 33.44 shows timings for transmission abort for two message buffers.

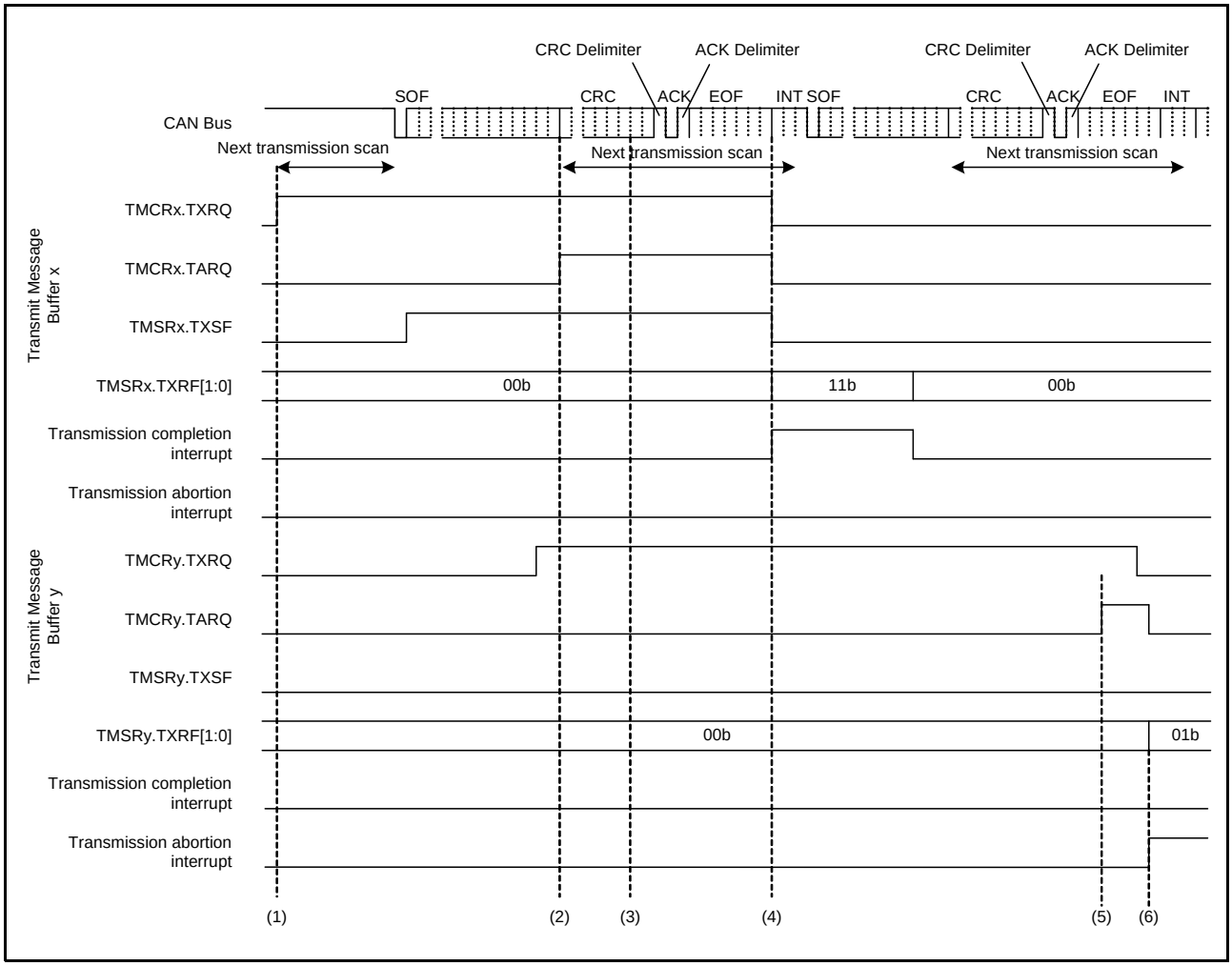


Figure 33.44 Timing of Request and Flag Bits for Transmission Abort

(1) If the TMCRx.TXRQ bit is set in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF flag is set (transmitting), and CAN channel starts the transmission*1.

(2) If the TMCRx.TARQ bit is set to 1 when the related message buffer is already selected for transmission or currently transmitting, the message will not be aborted, if no error occurs or arbitration is lost.

(3) At the first bit of CRC, the transmission scanning procedure starts for the next transmission. In this example timing chart message buffer y is not selected as next transmit message buffer.

(4) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 11b and the TMSRx.TXSF flag and TMCRx.TXRQ bit are cleared.
When the TMIER0.TMIEx bit is set to 1 (transmit message buffer interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the TMSRx.TXRF[1:0] flags has to be cleared.

(5) Another CAN node is transmitting on the CAN bus (the TMSRy.TXSF flag is not set): if the TMCRy.TARQ bit is set to 1 when the related channel is under transmission scan, the transmission request cannot be cleared.

(6) After internal processing time the transmission is aborted and the TMSRy.TXRF[1:0] flags are set to 01b.
If the message buffer is not transmitting or selected as next transmit message buffer or under transmit scan, the abort is immediately accepted and the corresponding TMSRy.TXRF[1:0] flags are set to 01b.
In addition, the TMCRy.TXRQ and TMCRy.TARQ bits are cleared automatically.

When the CHCR.TAIE bit is set to 1 (transmission abort interrupt enabled), an interrupt is generated for successful transmission abort.
To clear the related interrupt the TMSRy.TXRF[1:0] flags have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs, either during the transmission, or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

33.7.2.3 Message Transmission from FIFO Buffer

The CANFD module has one common FIFO. The common FIFO can be linked to the transmit message buffer by the CFCR0.LTM[1:0] bits if configured in transmit FIFO mode.
When a transmission scan is started and the common FIFO corresponding to that transmit message buffer is enabled, the relevant message in the common FIFO will participate in the transmission scan.
Do not configure the transmit message buffer linked to the common FIFO configured in transmit FIFO mode.

(1) Transmit FIFO Operation

Messages can be written into the transmit FIFO by writing to the common FIFO buffer 0 (CFB0).
When the value 000000FFh is written into the CFPCR0 register, the message count of the FIFO is incremented by 1.
Before writing to the CFPCR0 register, wait until the message has been completely written to the CFB0. If the message count matches the FIFO depth, the CFSR0.FULL flag is set to 1.
The oldest message in the transmit FIFO is included in the scan for transmission.
When a message is successfully transmitted from the transmit FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the CFSR0.EMPTY flag is set to 1.
The interrupt generation conditions for the transmit FIFO buffers can be set by the CFCR0.CFIM bit.
If the CFCR0.CFIM bit is set to 0, interrupt is generated when last message is successfully transmitted from the transmit FIFO buffer. If the CFCR0.CFIM bit is set to 1, interrupt is generated for every successfully transmitted message from the transmit FIFO buffer.
Common FIFO can set interrupt, when CAN frame transmission is completed.
The common FIFO configured in transmit FIFO mode can be disabled by setting the CFCR0.CFE bit to 0. If this bit is set to 0, the CFSR0.EMPTY flag is set to 1 as described below:

- immediately if the message from the transmit FIFO is neither scheduled for the next transmission nor in transmission
- following the transmission completion, the detection of an error on the CAN bus, arbitration lost or transition to CH_HALT or GL_HALT mode if the transmission from the transmit FIFO is already scheduled for transmission or already in transmission.

Note: The common FIFO is considered as disabled after setting the CFCR0.CFE bit to 0 only when the CFSR0.EMPTY flag is set to 1 for the corresponding common FIFO.

If there are other pending messages in the transmit FIFO buffer, they will be lost and the transmission needs to be requested again. Before the CFCR0.CFE bit is set to 1 again ensure that the CFSR0.EMPTY flag is set to 1 and that there is no pending abort request from the transmit FIFO.
When the CFCR0.CFE bit is set to 0, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after a configuration is shown in Figure 33.45.

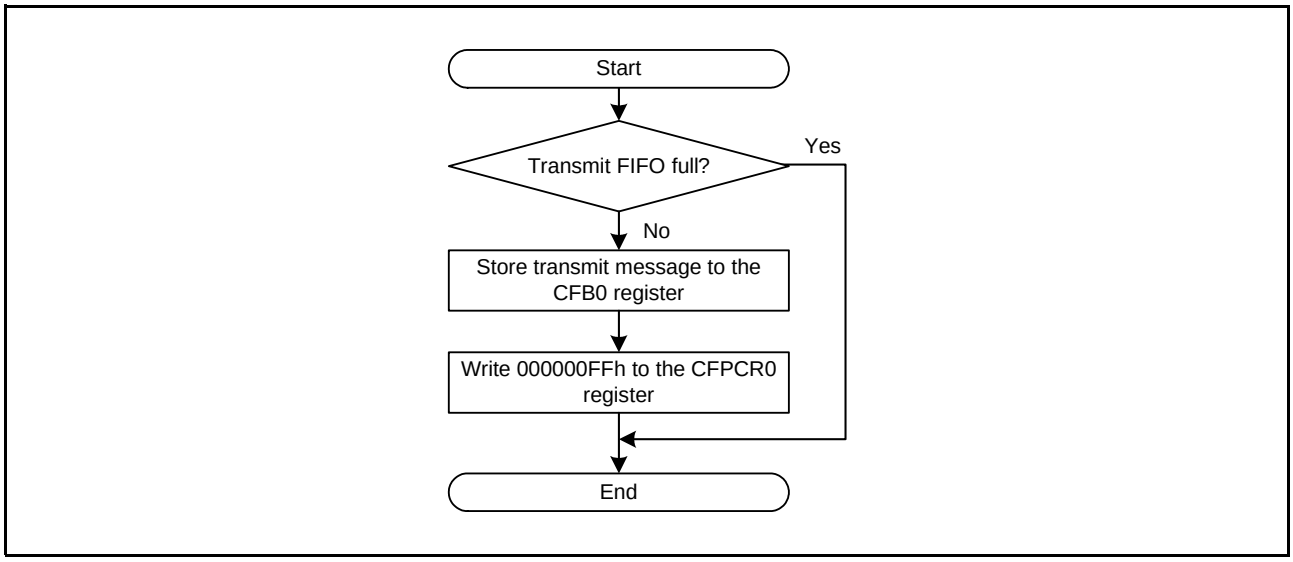


Figure 33.45 Transmit FIFO Transmission Request Procedure

(2) Interval Timer for FIFO Transmission

For the common FIFO in transmit mode it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFCR0.CFE bit is set to 1.
When the common FIFO in transmit mode is enabled, then the first message will be transmitted without considering this interval time.
The interval timer will stop counting when:

- FIFO is disabled by set the CFCR0.CFE bit to 0
- CAN channel is in CH_RESET mode.

The interval time is specified by the value of the CFCR0.TINT[7:0] bits and can be specified from 0 to 255 timer units. The timer unit can be defined based on two different count sources for the interval timer. Select the value 0 to disable the interval timer for FIFO transmission.
The count source can be selected by the CFCR0.ITCS bit. For the count source the CAN bit timing clock of the related channel or a reference clock could be selected.
If CAN channel bit time clock is configured as count source and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, then the interval timer is stopped for that channel.
If peripheral clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.
The reference clock can be used to configure the interval time in fixed time units. It is based on the PCLKB. The GCFG.ITP[15:0] bits define the relation between the PCLKB frequency/period and the reference clock period.
Refer to Table 33.23 for configuration values of the GCFG.ITP[15:0] bits to achieve different reference clock periods based on the PCLKB frequency/period.

Table 33.23 Configuration Example for Interval Timer Prescaler

	Reference clock period		
PCLKB frequency (period)	**1 µs**	**100 µs**	**500 µs**
16 MHz (62.5 ns)	16	1600	8000
20 MHz (50 ns)	20	2000	10000
32 MHz (31.25 ns)	32	3200	16000
50 MHz (20 ns)	50	5000	25000

Additionally the reference clock resolution can be specified by the CFCR0.ITR bit.
The interval time is based on the reference clock period multiplied by the configured value (×1 or ×10).
The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time.
The whole range for the separation time from 100 µs to 127 ms can be covered.
The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).
When the interval time has elapsed, the next transmission request is raised by the related transmit FIFO. Hence, the interval time defines the minimum time between two messages transmitted from one FIFO.
The next message will earliest be sent after this interval time. The Figure 33.46 shows an example timing of the internal processing.

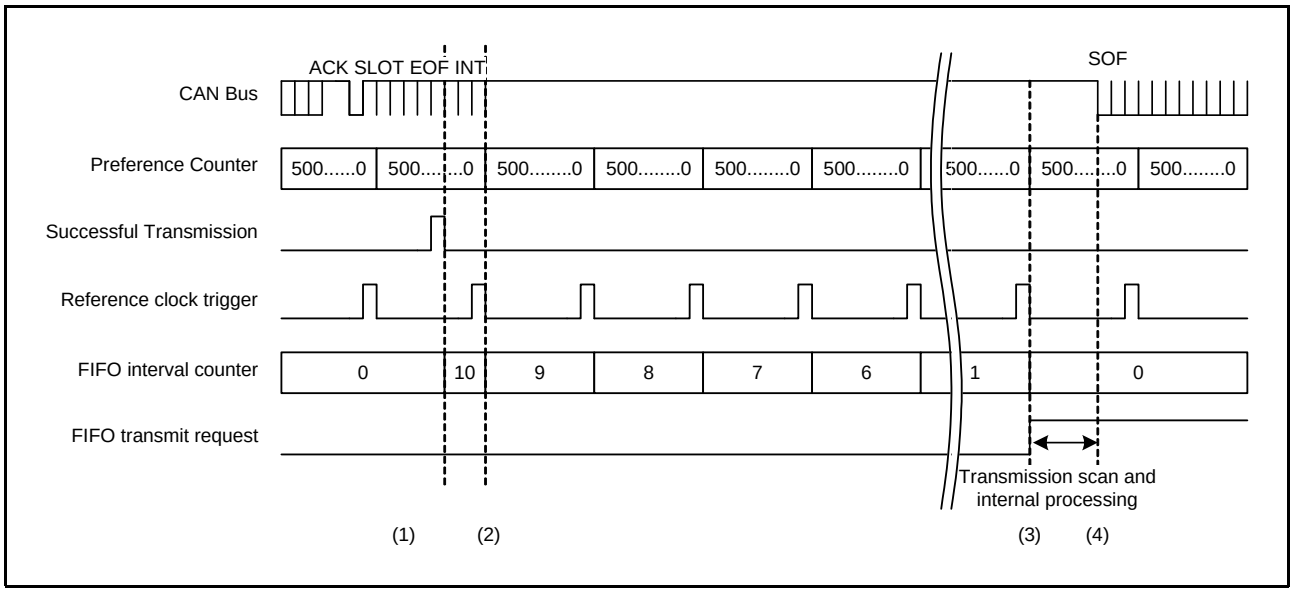


Figure 33.46 Example for Interval Processing Time

The configuration for this timing above is following:

- PCLKB frequency = 50 MHz
- Interval timer prescaler (GCFG.ITP[15:0]) = Divided by 500
- Reference clock due to the settings above = 10 µs
- Common FIFO interval timer count source selection (CFCR0.ITCS) = 0
- Common FIFO interval timer resolution (CFCR0.ITR) = 0
- Common FIFO transmission interval (CFCR0.TINT) = 10 counts
- Theoretical message separation interval = 100 µs

(1) Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore the first interval is counting less or equal to one reference clock interval.
(2) With the next reference clock trigger the FIFO interval timer is decremented.
(3) When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
(4) When the FIFO is selected for transmission then the transmission will start soon. Due to internal processing this usually takes less than 3 bit time, between internal FIFO transmit request set 3. (shown above) and actual transmission.
When multi events like reception scan, internal message routing, transmit scan happen, then it could take up to 126 PCLKB cycles.

As shown in Figure 33.46, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure the CFCR0.TINT bit to required minimum value + 1.
If transmit message buffers or transmit FIFOs are configured for transmission for the channel the real delay between two messages transmitted from a transmit FIFO can be much longer than specified by the interval time due to higher priority message transmission from these transmit message buffers or transmit FIFOs.
Figure 33.47 shows a block diagram of the FIFO interval timer.

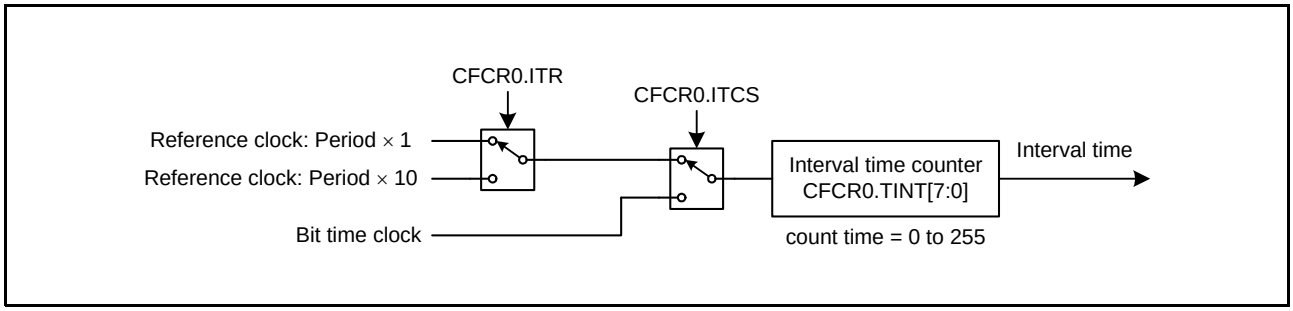


Figure 33.47 Block Diagram of FIFO Interval Timer

33.7.2.4 Transmit Queue

Each enabled transmit queue consists of three or four transmit message buffers, which are accessed via one access window.
The transmit queue 0 (hereafter TXQ0) can be configured with a depth of three to four, and it is using the transmit message buffer 0 as access window. All the message of TXQ0 enter the priority comparison for the transmission, which should be ID priority (GCFG.TPRI = 0).
The registers for TXQ0 are TQCR0, TQSR0, and TQPCR0.
As access window transmit message buffer 0 (TXQ0) is used, refer to related access registers TMBn.HF0, TMBn.HF1, TMBn.HF2, and TMBn.DF0 to TMBn.DF15.
The depth of each TXQ0 buffer can be configured by writing to the TQCR0.QDS[1:0] bits. TXQ0 can be set from TMB0 to TMB3 as a queue buffer at the maximum.
The available options for depth configuration are:

- 10b: 3 messages
- 11b: 4 messages.

Do not access all transmit message buffers except transmit message buffer 0 that configure the transmit queue. Also, do not access the TMCRn registers that correspond to the transmit message buffers that configure the transmit queue.
When writing data to TXQ0, check the status of TXQ0 before writing the data to be transmitted.
The messages stored to the transmit queue access window are internally stored to a free buffer of the transmit queue. When the TXQ0 buffer is full, do not access the queue anymore. If the transmit data is written when the TXQ0 is full, the

transmit data will be overwritten.
The transmit queue can be disabled by setting the TQCR0.TQE bit to 0. If this bit is set to 0, the TQSR0.EMPTY flag is set as described below:

- immediately: if a message from the transmit queue is not scheduled for the next transmission and is not being transmitted
- after the transmission completion, error detection on the CAN bus, loss of arbitration, or transition to CH_HALT or GL_HALT mode: if a message from the transmit queue is scheduled for transmission or is being transmitted.

Note: The transmit queue is disabled only when the TQSR0.EMPTY flag is set to 1 after setting the TQCR0.TQE bit to 0.

If there are other pending messages in the transmit queue, they will be lost, so their transmission needs to be requested again.
Before the TQCR0.TQE bit is set to 1 again, ensure that the TQSR0.EMPTY bit is set to 1 and that there is no pending abort request from the transmit queue.
When the TQE bit is set to 0, all messages in the transmit queue buffers will be lost and no further message should be stored into the transmit queue.
When a message has been stored to the transmit queue, 000000FFh must be written in to the TQPCR0 register. This will set the transmit request automatically and change the internal message buffer pointer to the next free message buffer location of the transmit queue.

Note: If two messages with the same ID are stored in the transmit queue, then the order of transmission of these messages could be different from the order in which they were stored in the transmit queue.
To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the transmit queue.

For the transmit queue a dedicated interrupt can be enabled by setting the TQCR0.TQIE bit.
The interrupt mode can be configured with the TQCR0.TQIM bit either to generate an interrupt for every transmitted message or for the last transmitted message.
The transmit queue transmission request procedure after configuration is shown in Figure 33.48.

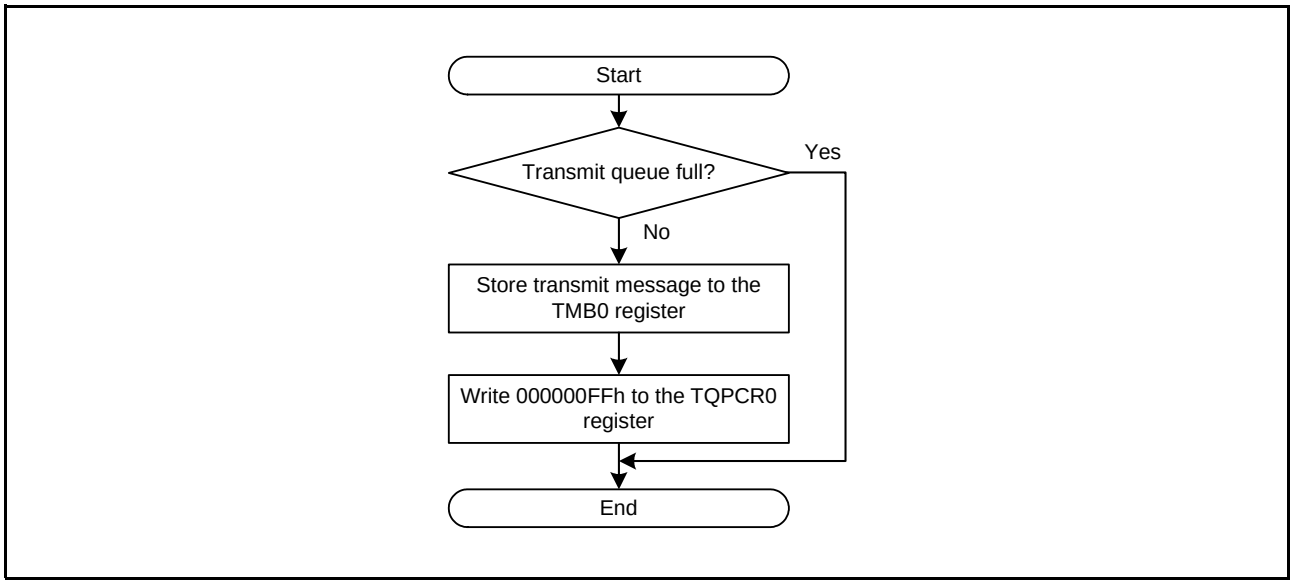


Figure 33.48 Transmit Queue Transmission Request

33.7.2.5 Transmission History

The transmission history function records the information of the successfully transmitted message in the transmission history buffer. The transmission history buffer can store up to eight transmission history entries.
The THCR.THRC bit can select whether to store information only for messages transmitted from transmit FIFO and transmit queue, or also for messages transmitted from the transmit message buffers.
Each transmit message can be individually configured for acceptance to the transmission history by the CFB0.HF0.THENT bit.
The message information is stored to the transmission history buffer of a CAN channel after the message is successfully transmitted on that CAN channel.
Storing to the list is not synchronized with the status of TMSRn.TXRF[1:0] flags.
Due to internal processing, the storage to the list could happen with a delay after the successful transmission indication.
Storing the transmission history data can be recognized by the condition that the THSR.THIF flag is set to 1 when the THCR.THIE bit is set to 1 or when the transmission history counter (THSR.FLVL[3:0]) is increased.
Multiple events could occur, such as reception scans and internal message routing.
Maximum delay time from setting the TMSRn.TXRF[1:0] flags to storing the transmission history data is 76 PCLKB cycles.

The transmission history records following information of the transmitted message:

- Transmitted Buffer type:
  – 001: Transmit message buffer
  – 010: Common FIFO in transmit FIFO mode (hereinafter referred to as transmit FIFO)
  – 100: Transmit queue
- Transmitted Buffer number:
  Transmit message buffer, transmit queue message buffer or transmit message buffer link for the common FIFO from which the transmission occurred. The number depends upon the buffer type, refer to Table 33.24.
- Transmitted pointer:
  Pointer set in header field 2 of transmit message (HF2.PTR[15:0])
- Transmitted timestamp:
  Message timestamp captured at capture point as set by the GFDCFG.TSCPS[1:0] bits
- Transmitted information label:
  Information label set in header field 2 of transmit message (HF2.IFL[1:0])

Table 33.24 Transmission History Buffer Number Entry

Transmitted Buffer Number (THACR0.BN[1:0])	Transmited Buffer Type (THACR0.BT[2:0])		
	001b	010b	100b
	Transmit Message Buffer	Transmit FIFO	Transmit Queue
00b	TXMB0	The value of the BN[1:0] bits corresponds to the setting of the CFCR0.LTM[1:0] bits.	The value of the BN[1:0] bits indicates the number of the transmit message buffer from which the message was transmitted.
01b	TXMB1		
10b	TXMB2		
11b	TXMB3		

The transmission pointer is used to identify which message of a transmit FIFO or transmit queue has been successfully transmitted because the transmit FIFO or transmit queue number alone is not sufficient.
Therefore, a unique number (pointer) can be attached to each transmission message stored in a transmit FIFO or transmit queue. This unique pointer should be written to the CFB0.HF2.PTR[15:0] part for a transmit FIFO or to the TMB0.HF2.PTR[15:0] part of the transmit queue.
When the message is successfully transmitted then this pointer is stored together with the other message related

information to the transmission history and can be read via the transmission pointer field (PTR[15:0]) in the Transmission History Access Register.
Also for normal transmit message buffers, the TMBn.HF2.PTR[15:0] part will be stored in the transmission history. An information label is the same.
Read access to the Transmission History Access Register will be done for every single entry.
After reading one entry, 000000FFh has to be written to the THPCR register to be able to access the next entry until transmission history is empty.
Figure 33.49 shows an example flow for processing the transmission history.
The transmission history has dedicated interrupts, which can be configured with the THCR.THIM bit and enabled with the THCR.THIE bit, either to generate an interrupt when the transmission history reached a filling level of 75% or for every new transmission history entry.
Loss of transmission history is indicated by the THSR.LOST flag. Status of this flag is also shown by the GESR.THLDF flag.

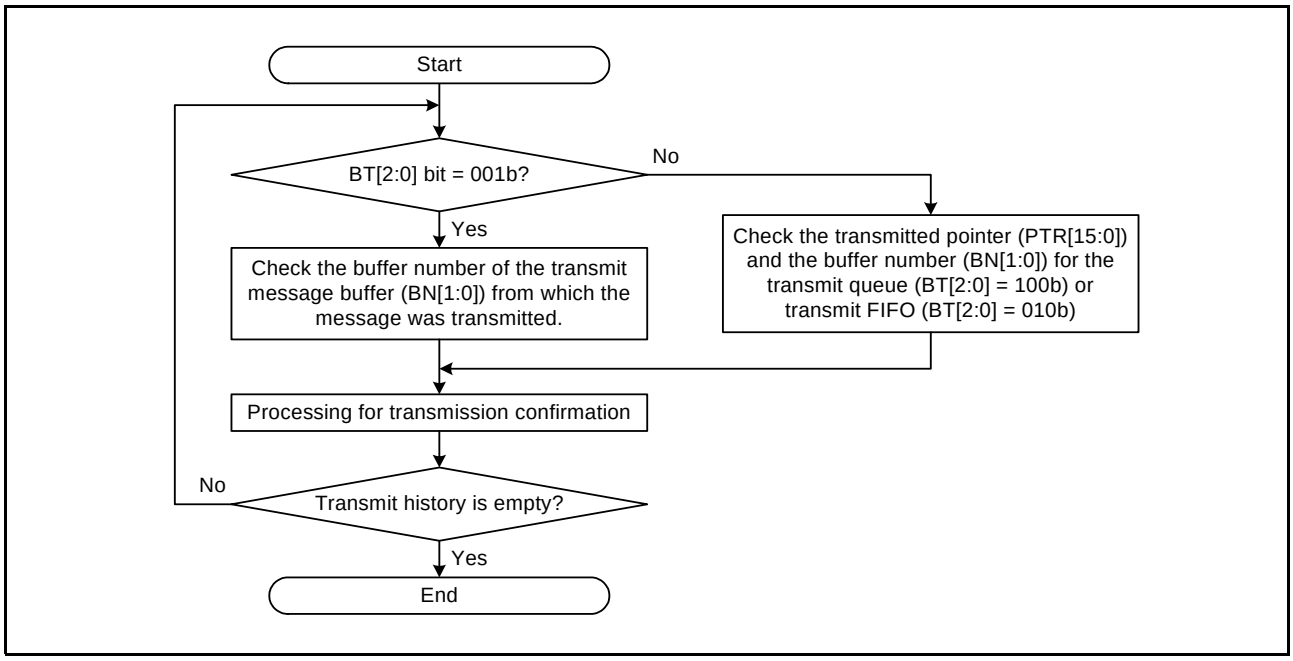


Figure 33.49 Example of Transmission History Processing Flow

33.7.2.6 Transmit Data Padding

If the data length code (DLC) of the transmit message is higher number of data bytes than the buffer size, the data bytes beyond the restricted range will be replaced by bytes with the value of CCh.
This could occur for common FIFO configured in transmit FIFO mode when the DLC of the transmit message is larger than the payload size set in the CFCR0.PLS[2:0] bits.
This can also occur in FD only mode, if the DLC value for a Classical CAN frame is greater than 8.

33.8 ECC Check

Message buffer RAM has ECC function of the 2-bit error detection and 1-bit error detection/correction.*1 The ECC module appends 7-bit ECC data to 32-bit RAM data.

Note 1. The ECC module cannot detect errors larger than 3 bits. In this case, the ECC module detects a 1-bit or 2-bit error, does not detect the error, or corrects the error bit to error data by configuration. If all RAM data is fixed at 0 or 1, it will be detected as a 2-bit ECC error.

33.8.1 ECC Function Setting

Figure 33.50 shows a procedure for ECC function setting.

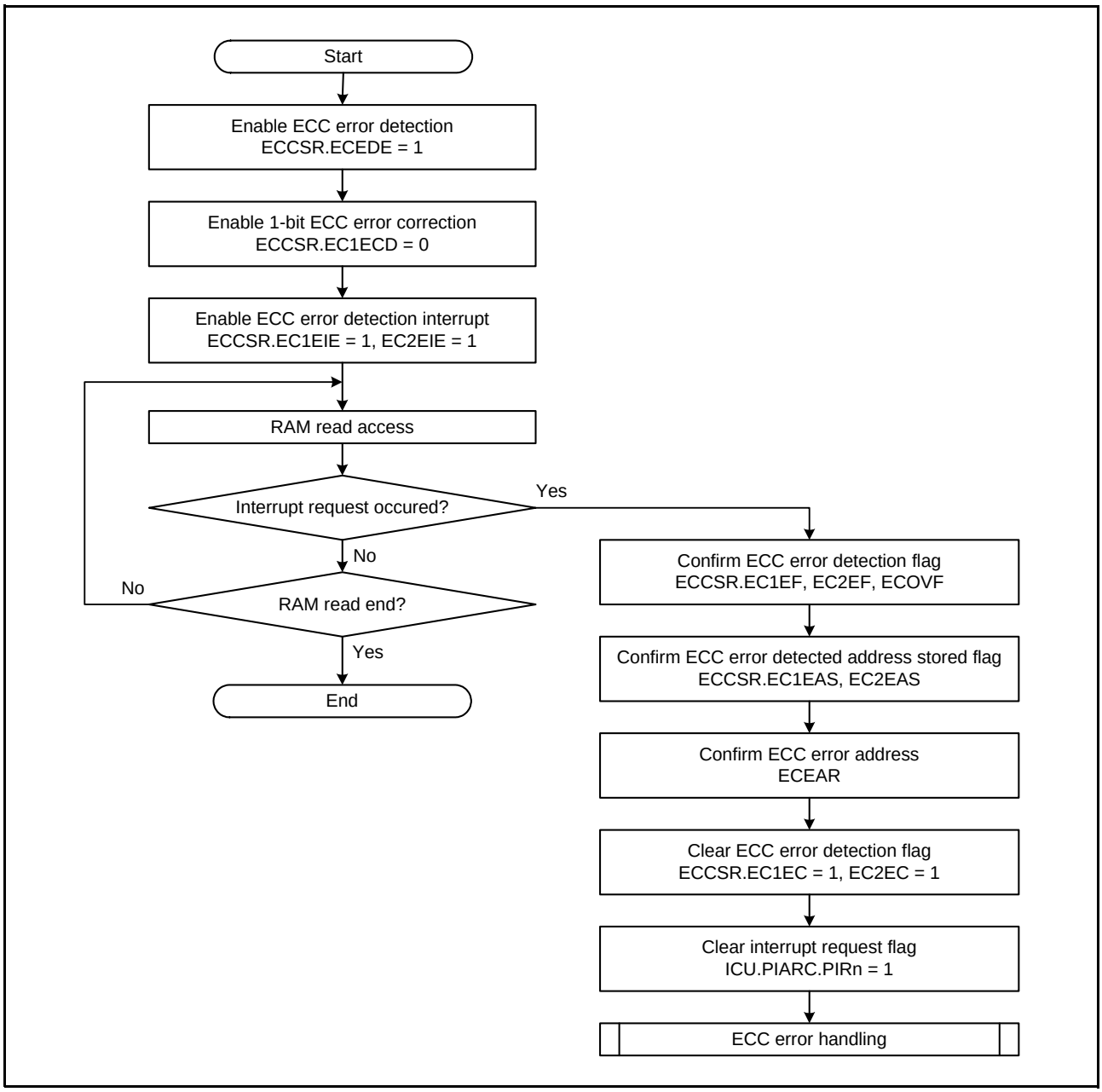


Figure 33.50 Setting Procedure for ECC Function

33.8.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 33.51 shows a procedure for ECC decoder testing.

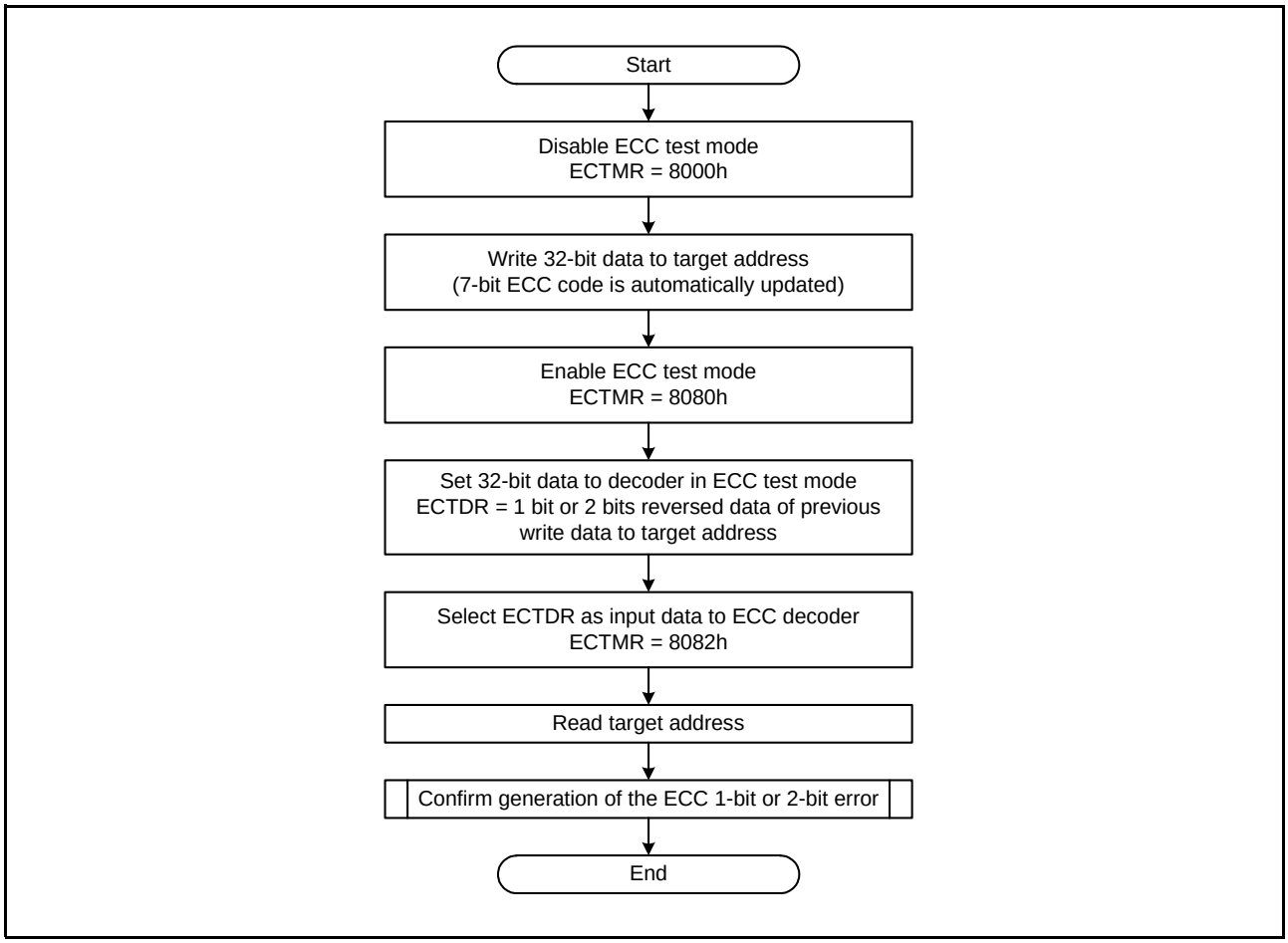


Figure 33.51 Testing Procedure for ECC Decoder

33.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in the test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.
The test modes can be broadly split into two groups:

- Channel specific test modes
- Global test modes

33.9.1 Channel Specific Test Modes

CAN channel can be configured into following test modes:

- Basic test mode
- Listen-only mode
- Self test mode 0 (external loop back mode)
- Self test mode 1 (internal loop back mode)
- Restricted operation mode

33.9.1.1 Basic Test Mode

The basic test mode should be used when a particular test setting needs to be enabled other than when in listen-only and self-test modes.

33.9.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional bus monitoring mode. In this mode, the CAN channel can receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit. If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN engine monitors this dominant bit, although the CTX0 pin remain in recessive state.
This mode can be used for bit rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.
In this mode, it is not permitted to request transmission from any transmit message buffer, transmit queue, or common FIFO.

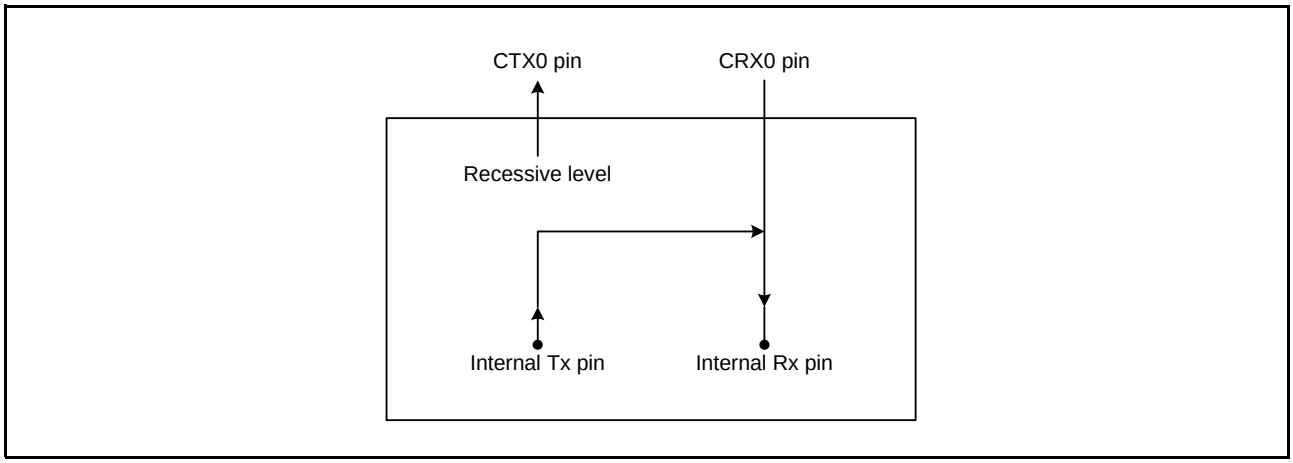


Figure 33.52 Listen-only Mode Configuration

33.9.1.3 Self Test Mode 0 (External Loop Back Mode)

In self test mode 0, the CAN engine treats its own transmitted messages as received messages via the CAN transceiver and can store them into its receive message buffers.
To be independent from external stimulation the engine generates its own acknowledge bit. This test can be used for CAN transceiver tests.
The CRX0 and CTX0 pins should be connected to the transceiver.

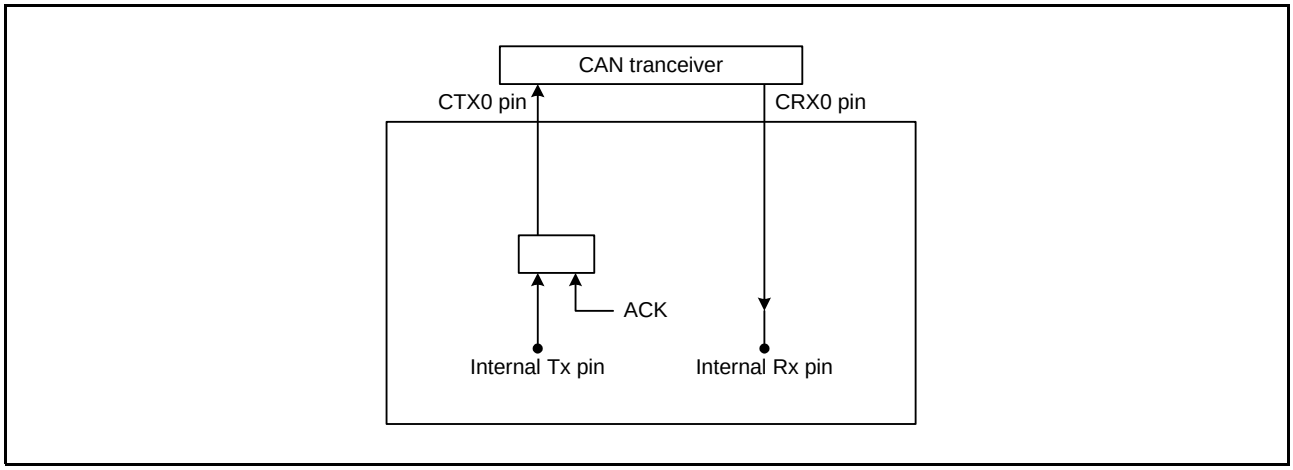


Figure 33.53 Self Test Mode 0 Configuration

33.9.1.4 Self Test Mode 1 (Internal Loop Back Mode)

In self test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation the CAN engine generates its own acknowledge bit. In this mode the CAN engine performs an internal feedback from internal Tx pin to internal Rx pin. The actual input level of the CRX0 pin is disregarded by the CAN engine.
The CTX0 pin outputs only recessive bits. The CRX0 and CTX0 pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the Internal CAN bus communication line.

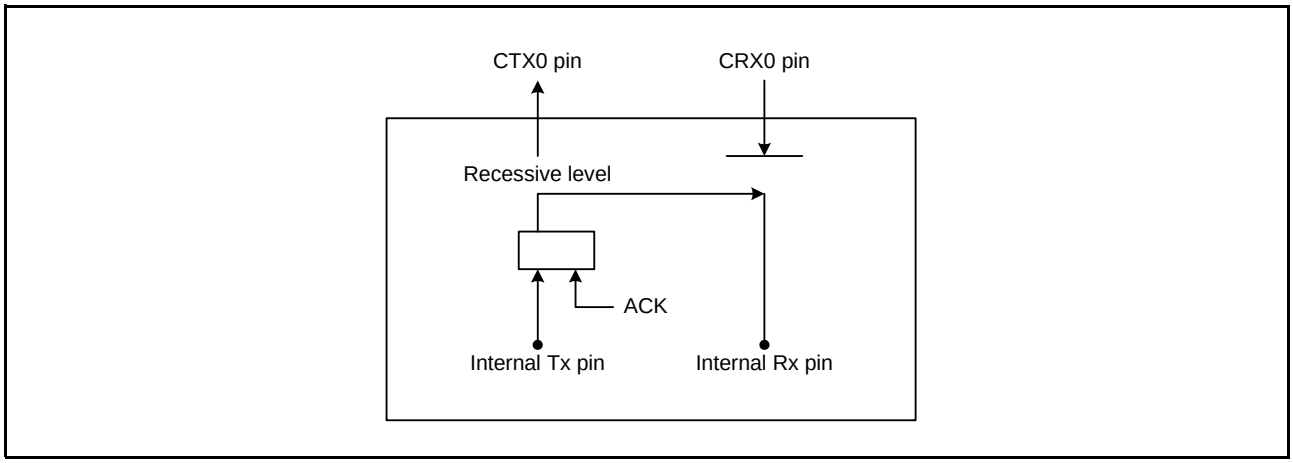


Figure 33.54 Self Test Mode 1 Configuration

33.9.1.5 Restricted Operation Mode

In restricted operation mode the CAN node is able to receive valid data and remote frames generating the acknowledge bit.
Active error and overload frames cannot be transmitted instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.
Moreover the receive error counter (CHSR.REC[7:0]) and transmit error counter (CHSR.TEC[7:0]) are frozen independently from the occurrence of errors.
The mode is specified as in ISO 11898-1; however it is permitted to set any transmit requested.

33.9.2 Global Test Modes

The CANFD module can be configured into following test modes:

- RAM test mode
- Bit flip test

For following test modes are protected by a special software procedure to enable the mode. This software procedure enables the write access to the test mode by specific unlock key, the related unlock key can be seen in the table below:

Table 33.25 Test Mode Unlock Key

Test Mode	Unlock Key 1	Unlock Key 2
RAM Test Mode	00007575h	00008A8Ah

If the software sequence of the two consecutive unlock key write accesses is interrupted by any other write access to the SFR or if incorrect data is written to the Global Test Mode Lock Key Register then the corresponding test mode cannot be set and the sequence should be re-started.
After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism resets and the test mode enable bit cannot be set and then the unlock sequence should be restarted.

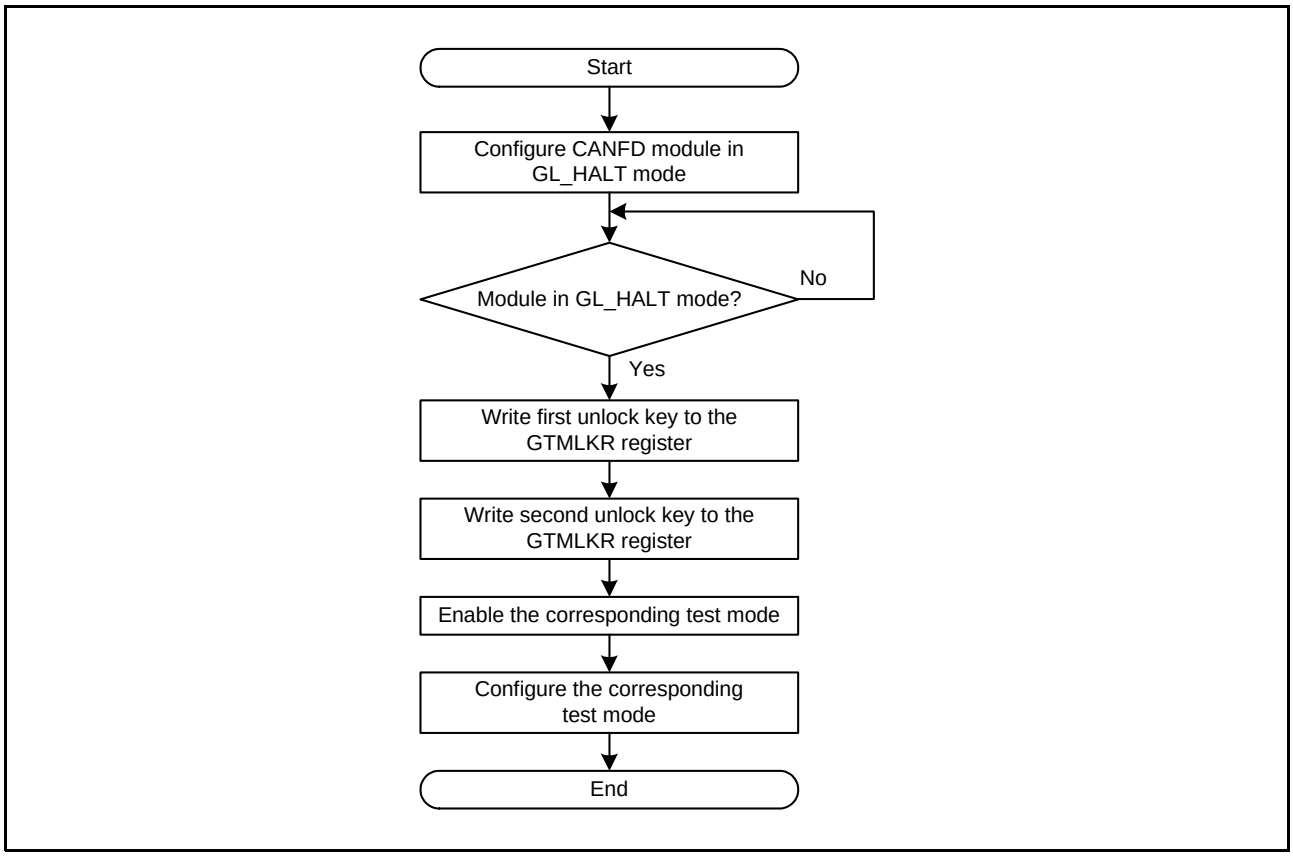


Figure 33.55 Unlock Software Protection Routine

33.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the GTMER.RTME bit when the corresponding lock key is written before. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after MCU reset. Hence, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages of 256 bytes each. Which can be accessed via RTPARk register (k = 0 to 63).
The page should be selected for read/write access by writing to the GTMCR.RTPS[3:0] bits. Then, data can be read from or written in to the RAM Test Page Access Register.
Figure 33.56 shows the structure of the pages in the RAM when performing a RAM test mode.

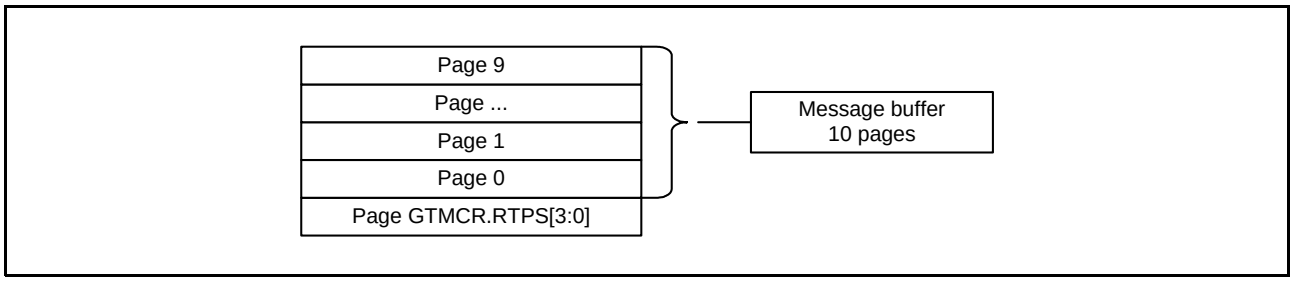


Figure 33.56 RAM Page Structure

The total available RAM size is, 2328 bytes for the message buffer RAM.
Total number of pages for the RAMs and the GTMCR.RTPS[3:0] values are calculated in the following way:

Total number of pages = ceil (total RAM size in bytes / number of bytes per page)

Message buffer RAM:

Total number of pages = ceil (2328 / 256) = 10 pages
GTMCR.RTPS[3:0] = 0 to 9

Figure 33.57 below shows the software flow for RAM test mode.

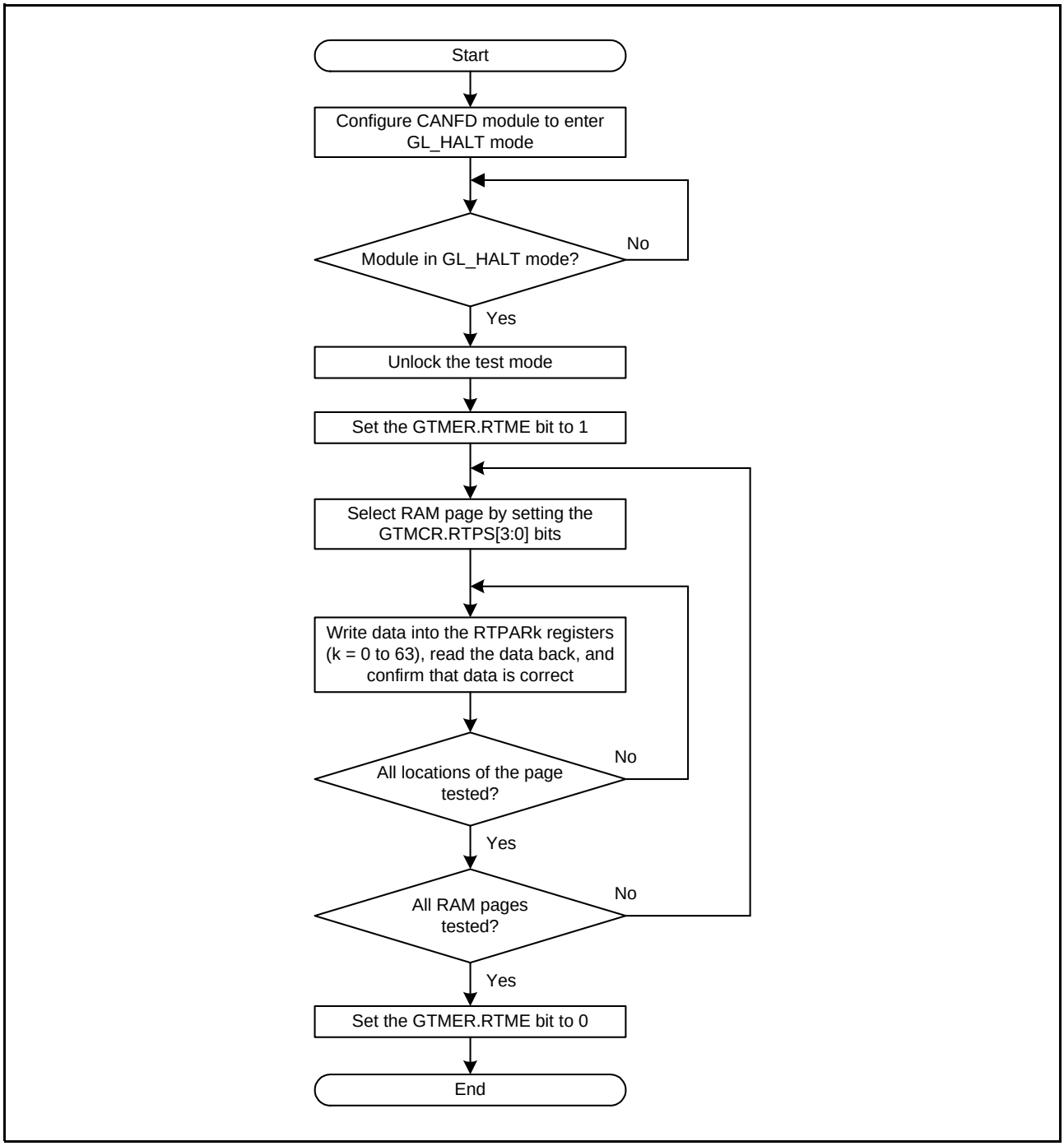


Figure 33.57 RAM Test Mode Software Flow

To exit RAM test mode, set the GTMER.RTME bit to 0.
The GTMER.RTME bit is automatically set to 0 when the CANFD module enters GL_RESET mode from the test mode.

33.9.2.2 Bit Flip Test

Bit flip test can invert the bit (the first bit of ID) of the beginning of the bit stream to receive.
If this function is used by a transmitting node, a bit error or an arbitration lost will occur.
If this function is used by a receiving node, a CRC error or a stuff error will occur.
Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.
The following sequence should be used to perform CRC error testing. In the sequence below CANFD module is the receiver.

1. Set the CTR.BFT bit to 1, in order to invert the first bit of the incoming bit stream from sending node
2. Wait for the can_cherr_int output signal to set to 1
3. Read either the CHESR.CRC15[14:0] or FDCRC.CRC21[20:0] bits (depending on the received frame type: Classical CAN or CAN FD). The value should be different from the received CRC value of the reference message from sending node.
4. Check that CHESR.CEDF is 1
   As the CRC generator logic is shared for receive and transmit there is no need to create a separate transmit CRC error test.

33.10 Interrupts and DTC/DMA Requests

33.10.1 CANFD Interrupts

The CANFD module generates several Interrupts. The interrupt output, which is connected to the interrupt controller, can be controlled by the corresponding interrupt enable bit.
The status flag will be set independent from this enable bit.
The channel transmit interrupt has an additional status flag register; these status bits will only be set when the corresponding interrupt enables are set.
This register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.
The interrupts in the CANFD module can be classified into two groups: global interrupts and channel interrupts.

(1) Global Interrupts

The CANFD module can generate the following three global interrupts.

1. Receive FIFO interrupt
2. Global error interrupt
3. Receive message buffer interrupt

(2) Channel Interrupts

The CAN channel can generate the following three channel interrupts.

1. Channel transmit interrupt
   (a) Successful transmission interrupt
   (b) Transmission abort interrupt
   (c) Transmit queue interrupt
   (d) Common FIFO transmission interrupt
   (e) Transmission history interrupt
2. Channel error interrupt
3. Common FIFO receive interrupt

The interrupts are cleared when the corresponding flag bits are cleared or interrupt enable bits are cleared.

Table 33.26 lists the interrupt sources for CANFD module.
To clear each interrupt request, clear all flags set to 1 from among the sources for which interrupts are enabled. The interrupt request can also be cleared by setting all corresponding interrupt enable bits to 0.

Table 33.26 Interrupt Sources

Interrupt Name			Interrupt Source Flag	Interrupt Enable Bit	Interrupt Status Flag
Global interrupts	Receive FIFO interrupt (RFRI)		RFSRn.RFIF	RFCRn.RFIE	—
	Global error interrupt (GLEI)		GESR.DEDF GESR.MLDF GESR.THLDF GESR.PODF	GCR.DEIE GCR.MLIE GCR.THLIE GCR.POIE	—
	Receive message buffer interrupt (RMRI)		RMNDR.NDF[n]	RMIER.RMIEn	—
Channel interrupts	Channel transmit interrupt (CHTI)	Transmission successful interrupt*1	TMSRn.TXRF[1]	TMIER0.TMIEn	TISR.TSIF0
		Transmission abort interrupt*1	TMSRn.TXRF[1:0] (TXRF[1:0] = 01b)	CHCR.TAIE	TISR.TAIF0
		Transmit queue interrupt	TQSR0.TQIF	TQCR0.TQIE	TISR.TQIF0
		Common FIFO transmission interrupt	CFSR0.CFTIF	CFCR0.CFTIE	TISR.CFTIF0
		Transmission history interrupt	THSR.THIF	THCR.THIE	TISR.THIF0
	Channel error interrupt (CHEI)		CHESR.BEDF CHESR.EWDF CHESR.EPDF CHESR.BOEDF CHESR.BORDF CHESR.OLDF CHESR.BLDF CHESR.ALDF FDSTS.ECOV FDSTS.SCOV FDSTS.TDCV	CHCR.BEIE CHCR.EWIE CHCR.EPIE CHCR.BOEIE CHCR.BORIE CHCR.OLIE CHCR.BLIE CHCR.ALIE CHCR.ECOVIE CHCR.SCOVIE CHCR.TDCVIE	—
	Common FIFO receive interrupt (CFRI)		CFSR0.CFRIF	CFCR0.CFRIE	—

Note 1. These interrupts are only generated for transmit message buffers that are not part of a valid transmit queue and are not linked to a common FIFO. The common FIFO and the transmit queue have other interrupts respectively.

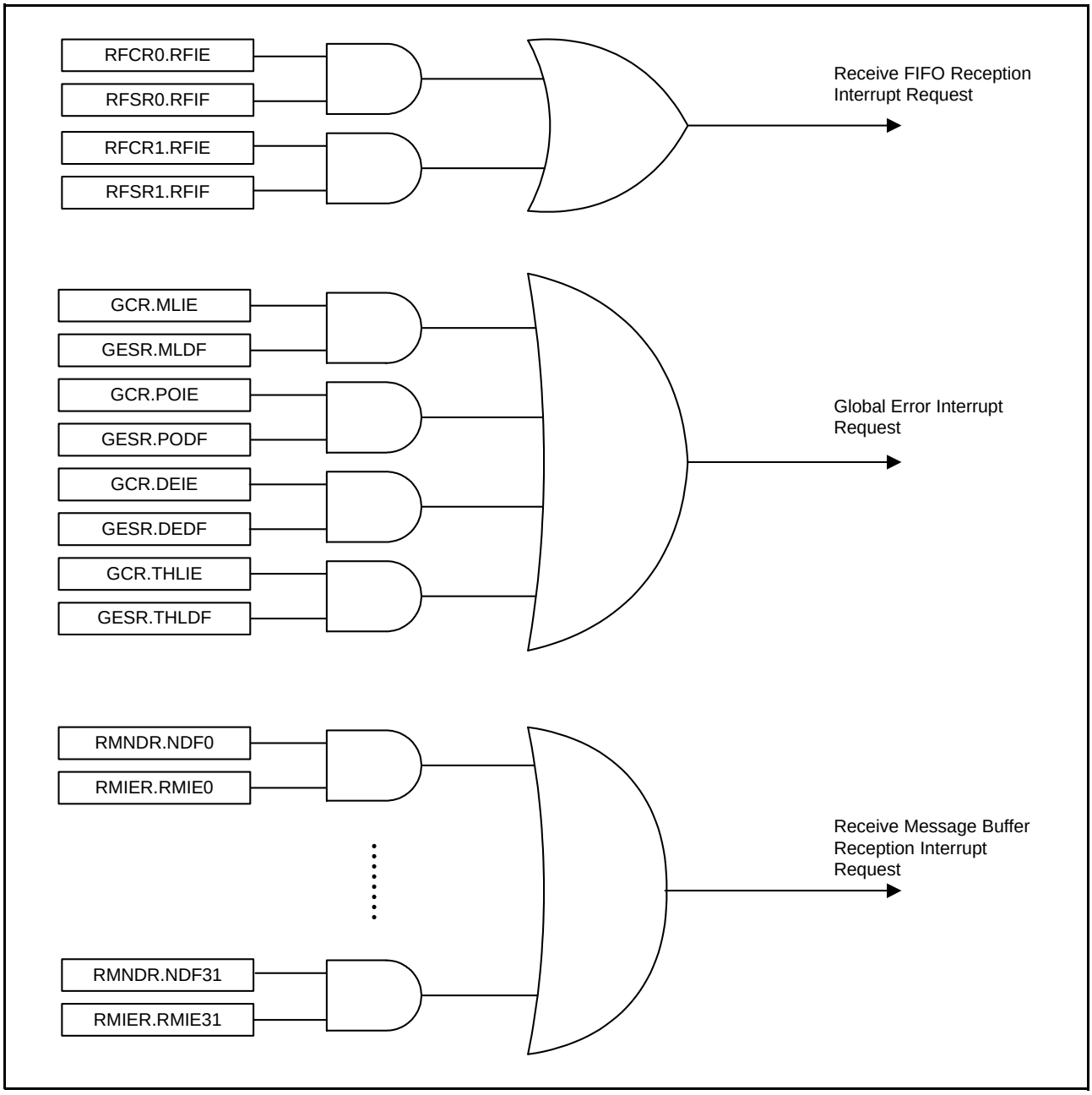


Figure 33.58 Global Interrupt Block Diagram

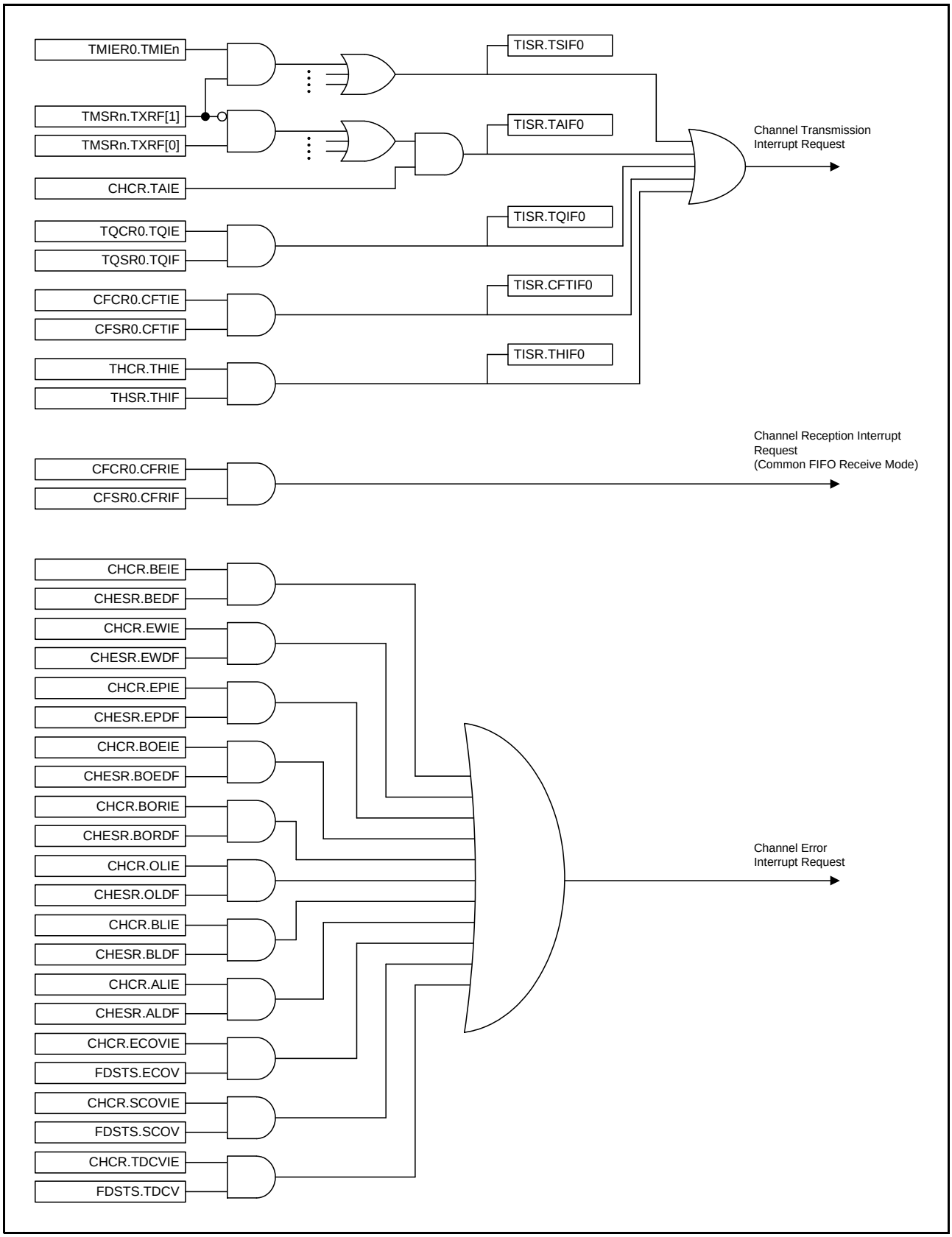


Figure 33.59 Channel Interrupt Block Diagram

33.10.2 ECC Interrupts

There are three types of interrupts generated by the ECC decoder:

- 1-bit ECC error detection interrupt
- 2-bit ECC error detection interrupt
- ECC overflow interrupt.

33.10.3 DTC/DMA Transfer Requests

The CANFD module has message buffers that can be read by DTC/DMA transfer:

- Two receive FIFO message buffers
- Common FIFO message buffer

A DTC/DMA transfer request is generated when the DTCR.RFDTE0, RFDTE1, or CFDTE0 bits are set to 1 and the corresponding FIFO is not empty.
For FIFOs with DTC / DMA transfer enabled, disable receive FIFO interrupts (the RFCR0.RFIE, RFCR1.RFIE or CFCR0.CFRIE bits).
Use the regular start address for the DMA access window address.
When the data of the specified payload size (the RFCR0.PLS[2:0], RFCR1.PLS[2:0] or CFCR0.PLS[2:0] bit) is read*1, the FIFO read pointer is automatically read.
When DTC/DMA transfer is permitted, do not write to the FIFO pointer control register (RFPCR0, RFPCR1, or CFPCR0).

Note 1. The DTC/DMA should read the exact length of the specified data payload size (the RFCR0.PLS[2:0], RFCR1.PLS[2:0] or CFCR0.PLS[2:0] bit).

The permission for DTC/DMA transfer (the DTCR.RFDTE0, RFDTE1 or CFDTE0 bit) can be set to 1 at any time. Figure 33.60 shows the DTC/DMA transfer configuration flow.

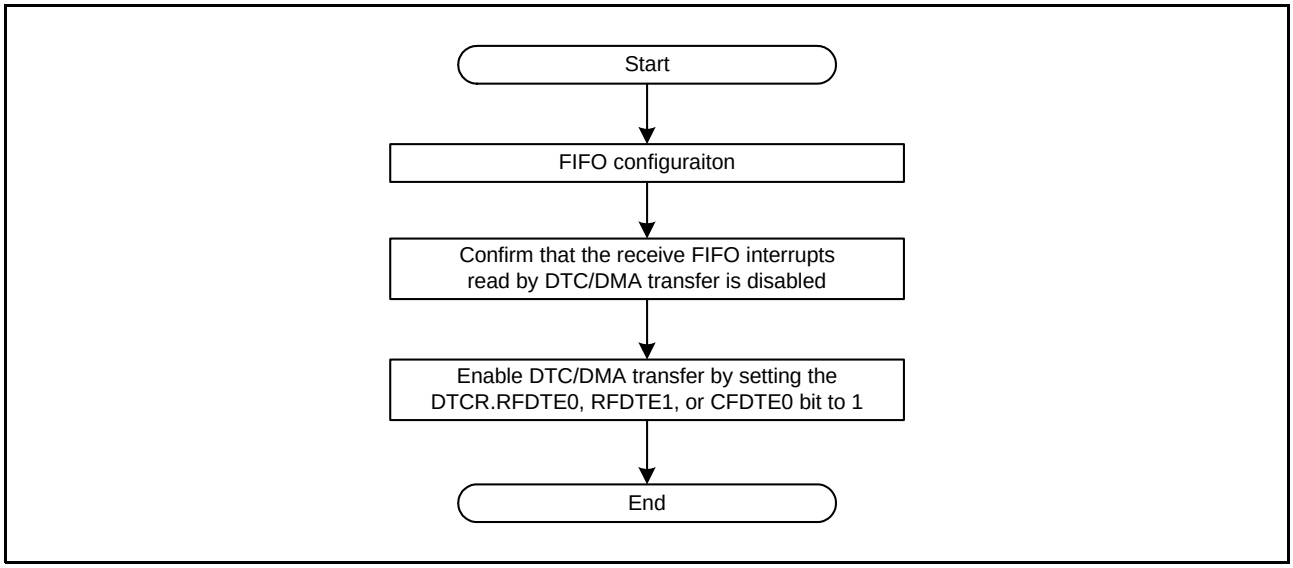


Figure 33.60 DTC/DMA Transfer Enable Flow

To disable the DTC/DMA transfer, set the corresponding DMA transfer enable bit (the DTCR.RFDTE0, RFDTE1 or CFDTE0 bit) to 0. If the disable is made during a DTC/DMA transfer, wait until the ongoing transfer is complete before performing the following operations. The transfer status can be confirmed by the DTSR.RFDTS0, RFDTS1 or CFDTS0

bit. Figure 33.61 shows the DTC/DMA transfer disable flow.
When the DTC/DMA transfer is disabled then consider what to do with the remaining or new incoming messages to this particular reception FIFOs. When the FIFO is not disabled then reception to the FIFO will continue.

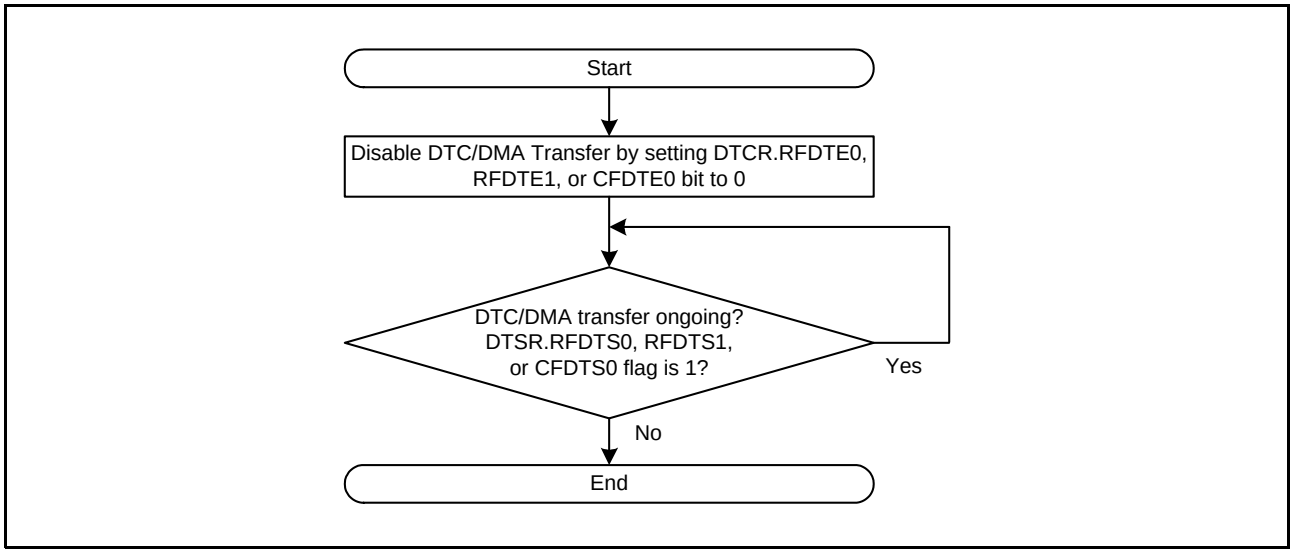


Figure 33.61 DTC/DMA Transfer Disable Flow

33.11 Usage Notes

33.11.1 Setting the Module Stop Function

The module-stop control register D (MSTPCRD) can be used to stop and start the CANFD module operations. After release from a reset, the CANFD module is placed in the module-stop state. The registers of the module become accessible after release from the module-stop state. For details, refer to section 11, Low Power Consumption.

33.11.2 Note on Configuration of Receive Message Buffers and FIFO Buffers

The maximum memory size available for receive message buffers and FIFO buffers is 1216 bytes.
For example, if all payload sizes are set to 8 bytes, each message size becomes 20 bytes, so the total number of messages must be 60 or less. If 32 receive message buffers are reserved, the total FIFO depth must be 28 or less.
Similarly, if all payload sizes are set to 64 bytes, each message size becomes 76 bytes, so the total number of messages must be 16 or less. If all FIFO depths are set to 4 messages, up to four receive message buffers can be used.
Operation is not guaranteed if the setting exceeds 1216 bytes.

34. Serial Peripheral Interface (RSPId)

In this section, “PCLK” is used to refer to PCLKA.

34.1 Overview

This MCU includes one channel 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 34.1 lists the specifications of the RSPI, and Figure 34.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

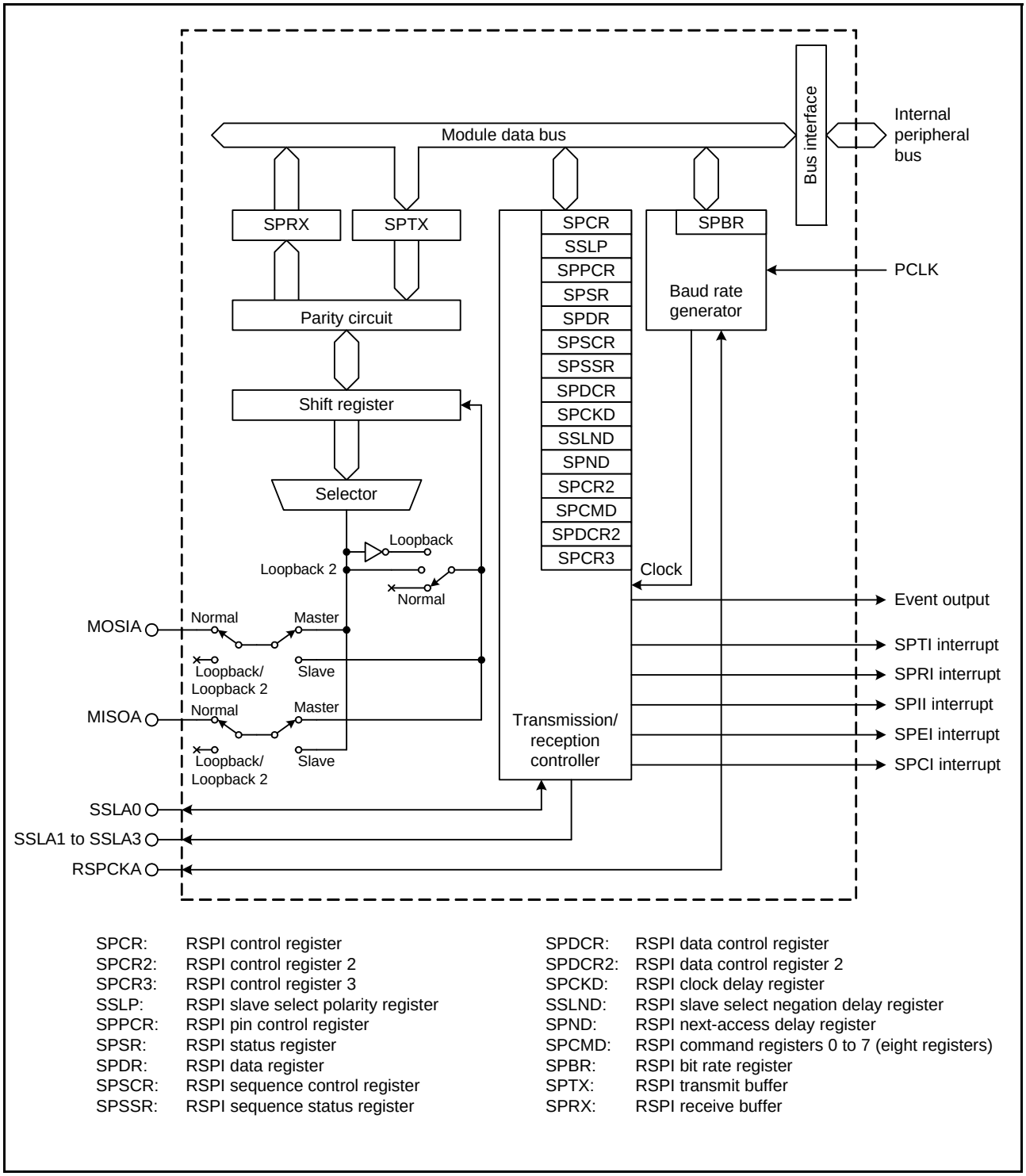
Table 34.1 RSPI Specifications (1/2)

Item	Description
Number of channels	One channel
RSPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • 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 (SSLA0 to SSLA3) for each channel • In single-master mode, SSLA0 to SSLA3 pins are output. • In multi-master mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for either output or unused. • In slave mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for unused. • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function • Delay between data bytes under burst transfer can be reduced.

Table 34.1 RSPI Specifications (2/2)

Item	Description
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	SPDCR:	RSPI data control register
SPCR2:	RSPI control register 2	SPDCR2:	RSPI data control register 2
SPCR3:	RSPI control register 3	SPCKD:	RSPI clock delay register
SSLP:	RSPI slave select polarity register	SSLND:	RSPI slave select negation delay register
SPPCR:	RSPI pin control register	SPND:	RSPI next-access delay register
SPSR:	RSPI status register	SPCMD:	RSPI command registers 0 to 7 (eight registers)
SPDR:	RSPI data register	SPBR:	RSPI bit rate register
SPSCR:	RSPI sequence control register	SPTX:	RSPI transmit buffer
SPSSR:	RSPI sequence status register	SPRX:	RSPI receive buffer

Figure 34.1 RSPI Block Diagram

Table 34.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 34.3.2, Controlling RSPI Pins for details.

Table 34.2 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

34.2 Register Descriptions

34.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 000D 0100h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select*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 SSLA0 to SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). 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 34.3.6, 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 34.3.10, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 34.3.2, Controlling RSPI Pins).

MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, and SSLA0 to SSLA3.

SPEIE Bit (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 34.3.10, 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 34.3.10, 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 34.3.11, 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.

34.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 000D 0101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active low 1: SSL0 signal is active high	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active low 1: SSL1 signal is active high	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is active low 1: SSL2 signal is active high	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active low 1: SSL3 signal is active high	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not change the SSLP register while the SPCR.SPE bit is 1.

34.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 000D 0102h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIA pin during MOSI idling corresponds to low 1: The level output on the MOSIA pin during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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 MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

34.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 000D 0103h

	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 34.3.10.1, 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 SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error.

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error.
- 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 SSLAi 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 SSLA0 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 SSLA0 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 SSLA0 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.

34.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 000D 0104h

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

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

- When accessing in word size

Address(es): RSPI0.SPDR.H 000D 0104h

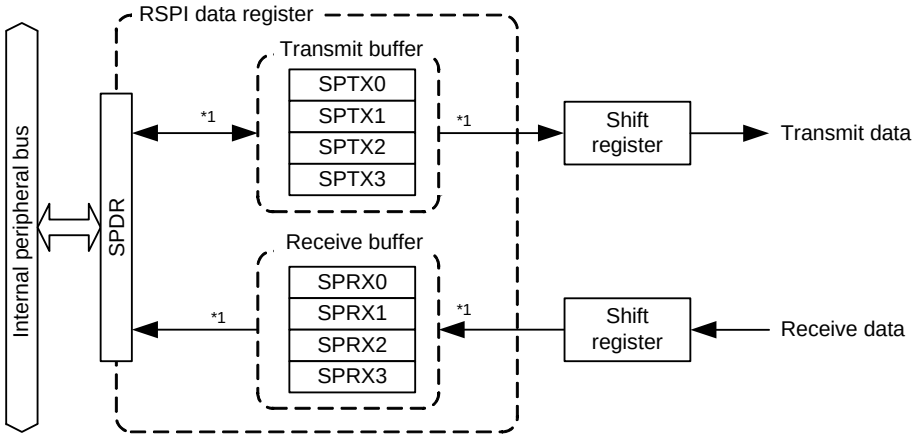
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): RSPI0.SPDR.HH 000D 0104h

	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 34.2 shows the Configuration of the SPDR Register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 34.2 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 34.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

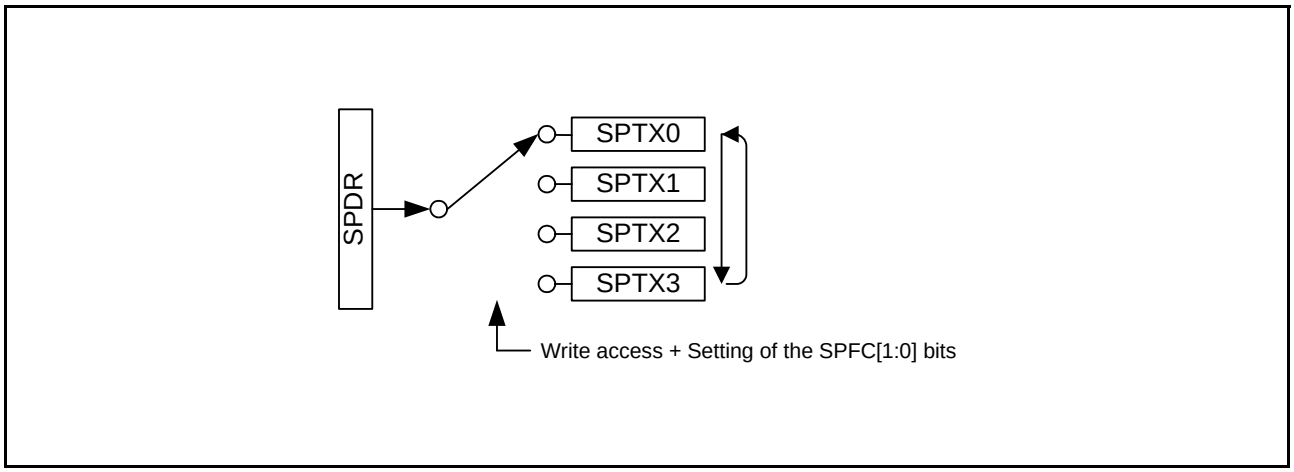


Figure 34.3 Configuration of the SPDR Register (Writing)

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 34.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

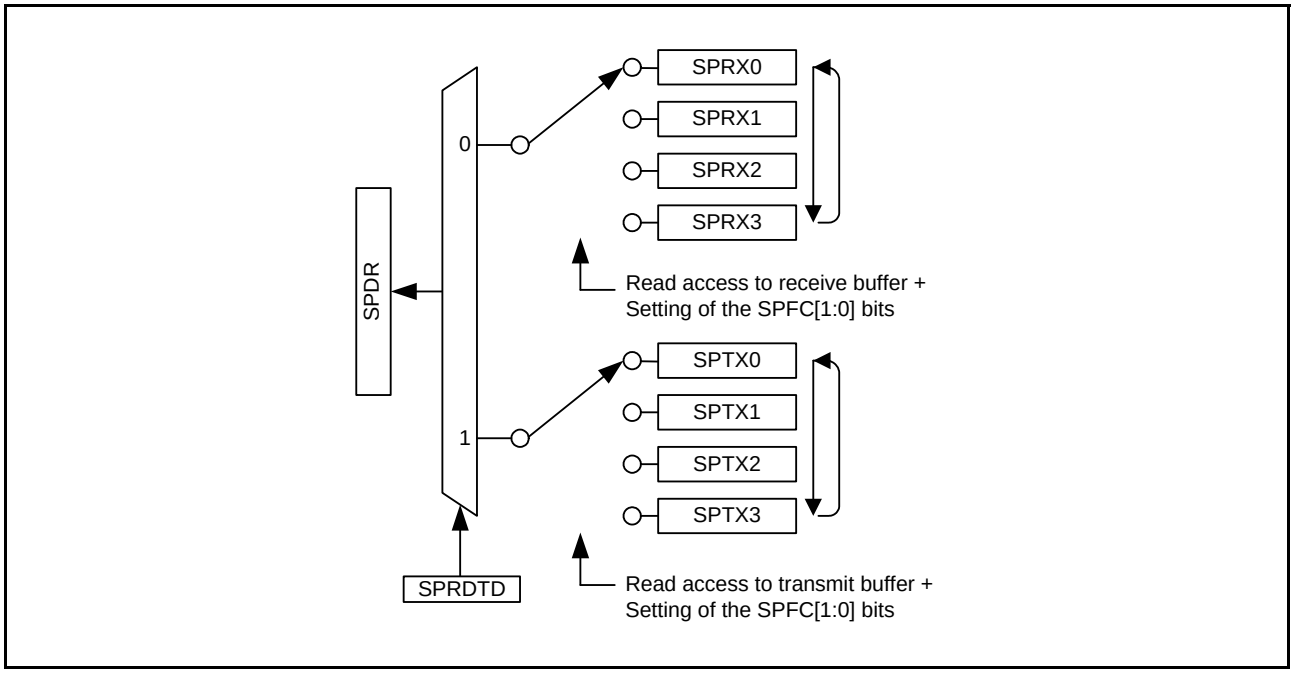


Figure 34.4 Configuration of the SPDR Register (Reading)

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).

34.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 000D 0108h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 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.

34.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 000D 0109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

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 34.3.12.1, 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 34.3.10, Error Detection. For the RSPI's sequence control, refer to section 34.3.12.1, Master Mode Operation.

34.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 000D 010Ah

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

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 34.3 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 34.3 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

34.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 000D 010Bh

	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 34.4 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 34.4 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 34.2.5, 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.

34.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 000D 010Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPCKD register sets a period from the beginning of SSLAi 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.

34.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 000D 010Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. 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.

34.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 000D 010Eh

	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 SSLAi 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.

34.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 000D 010Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable*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 34.3.10.1, Overrun Error.

34.2.14 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

	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 34.2.8, 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 SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 34.3.12.1, 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 SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

34.2.15 RSPI Data Control Register 2 (SPDCR2)

Address(es): RSPI0.SPDCR2 000D 0120h

	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 34.3.4.3 Byte Swap Transmission and 34.3.4.4 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.

34.2.16 RSPI Control Register 3 (SPCR3)

Address(es): RSPI0.SPCR3 000D 0121h

	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.

34.3 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.

34.3.1 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 34.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 34.5 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
RSPCKA signal	Input	Output	Output/Hi-Z*1	Input	Output
MOSIA signal	Input	Output	Output/Hi-Z*1	Input	Output
MISOA signal	Output/Hi-Z*2	Input	Input	Output	Input
SSLA0 signal	Input	Output	Input	Hi-Z*3	Hi-Z*3
SSLA1 to SSLA3 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 SSLA0 is asserted by another master device, the pin becomes Hi-Z.
Note 2. When SSLA0 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.

34.3.2 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 34.6.

Table 34.6 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIA Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

34.3.3 RSPI System Configuration Examples

34.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 34.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this MCU should be connected to the SSL input of the slave device.

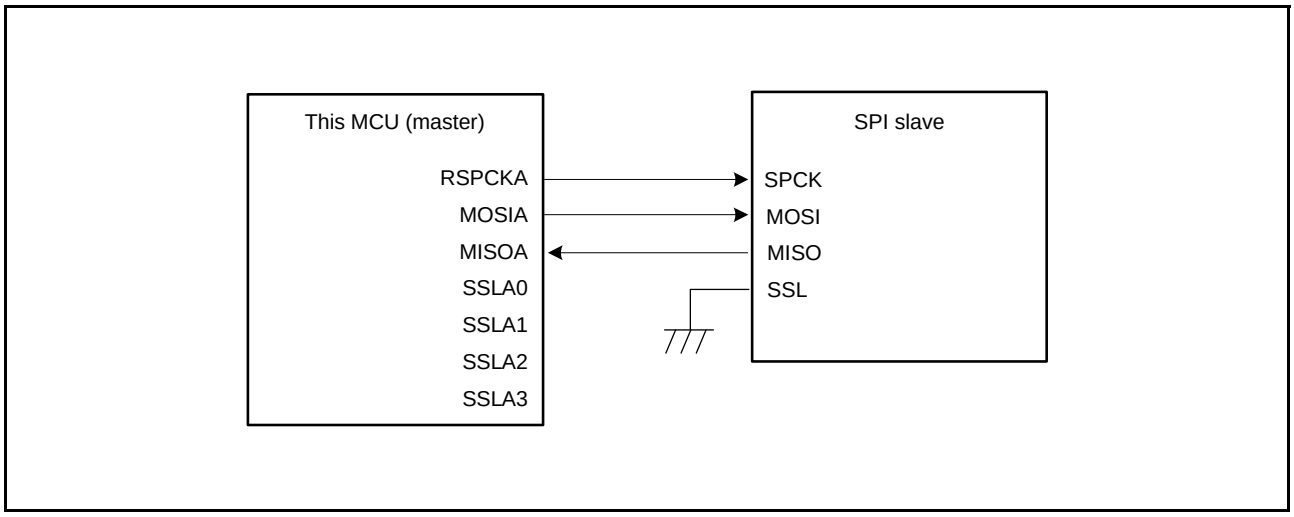


Figure 34.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

34.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 34.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 34.7). In this configuration, the communication end interrupt and the communication end event are not generated.

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

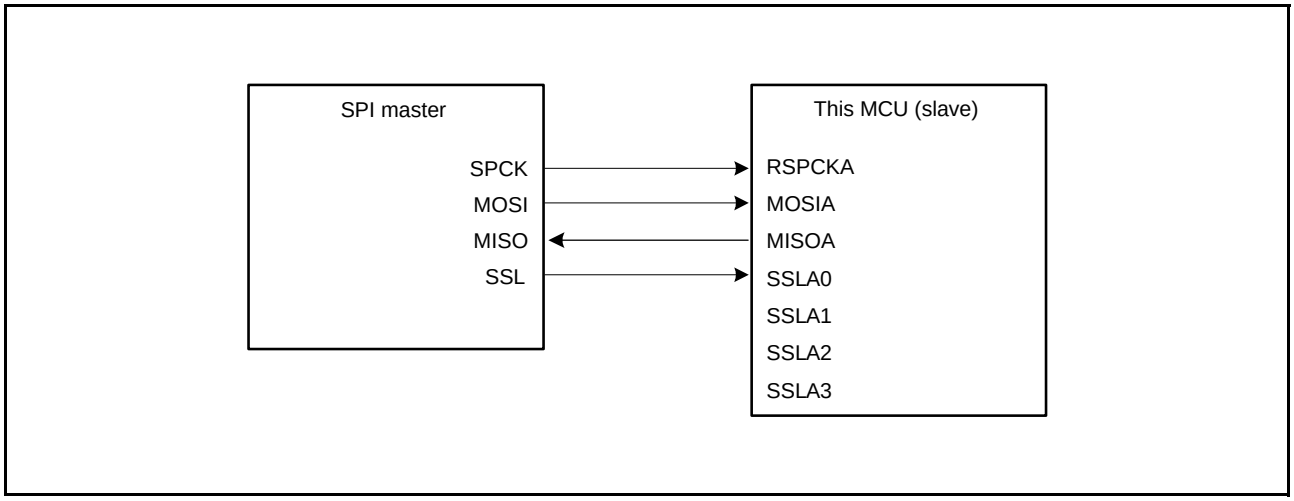


Figure 34.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

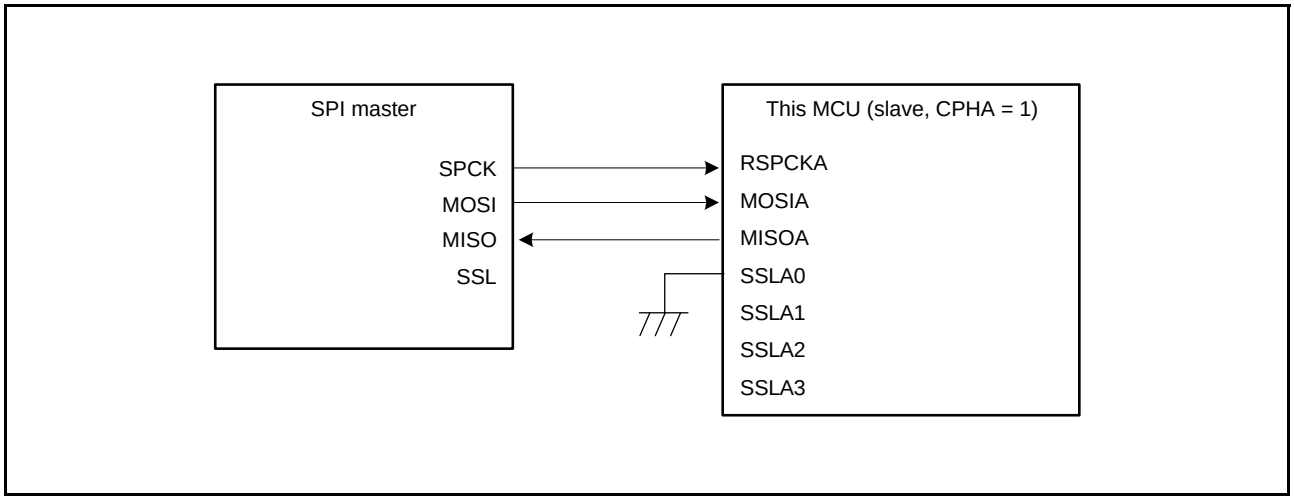


Figure 34.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

34.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 34.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 34.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0 to SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCKA, MOSIA, and SSLA0 to SSLA3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

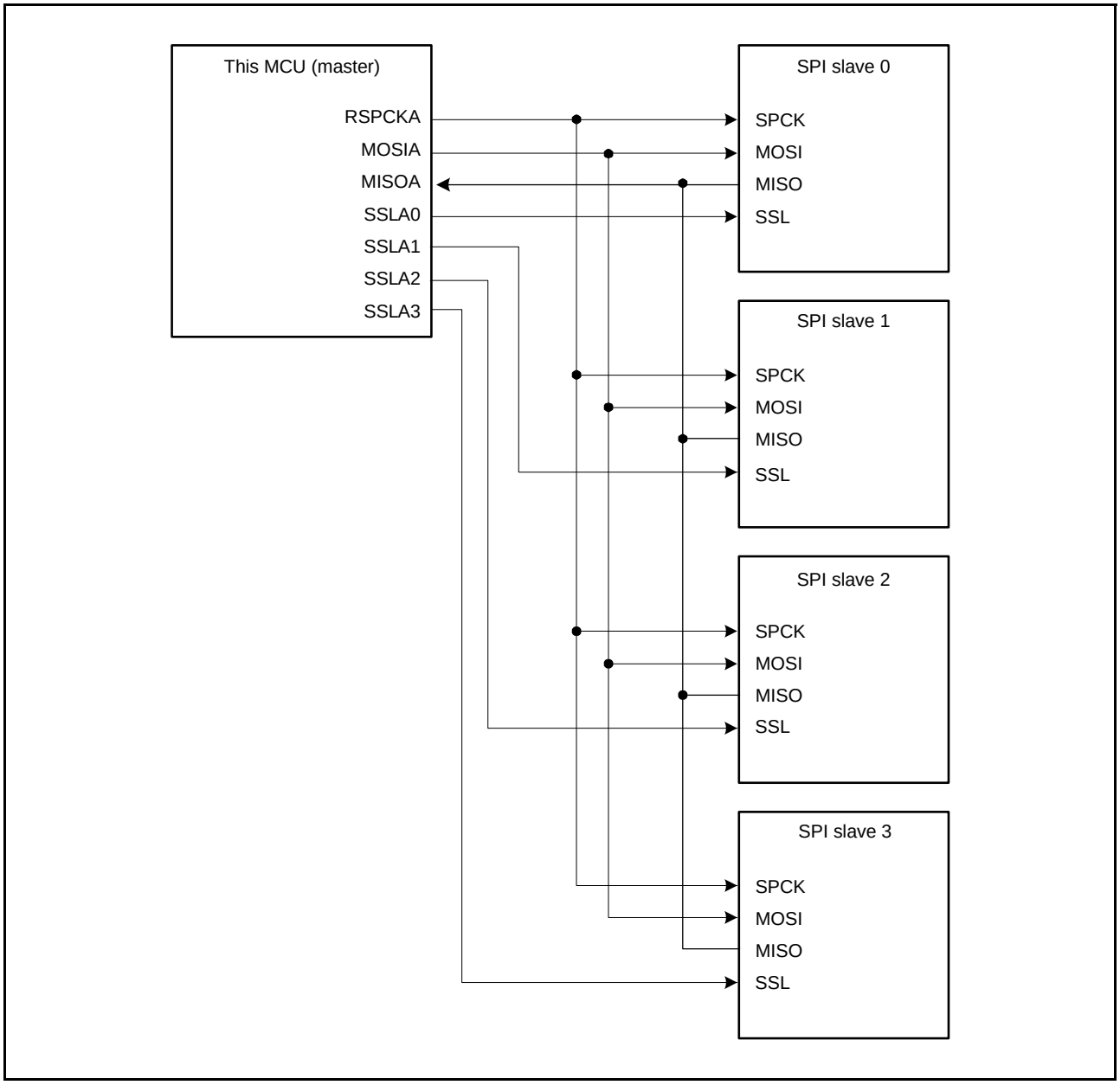


Figure 34.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

34.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 34.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 34.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).
The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

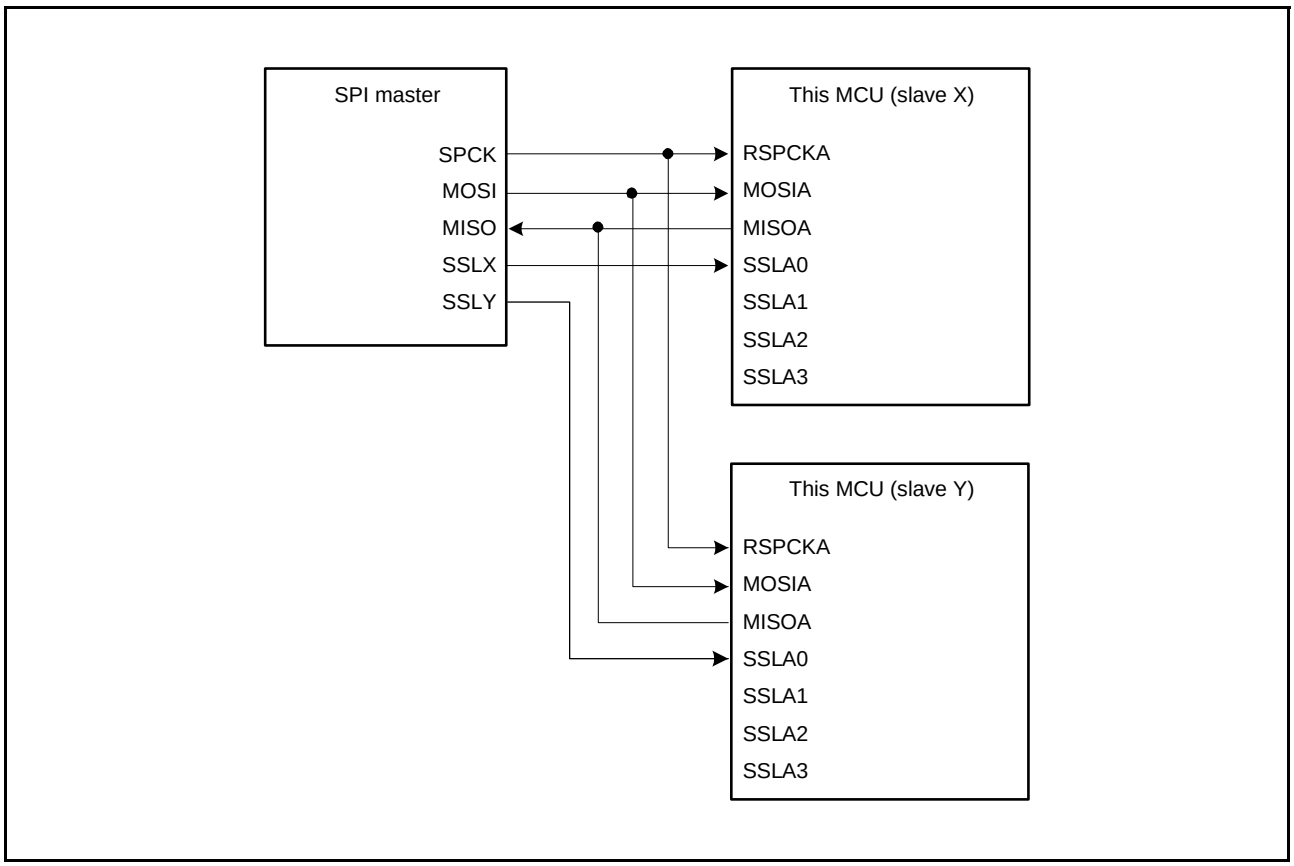


Figure 34.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

34.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 34.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 34.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this MCU is not required.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

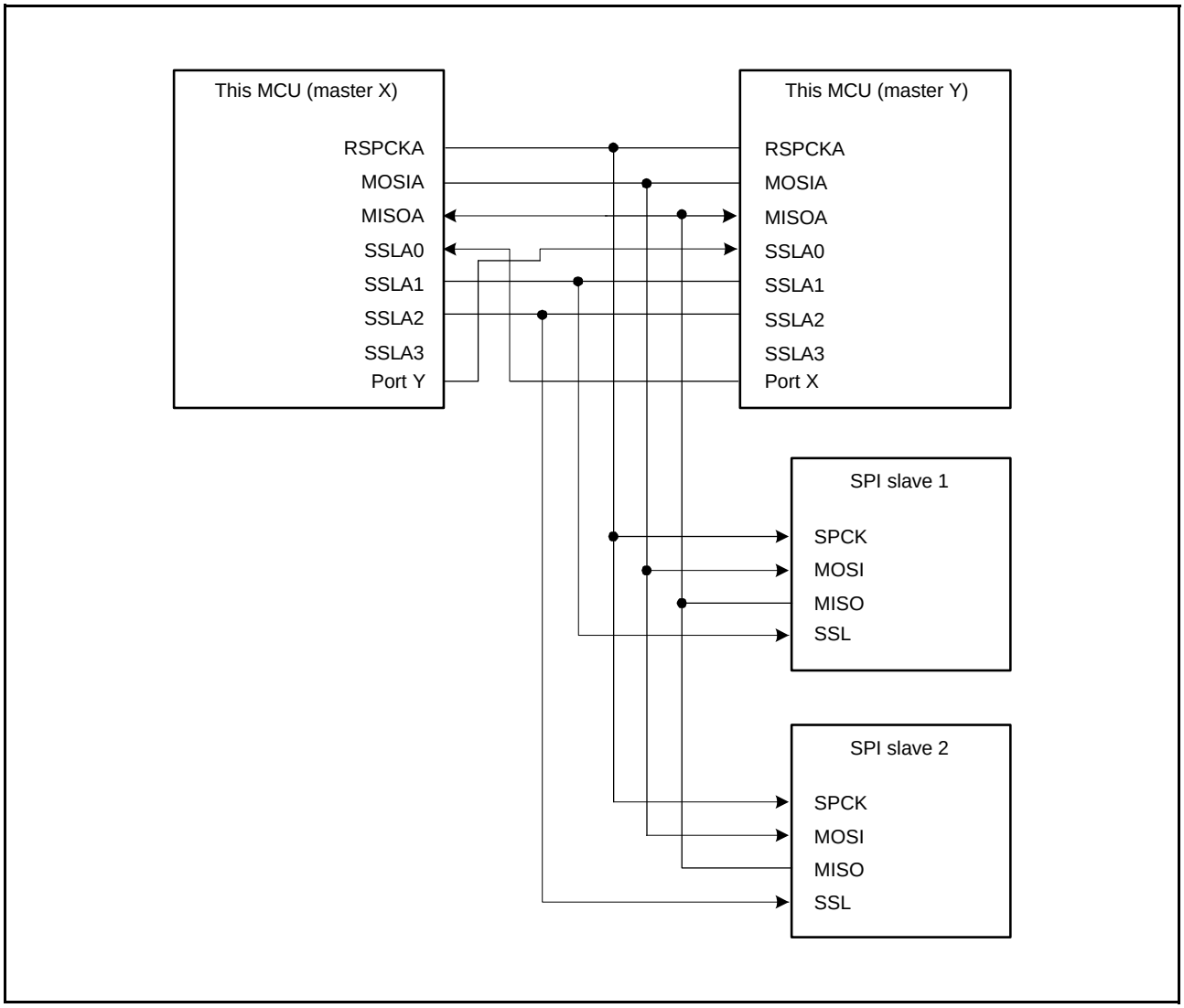


Figure 34.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

34.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 34.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

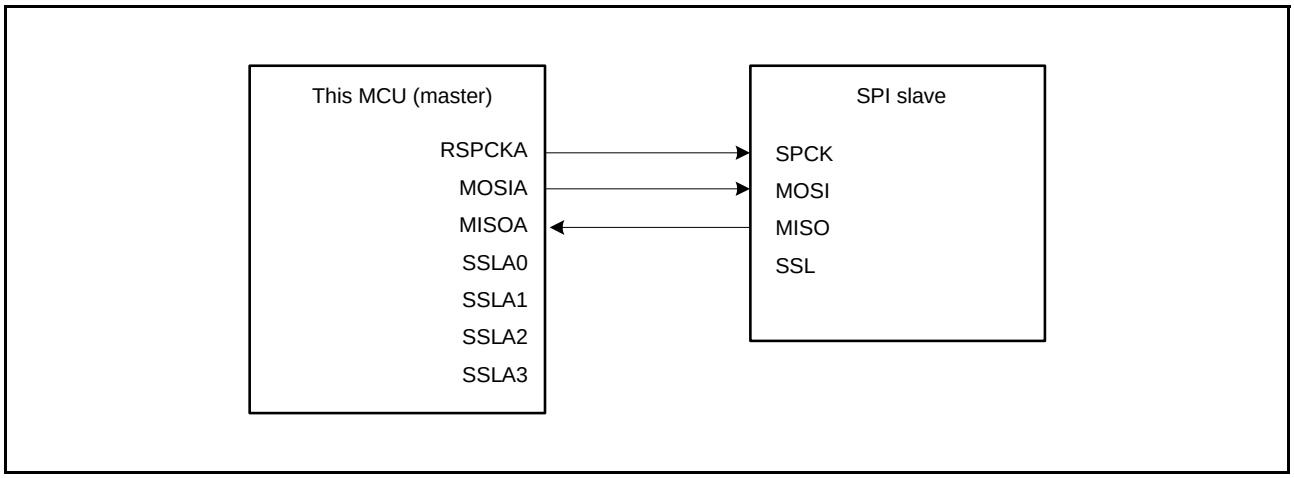


Figure 34.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

34.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 34.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0 to SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

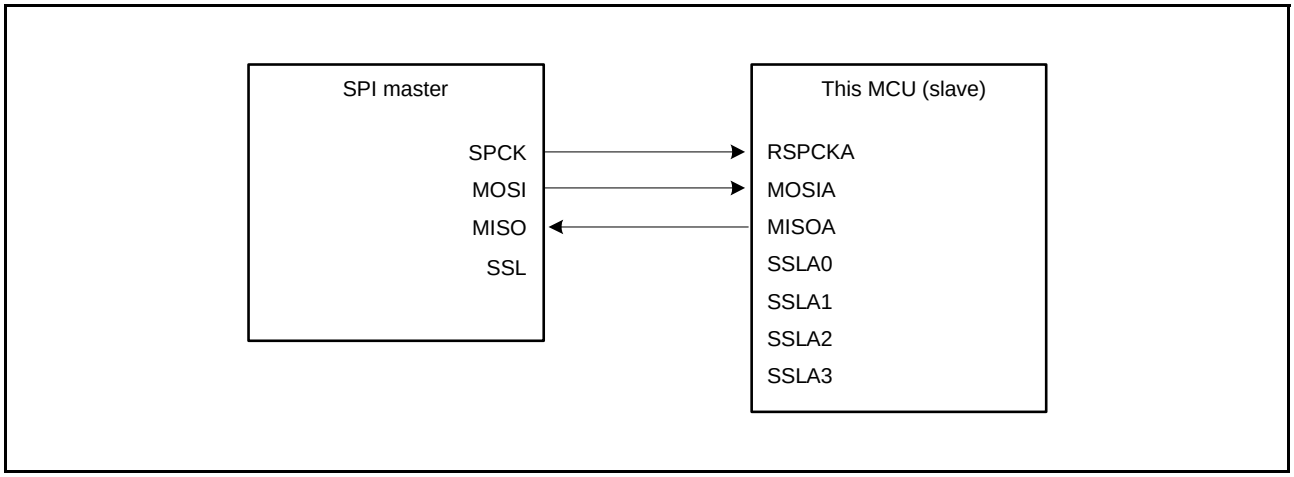


Figure 34.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

34.3.4 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.

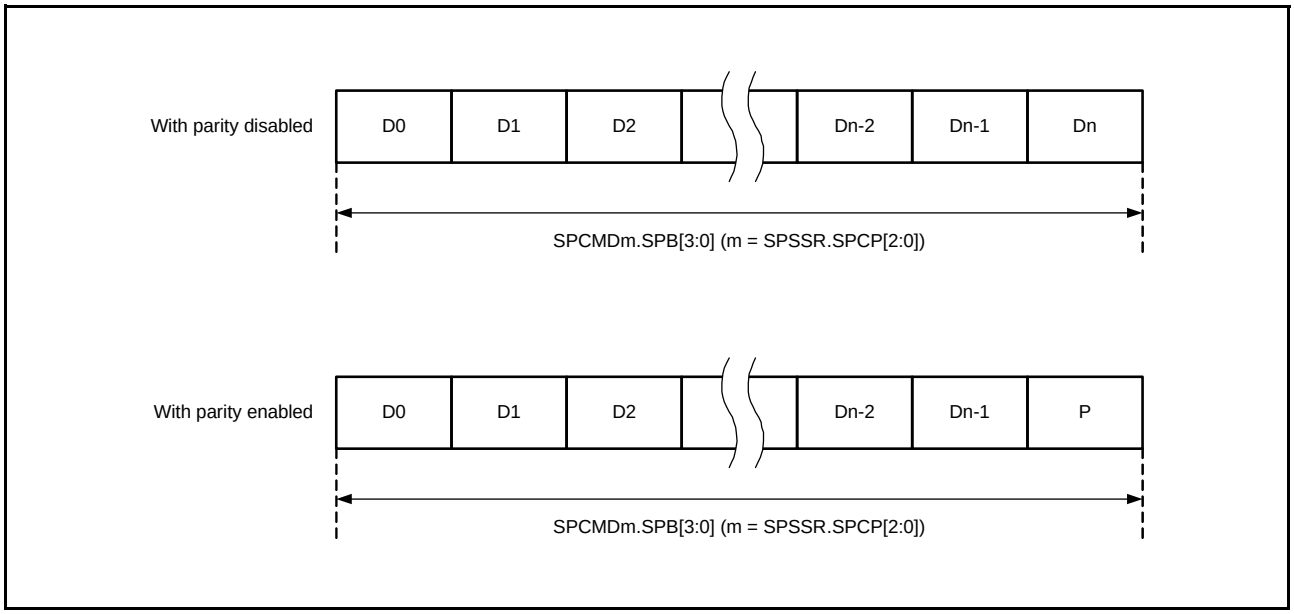


Figure 34.13 Outline of the Data Format (with Parity Disabled/Enabled)

34.3.4.1 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 34.14 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.

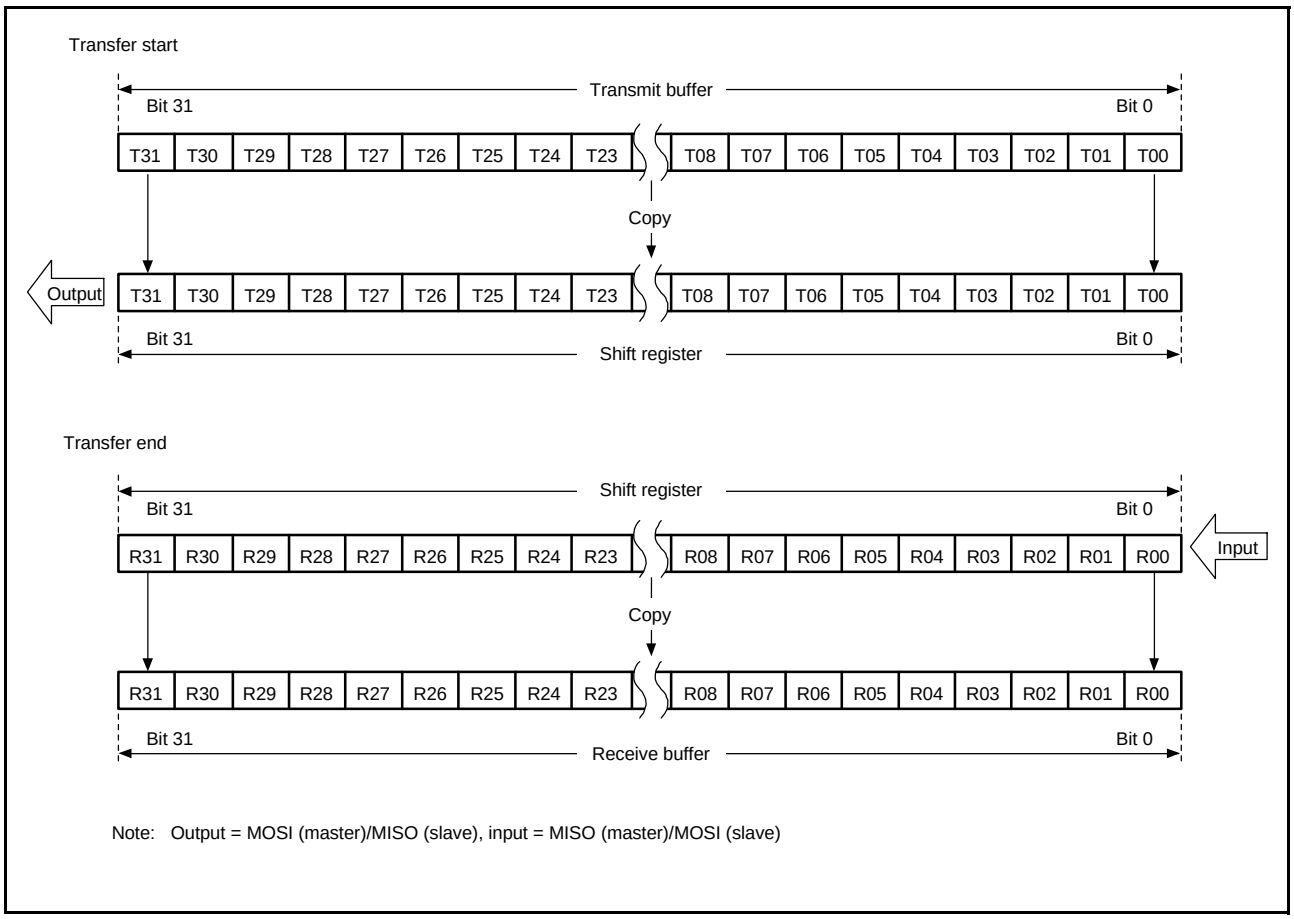


Figure 34.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 34.15 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.

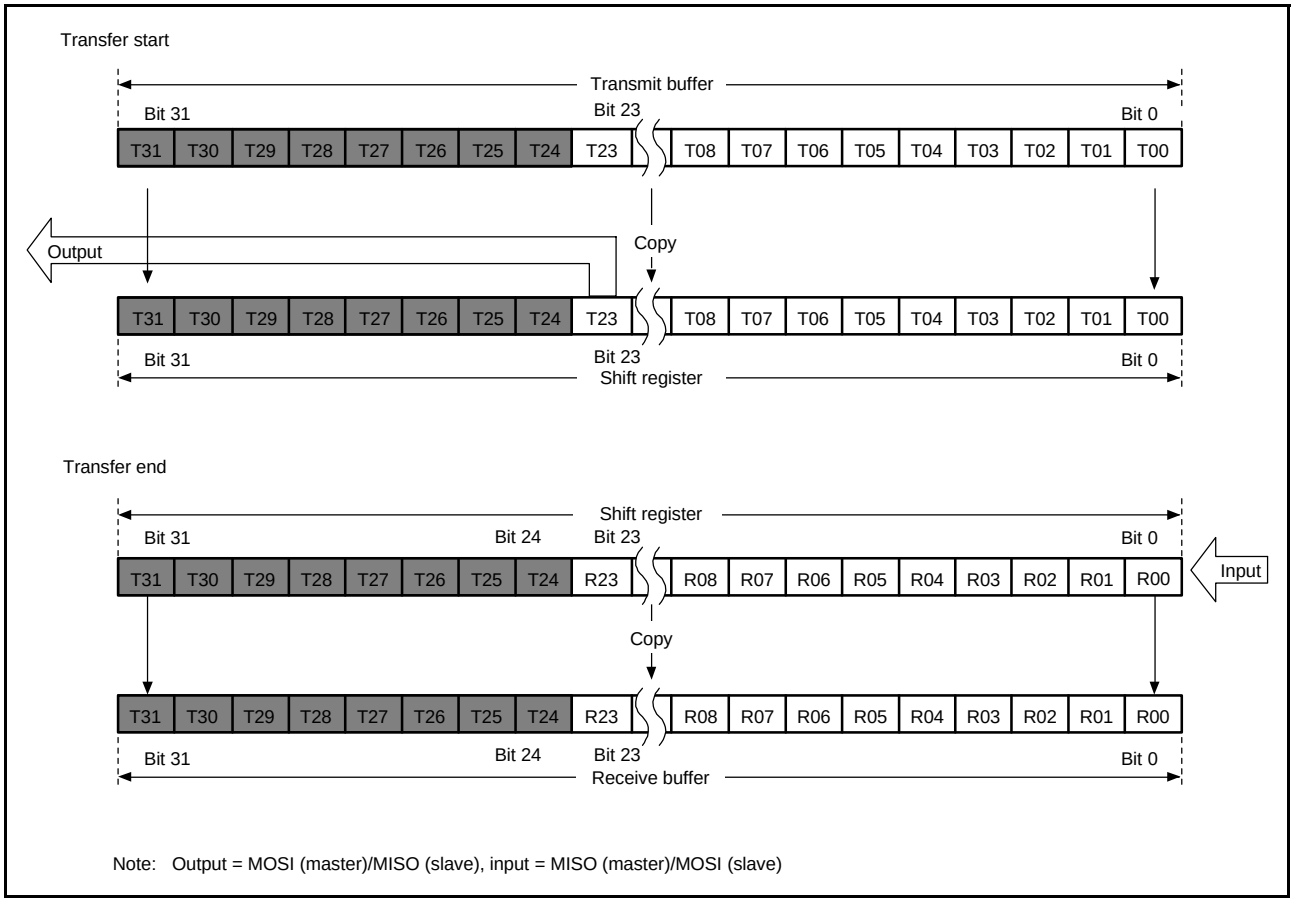


Figure 34.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 34.16 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.

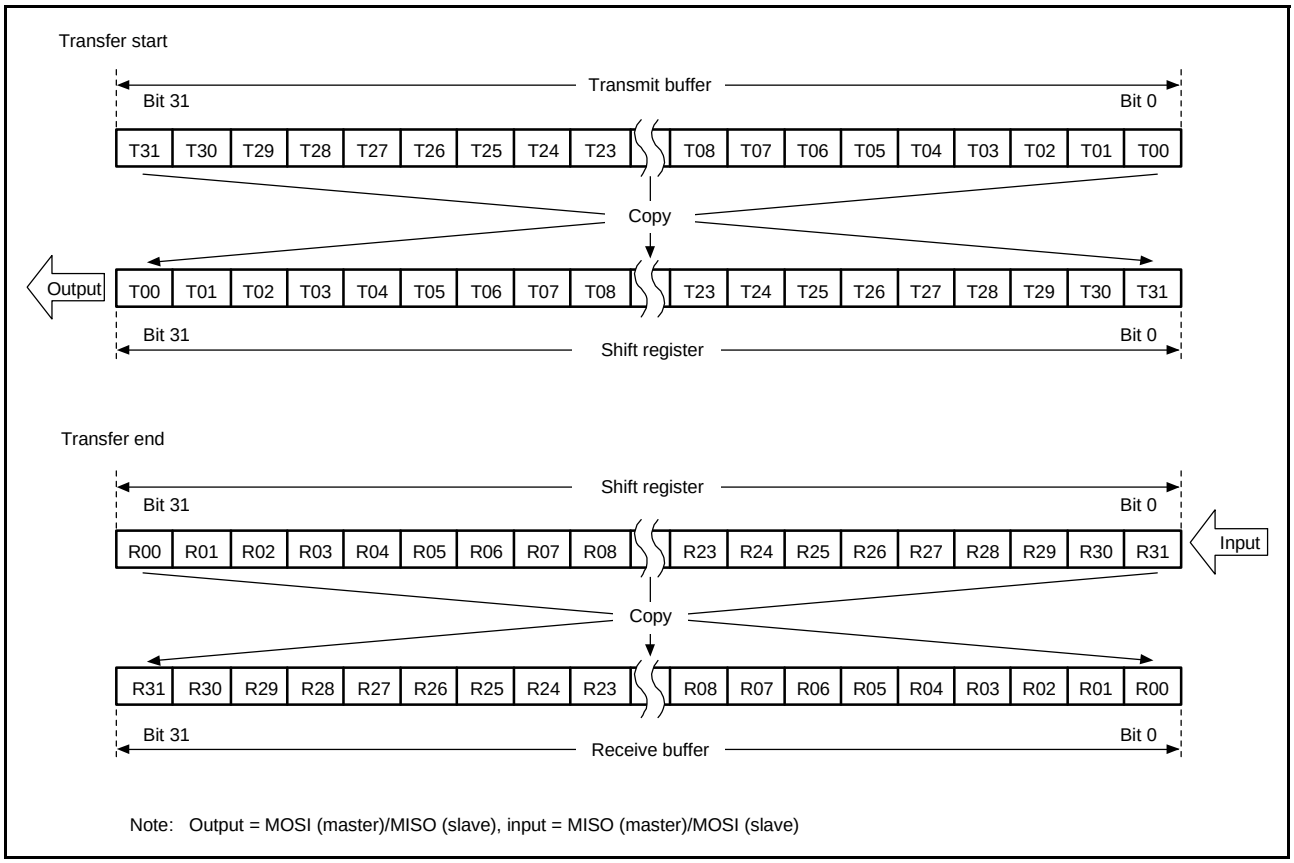


Figure 34.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 34.17 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.

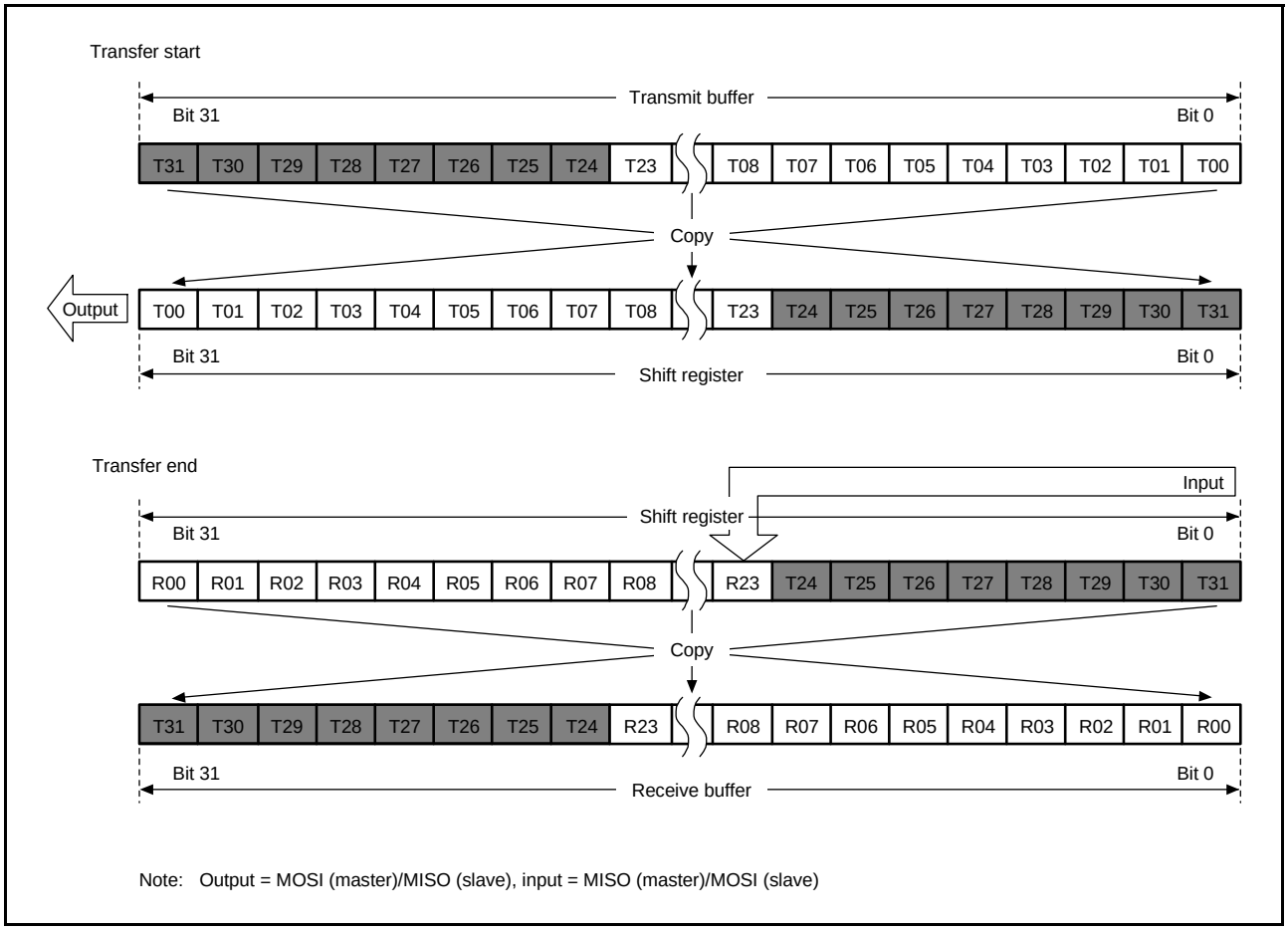


Figure 34.17 LSB First Transfer (24-Bit Data, Parity Disabled)

34.3.4.2 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 34.18 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.

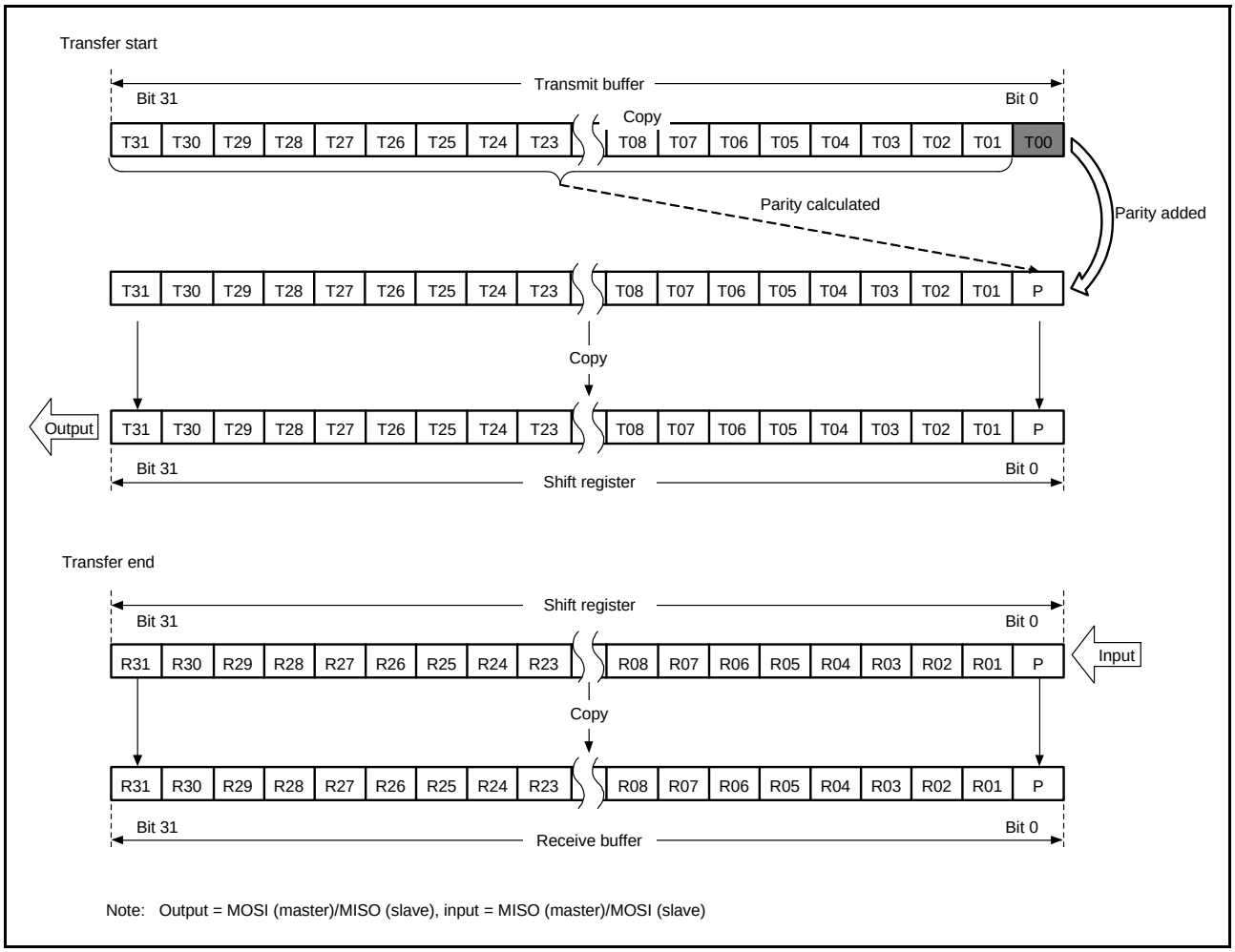


Figure 34.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 34.19 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.

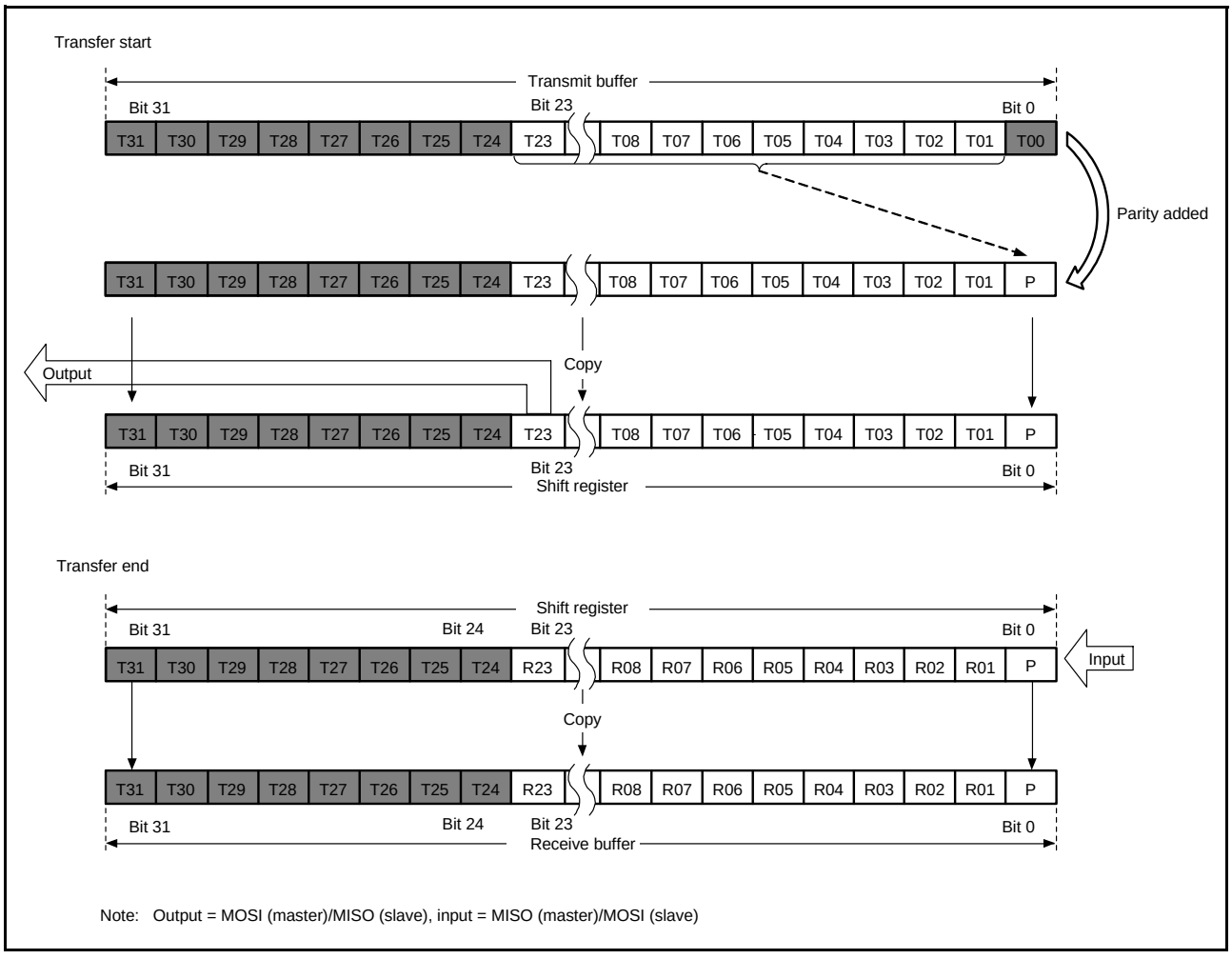


Figure 34.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 34.20 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.

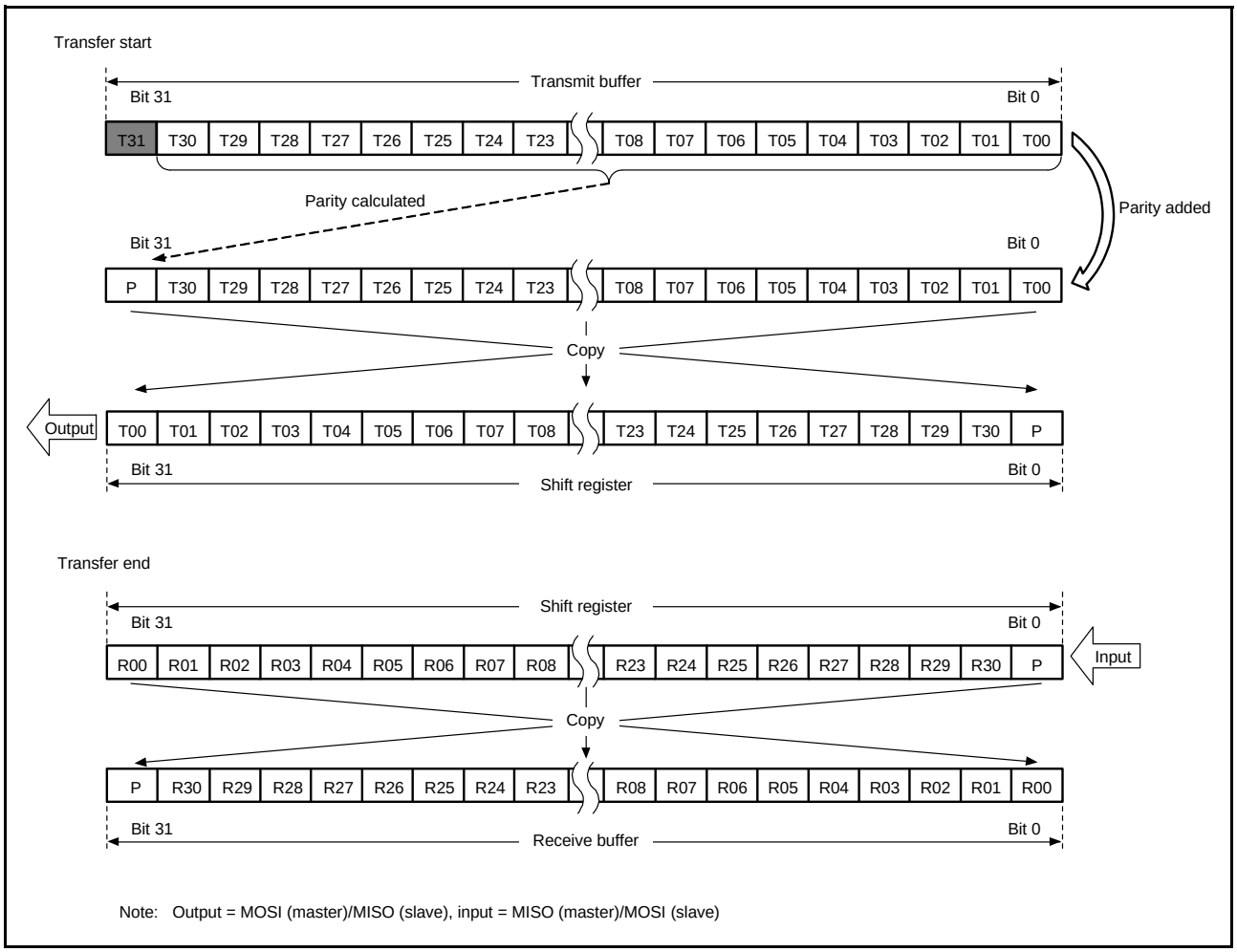


Figure 34.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 34.21 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.

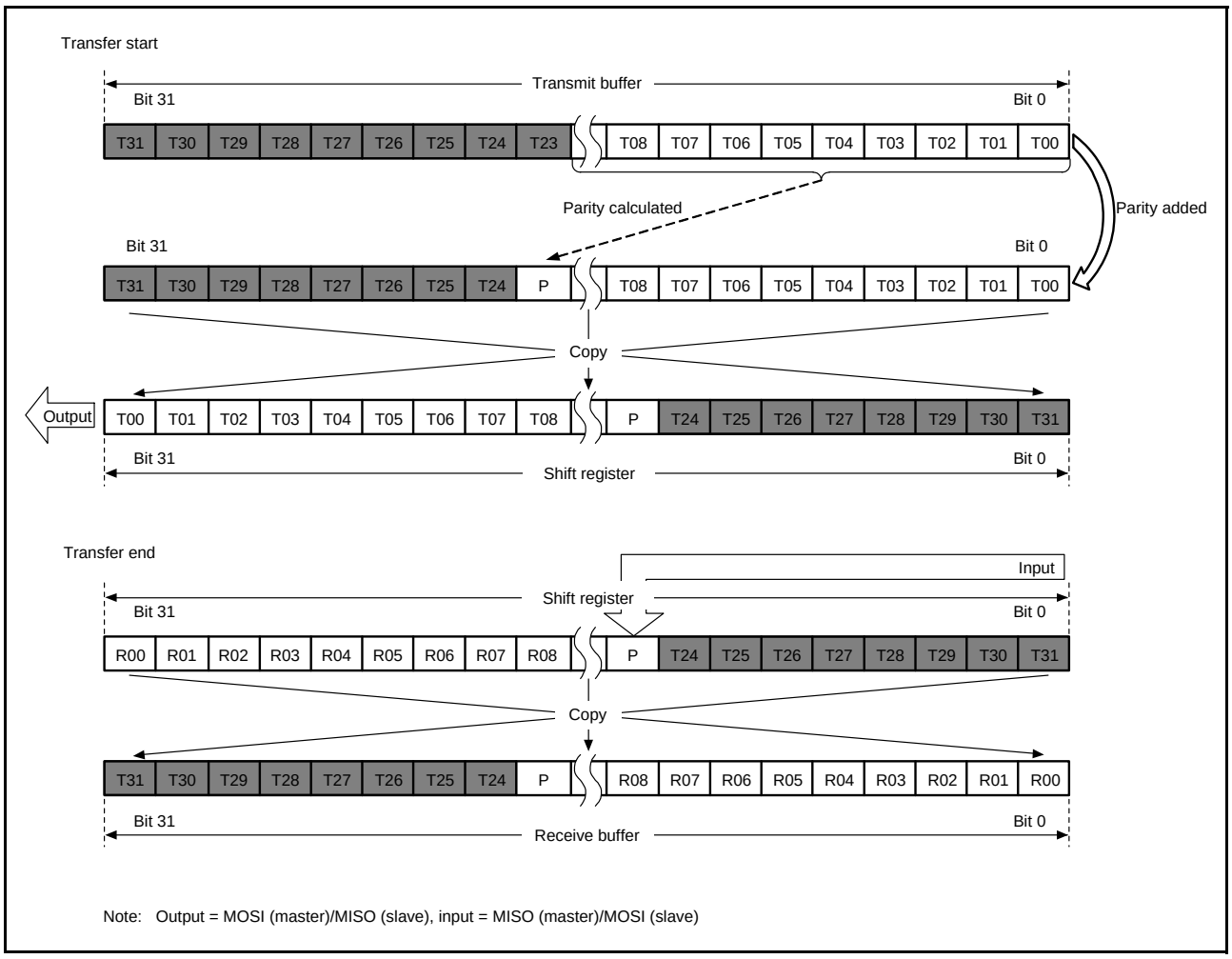


Figure 34.21 LSB First Transfer (24-Bit Data, Parity Enabled)

34.3.4.3 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 34.22 shows data transfer between the SPDR register and the shift register when data length is 32 bits.

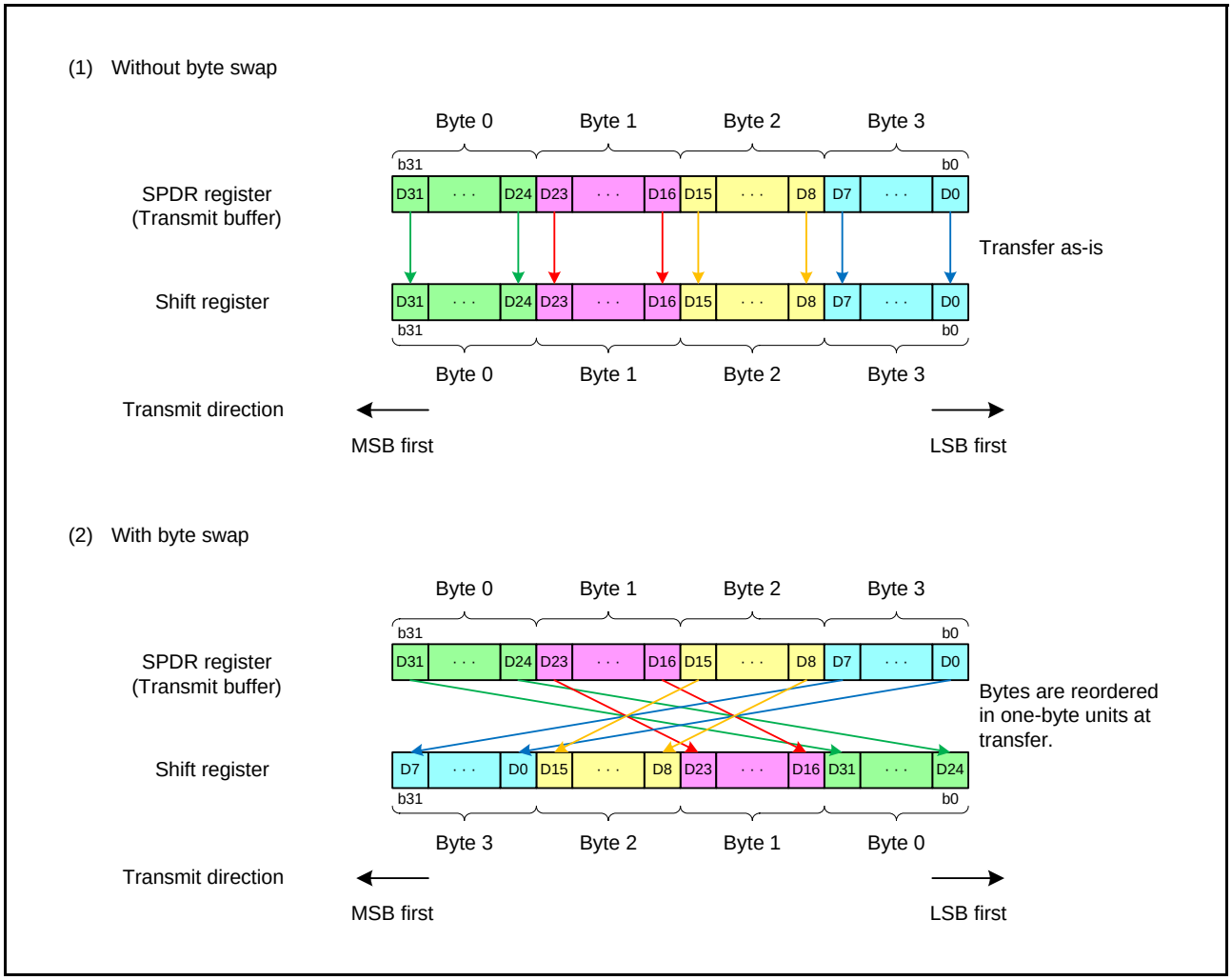


Figure 34.22 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

34.3.4.4 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 34.23 shows data transfer between the shift register and the SPDR register when data length is 32 bits.

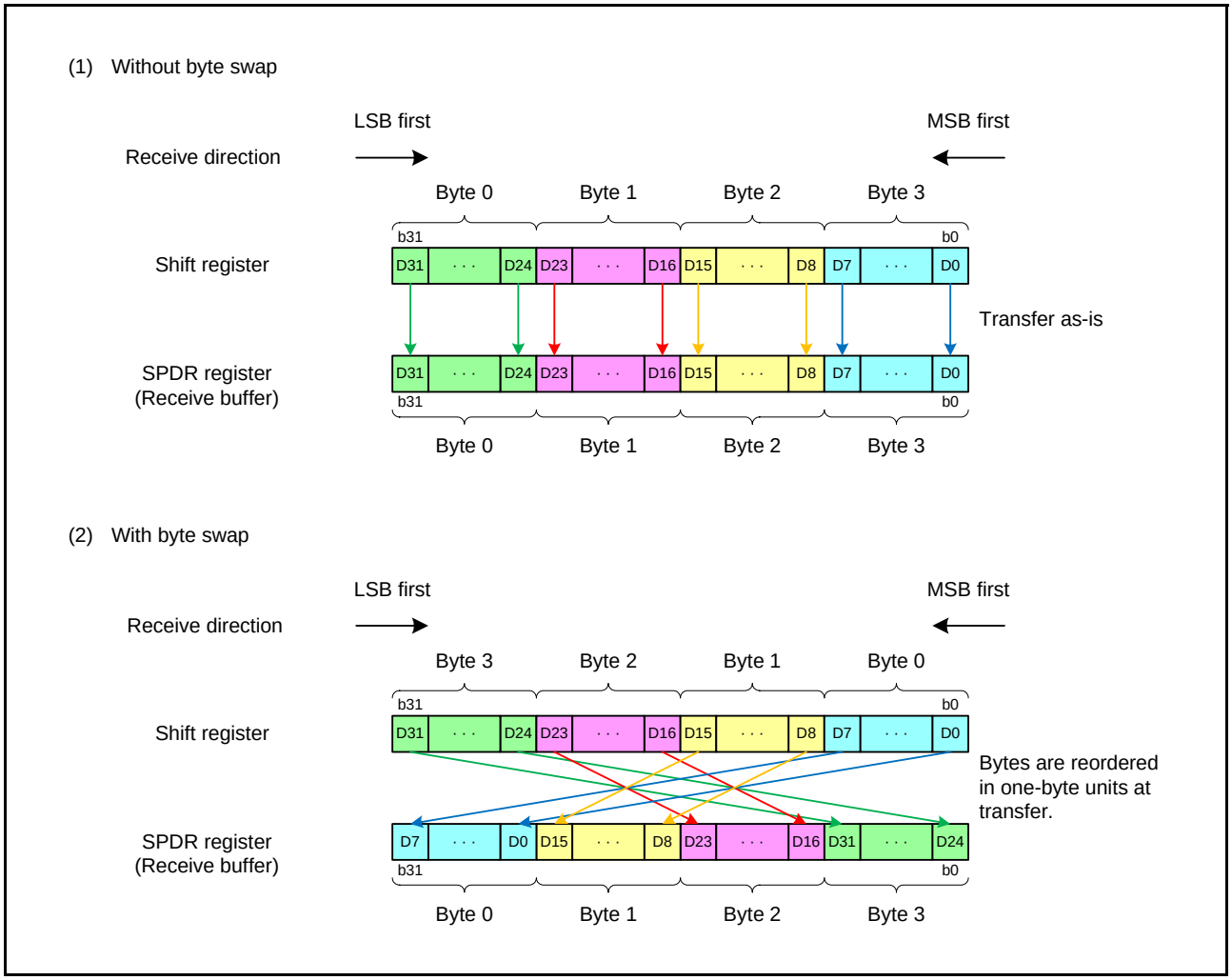


Figure 34.23 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

34.3.5 Transfer Format

34.3.5.1 CPHA = 0

Figure 34.24 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 34.24, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 34.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 34.3.12.1, Master Mode Operation.

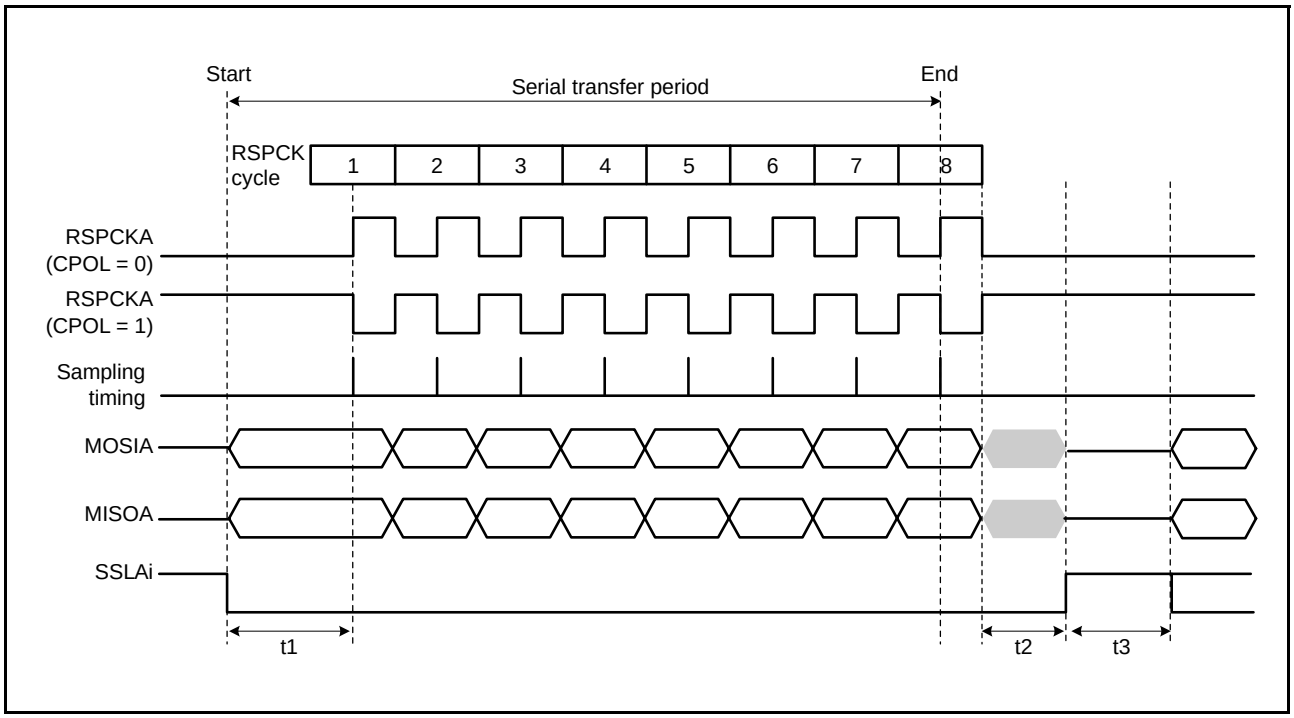


Figure 34.24 RSPI Transfer Format (CPHA = 0)

34.3.5.2 CPHA = 1

Figure 34.25 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 SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 34.25, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 34.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 34.3.12.1, Master Mode Operation.

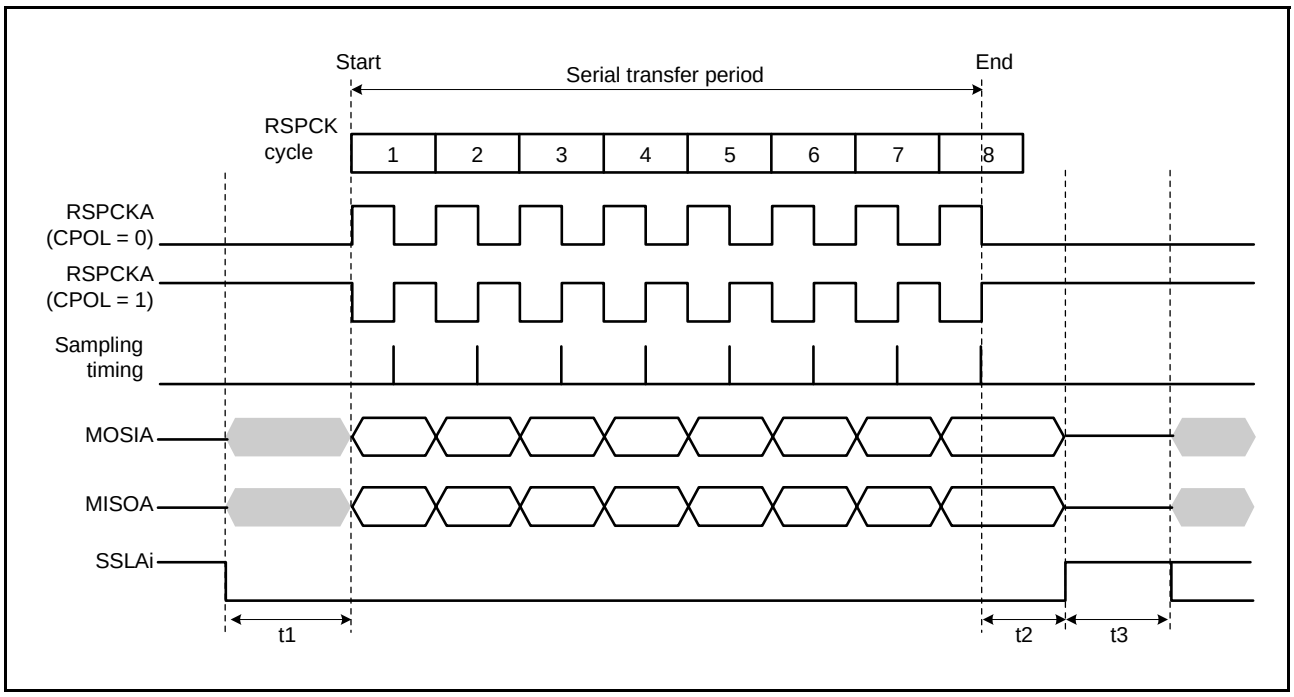


Figure 34.25 RSPI Transfer Format (CPHA = 1)

34.3.6 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 34.26 and Figure 34.27 indicate the condition of access to the SPDR register, where 'W' denotes a write cycle.

34.3.6.1 Full-Duplex Communications (SPCR.TXMD = 0, SPCR3.RXMD = 0)

Figure 34.26 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 34.26, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

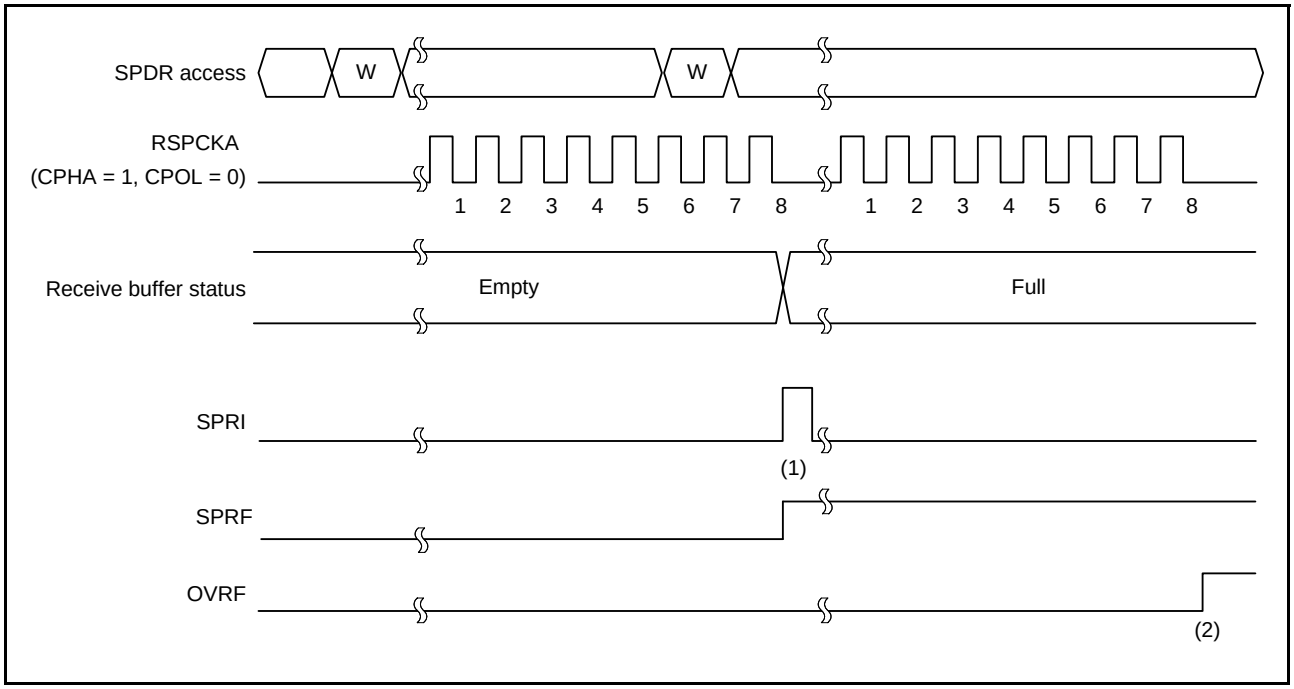


Figure 34.26 Operation Example of SPCR.TXMD = 0 and SPCR3.RXMD = 0

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.

34.3.6.2 Transmit-only Simplex Communications (SPCR.TXMD = 1, SPCR3.RXMD = 0)

Figure 34.27 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 34.27, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

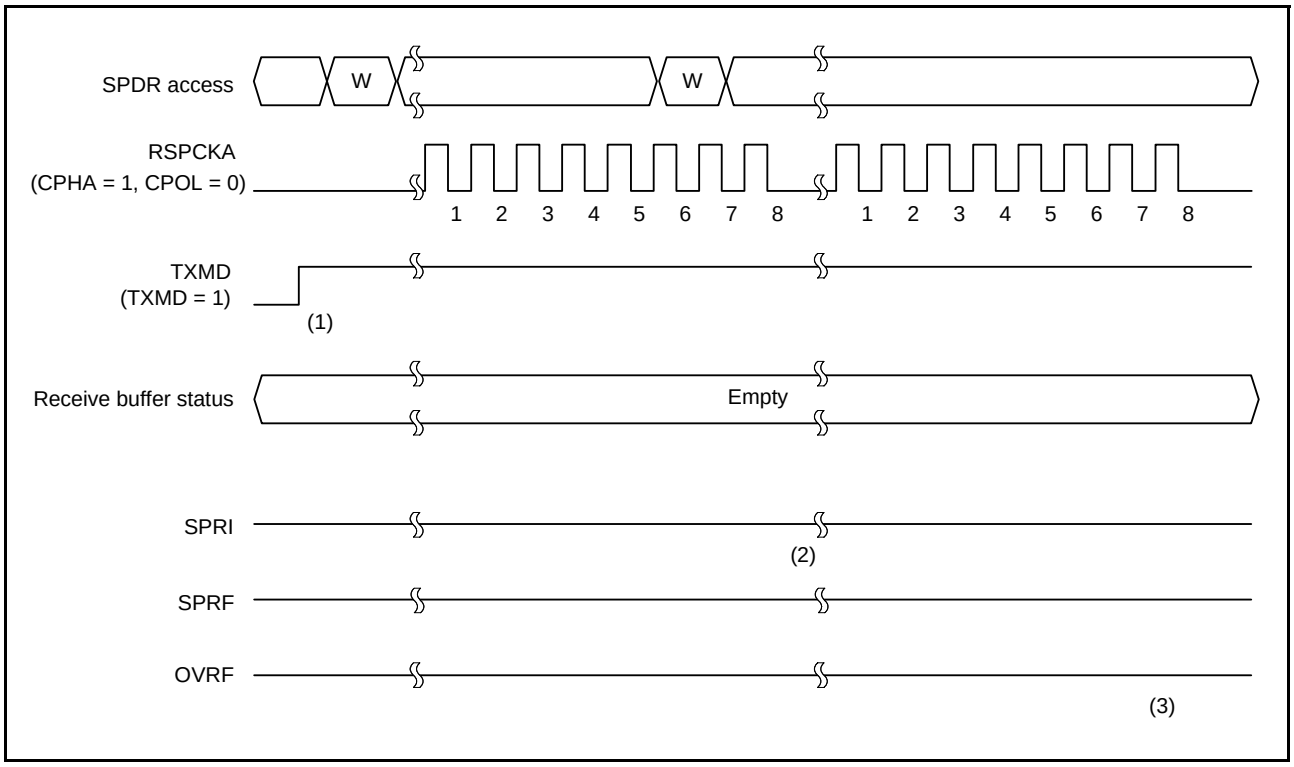


Figure 34.27 Operation Example of SPCR.TXMD = 1 and SPCR3.RXMD = 0

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).

34.3.6.3 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 34.28 shows an example of operation when the SPCR3.RXMD bit is set to 1. In the example in Figure 34.28, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

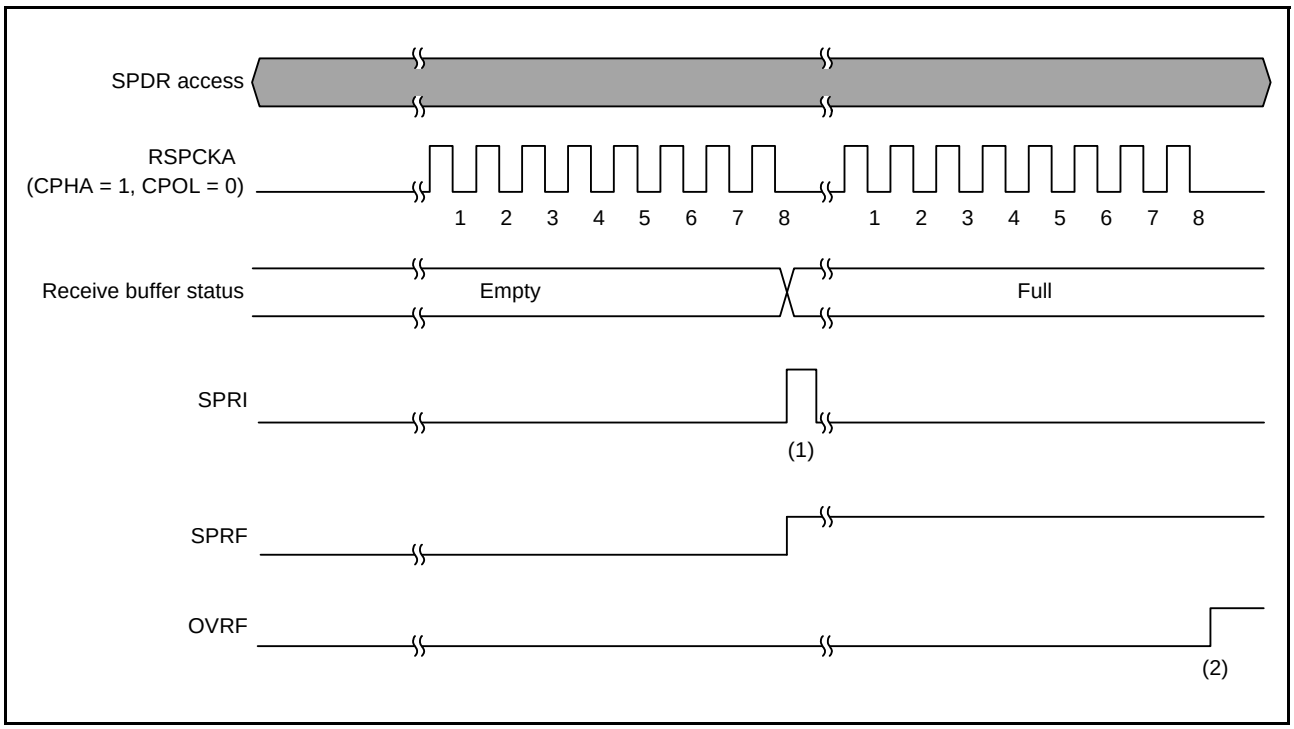


Figure 34.28 Operation Example of SPCR3.RXMD = 1

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.

34.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 34.29 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). 'SPDR access' shown in Figure 34.29 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 34.29, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

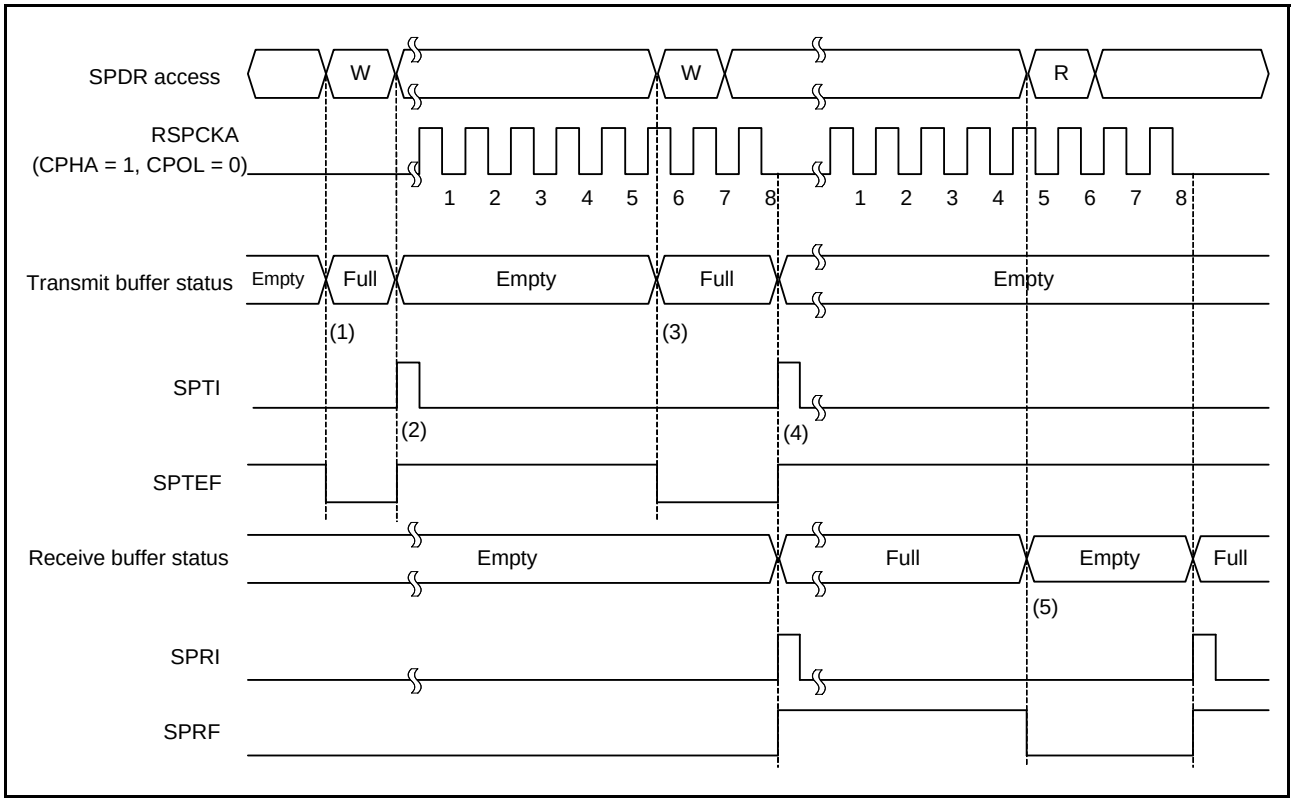


Figure 34.29 Operation Example of SPTI and SPRI Interrupts

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 34.3.12, SPI Operation, and section 34.3.13, 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 34.3.10, 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 14, Interrupt Controller (ICUF), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

34.3.8 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 SSLA0 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.

34.3.9 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 34.2.4, RSPI Status Register (SPSR).

34.3.9.1 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.

34.3.9.2 Full-duplex or transmit-only simplex communications in slave mode under SPI operation

When the SSLA0 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.

34.3.9.3 Receive-only simplex communications in slave mode under SPI operation

When the SSLA0 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 SSLA0 signal is asserted when the next communication is started.

34.3.9.4 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.

34.3.9.5 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.

34.3.10 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 34.7 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 34.7 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 SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLA0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLA0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOA output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 34.7, 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 34.3.10.4, Underrun Error. An overrun error shown in 4 is described in section 34.3.10.1, Overrun Error. A parity error shown in 5 is described in section 34.3.10.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 34.3.10.3, Mode Fault Error.

For the transmit and receive interrupts, refer to section 34.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

34.3.10.1 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 34.30 shows an example of operations of the SPRF and OVRF flags. 'SPSR access' and 'SPDR access' shown in Figure 34.30 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 34.30, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

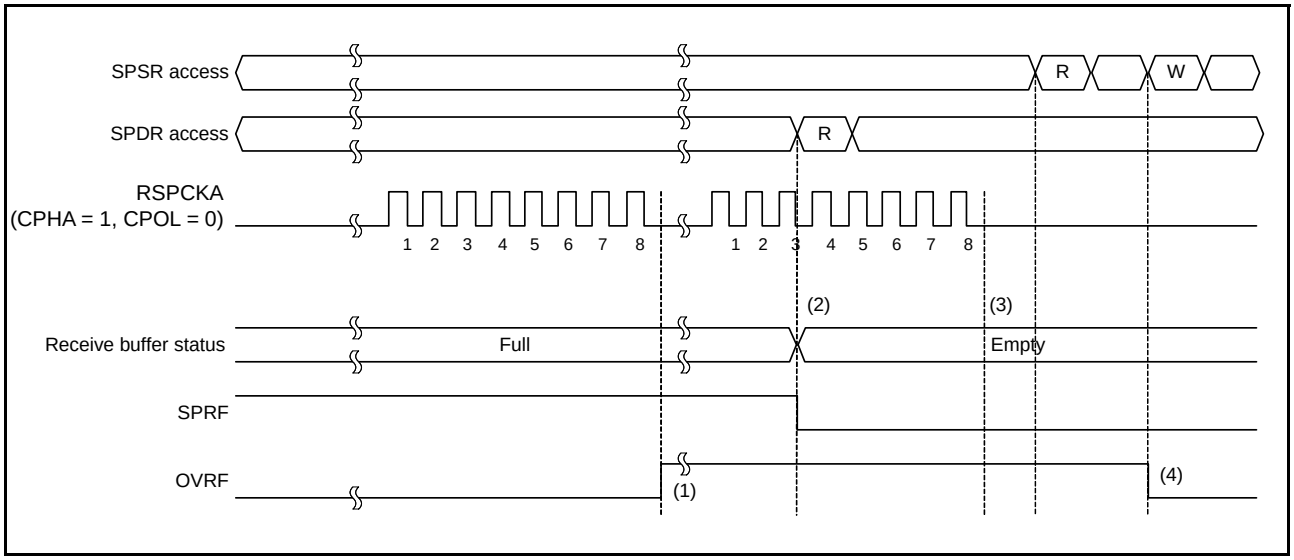


Figure 34.30 Operation Example of the SPRF and OVRF Flags

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 34.31 and Figure 34.32 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

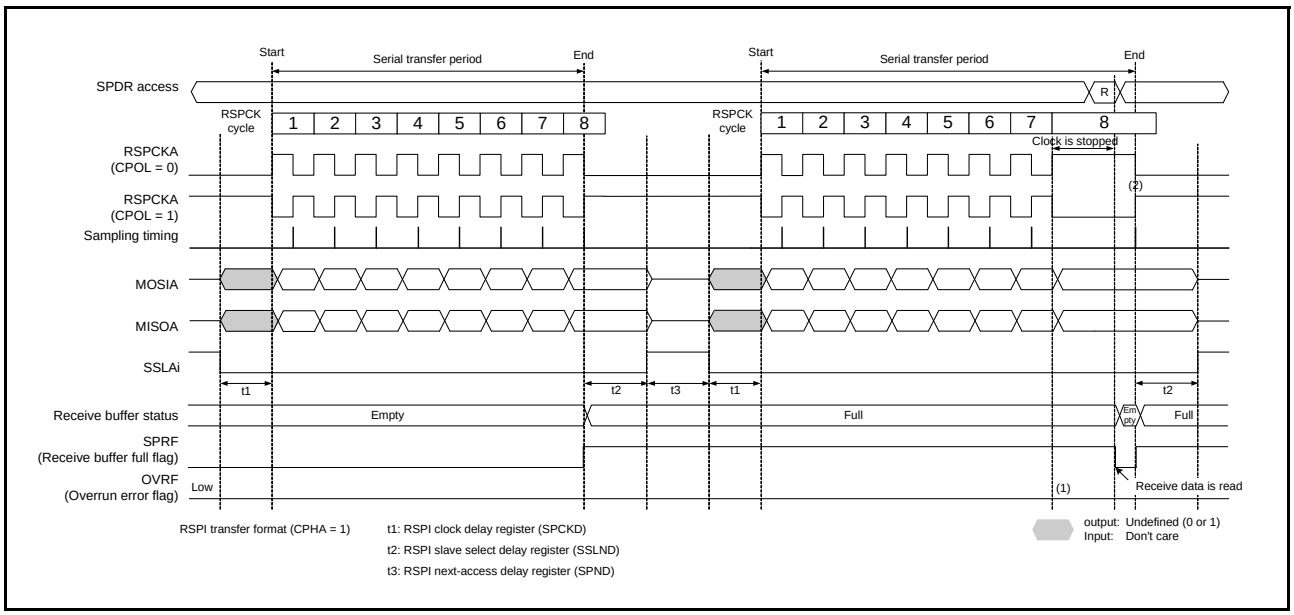


Figure 34.31 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

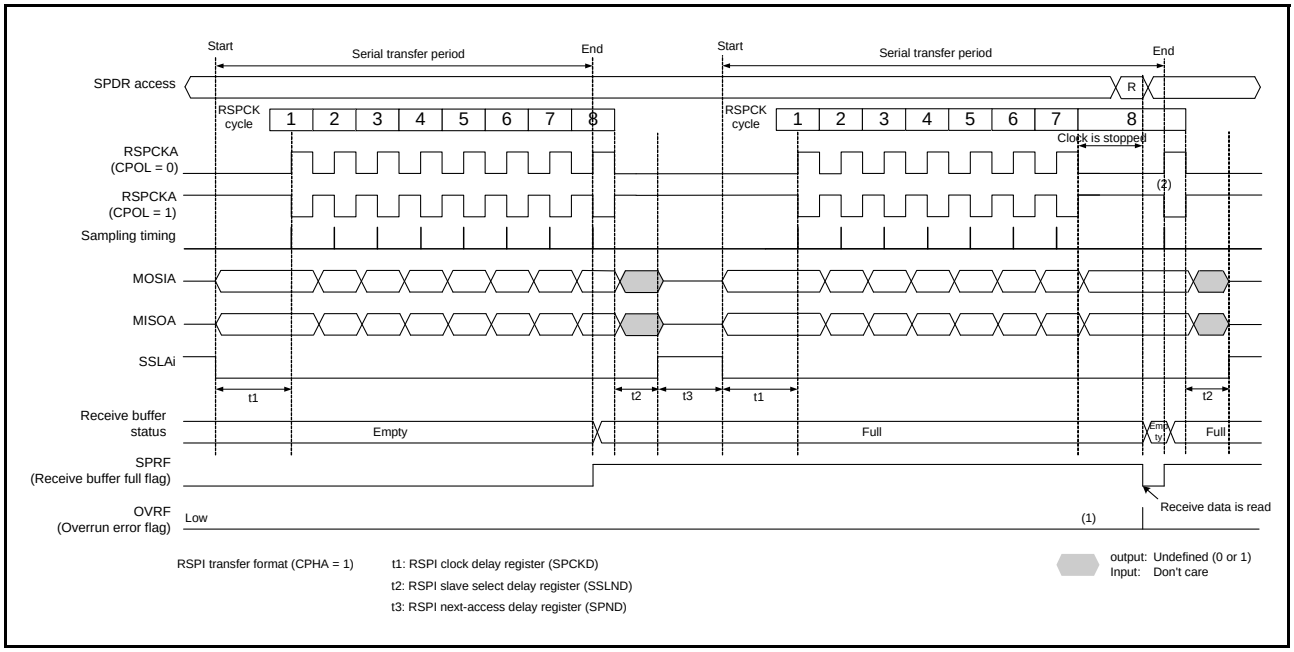


Figure 34.32 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

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).

34.3.10.2 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 34.33 shows an example of operation of the OVRF and PERF flags. 'SPSR access' shown in Figure 34.33 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 34.33, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

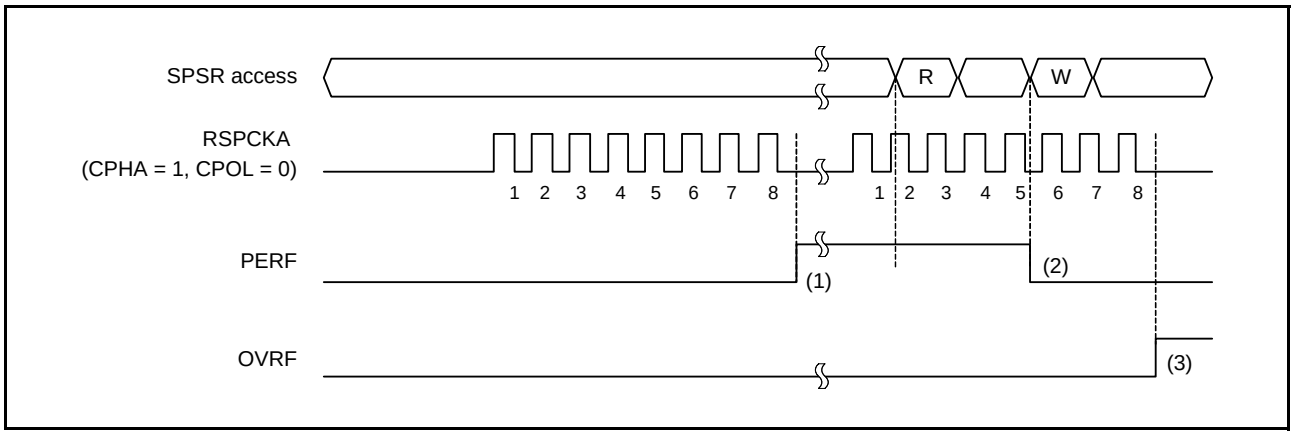


Figure 34.33 Operation Example of PERF Flag

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.

34.3.10.3 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 SSLA0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to the SPCMDm register to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 34.3.11, 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.

34.3.10.4 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 34.3.11, 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.

34.3.11 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.

34.3.11.1 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.

34.3.11.2 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 34.3.11.1, Initialization by Clearing the SPE Bit.

34.3.12 SPI Operation

34.3.12.1 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 34.3.10, 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 34.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 34.3.5, 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: SSLAi 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.

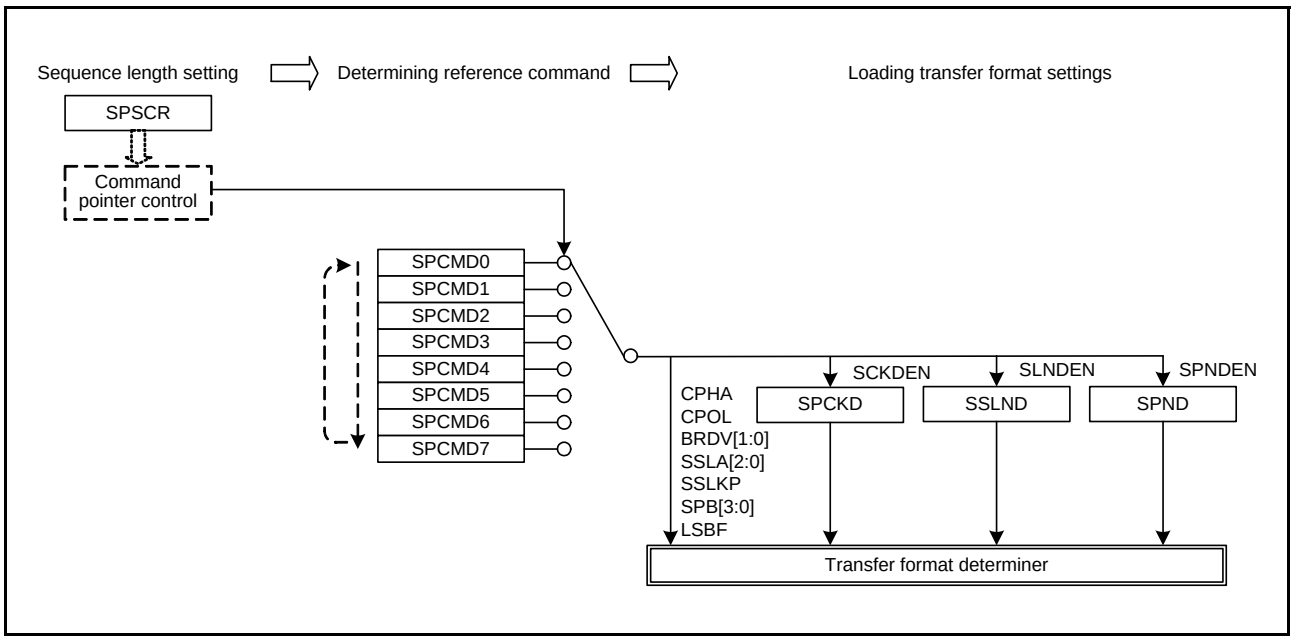


Figure 34.34 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

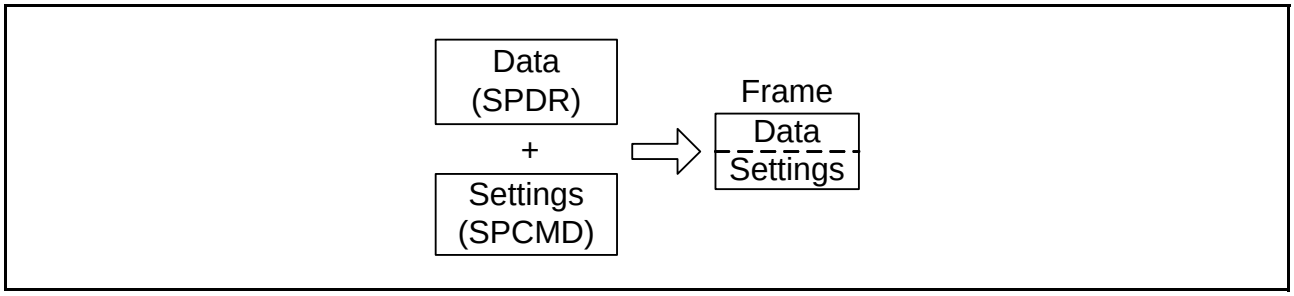


Figure 34.35 Concept of a Frame

Figure 34.36 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 34.4.

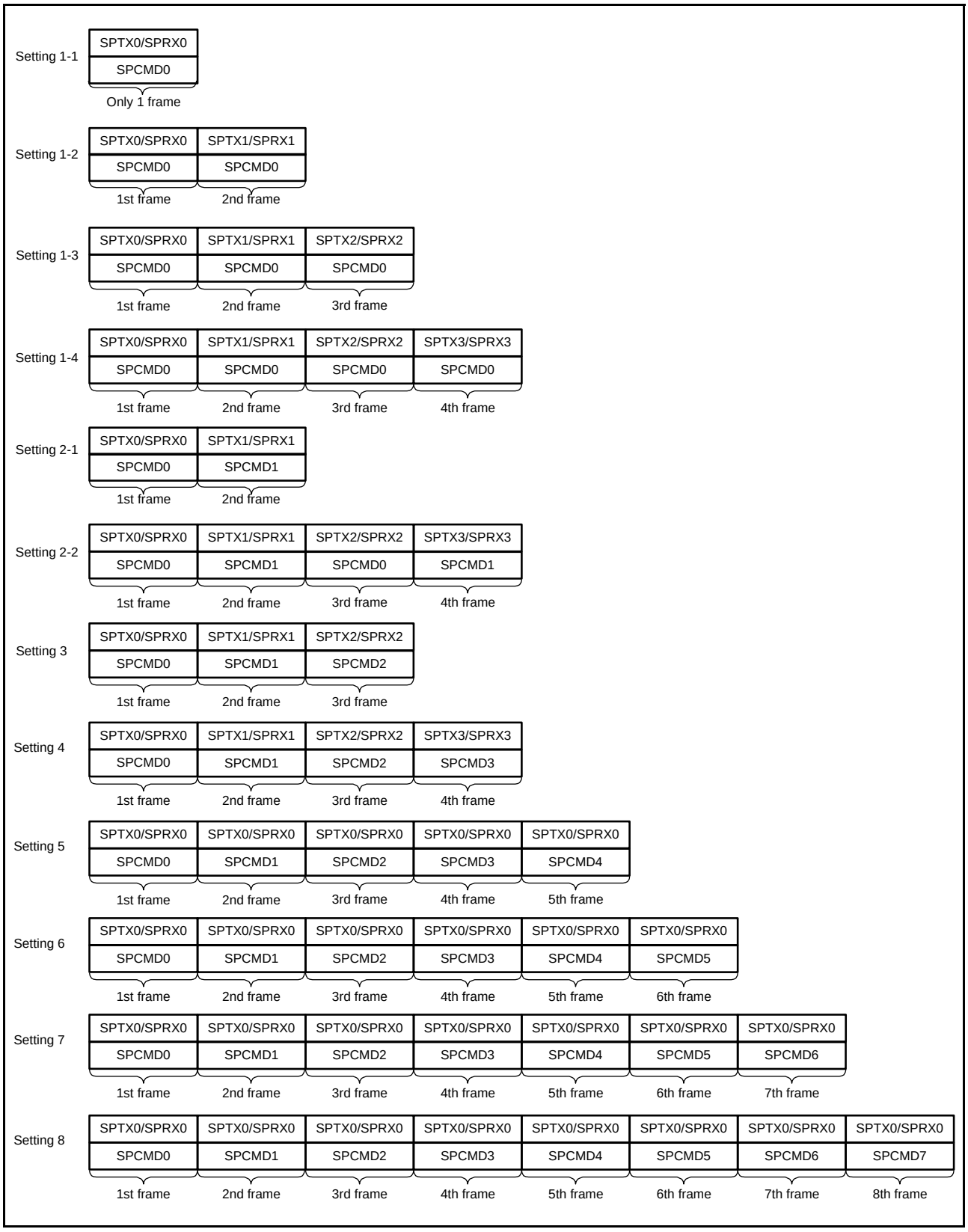


Figure 34.36 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 34.37 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 34.37. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

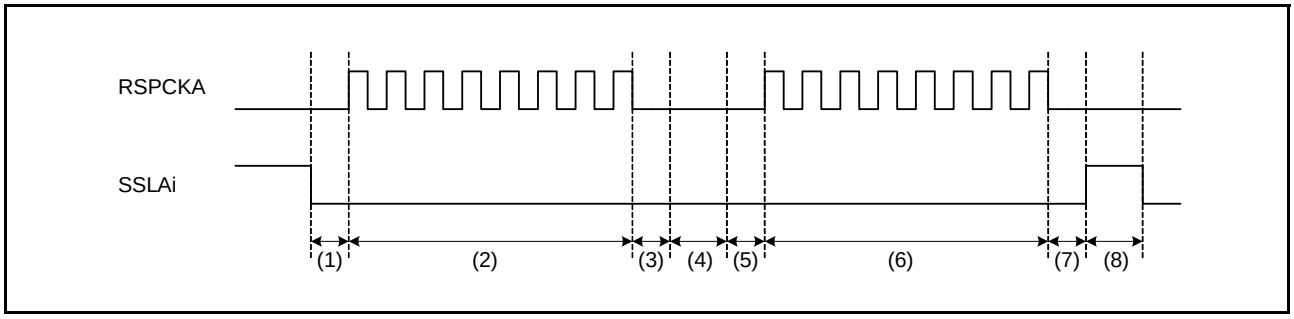


Figure 34.37 Example of Burst Transfer Operation Using SSLKP Bit (CPHA = 1, CPOL = 0)

(1) Based on the SPCMD0 register, the RSPI asserts the SSLAi 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 SSLAi 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 SSLAi signal. In addition, a next-access delay is inserted according to the SPCMD1 register.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 34.37) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.
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 34.8. For a definition of RSPCK delay, refer to section 34.3.5, 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 34.8 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 34.9. For a definition of SSL negation delay, refer to section 34.3.5, 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 34.9 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 34.10. For a definition of next-access delay, refer to section 34.3.5, 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 34.10 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 34.38 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.

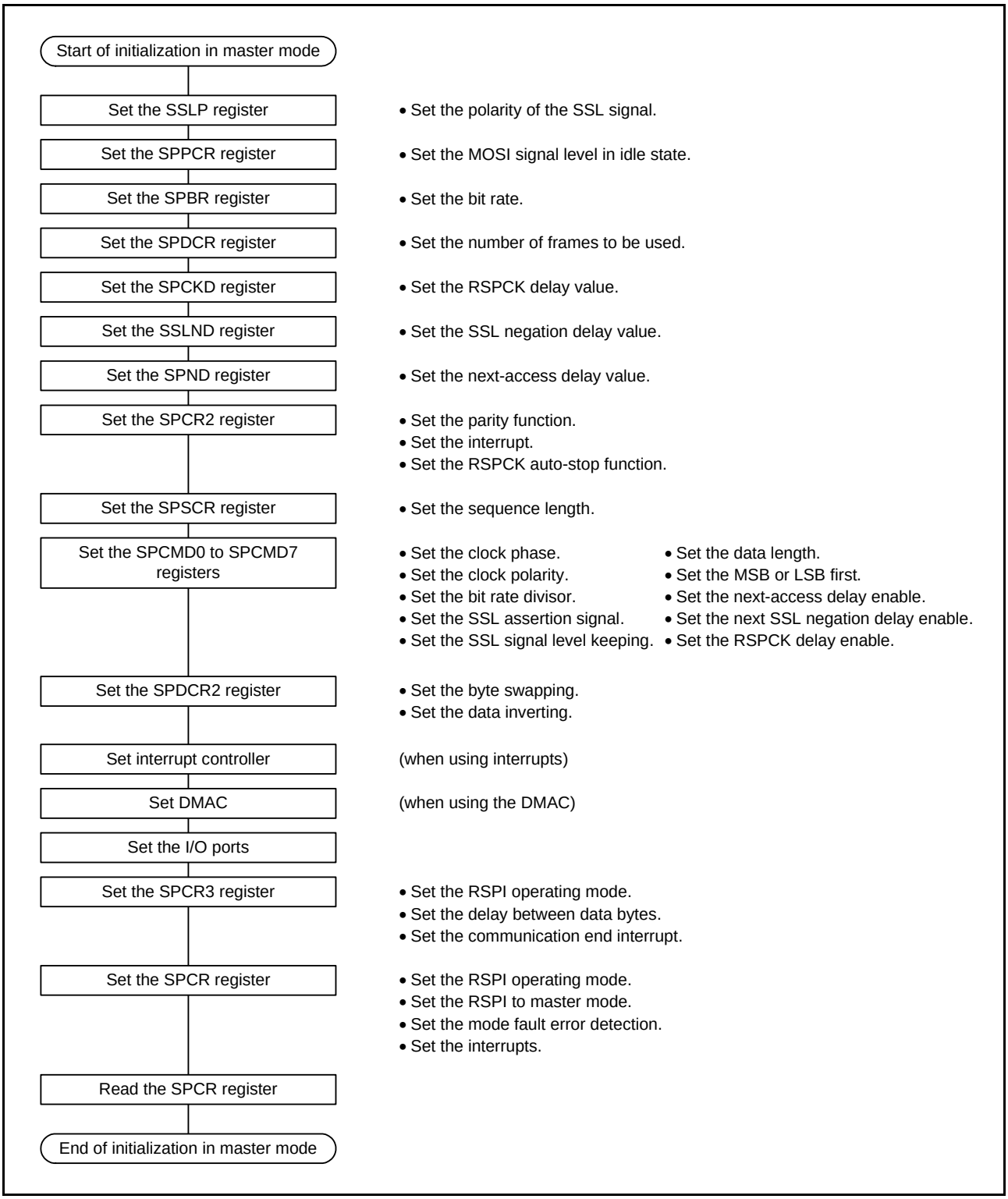


Figure 34.38 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 34.40 to Figure 34.42 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.

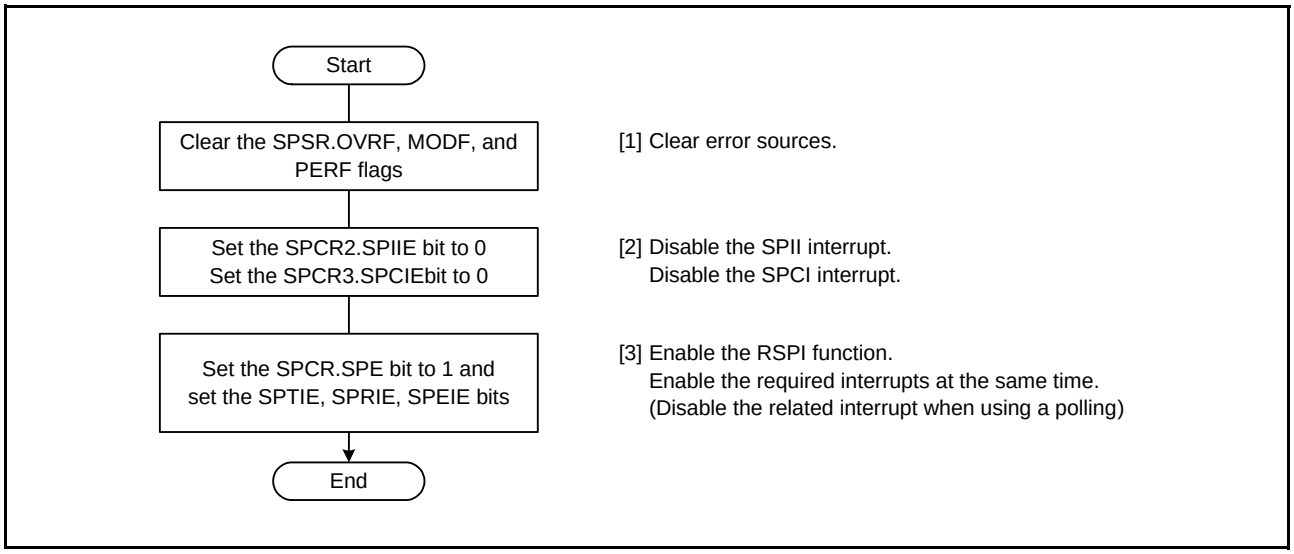


Figure 34.39 Flowchart in Master Mode (Communication Preprocess)

(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.

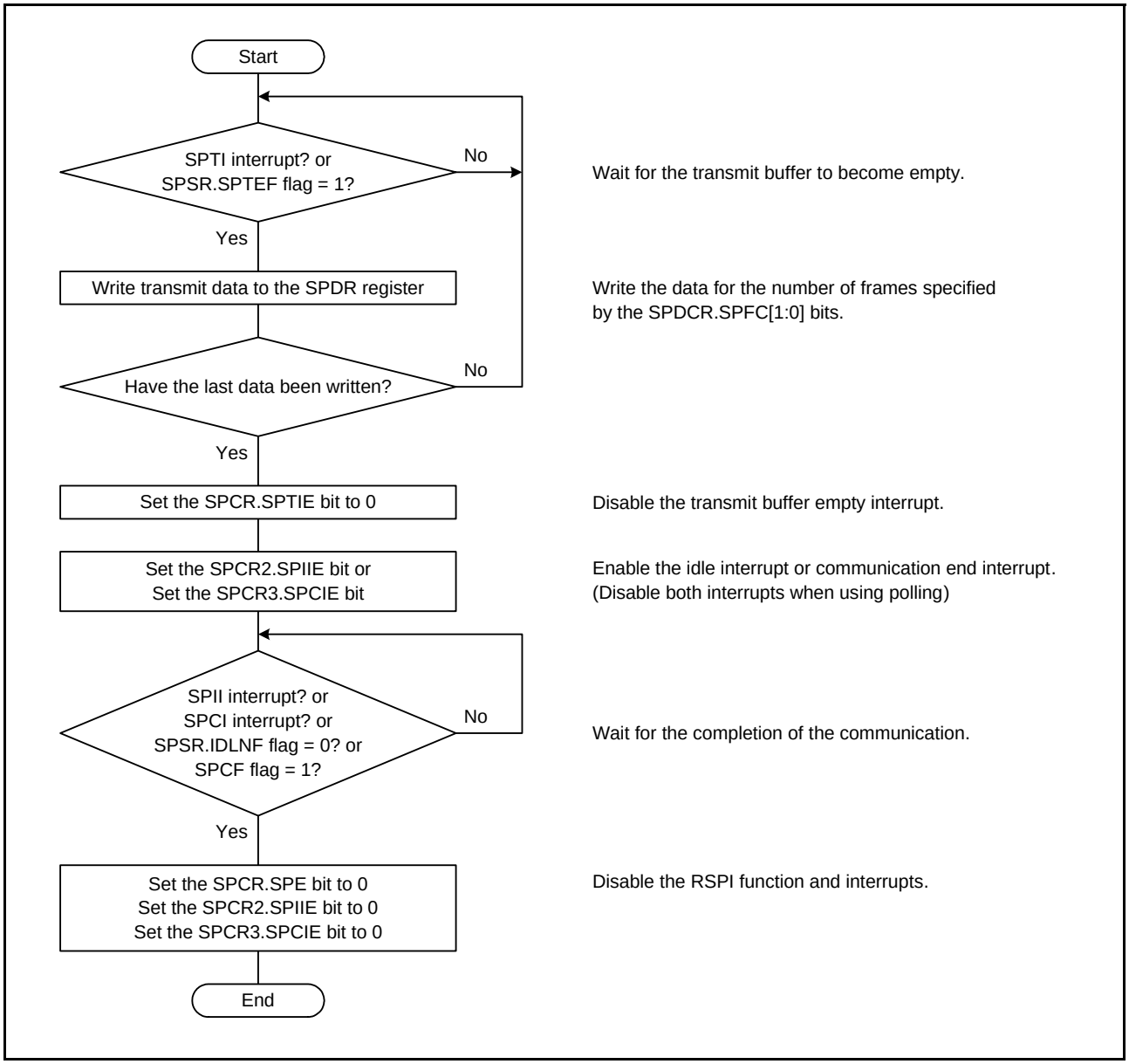


Figure 34.40 Flowchart in Master Mode (Transmission)

(c) Receive Processing Flow

The RSPI does not support receive-only simplex communications in master mode, so processing for transmission is required.

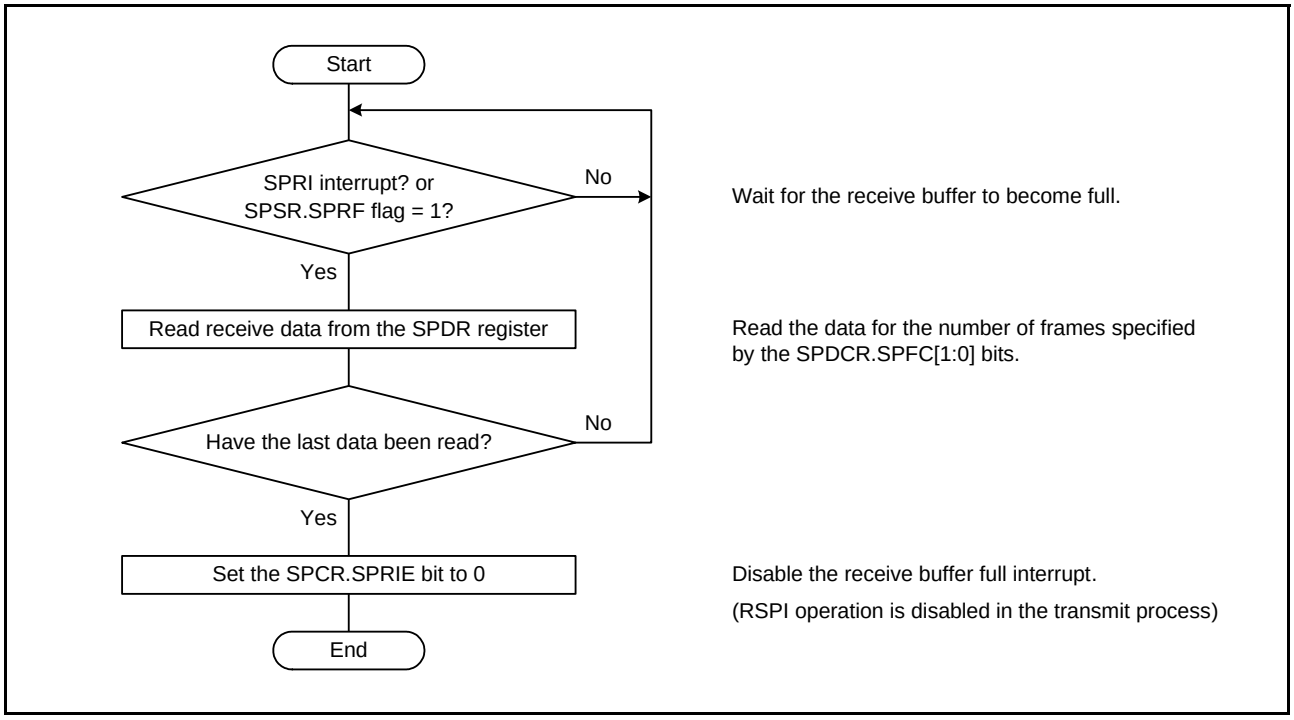


Figure 34.41 Flowchart in Master Mode (Reception)

(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.

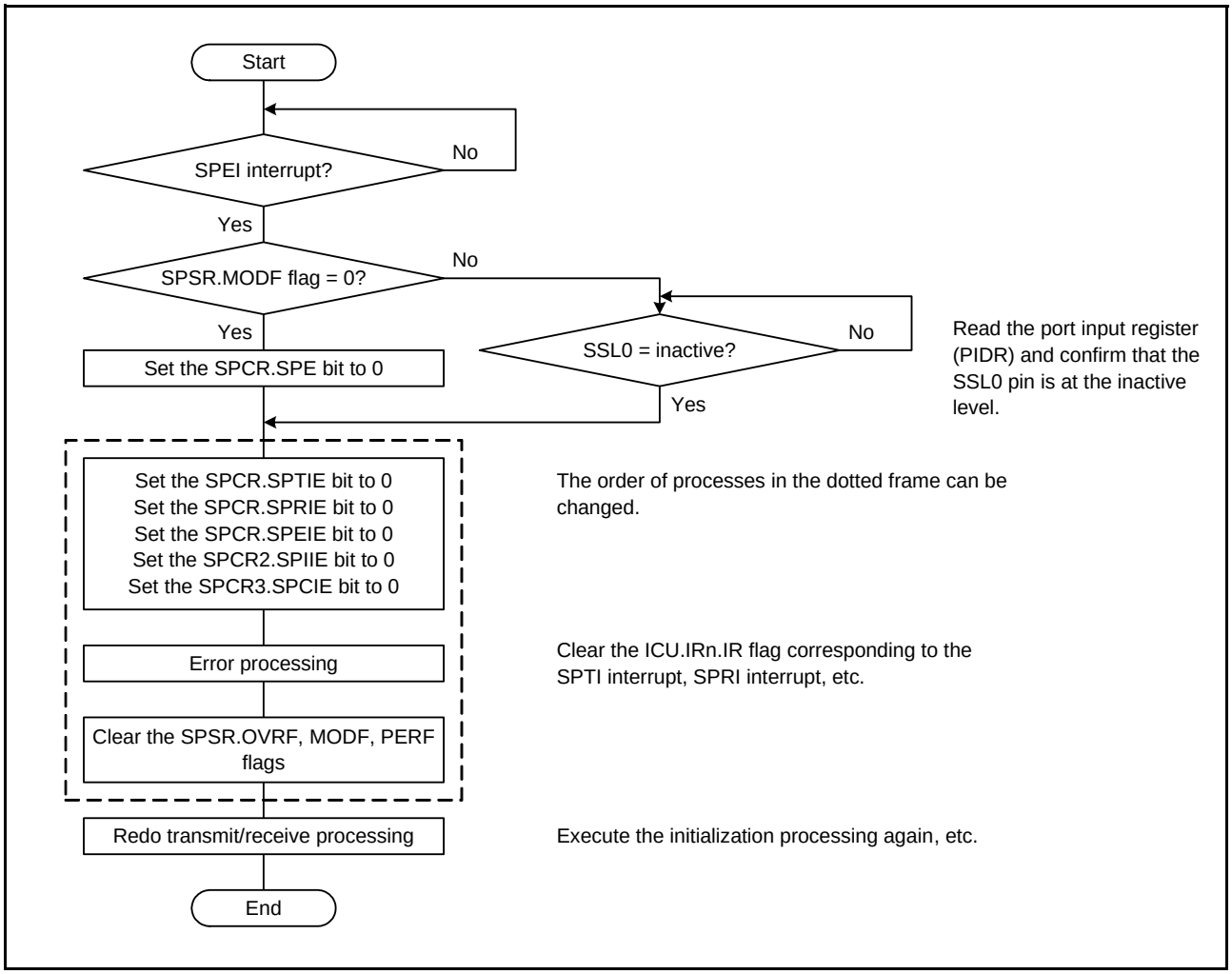


Figure 34.42 Flowchart for Master Mode (Error Processing)

34.3.12.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.
Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 34.3.10, 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 SSLA0 input signal depends on the SSLP.SSL0P bit setting.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.

(3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 34.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

(4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 34.43 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.

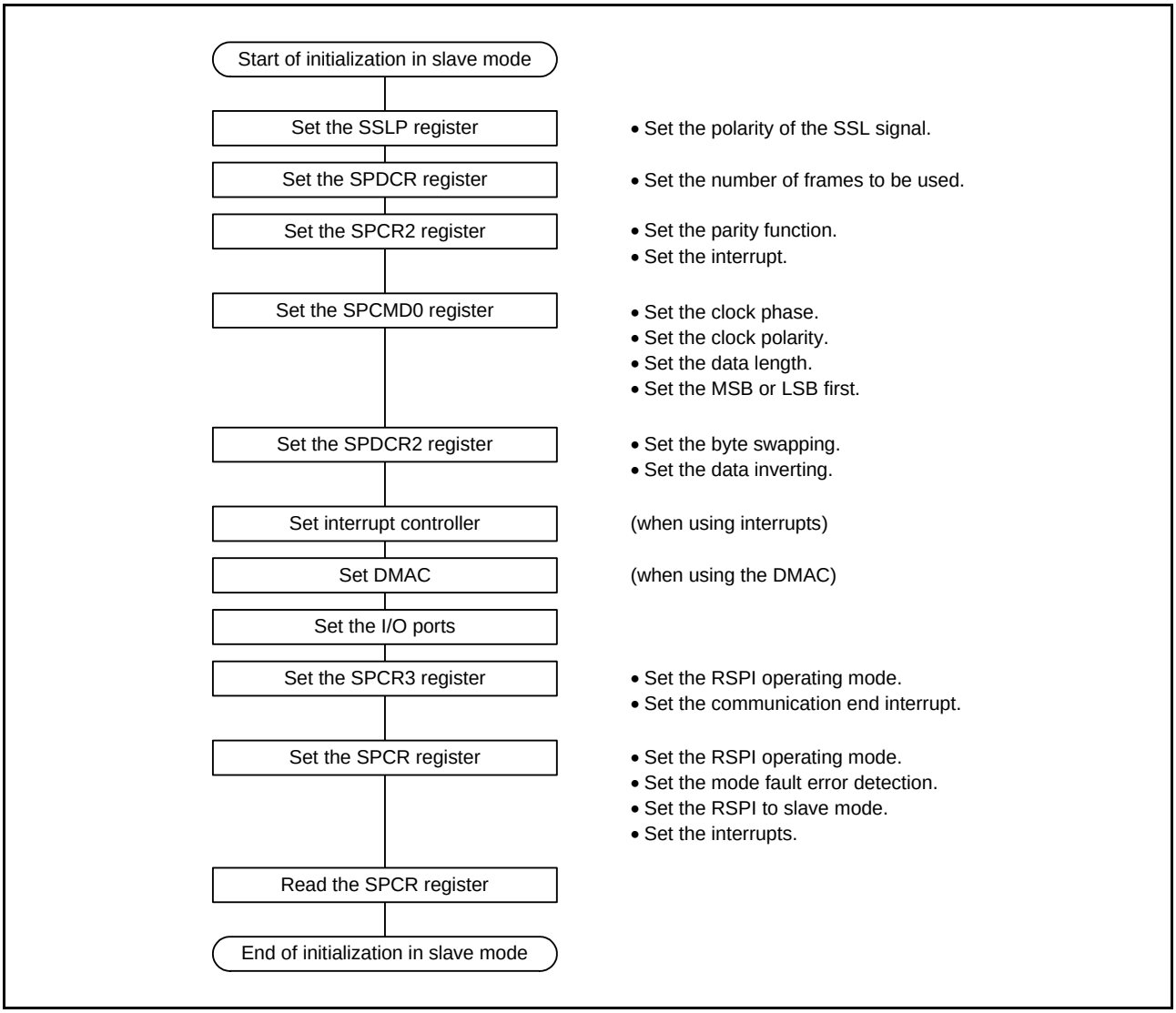


Figure 34.43 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 34.45 to Figure 34.47 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.

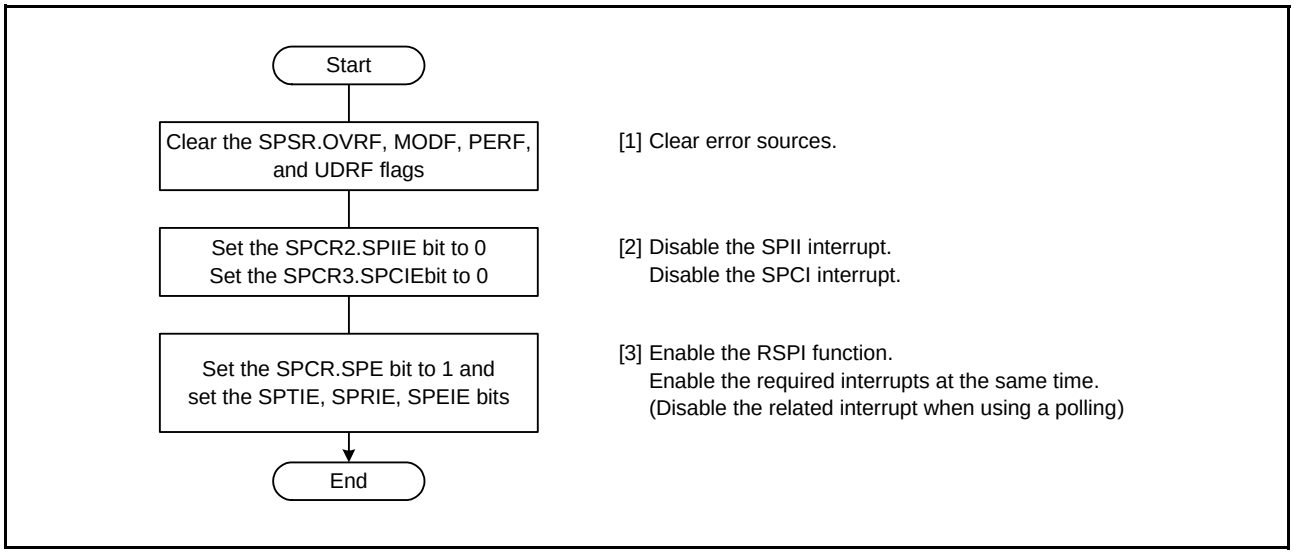


Figure 34.44 Flowchart in Slave Mode (Communication Preprocess)

(b) Transmit Processing Flow

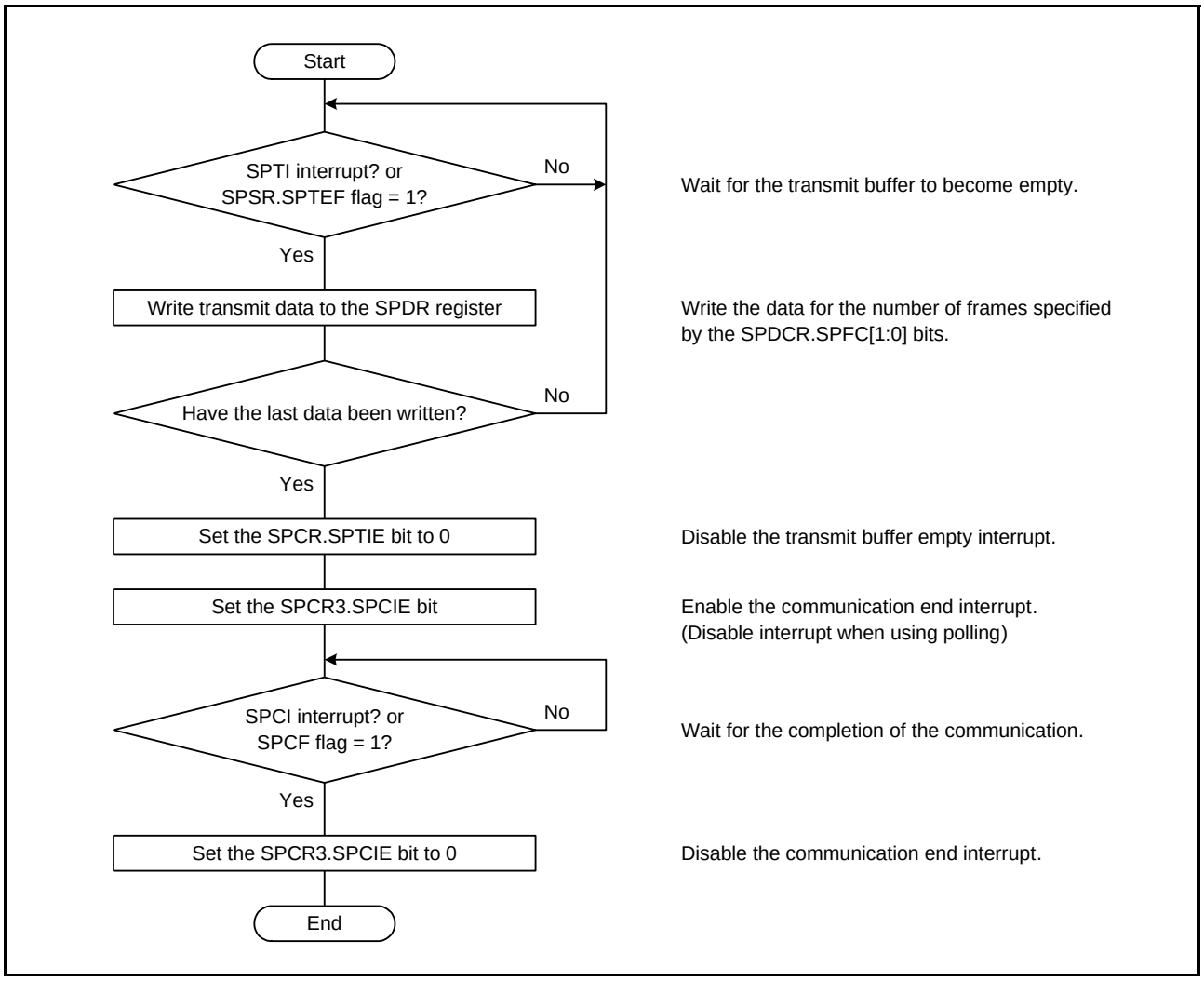


Figure 34.45 Flowchart in Slave Mode (Transmission)

(c) Receive Processing Flow

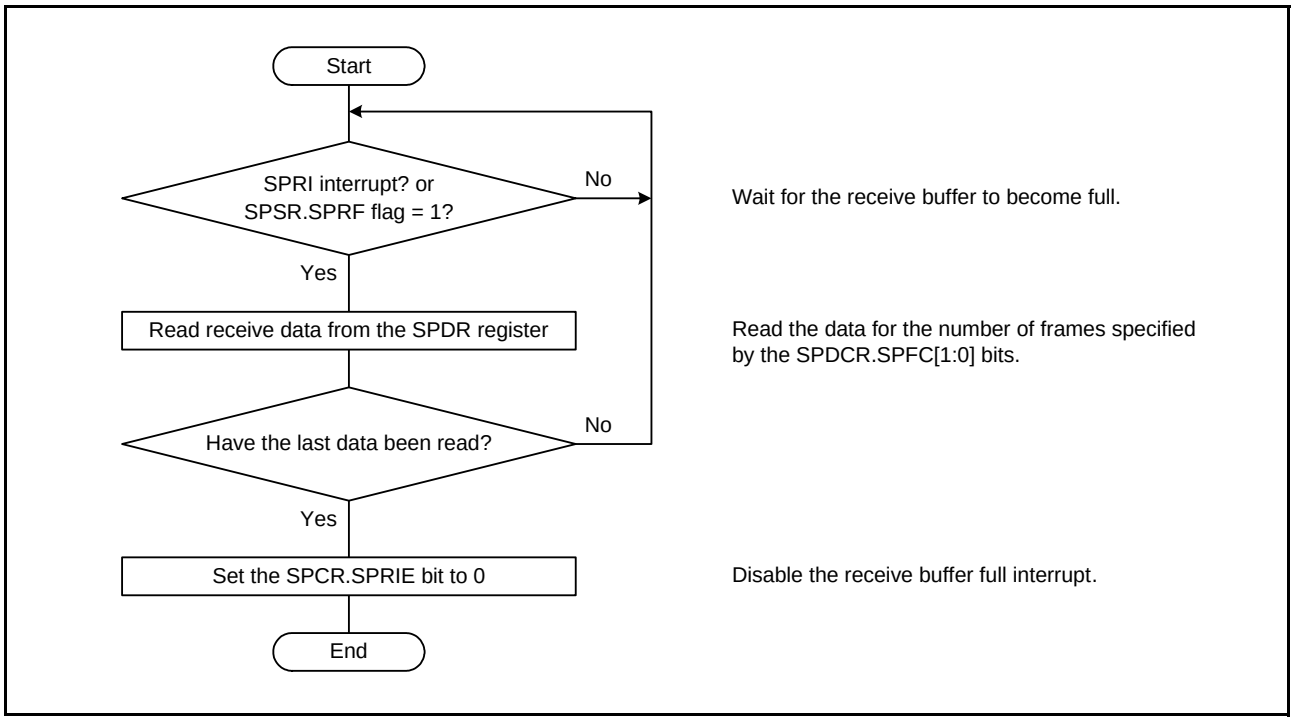


Figure 34.46 Flowchart in Slave Mode (Reception)

(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 SSLA0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

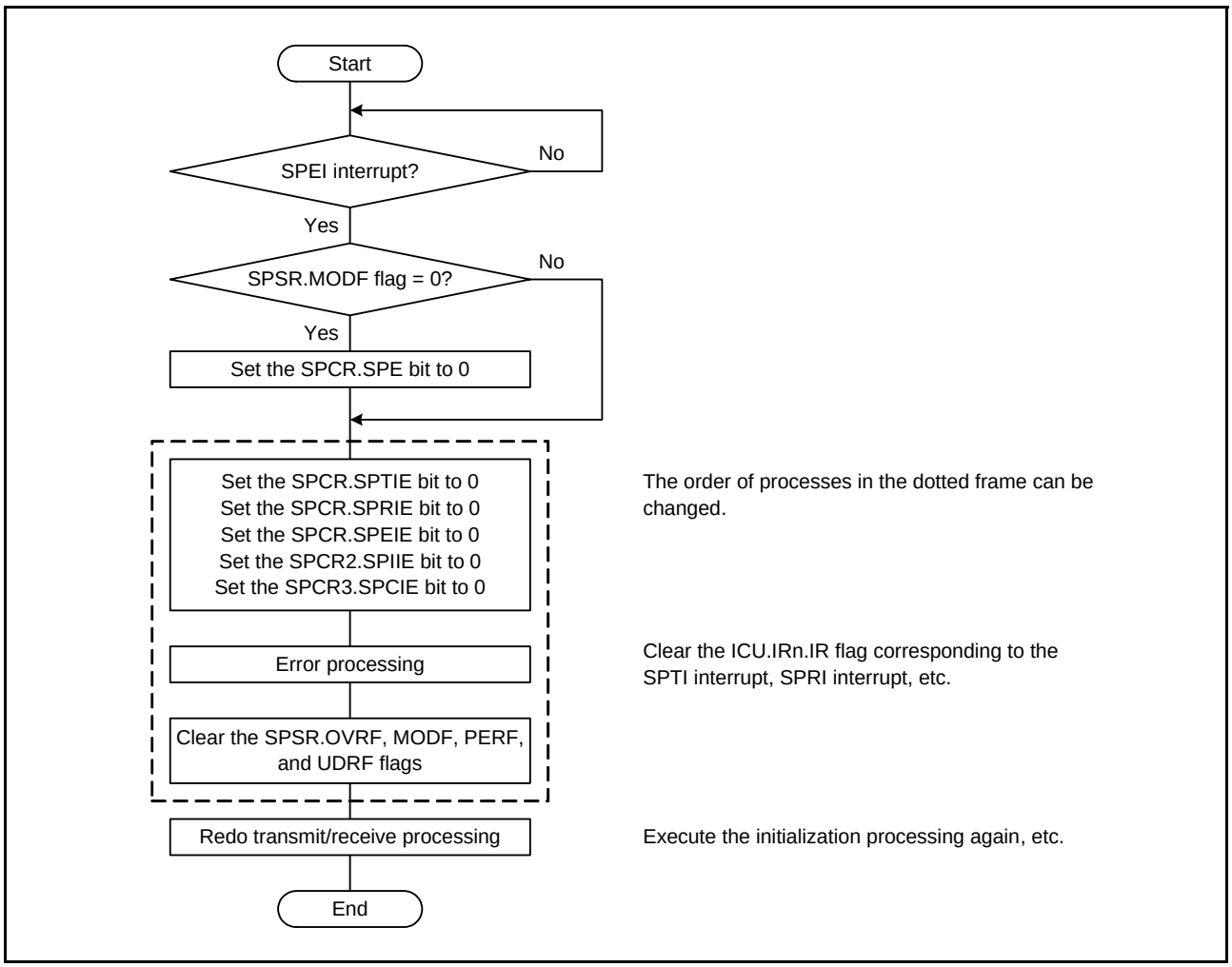


Figure 34.47 Flowchart for Slave Mode (Error Processing)

34.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

34.3.13.1 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 34.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
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: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA 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.

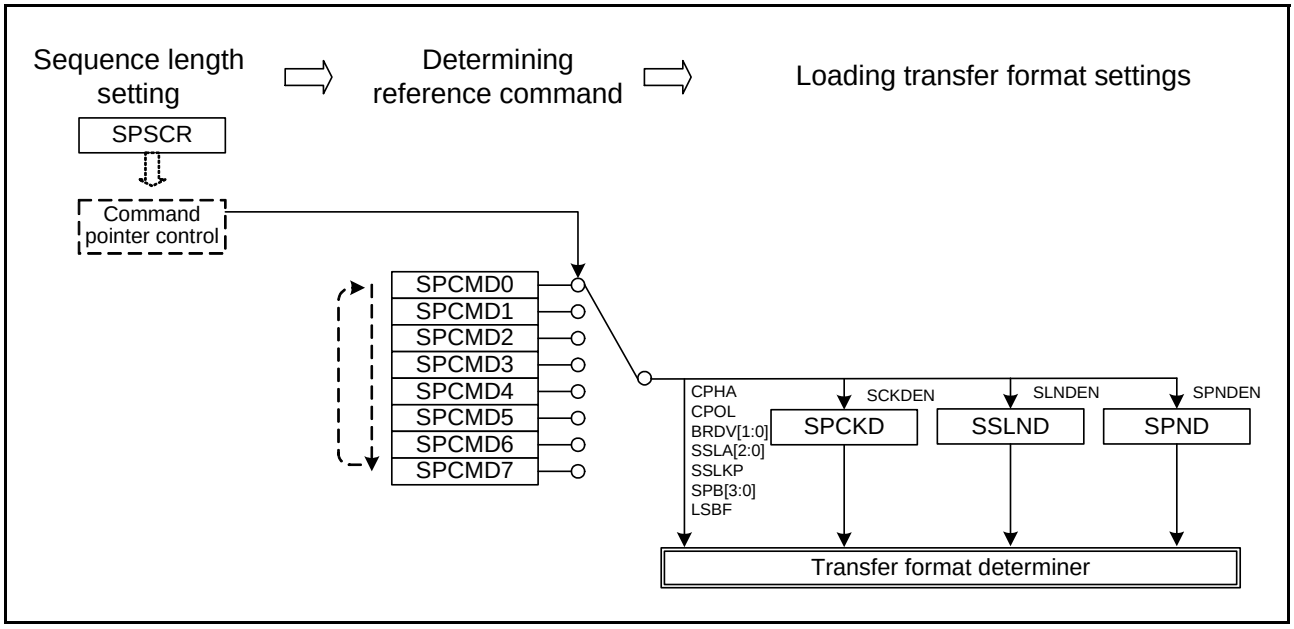


Figure 34.48 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

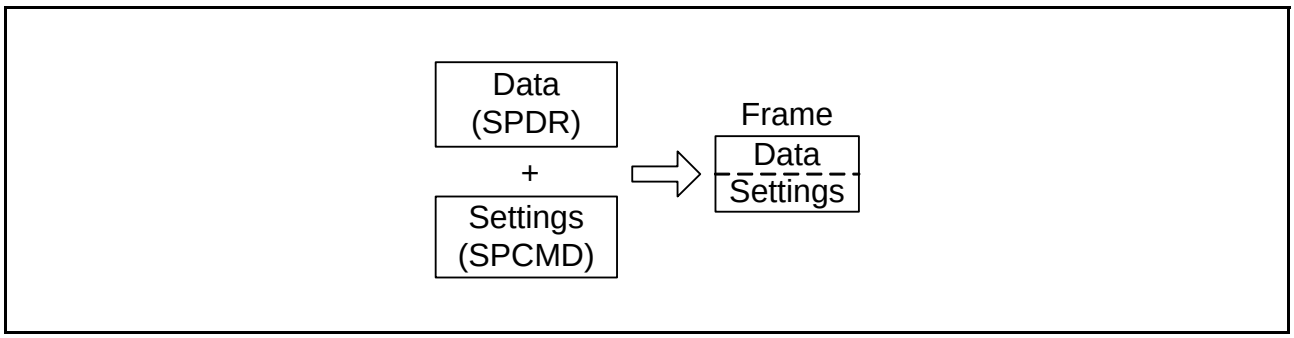


Figure 34.49 Concept of a Frame

Figure 34.50 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 34.4.

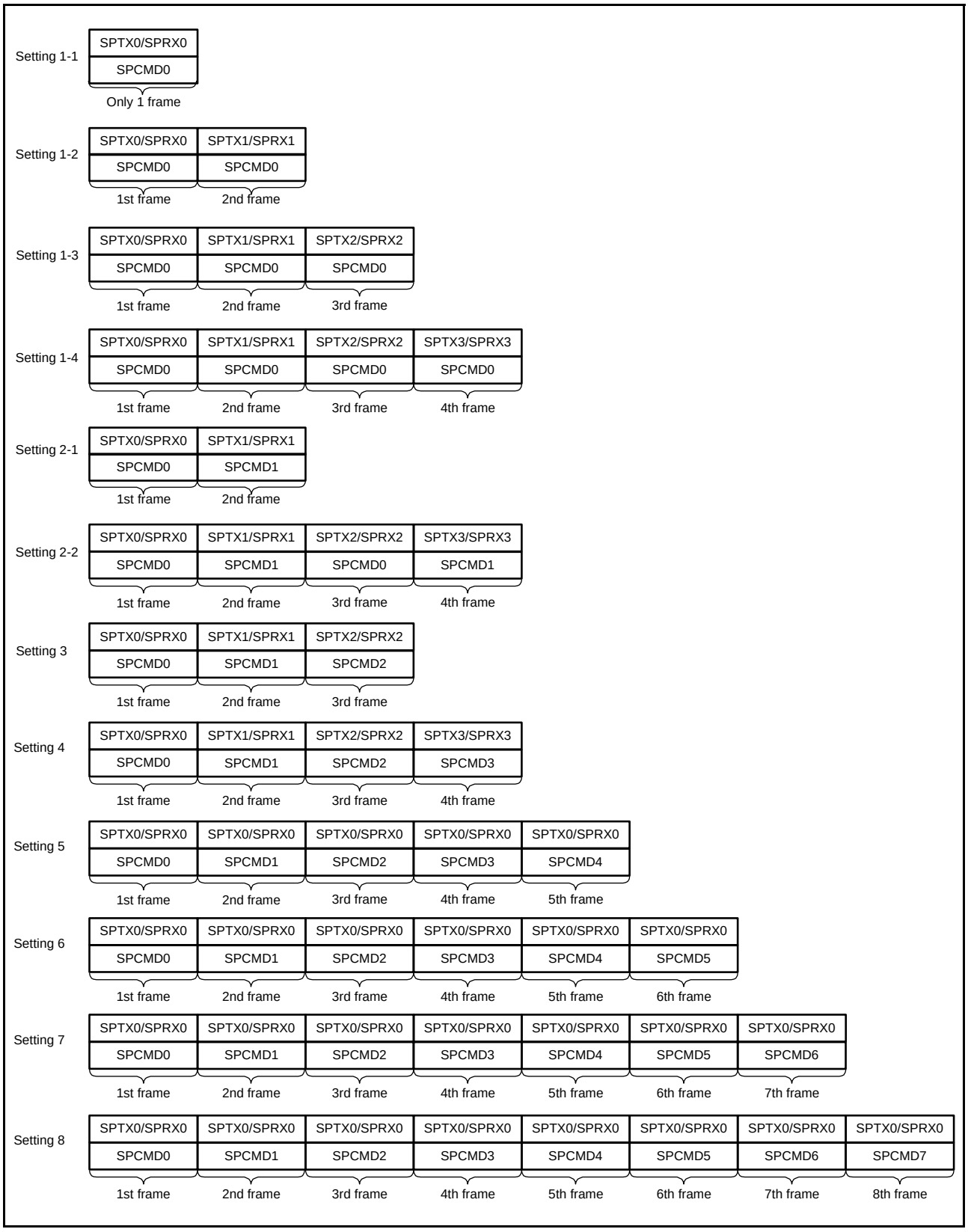


Figure 34.50 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 34.51 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.

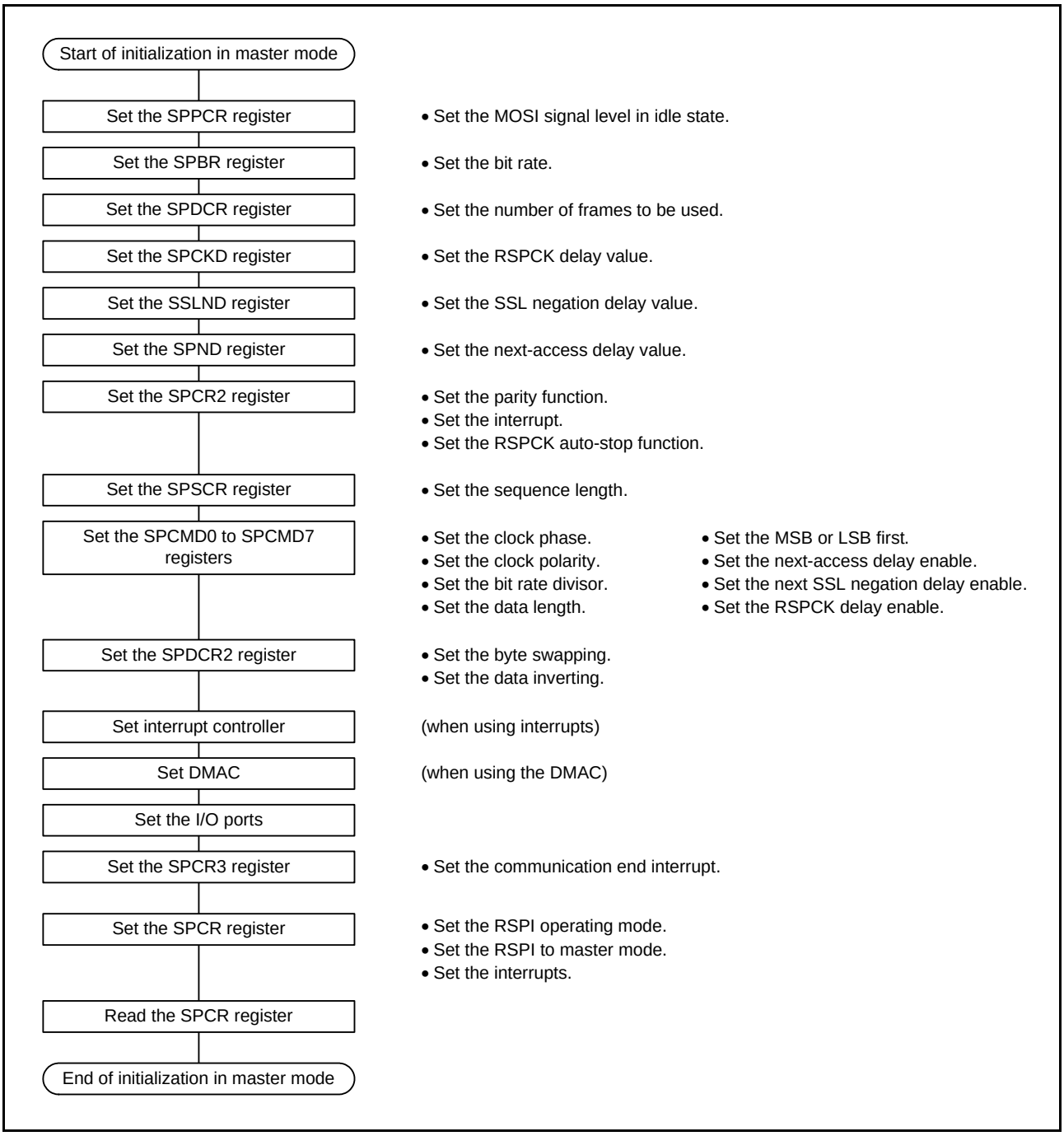


Figure 34.51 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(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 34.3.12.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

34.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When the SPMS bit is 1, the RSPI drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.
It should be noted that the SSLA0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 34.52 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.

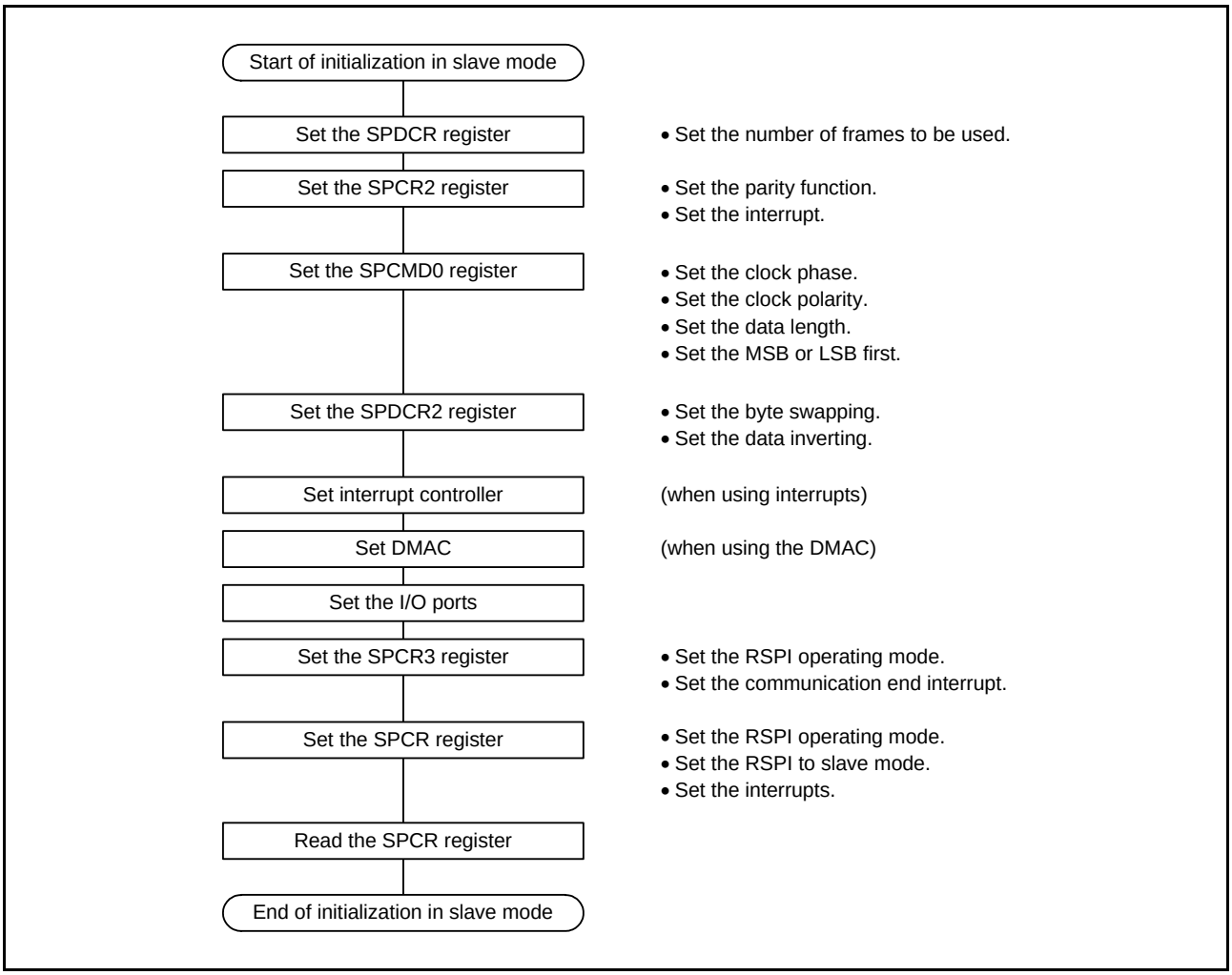


Figure 34.52 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(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 34.3.12.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

34.3.14 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 34.11 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 34.53 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 34.11 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIA pin or MISOA pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

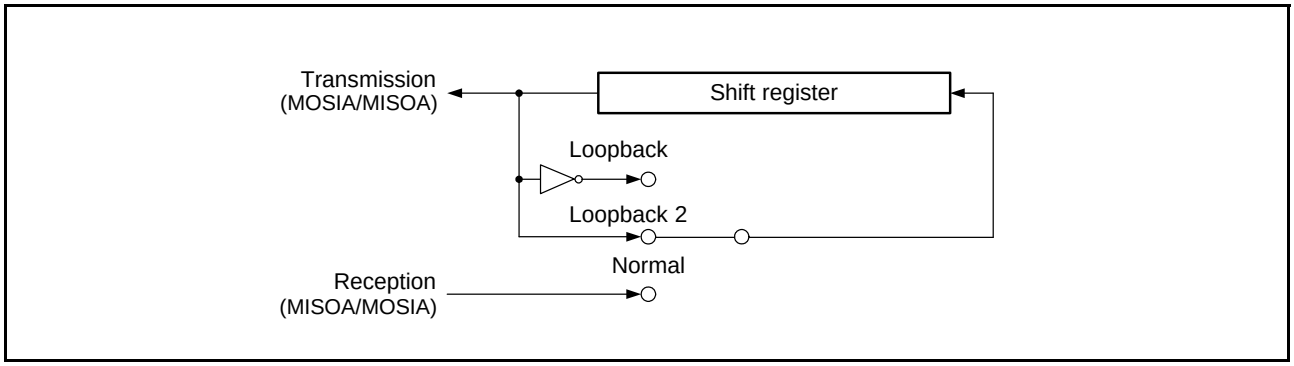


Figure 34.53 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

34.3.15 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 34.54.

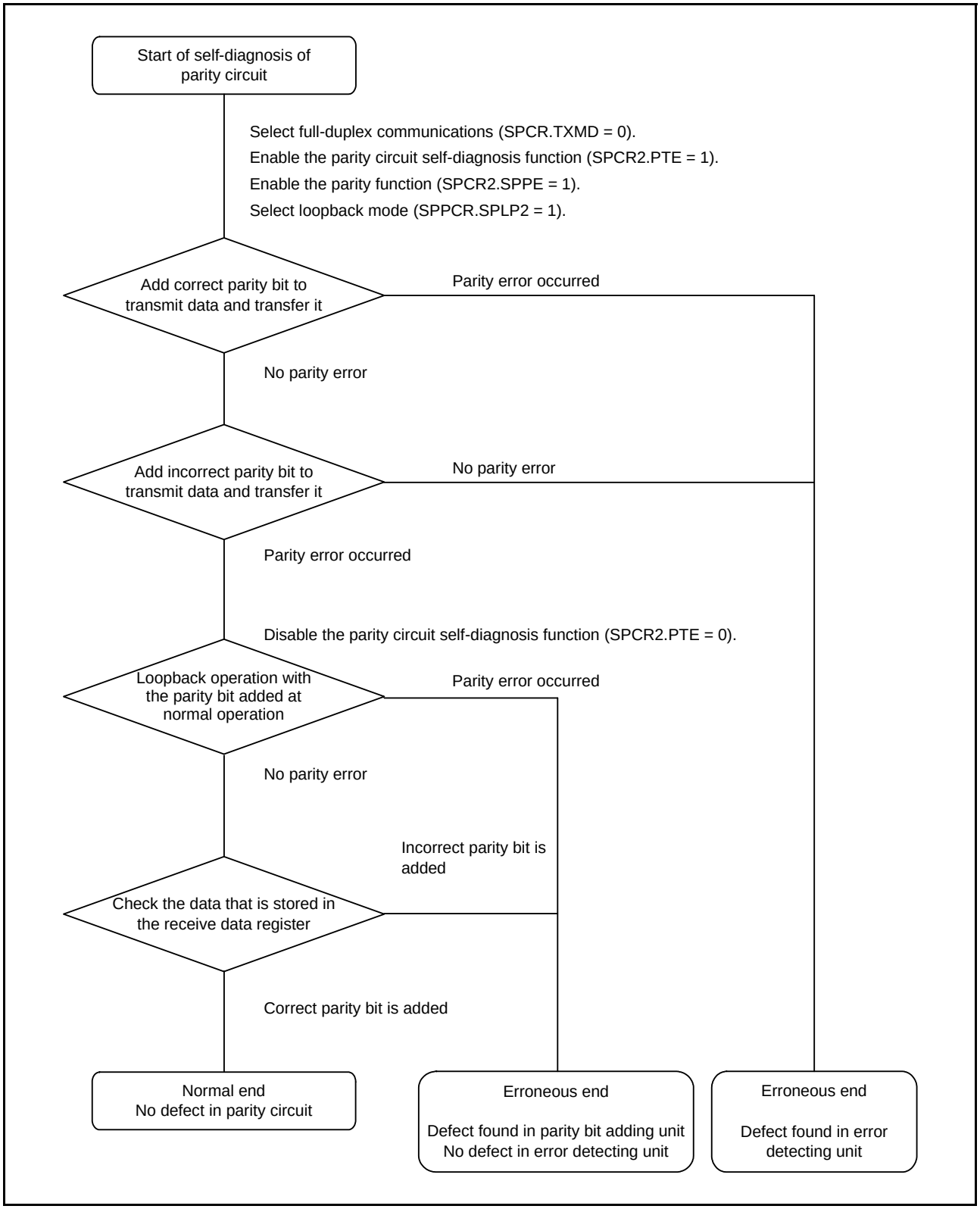


Figure 34.54 Flowchart for Self-Diagnosis of Parity Circuit

34.3.16 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 34.12. An interrupt is generated on satisfaction of an interrupt condition in Table 34.12. 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 17, DMA Controller (DMACAa), or section 18, 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 34.12 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

34.4 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.

34.4.1 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.

34.4.2 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.

34.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 34.13 lists the occurrence conditions of a mode fault event.

Table 34.13 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLA0 Pin	Remarks
Master (SPCR.MSTR bit = 1)	1	Active	Under the condition (the SPCR.MSTR bit is 1 and the MODFEN bit is 1), if the SPCR.SPMS bit is 0, mode fault error, overrun error, and parity error event output cannot be used. Do not set the ELSRn register to 52h.
Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the pin is deactivated during transmission.

(2) 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.

34.4.4 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).

34.4.5 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 34.14 and Table 34.14.

Table 34.14 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 SSLA0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Even edge detection of the last RSPCKA for the last data

Table 34.15 Generating Conditions of Communication End Event (Slave mode, receive-only simplex communications)

RSPI Mode	Condition
SPI operation (SPMS = 0)	Negation of the SSLA0 input after the last data was received
Clock synchronous operation (SPMS = 1)	Even edge detection of the last RSPCKA 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.

34.5 Usage Notes

34.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

34.5.2 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.

34.5.3 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.

34.5.4 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.

35. CRC Calculator (CRCA)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

35.1 Overview

Table 35.1 lists the specifications of the CRC calculator, and Figure 35.1 shows a block diagram of the CRC calculator.

Table 35.1 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.

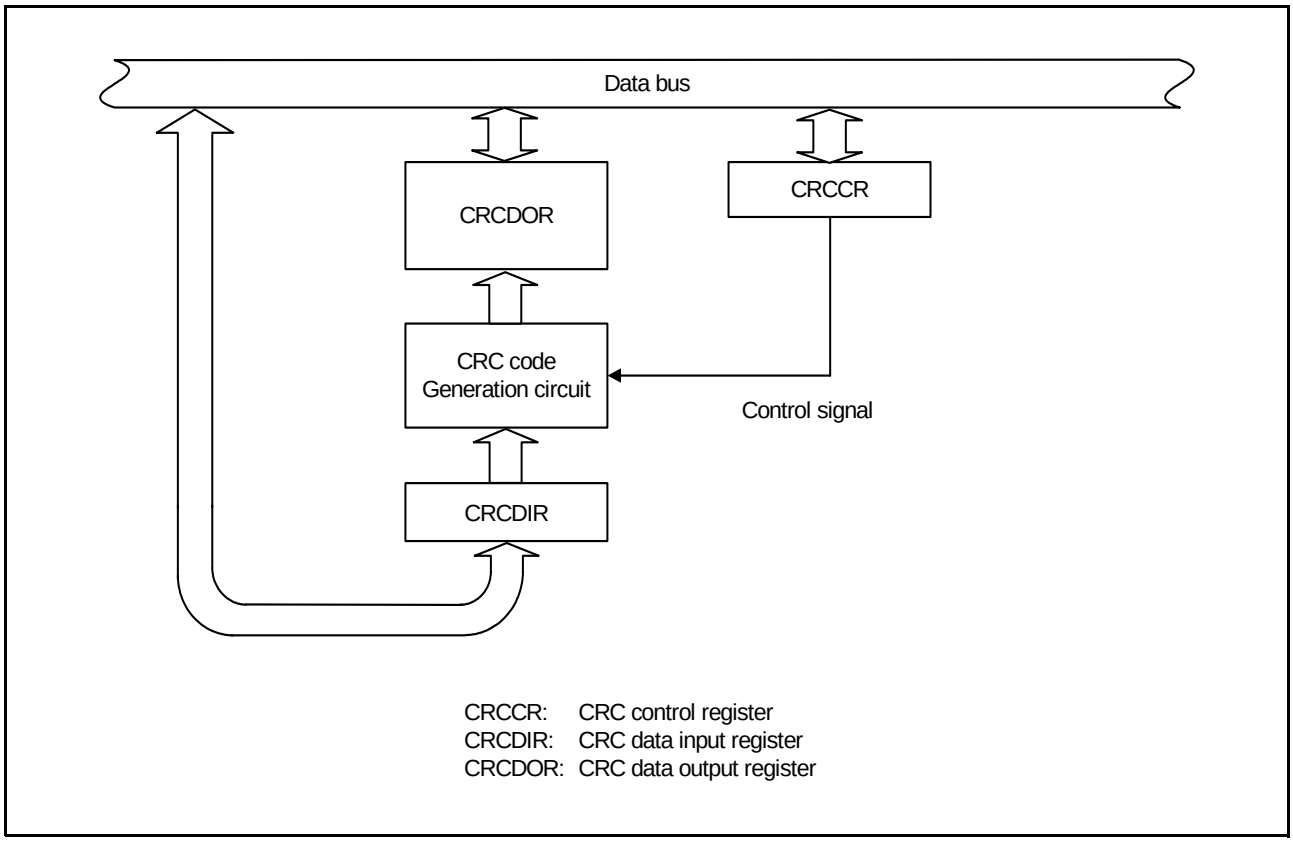


Figure 35.1 CRC Block Diagram

35.2 Register Descriptions

35.2.1 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 35.3, 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.

35.2.2 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.

35.2.3 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.

35.3 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.

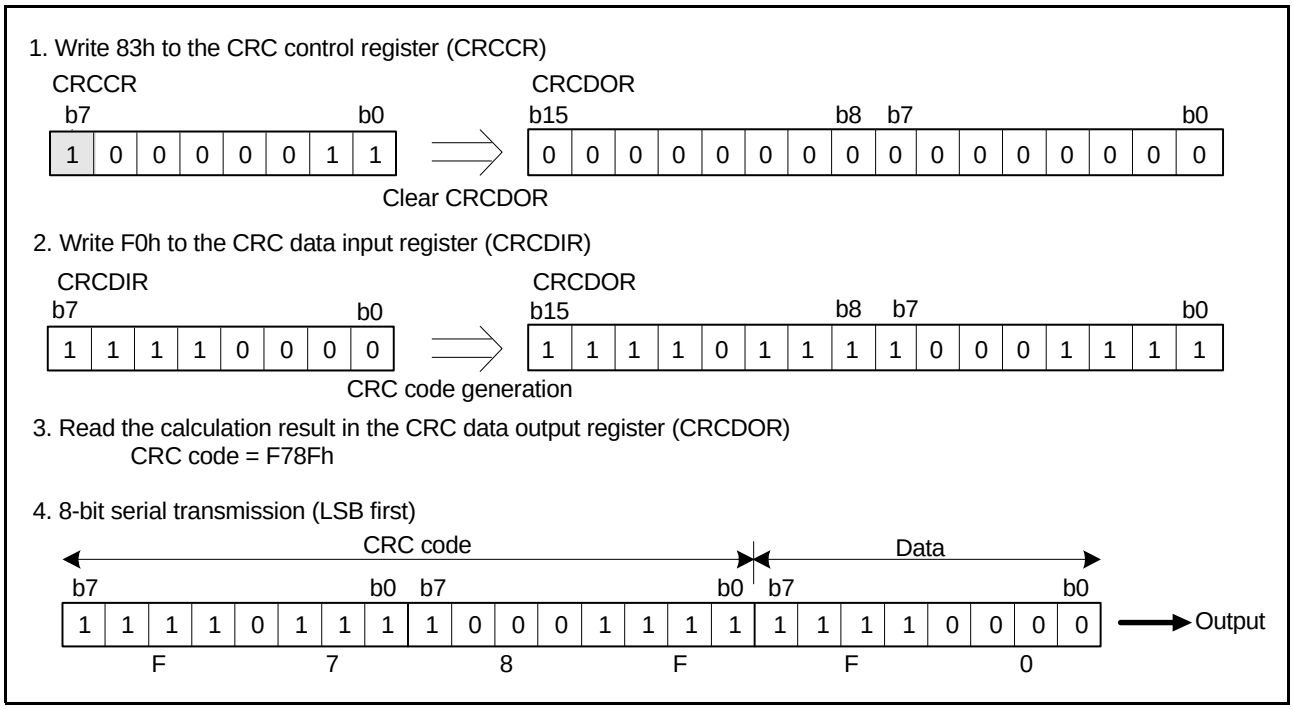


Figure 35.2 LSB First Data Transmission

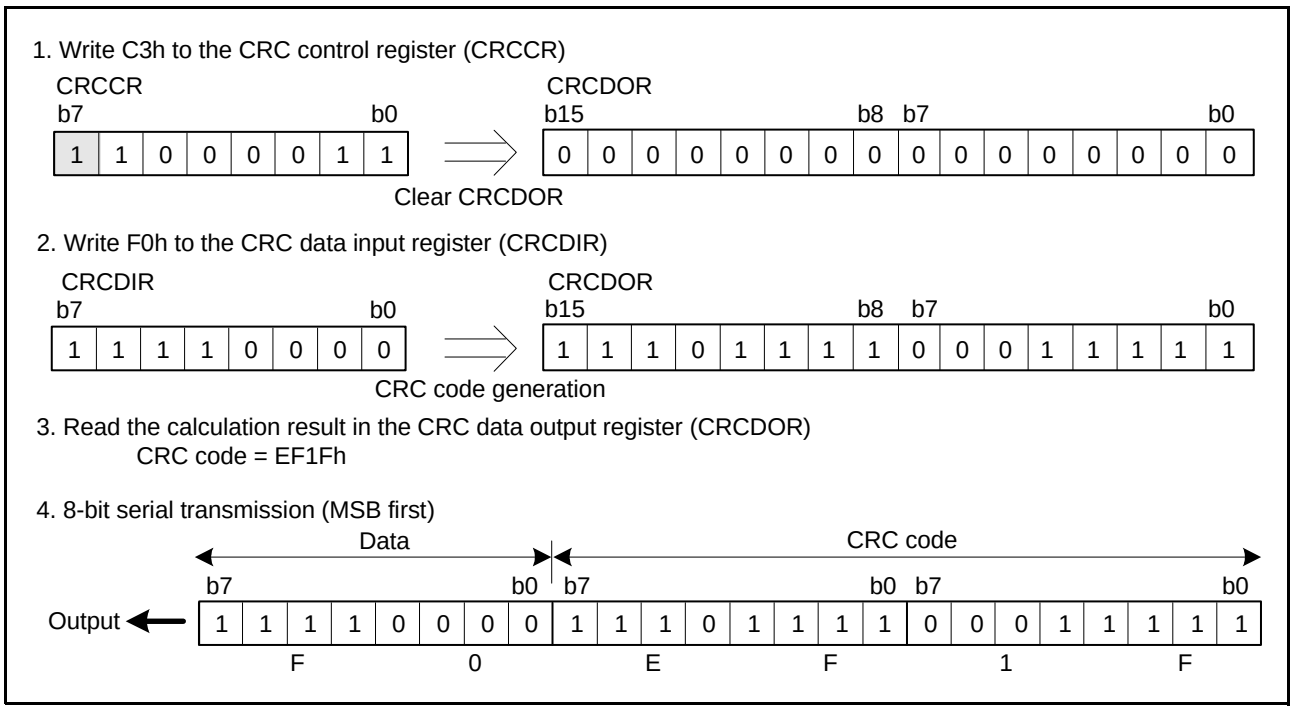


Figure 35.3 MSB First Data Transmission

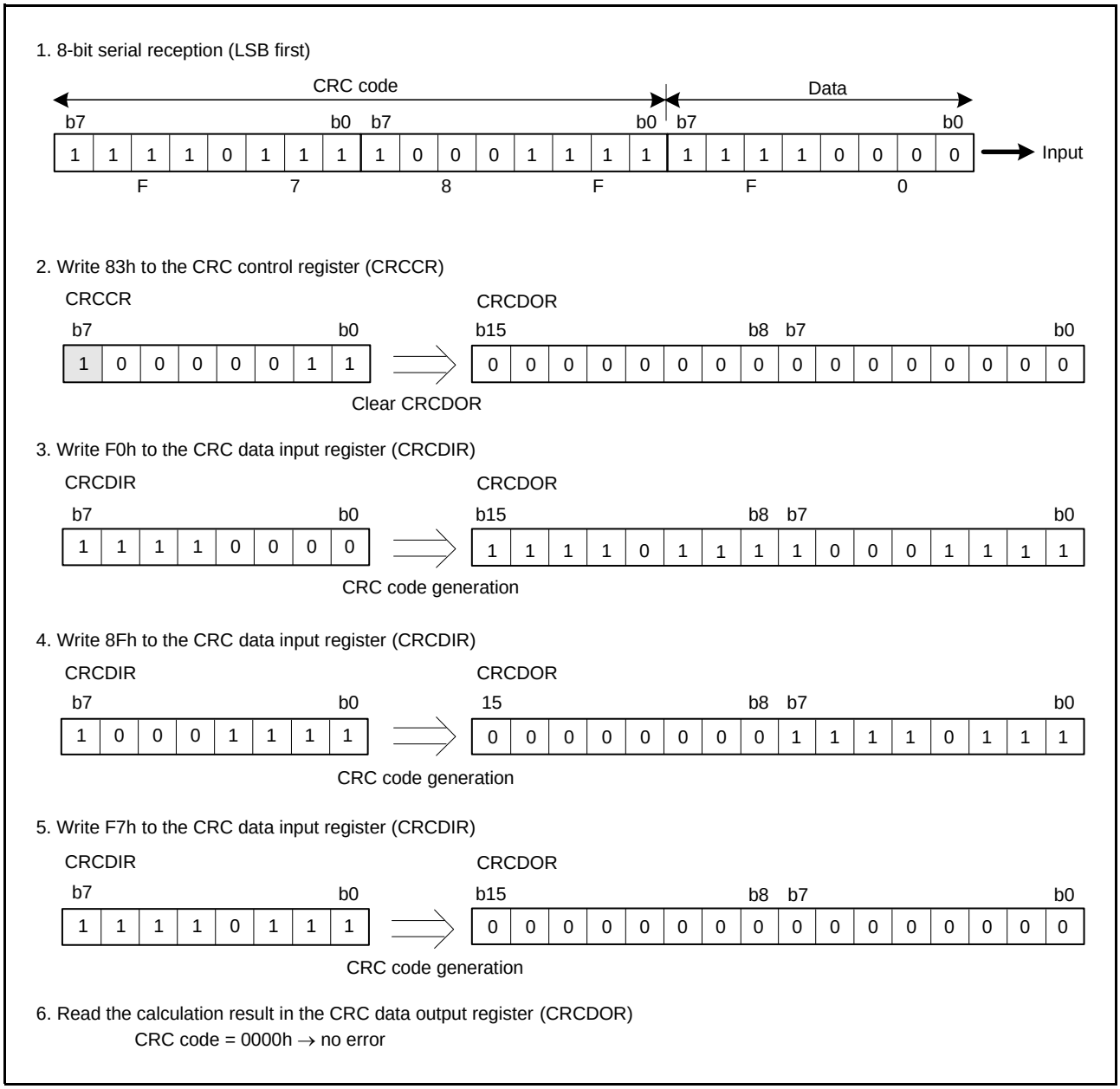


Figure 35.4 LSB First Data Reception

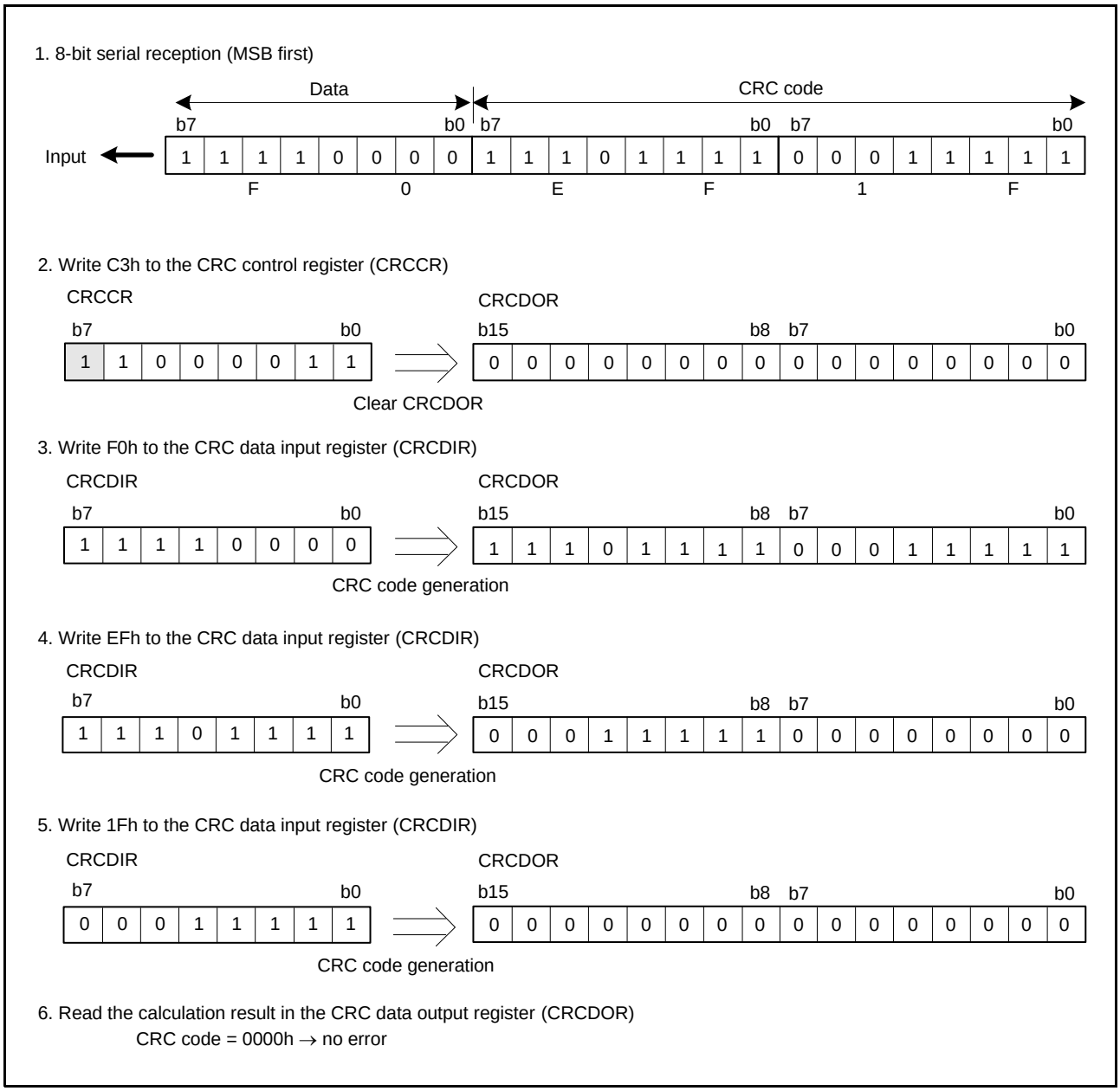


Figure 35.5 MSB First Data Reception

35.4 Usage Notes

35.4.1 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.

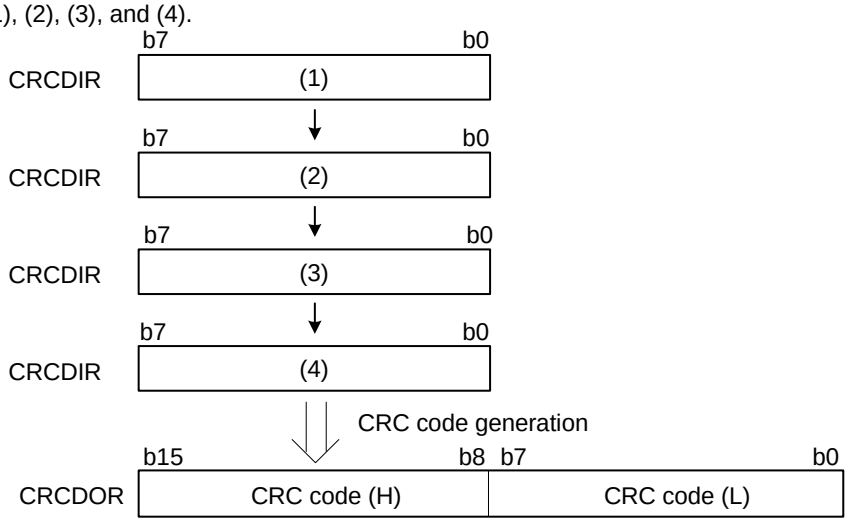
35.4.2 Note on Transmission

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

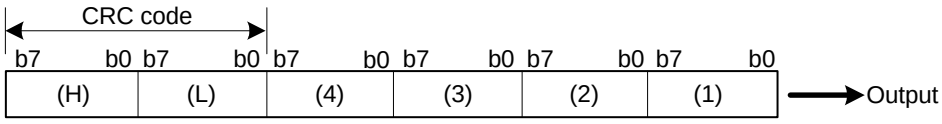
1. CRC code

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

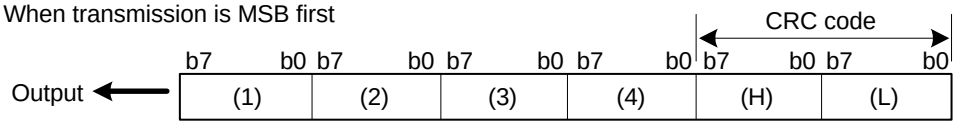


Figure 35.6 LSB First and MSB First Data Transmission

36. 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.

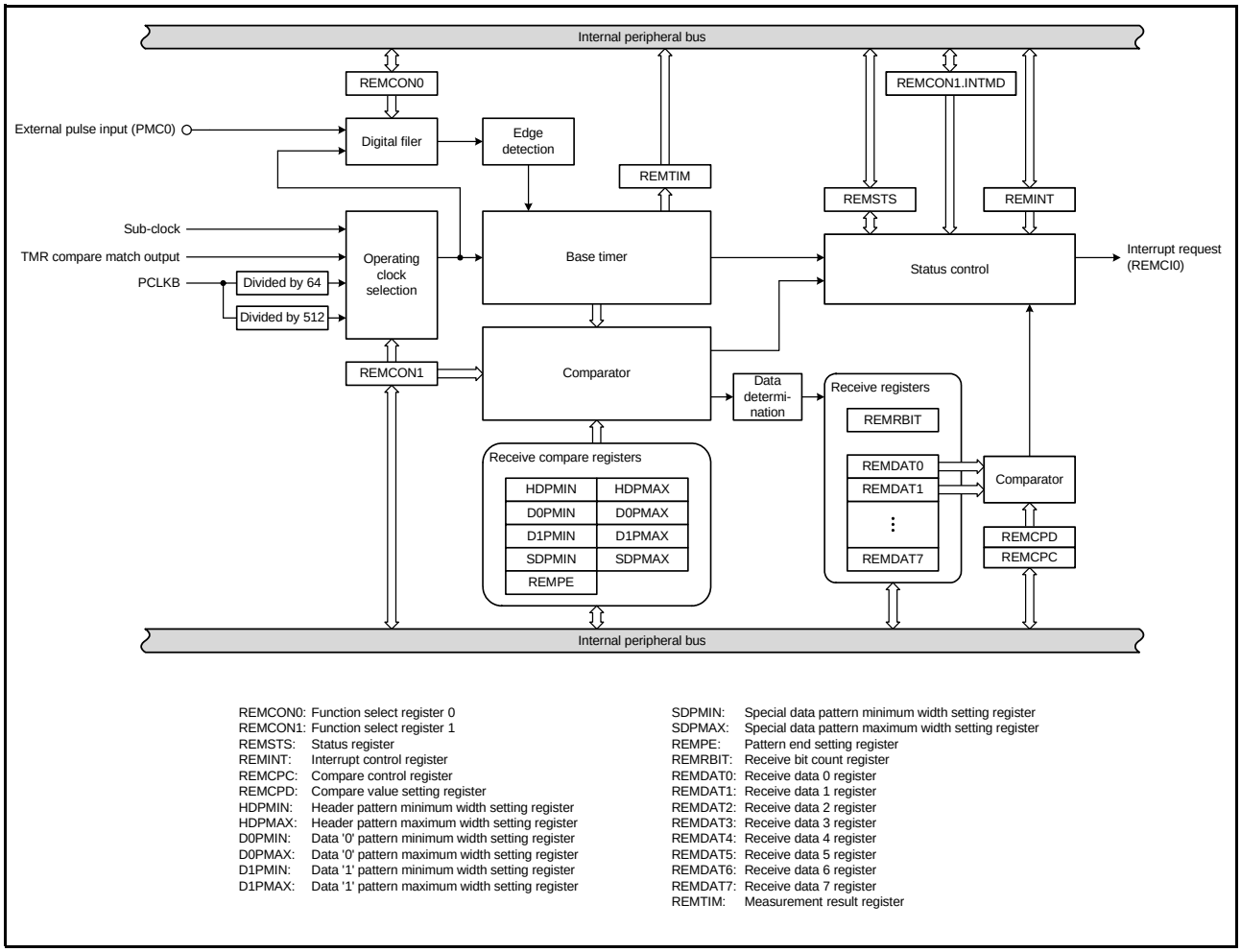
36.1 Overview

Table 36.1 lists the REMC specifications. Figure 36.1 shows a block diagram of the REMC.

Table 36.1 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.



REMCON0: Function select register 0
REMCON1: Function select register 1
REMSTS: Status register
REMINT: Interrupt control register
REMCPC: Compare control register
REMCPD: Compare value setting register
HDPMIN: Header pattern minimum width setting register
HDPMAX: Header pattern maximum width setting register
D0PMIN: Data '0' pattern minimum width setting register
D0PMAX: Data '0' pattern maximum width setting register
D1PMIN: Data '1' pattern minimum width setting register
D1PMAX: Data '1' pattern maximum width setting register

SDPMIN: Special data pattern minimum width setting register
SDPMAX: Special data pattern maximum width setting register
REMPE: Pattern end setting register
REMRBIT: Receive bit count register
REMDAT0: Receive data 0 register
REMDAT1: Receive data 1 register
REMDAT2: Receive data 2 register
REMDAT3: Receive data 3 register
REMDAT4: Receive data 4 register
REMDAT5: Receive data 5 register
REMDAT6: Receive data 6 register
REMDAT7: Receive data 7 register
REMTIM: Measurement result register

Figure 36.1 REMC Block Diagram

Table 36.2 lists the input pins used for the REMC.

Table 36.2 REMC Pin Configuration

Channel	Pin Name	I/O	Function
REMC0	PMC0	Input	External pulse signal input

36.2 Registers

36.2.1 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.

36.2.2 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 36.3.3, Pattern Setting. 0 1: Format B shown in section 36.3.3, Pattern Setting. 1 0: Format C shown in section 36.3.3, 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 36.3.12, Interrupts.

36.2.3 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 36.4.7, 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 36.3, 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 36.3, Measurement Results and Flags.

D0FLG Flag (Data ‘0’ Pattern Match Flag)

[Setting condition]

- See Table 36.3, 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 36.3, Measurement Results and Flags.

D1FLG Flag (Data ‘1’ Pattern Match Flag)

[Setting condition]

- See Table 36.3, 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 36.3, Measurement Results and Flags.

SDFLG Flag (Special Data Pattern Match Flag)

[Setting condition]

- See Table 36.3, 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 36.3, Measurement Results and Flags.

Table 36.3 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.

36.2.4 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).

36.2.5 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).

36.2.6 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).

36.2.7 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).

36.2.8 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).

36.2.9 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).

36.2.10 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).

36.2.11 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).

36.2.12 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).

36.2.13 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).

36.2.14 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).

36.2.15 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).

36.2.16 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 36.4.7, Reading Registers.

Note: The values of this register are initialized when the REMCON1.EN bit is 0.

36.2.17 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 36.4.7, Reading Registers.
Note: The values of this register are initialized when the REMCON1.EN bit is 0.

36.2.18 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 36.4.7, 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 36.3.8, Receive Data Buffer.

36.2.19 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 36.4.7, Reading Registers.

Note: The values of this register are initialized when the REMCON1.EN bit is 0.

36.3 Operation

36.3.1 Overview of REMC Operation

Figure 36.2 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.

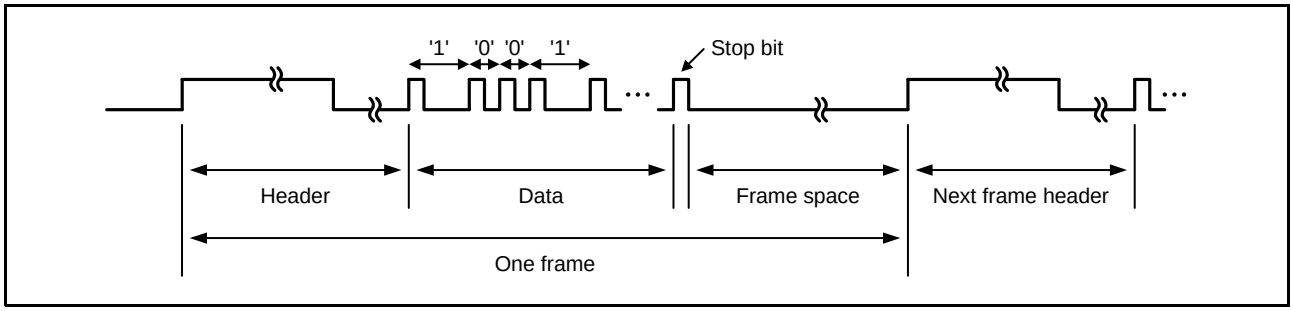


Figure 36.2 Example of Remote Control Signal

36.3.2 Initial Setting

Initialize the REMC according to the procedure shown in Figure 36.3 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.

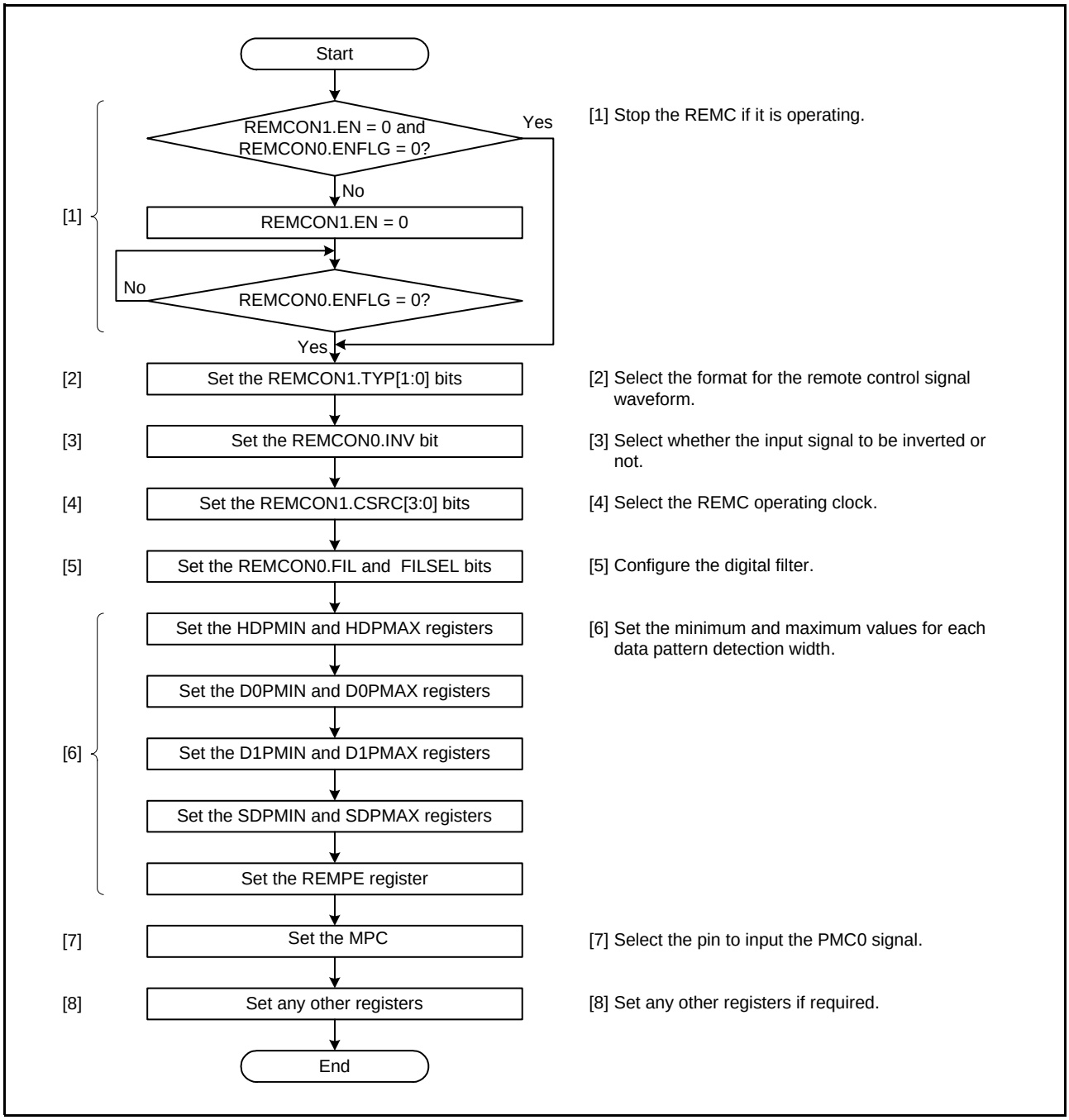


Figure 36.3 Example of Flowchart for Initial Settings of REMC

36.3.3 Pattern Setting

The format for capturing the remote control signal reception waveform can be set by setting the REMCON1.TYP[1:0] bits. Figure 36.4 and Figure 36.5 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)

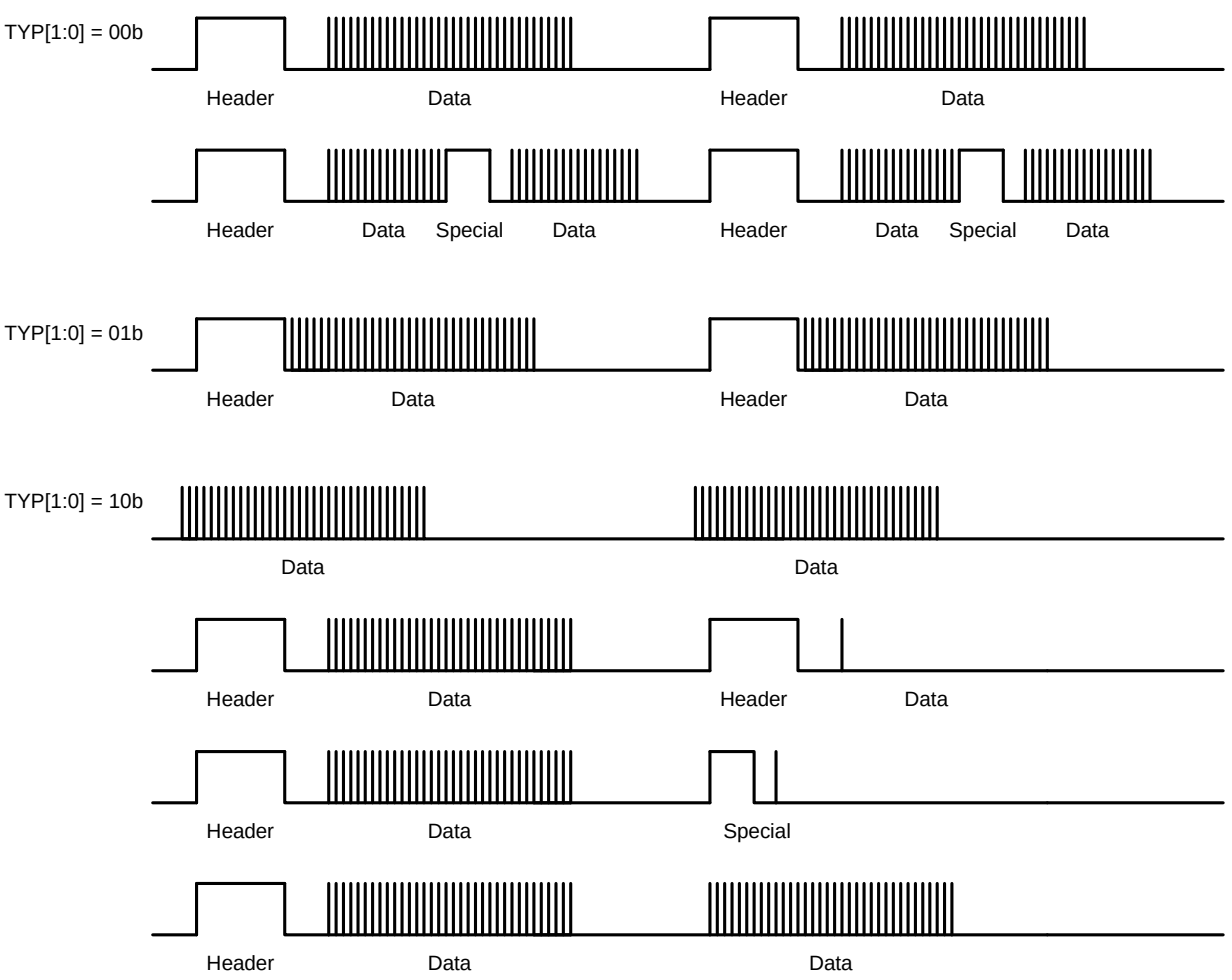
The measured result is determined from the setting value of the header pattern at the rising edge of the internal input signal.
When the header pattern is received, the measured result is determined from the setting values of the data ‘0’, data ‘1’ and special data patterns at the rising edge of the 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				

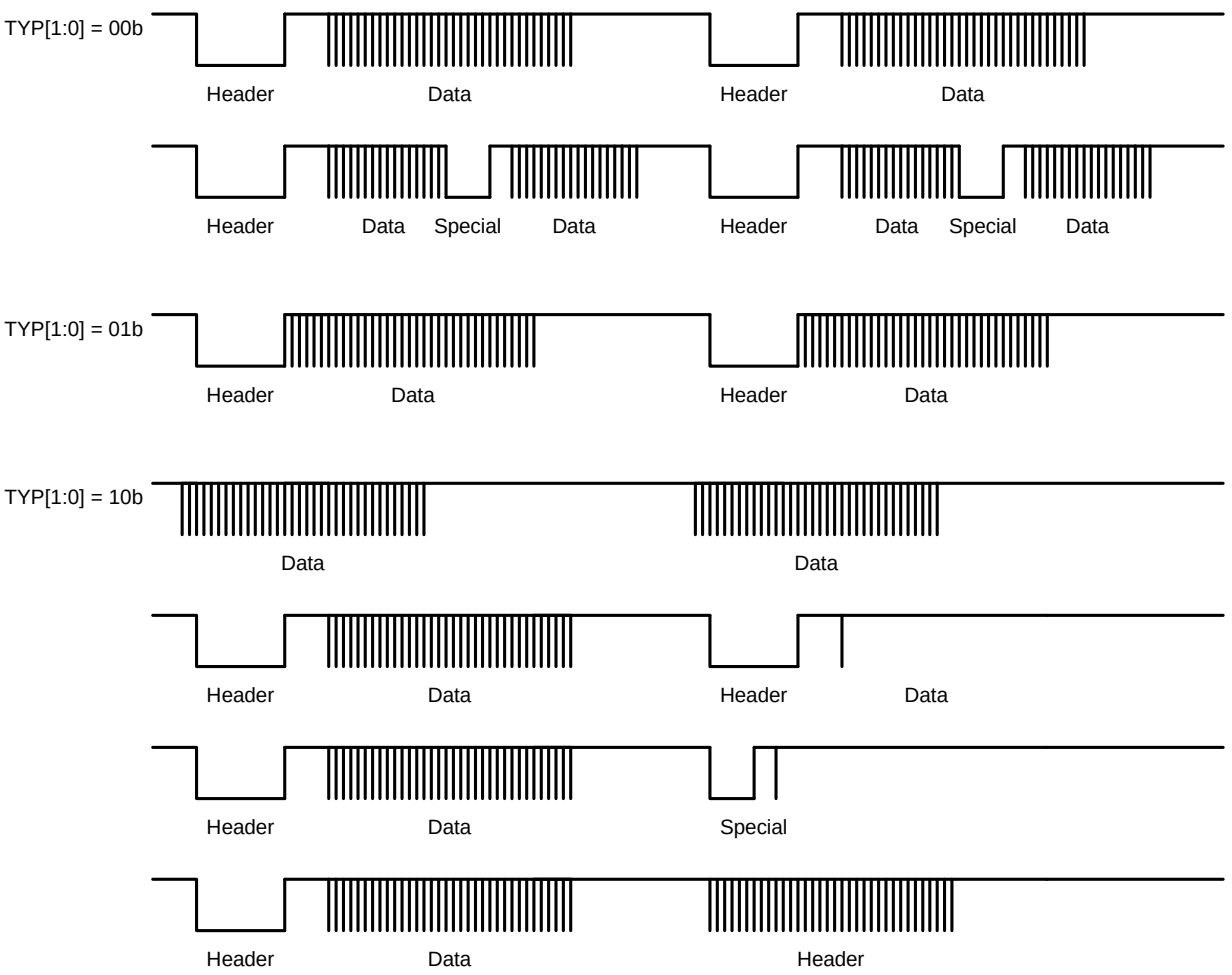
TYP[1:0]: Bits in the REMCON1 register

↑↓・・・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 36.4 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 36.5 Example of Remote Control Signal Reception Waveform Captured by Setting REMCON1.TYP[1:0] Bits (REMCON0.INV = 1)

36.3.4 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.

36.3.4.1 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).

36.3.4.2 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 24, 8-Bit Timer (TMRb).

36.3.5 PMC0 Input

The options below can be selected in PMC0 input.

- Input polarity
- Digital filter

Figure 36.6 shows the configuration of PMC0 internal input signal generation.

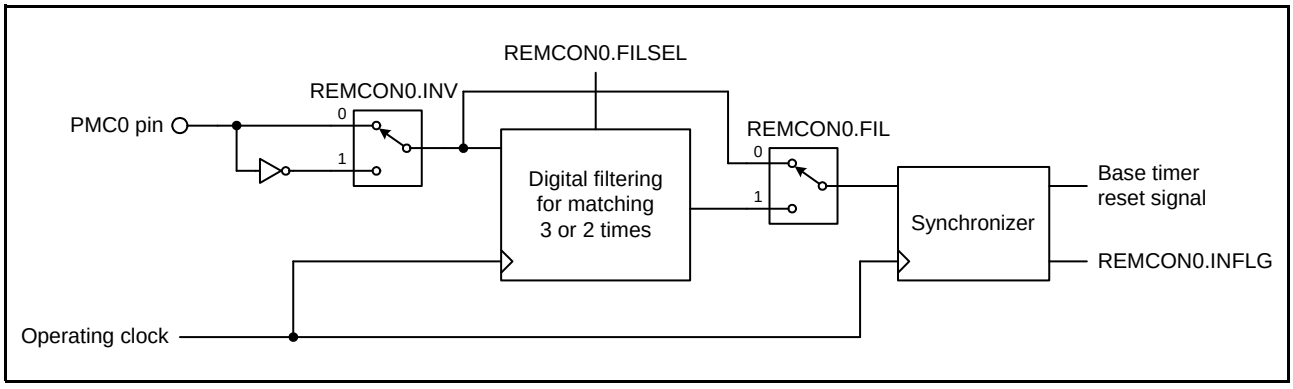
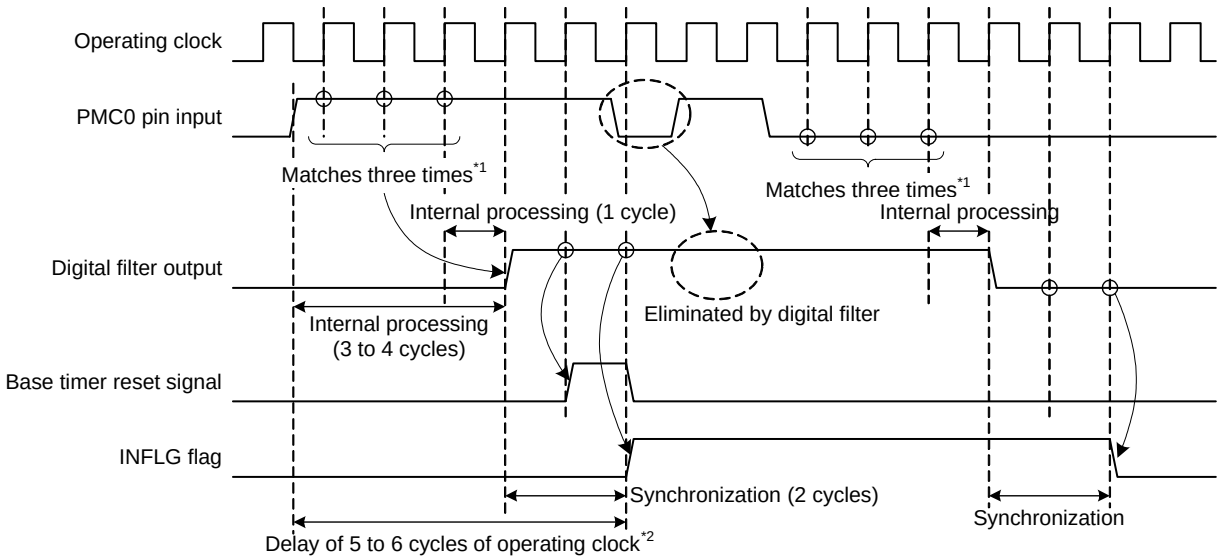


Figure 36.6 PMC0 Internal Input Signal Generation Configuration

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 36.7 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 36.7 Digital Filtering for PMC0 Input

36.3.6 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 36.8 shows the waveform of REMC operation.

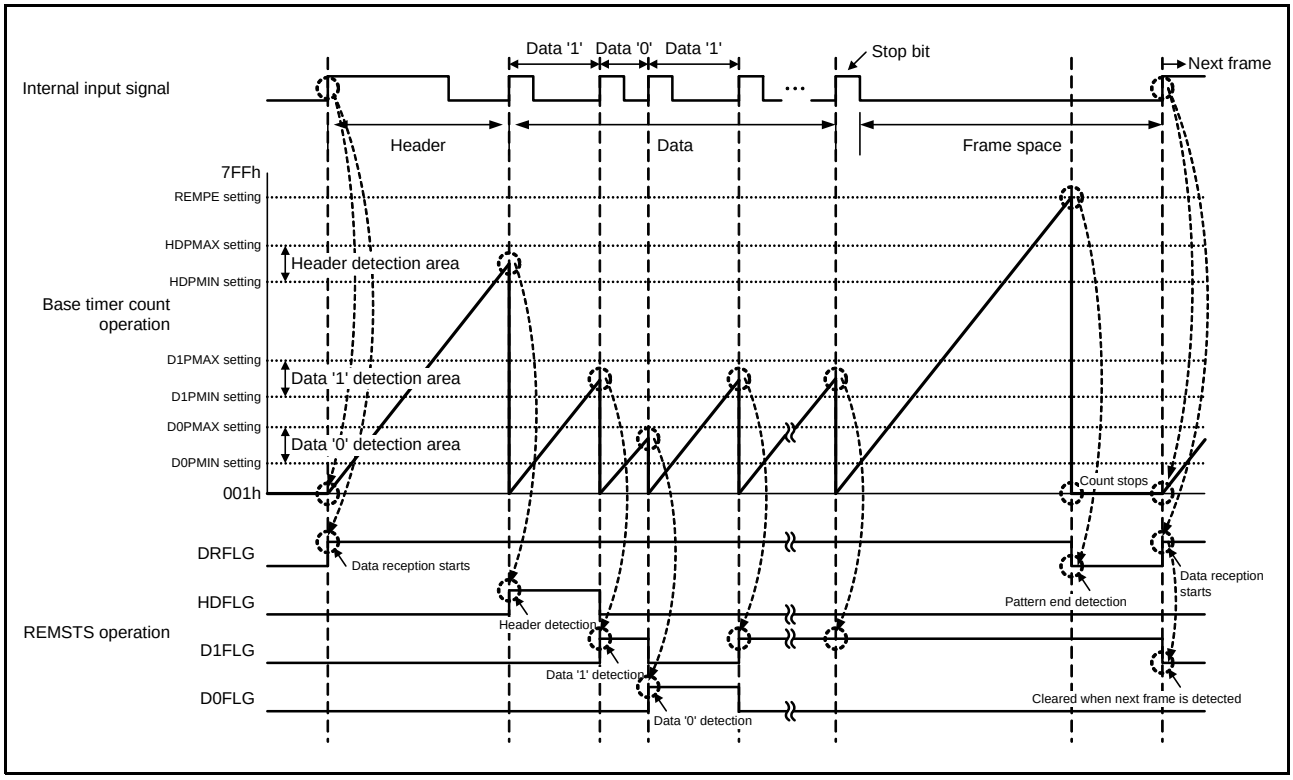


Figure 36.8 Waveform of REMC Operation

36.3.6.1 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.

36.3.6.2 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.

36.3.6.3 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.

36.3.6.4 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.

36.3.6.5 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 36.9.

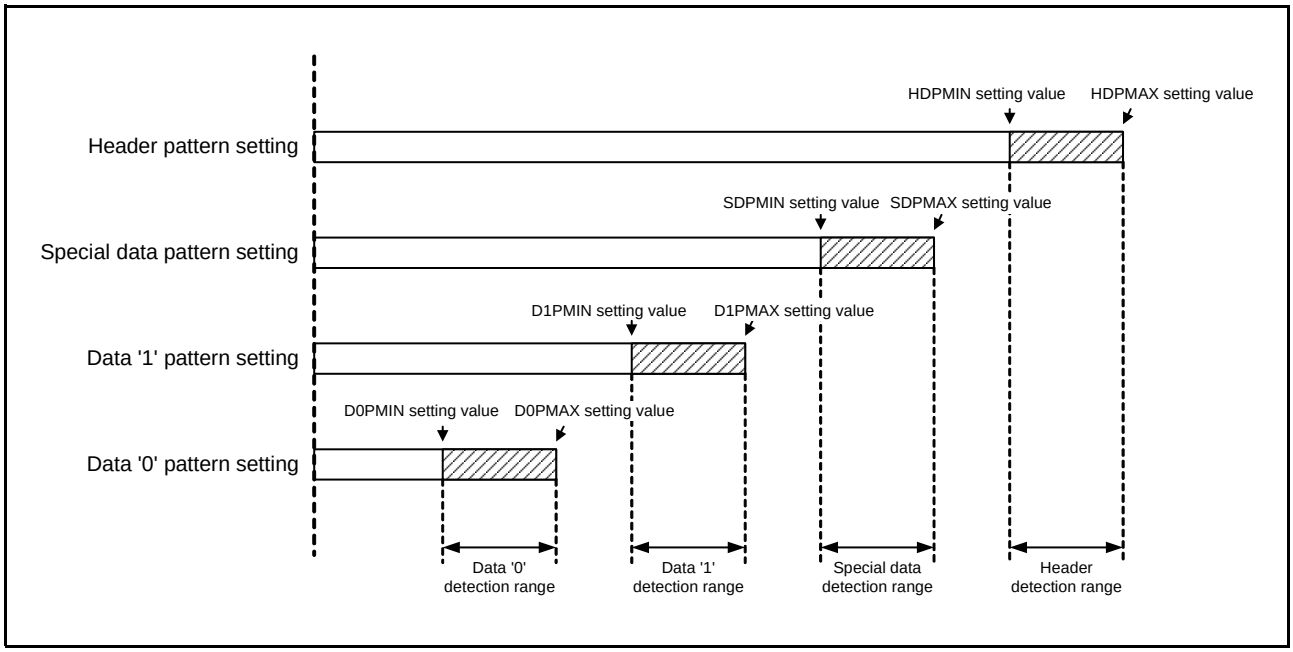


Figure 36.9 Examples of Setting Pattern Setting Registers

36.3.6.6 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 36.10 shows pattern detection and an example of flag operation.

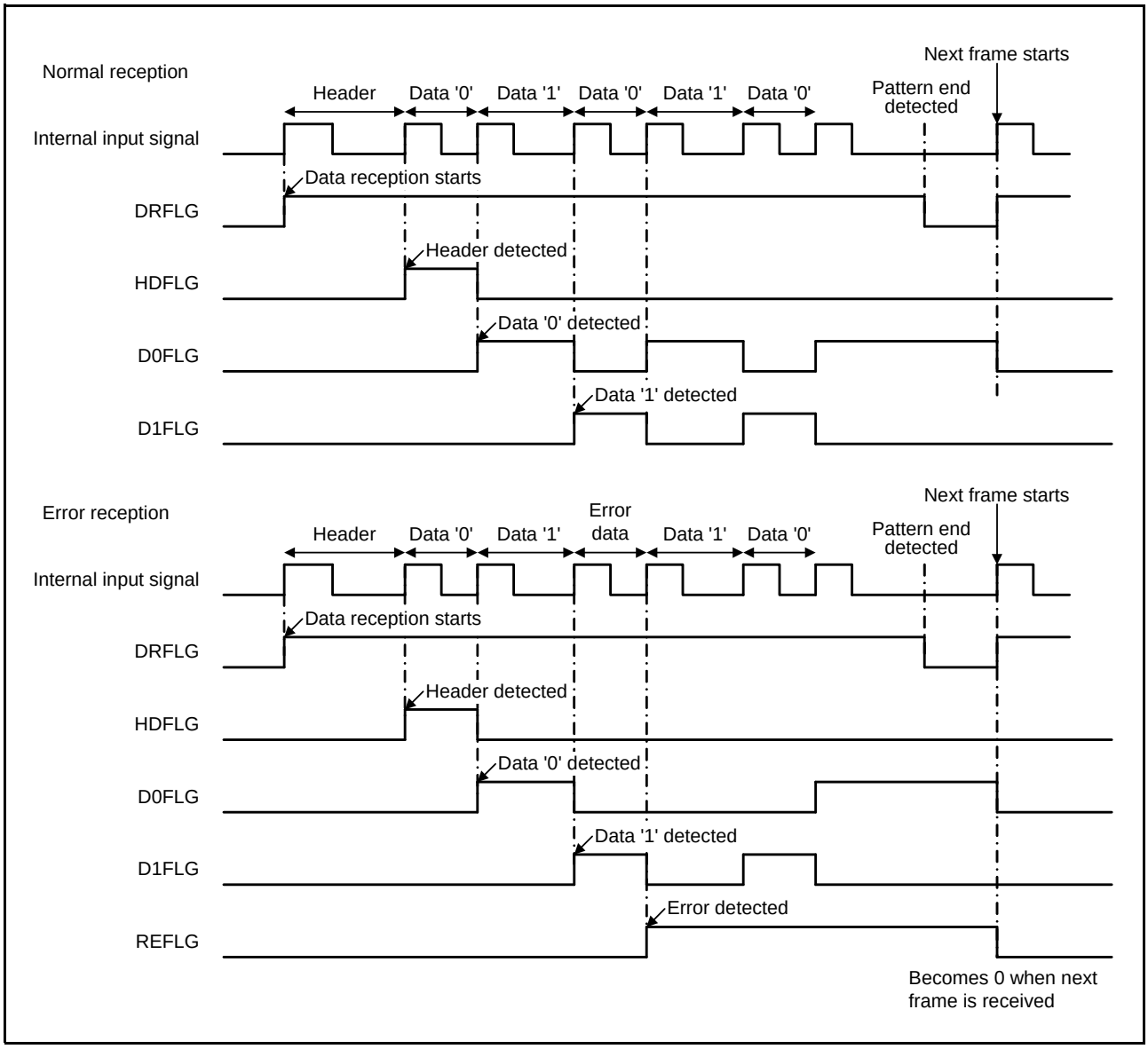


Figure 36.10 Example of Flag Operation

36.3.7 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 36.11 shows operation of the data reception complete flag for each pattern end setting.

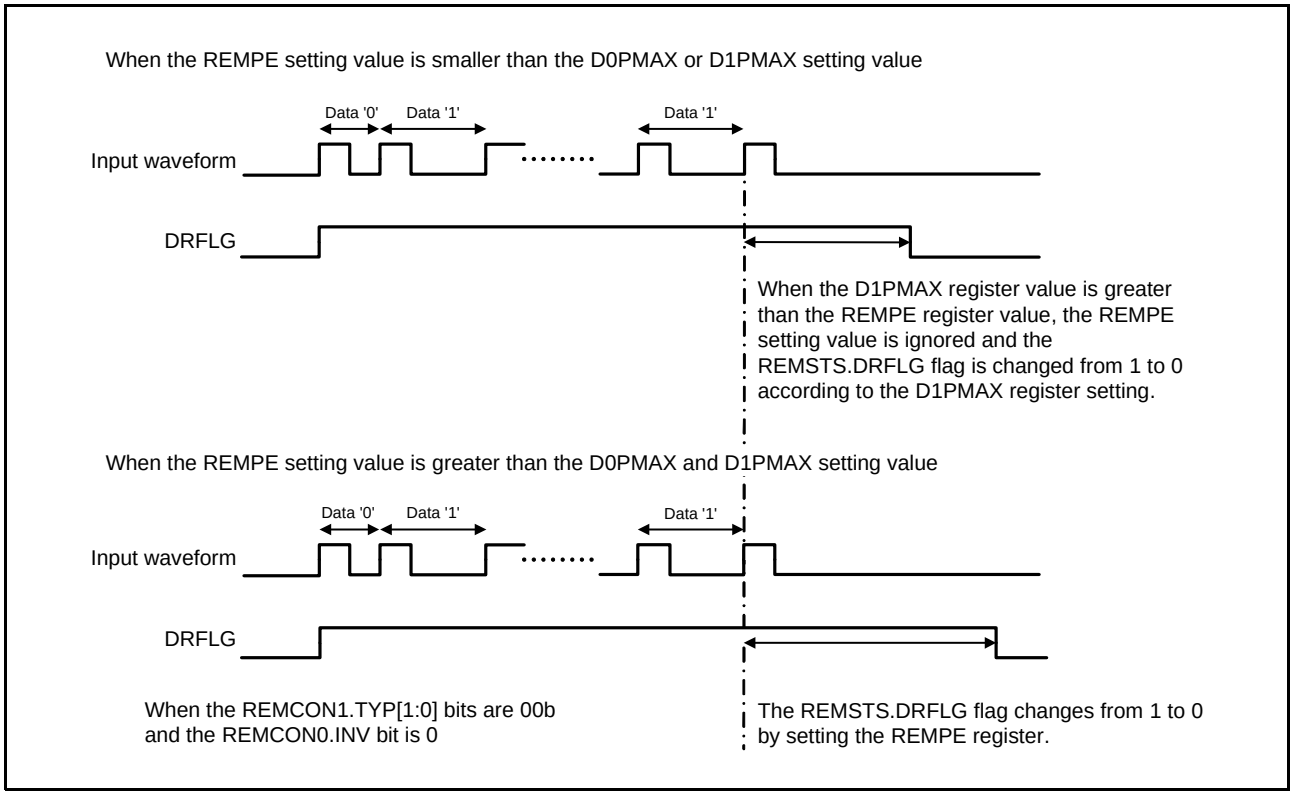


Figure 36.11 Operation of Data Reception Complete Flag for Each Pattern End Setting

36.3.8 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 36.12. 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 36.4 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.

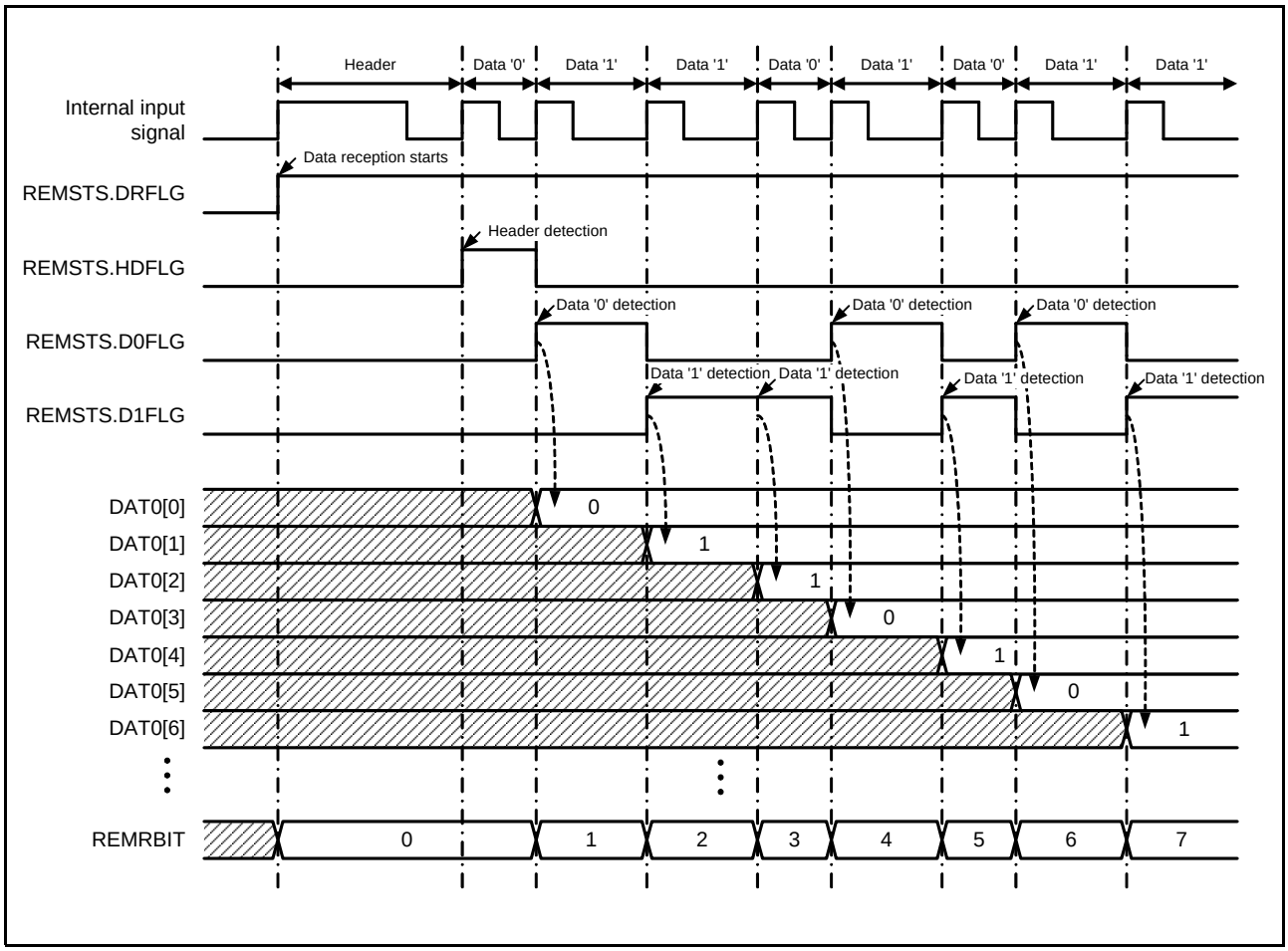


Figure 36.12 Operation of Receive Data Buffer

Table 36.4 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 36.13 shows the REMDATj/REMRBIT register operation when 00h is written to the REMDAT0 register.

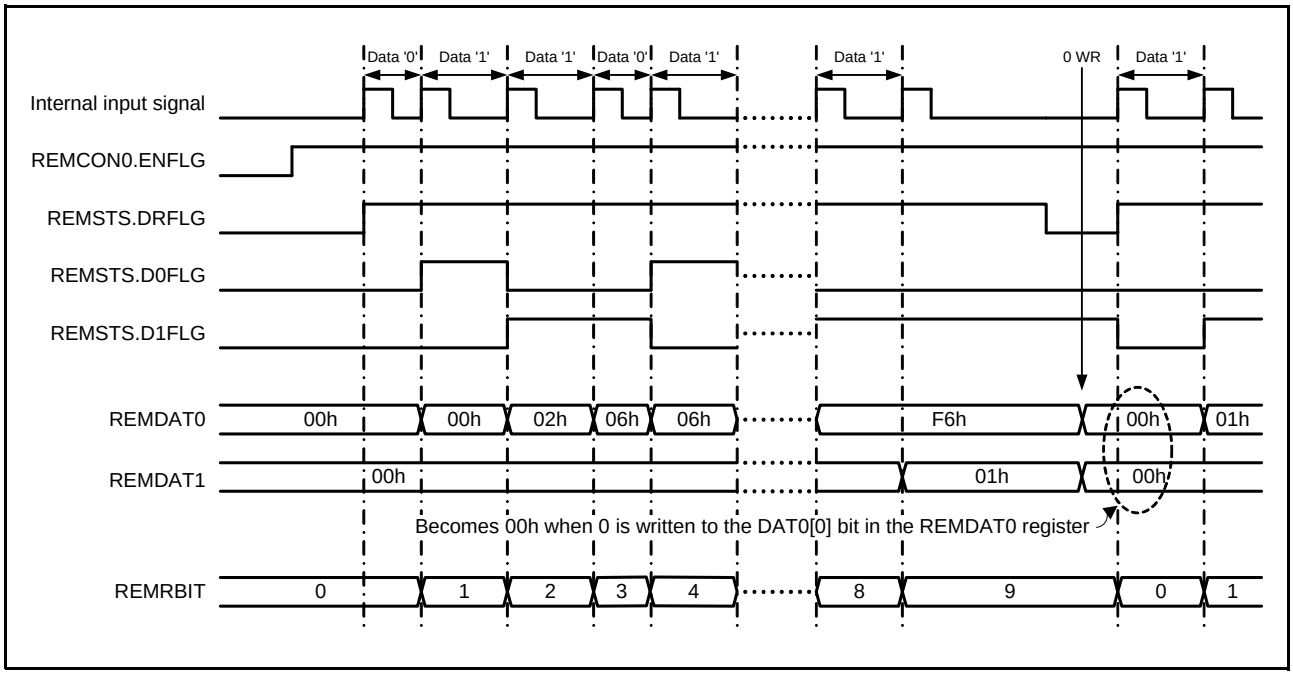


Figure 36.13 REMDATj/REMRBIT Register Operation (00h is Written to REMDAT0 Register)

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 36.14 shows operation of header pattern detection during data reception.

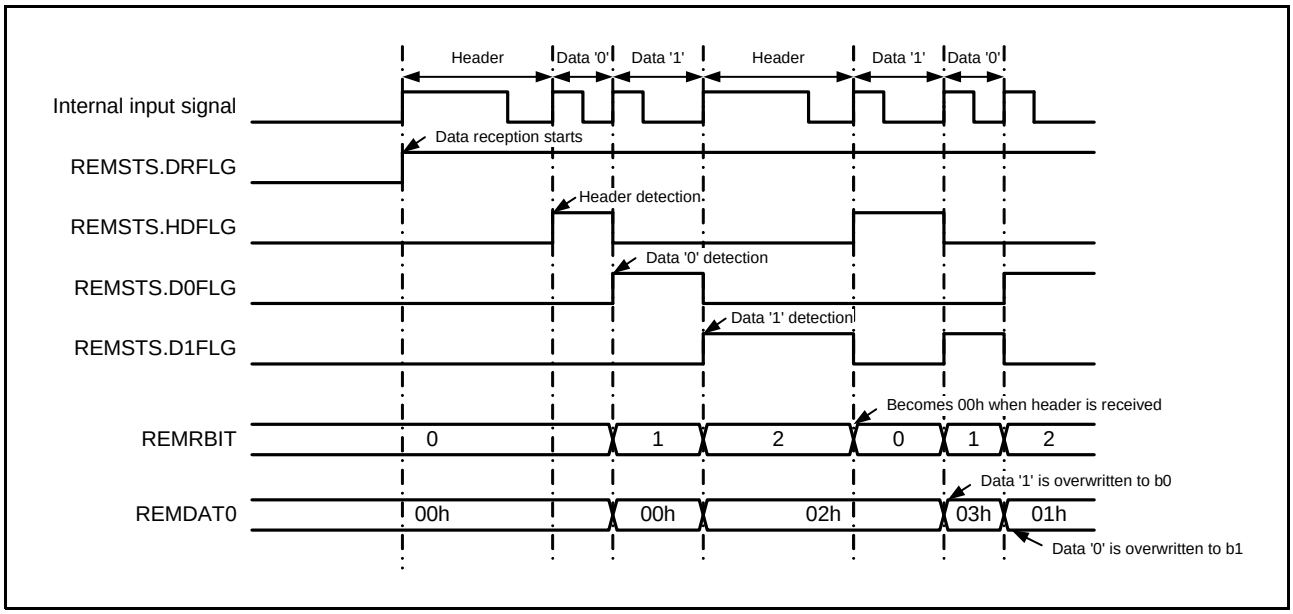


Figure 36.14 Operation of Header Pattern Detection during Data Reception

When the data exceeds 64 bits, the buffer is sequentially overwritten from the first bit. Figure 36.15 shows the REMRBIT register operation when the REMSTS.BFULFLG flag becomes 1.

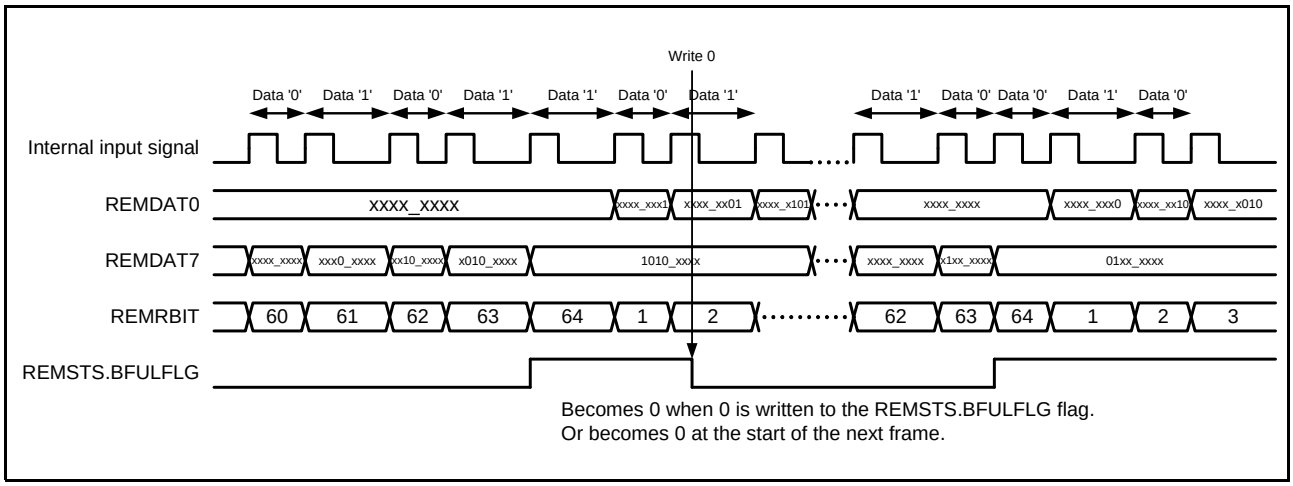


Figure 36.15 REMRBIT Register Operation (REMSTS.BFULFLG Flag = 1)

36.3.9 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 36.16 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).

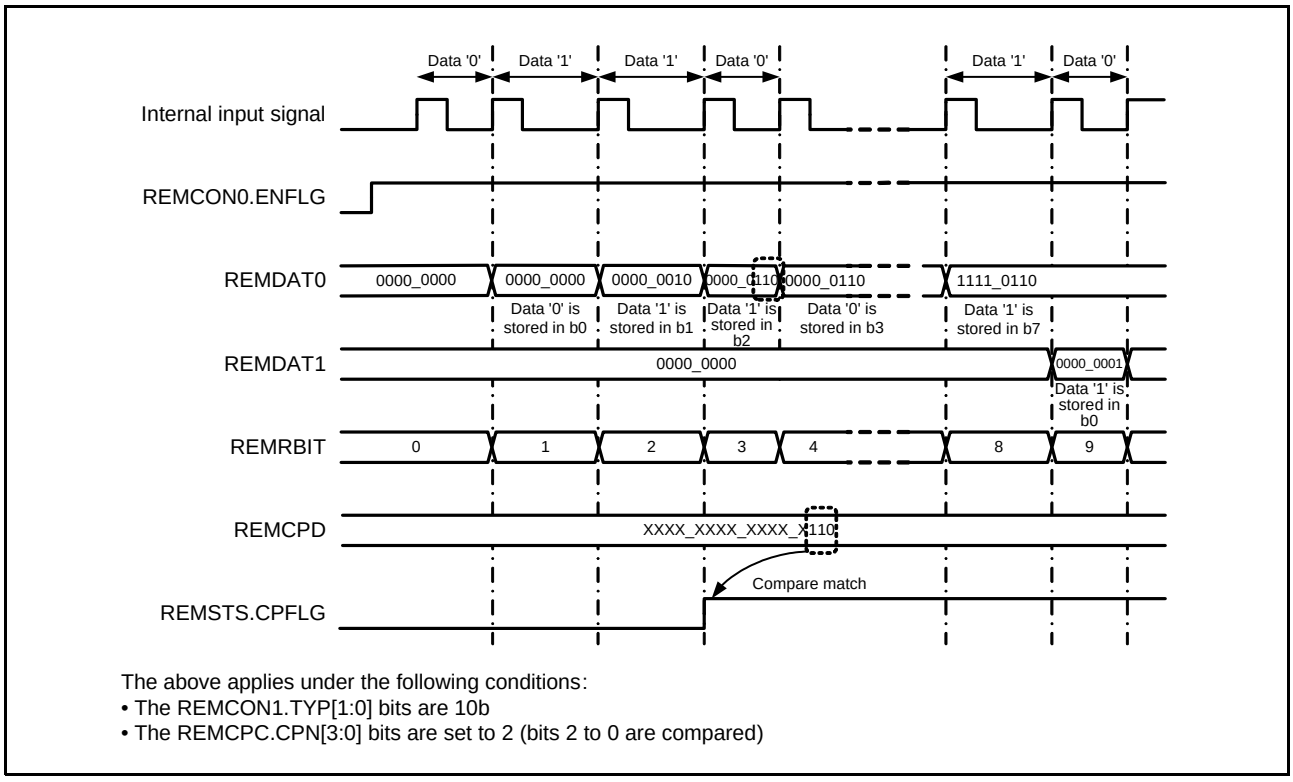


Figure 36.16 Receive Buffer and Compare Function

36.3.10 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 36.17 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 36.18 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.

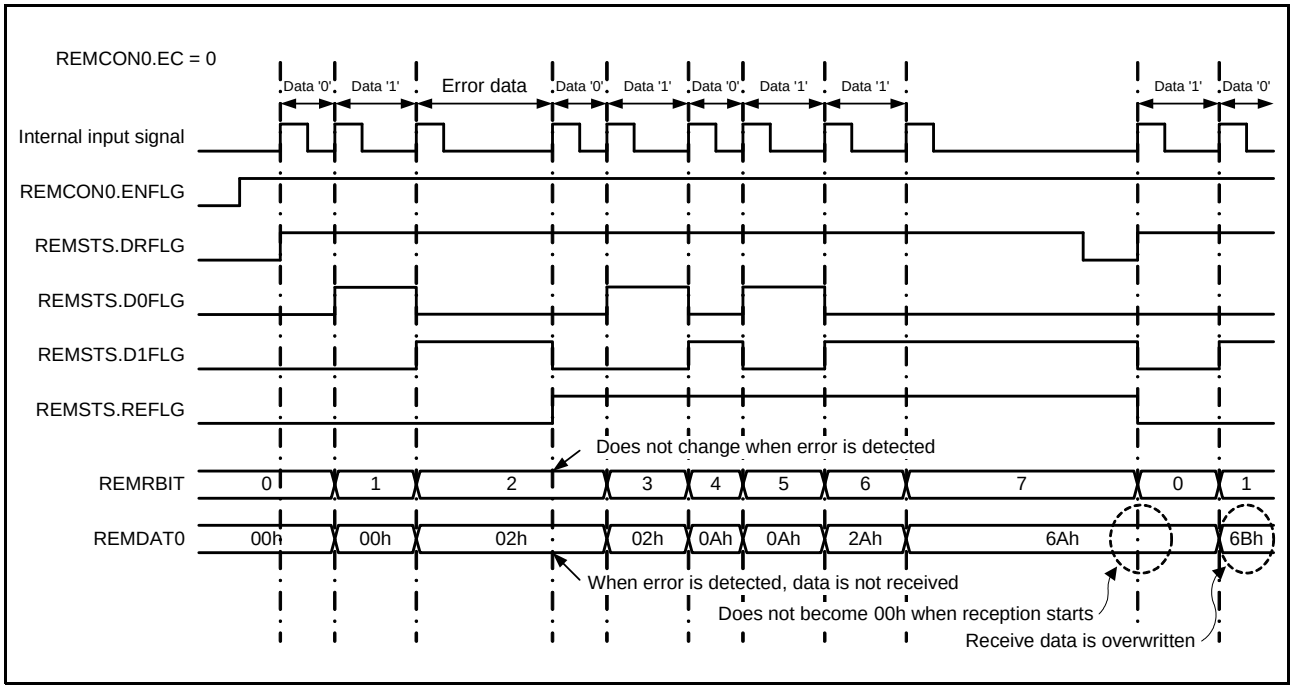


Figure 36.17 REMDAT0 and REMRBIT Registers Operation upon Error Detection (REMCON0.EC Bit = 0)

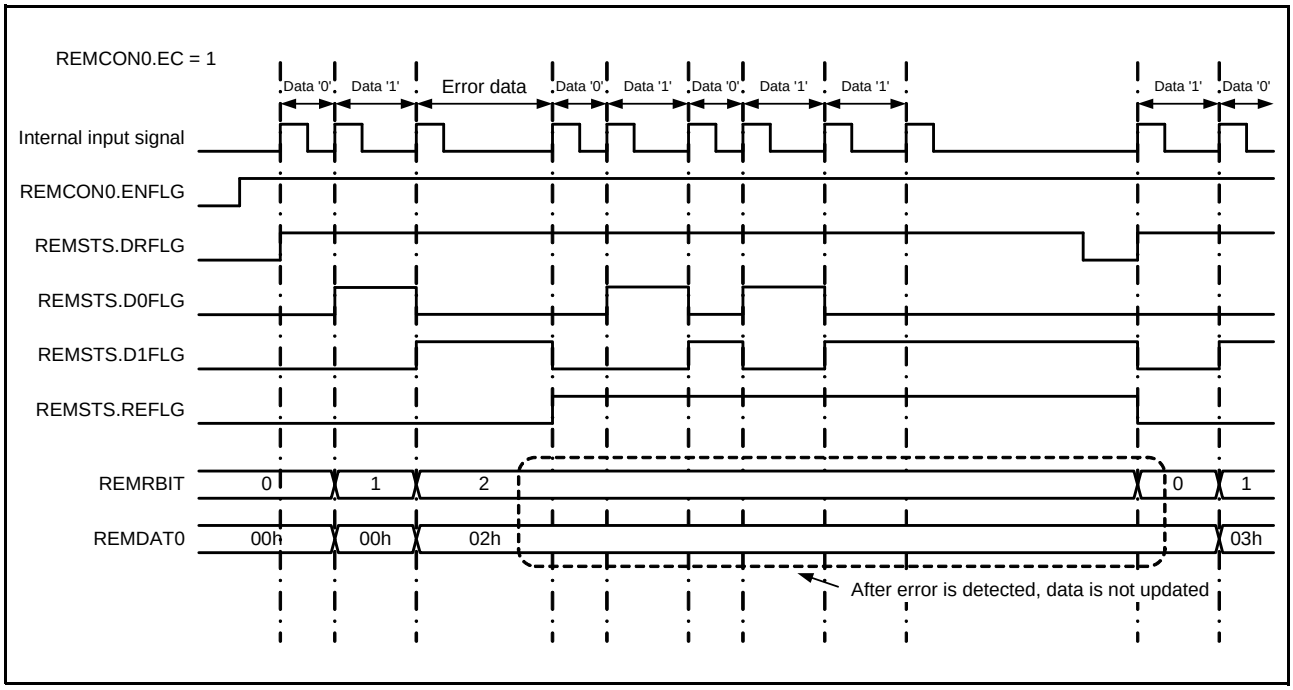


Figure 36.18 REMDAT0 and REMRBIT Registers Operation upon Error Detection (REMCON0.EC Bit = 1)

36.3.11 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 36.19 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)

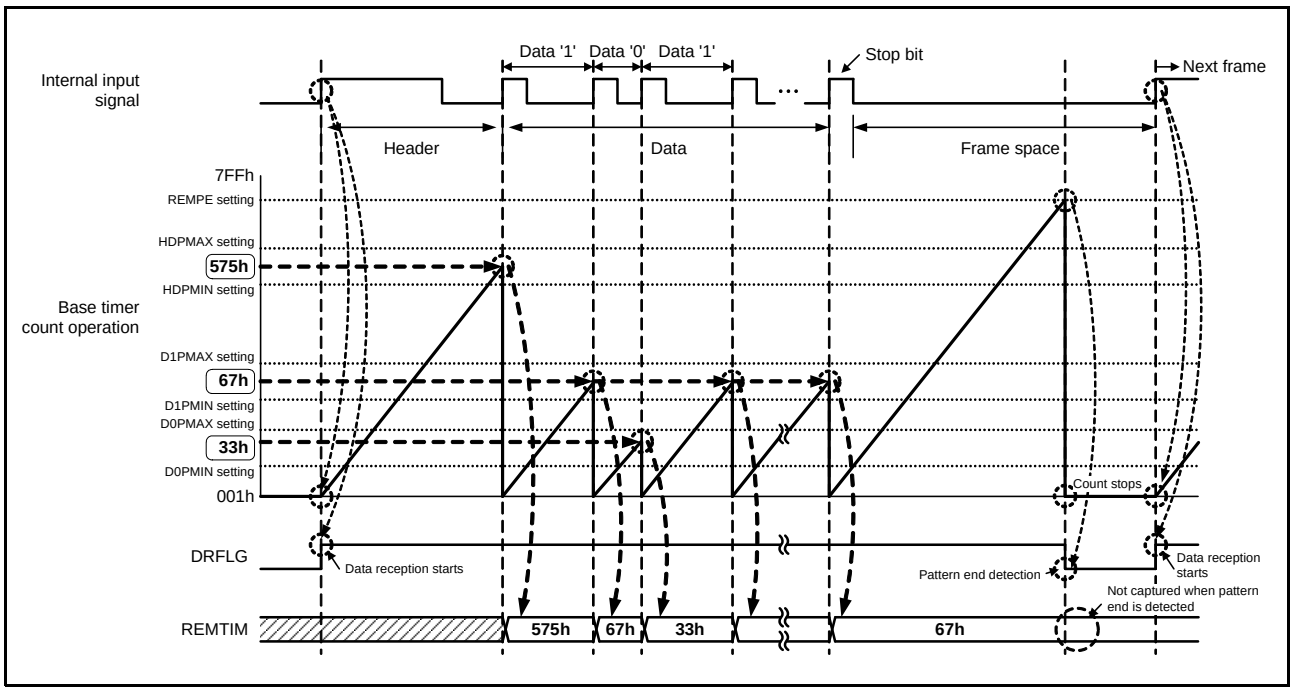


Figure 36.19 Operation Example of Measurement Function

36.3.12 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 36.5 lists the REMC interrupt sources, and Table 36.6 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 14, Interrupt Controller (ICUF) for details on interrupt control.

Table 36.5 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 36.6 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

36.3.13 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, or software standby mode).

To receive data in a low power consumption state, REMC communications should be set before transitioning to the state.

36.3.13.1 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 36.20). Pattern detection and compare function enable returning from the low power consumption state only when specified data is received.

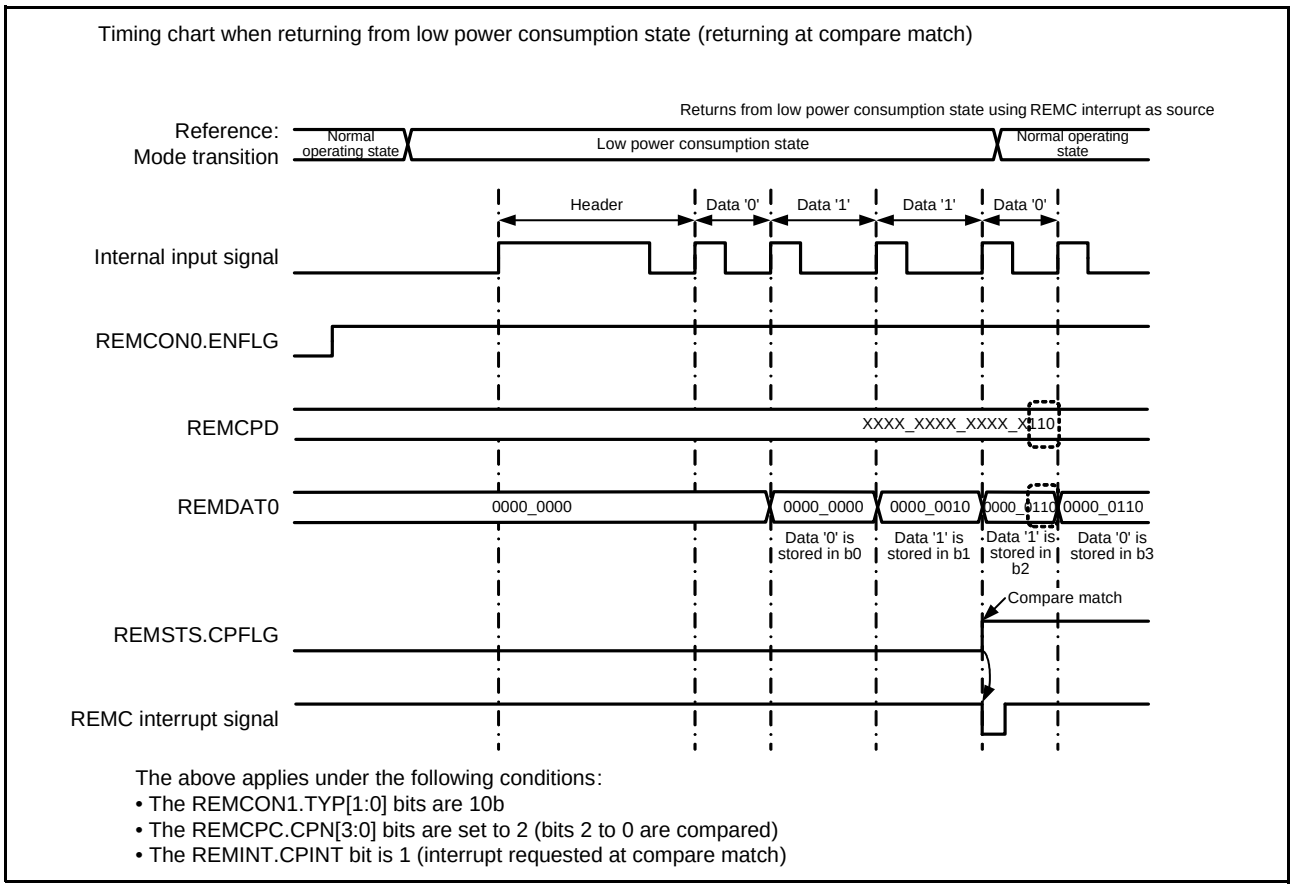
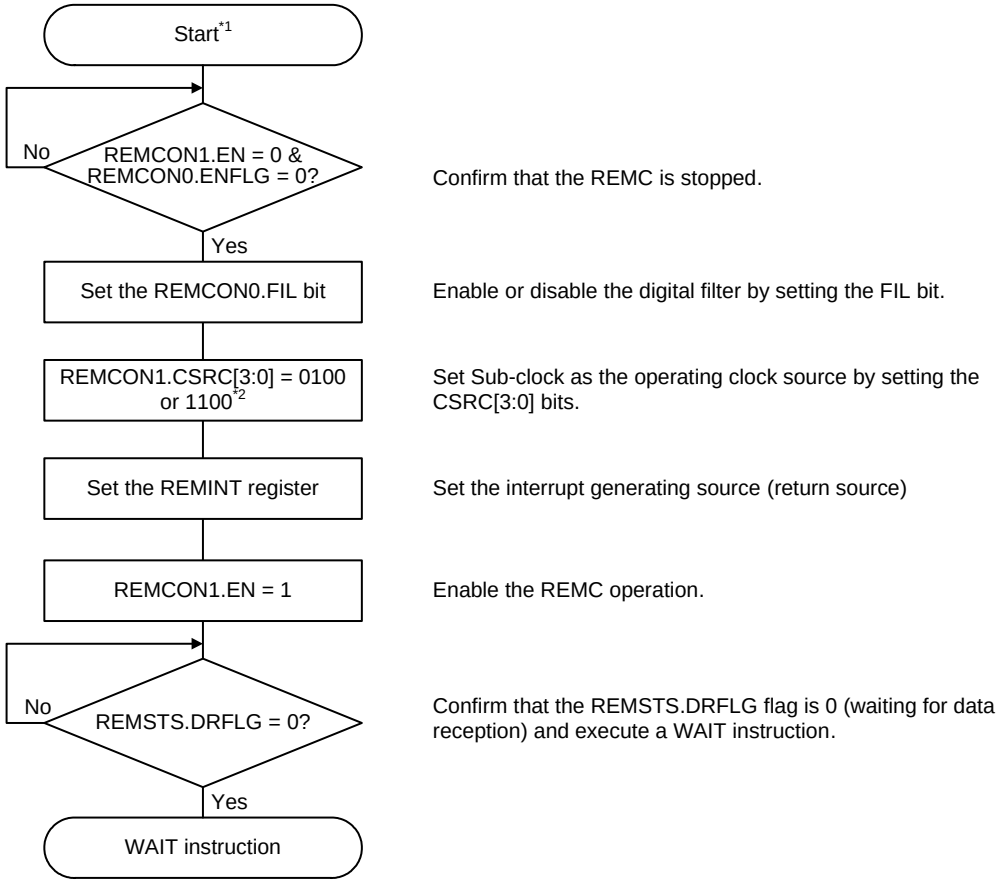


Figure 36.20 Using REMC Interrupt Request to Return from Low Power Consumption State (Normal Interrupt Mode)

36.3.13.2 Data Reception in Software Standby Mode

Data can be received in software standby mode while the sub-clock is selected as the REMC operating clock.
When data is to be received in software standby mode, set the SOFCR.SOFE bit to 1 to select continuous supply of the sub-clock to the REMC as the operating clock in software standby mode. For details on how to supply the REMC operating clock, see section 36.3.4, Operating Clocks.
To return the chip from the 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 36.21 shows an example of a flowchart of the procedure for setting up data reception in software standby mode.



Note 1. Continuous supply of the sub-clock to the REMC is required in software standby mode. This is selected by setting the SOFCR.SOFE 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 36.21 Flowchart for Setting Data Reception in Software Standby Mode

36.4 Usage Notes

36.4.1 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.

36.4.2 Settings for Peripheral Module Clock and REMC Operating Clock

Set the peripheral module clock (PCLKB) frequency to higher than the REMC operating clock frequency.

36.4.3 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 36.2.1 to section 36.2.19) except for the REMCON0.ENFLG flag.

36.4.4 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.

36.4.5 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.

36.4.6 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.

36.4.7 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.

37. Arithmetic Unit for Trigonometric Functions (TFU)

This MCU has an arithmetic unit for trigonometric functions (TFU).
The TFU handles the high-speed calculation of sinf, cosf, atan2f, and hypotf functions.
An option must be specified to make the C or C++ compiler produce instructions that use the TFU. For details, refer to the manual for the compiler you are using.
The ICLK is used as the operating clock for the TFU.

37.1 Overview

Table 37.1 lists the specifications of the TFU.

Table 37.1 TFU specifications

Item	Description				
Arithmetic Processing	Calculation of sine, cosine, arctangent, and $\sqrt{x^2+y^2}$ • A sine and cosine can be simultaneously calculated. • An arctangent and $\sqrt{x^2+y^2}$ can be simultaneously calculated.				
Range and Unit of Values	Arithmetic Processing	I/O		Range	Unit
	Calculating sine	Input	Angle θ	–float_max ≤ θ ≤ float_max*1	radian
		Output	sin(θ)	–1.0 ≤ sin(θ) ≤ 1.0	—
	Calculating cosine	Input	Angle θ	–float_max ≤ θ ≤ float_max*1	radian
		Output	cos(θ)	–1.0 ≤ cos(θ) ≤ 1.0	—
	Calculating arctangent	Input	x and y coordinates	–float_max ≤ x ≤ float_max*1 –float_max ≤ y ≤ float_max*1	—
		Output	arctan(y/x)	–π ≤ arctan(y/x) ≤ π	radian
	Calculating $\sqrt{x^2+y^2}$	Input	x and y coordinates	–float_max ≤ x ≤ float_max*1 –float_max ≤ y ≤ float_max*1	—
		Output	$\sqrt{x^2+y^2}$	0 ≤ $\sqrt{x^2+y^2}$ ≤ ∞	—
Data Type for Processing	Single-precision floating-point				
Number of cycles for calculation	Sine: 14 Cosine: 14 Arctangent: 14 $\sqrt{x^2+y^2}$: 14 + 2 for software processing				

Note 1. float_max is the maximum value that can be expressed as single-precision floating-point: $(2 - 2^{-23}) \times 2^{127}$.

38. 12-Bit A/D Converter (S12ADH)

38.1 Overview

This MCU has one unit of a 12-bit successive approximation A/D converter. It can select analog input of up to 24 channels, temperature sensor output, or internal reference voltage.

The 12-bit A/D converter converts analog input of up to 24 selected channels, 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 arbitrarily selected analog inputs from 24 channels are converted in ascending channel order; continuous scan mode in which the arbitrarily selected analog inputs from 24 channels are continuously converted in ascending channel order; and group scan mode in which the arbitrarily selected analog inputs from 24 channels 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, 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 interrupted. 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 interrupted, 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 interrupted 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 interrupted and scan for group A is started.

The interrupted 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 or internal reference voltage can be selected in single scan mode and group scan mode. A/D conversion is performed on the temperature sensor output prior to internal reference voltage when both the temperature sensor output and internal reference voltage are selected.

The temperature sensor output or internal reference voltage cannot be selected in continuous scan mode.

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 38.1 lists the specifications of the 12-bit A/D converter and Table 38.2 lists the functions of the 12-bit A/D converter. Figure 38.1 shows block diagrams of the 12-bit A/D converter.

Table 38.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	One unit (S12AD)
Input channels	24 channels
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.9 µs per channel (when A/D conversion clock ADCLK = 60 MHz)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock ADCLK*1 can be set so that the frequency ratio should be one of the following. PCLKB to ADCLK frequency ratio = 1:1, 2:1, 4:1, 1:2 ADCLK is set using the clock generation circuit. A/D conversion clock (ADCLK) can operate between 8 MHz at a minimum and 60 MHz at a maximum.
Data registers	• 24 registers for analog input, 1 for A/D-converted data duplication in double trigger mode, and 2 for A/D-converted data duplication during extended operation in double trigger mode. • One register for temperature sensor • One register for internal reference • One register for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • 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	• 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. A/D conversion is performed only once on the internal reference voltage. • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs arbitrarily selected. • 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, and internal reference voltage 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.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the multi-function timer pulse unit (MTU), 8-bit timer (TMR), or event link controller (ELC). • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# pin.
Functions	• Variable sampling time (can be set per 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) • Automatic clear function of A/D data registers • Comparison function (windows A and B) • Order of channel conversion can be set.

Table 38.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Description
Interrupt sources	• In the modes except double trigger mode and group scan mode, a scan end interrupt request (S12ADI) can be generated on completion of single scan. (independently for three units). • In double trigger mode, a scan end interrupt request (S12ADI) can be generated on completion of double scan. • In group scan mode, a scan end interrupt request (S12ADI) can be generated on completion of group A scan, whereas a group B scan end interrupt request (S12GBADI) can be generated on completion of group B scan, and a group C scan end interrupt request (S12GCADI) 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) can be generated on completion of double scan of group A, and the corresponding scan end interrupt request (S12GBADI/S12GCADI) can be generated on completion of group B and group C scan. • A compare interrupt request (S12CMPAI or S12CMPBI) can be generated upon a match with the comparison condition for the digital compare function. • The S12ADI, S12GBADI, and S12GCADI interrupts can trigger the DMA controller (DMAC) and data transfer controller (DTC).
Event link	• The event signal is generated when all scans are finished. • The event signal is generated depending on conditions for comparison function window in single scan mode. • 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 peripheral module clock PCLKB frequency is set according to the setting of the SCKCR.PCKB[3:0] bits and the A/D conversion clock ADCLK frequency is set according to the setting of the SCKCR.PCKD[3:0] bits.
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. Refer to 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 38.2 Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation Unit 0 (S12AD)
Analog input channels			AN000 to AN023, internal reference voltage, temperature sensor output
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger	Trigger input pin	ADTRG0#
	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/input capture from MTU0.TGRA, or compare match from MTU0.TGRE	TRGA0N or TRG0N
Conditions for A/D conversion start	Synchronous trigger	Compare match between TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0
		Compare match between TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1
		ELC trigger	ELCTRG00N
			ELCTRG01N
			ELCTRG00N or ELCTRG01N
Interrupts			S12ADI, S12GBADI, S12GCADI, S12CMPAI, S12CMPBI interrupt
Setting of module stop function*1, *2			MSTPCRA.MSTPA17 bit

Note: When setting an A/D conversion start trigger to ADTRG0#, 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#. Refer to section 20, I/O Ports for details.

Note 1. Refer to 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.

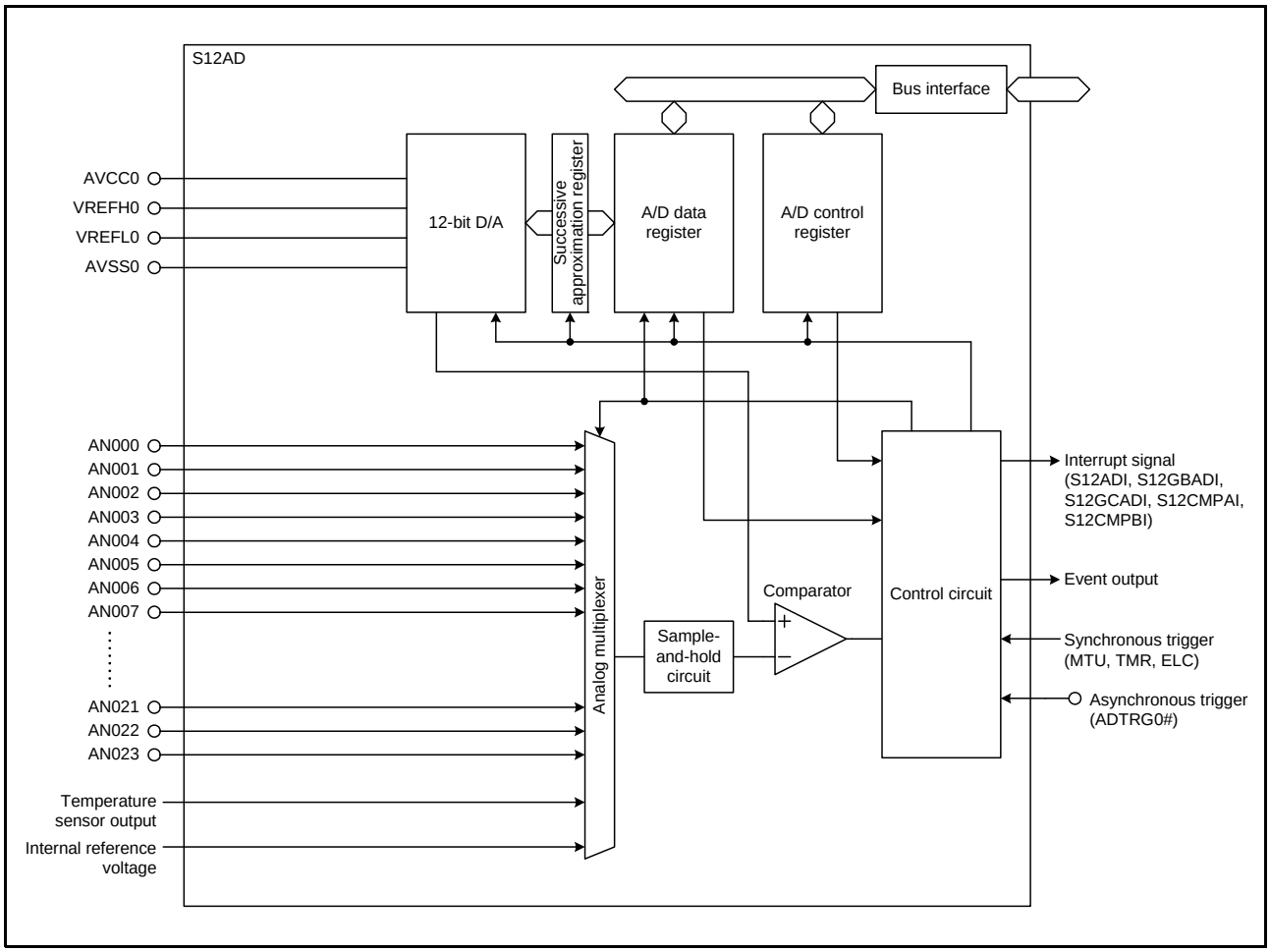


Figure 38.1 Block Diagram of 12-Bit A/D Converter

Table 38.3 lists the input/output pins of the 12-bit A/D converter.

Table 38.3 Input/Output Pins of 12-Bit A/D Converter

Unit	Pin Name	I/O	Function
Unit 0 (S12AD)	AVCC0	—	Analog power supply pin
	AVSS0	—	Analog ground pin
	VREFH0	—	Reference power supply pin
	VREFL0	—	Reference ground pin
	AN000 to AN023	Input	Analog input pins
	ADTRG0#	Input	External trigger input pin for starting A/D conversion
	ADST0	Output	Output pin for state of the ADST bit

38.2 Register Descriptions

38.2.1 A/D Data Registers y (ADDRy) (y = 0 to 23), 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.ADDR8 0008 9030h, S12AD.ADDR9 0008 9032h, S12AD.ADDR10 0008 9034h, S12AD.ADDR11 0008 9036h, S12AD.ADDR12 0008 9038h, S12AD.ADDR13 0008 903Ah, S12AD.ADDR14 0008 903Ch, S12AD.ADDR15 0008 903Eh, S12AD.ADDR16 0008 9040h, S12AD.ADDR17 0008 9042h, S12AD.ADDR18 0008 9044h, S12AD.ADDR19 0008 9046h, S12AD.ADDR20 0008 9048h, S12AD.ADDR21 0008 904Ah, S12AD.ADDR22 0008 904Ch, S12AD.ADDR23 0008 904Eh, S12AD.ADDBLDR 0008 9018h, S12AD.ADDBLDRA 0008 9084h, S12AD.ADDBLDRB 0008 9086h, S12AD.ADTSDR 0008 901Ah, S12AD.ADOCDR 0008 901Ch

	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 23) 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) (right-justified or left-justified)
- 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 Disabled

- Right-justified format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Left-justified format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Right-justified format
  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.
- Left-justified format
  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.

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

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

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.

38.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh

	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 ADRD register 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) (right-justified or left-justified)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. If temperature sensor output or internal reference voltage is selected, self-diagnosis function cannot be selected. For details of self-diagnosis, refer to section 38.2.14, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Right-justified format
  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.
- Left-justified format
  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.

Table 38.4 Self-Diagnosis Status Description

Bits 15 and 14 for Right-Justified Format Setting Bits 1 and 0 for Left-Justified 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 1/2 × AVCC has been executed.
11b	Self-diagnosis using the voltage of AVCC has been executed.

Note: For details of self-diagnosis, refer to section 38.2.14, A/D Control Extended Register (ADCER).

38.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h

	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: Group B scan end interrupt is disabled. 1: Group B scan end interrupt is enabled.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select	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: Scan end interrupt is disabled. 1: Scan end interrupt is enabled.	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

The ADCSR register 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 38.5 shows selection of the channel for double triggered operation.

When double trigger mode is selected, the channels selected by the ADANSA0 and ADANSA1 registers are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.

When double trigger mode is 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 38.5 Relationship Between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000b	AN000	10000b	AN016
00001b	AN001	10001b	AN017
00010b	AN002	10010b	AN018
00011b	AN003	10011b	AN019
00100b	AN004	10100b	AN020
00101b	AN005	10101b	AN021
00110b	AN006	10110b	AN022
00111b	AN007	10111b	AN023
01000b	AN008	11000b	—
01001b	AN009	11001b	—
01010b	AN010	11010b	—
01011b	AN011	11011b	—
01100b	AN012	11100b	—
01101b	AN013	11101b	—
01110b	AN014	11110b	—
01111b	AN015	11111b	—

—: Do not specify this value.

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 group B scan end interrupt is represented by S12GBADI.

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 and ADANSA1 registers are invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger 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, when the ADIE bit is set to 1, the interrupt request is generated upon completion of the second conversion instead of upon completion of the first 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 temperature sensor output or 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.

To use an external pin (asynchronous trigger) to start A/D conversion, set the ADCSR.TRGE and ADCSR.EXTRG bits to 1 while the external trigger pin (ADTRG0#) is being driven high. Then, if the ADTRG0# 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 PCLKB.

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 in scans except for groups B and C scan in group scan mode.
With double trigger mode deselected, the A/D scan conversion end interrupt is generated after the first scan is completed if the ADIE bit is set to 1.
With double trigger mode selected, the A/D scan conversion end interrupt is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger 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 represented by S12ADI.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits selects the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs for up to 24 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of arbitrary channel number, and the A/D conversion is stopped when one cycle of the A/D scan conversion is completed for all the selected channels.
The temperature sensor output or internal reference voltage can be selected; however, it cannot be selected with a channel input. Conversion of the temperature sensor output takes priority over that of internal reference voltage when both the temperature sensor output and internal reference voltage are selected.
In continuous scan mode, A/D conversion is performed for the analog inputs for up to 24 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of arbitrary channel number, and the A/D conversion is repeated from the first channel when one cycle of the A/D scan conversion is completed for all the selected channels.
In continuous scan mode, selecting the temperature sensor output and internal reference voltage is prohibited.
In group scan mode, A/D conversion is performed for the analog inputs (group A) for up to 24 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of arbitrary channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, and the A/D conversion is stopped when one cycle of the A/D scan conversion is completed for all the selected channels.
A/D conversion is also performed for the analog inputs (group B or C) for up to 24 channels selected with the ADANSB0 and ADANSB1 registers and the ADANSC0 and ADANSC1 registers in the ascending order of arbitrary channel number after A/D conversion is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits and the ADGCTRGR.TRSC[5:0] bits, and when A/D conversion is completed for all the selected channels, A/D conversion is stopped.
The temperature sensor output or internal reference voltage can be selected; however, it cannot be selected with a channel input in the same group. Conversion of the temperature sensor output takes priority over that of internal reference voltage when both the temperature sensor output and internal reference voltage are selected.
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).
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.

Table 38.6 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	Analog Input (Including Group A)	Analog Input (Group B and Group C)	Temperature Sensor Output	Internal Reference Voltage
Single scan	DBLE = 0	✓	✓	×	✓	✓
	DBLE = 1	×	✓ (1 channel only)	×	×	×
Continuous 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

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 the 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.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 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 the ADGSPCR.GBRP bit = 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.

38.2.4 A/D Channel Select Register A0 (ADANSA0)

(1) S12AD.ADANSA0

Address(es): S12AD.ADANSA0 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSA015	ANSA014	ANSA013	ANSA012	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	Select whether the voltages on AN000 to AN015 are to be converted or not. 0: Not selected for conversion 1: Selected for 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
b12	ANSA012			R/W
b13	ANSA013			R/W
b14	ANSA014			R/W
b15	ANSA015			R/W

The S12AD.ADANSA0 register selects analog input channels for A/D conversion from among AN000 to AN015. In group scan mode, this register selects group A channels.

ANSA0n Bit (A/D Conversion Channel Select) (n = 00 to 15)

The ANSA0n bit selects analog input channels for A/D conversion from among AN000 to AN015. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA015 bit corresponds to AN015.

When A/D conversion is performed for the temperature sensor output or internal reference voltage for group A either in single scan mode or in group scan mode, set the register to 0000h. Do not select analog input channels.

When double trigger mode is selected, the channel selected by the 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.

38.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): S12AD.ADANSA1 0008 9006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSA107	ANSA106	ANSA105	ANSA104	ANSA103	ANSA102	ANSA101	ANSA100
Value after 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	ANSA100	A/D Conversion Channel Select	Select whether the voltages on AN016 to AN023 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSA101			R/W
b2	ANSA102			R/W
b3	ANSA103			R/W
b4	ANSA104			R/W
b5	ANSA105			R/W
b6	ANSA106			R/W
b7	ANSA107			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADANSA1 register selects analog input channels for A/D conversion from among AN016 to AN023. In group scan mode, this register selects group A channels.

ANSA1n Bit (A/D Conversion Channel Select) (n = 00 to 07)

The ANSA1n bit selects analog input channels for A/D conversion from among AN016 to AN023. The channels to be selected and the number of channels can be arbitrarily set. The ANSA100 bit corresponds to AN016 and the ANSA107 bit corresponds to AN023.

When A/D conversion is performed for the temperature sensor output or internal reference voltage for group A either in single scan mode or in group scan mode, set the register to 0000h. Do not select analog input channels.

When double trigger mode is selected, the channel selected by the S12AD.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA1n bit setting is invalid.

The ANSA1n bit should be set while the S12AD.ADCSR.ADST bit is 0.

38.2.6 A/D Channel Select Register B0 (ADANSB0)

(1) S12AD.ADANSB0

Address(es): S12AD.ADANSB0 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSB015	ANSB014	ANSB013	ANSB012	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	Select whether the voltages on AN000 to AN015 are to be converted or not. 0: Not selected for conversion 1: Selected for 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
b12	ANSB012			R/W
b13	ANSB013			R/W
b14	ANSB014			R/W
b15	ANSB015			R/W

The S12AD.ADANSB0 register selects analog input channels for A/D conversion from among AN000 to AN015 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 (A/D Conversion Channel Select) (n = 00 to 15)

The ANSB0n bit selects analog input channels for A/D conversion from among AN000 to AN015 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 S12AD.ADANSA0 register), the channels specified in group C (the S12AD.ADANSC0 register), and the channels corresponding to group A, selected with 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.

When A/D conversion is performed for the temperature sensor output or internal reference voltage for group B in group scan mode, set the register to 0000h. Do not select analog input channels.

The ANSB000 bit corresponds to AN000 and the ANSB015 bit corresponds to AN015.

The ANSB0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

38.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): S12AD.ADANSB1 0008 9016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSB107	ANSB106	ANSB105	ANSB104	ANSB103	ANSB102	ANSB101	ANSB100
Value after 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	ANSB100	A/D Conversion Channel Select	Select whether the voltages on AN016 to AN023 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSB101			R/W
b2	ANSB102			R/W
b3	ANSB103			R/W
b4	ANSB104			R/W
b5	ANSB105			R/W
b6	ANSB106			R/W
b7	ANSB107			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADANSB1 register selects analog input channels for A/D conversion from among AN016 to AN023 in group B when group scan mode is selected. The S12AD.ADANSB1 register is not used in any scan mode other than group scan mode.

ANSB1n Bit (A/D Conversion Channel Select) (n = 00 to 07)

The ANSB1n bit selects analog input channels for A/D conversion from among AN016 to AN023 in group B when group scan mode is selected. The S12AD.ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD.ADANSA0 and S12AD.ADANSA1 registers), the channels specified in group C (the S12AD.ADANSC0 and S12AD.ADANSC1 registers), and the channels corresponding to group A, selected with 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.

When A/D conversion is performed for the temperature sensor output or internal reference voltage for group B in group scan mode, set the register to 0000h. Do not select analog input channels.

The ANSB100 bit corresponds to AN016 and the ANSB107 bit corresponds to AN023.

The ANSB1n bit should be set while the S12AD.ADCSR.ADST bit is 0.

38.2.8 A/D Channel Select Register C0 (ADANSC0)

(1) S12AD.ADANSC0

Address(es): S12AD.ADANSC0 0008 90D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSC015	ANSC014	ANSC013	ANSC012	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	Select whether the voltages on AN000 to AN015 are to be converted or not. 0: Not selected for conversion 1: Selected for 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
b12	ANSC012			R/W
b13	ANSC013			R/W
b14	ANSC014			R/W
b15	ANSC015			R/W

The S12AD.ADANSC0 register selects analog input channels for A/D conversion from among AN000 to AN015 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 (A/D Conversion Channel Select) (n = 00 to 15)

The ANSC0n bit selects analog input channels for A/D conversion from among AN000 to AN015 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 S12AD.ADANSA0 register), the channels specified in group B (the S12AD.ADANSB0 register), and the channels corresponding to group A, selected with 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.

When A/D conversion is performed for the temperature sensor output or internal reference voltage for group C in group scan mode, set the register to 0000h. Do not select analog input channels.

The ANSC000 bit corresponds to AN000 and the ANSC015 bit corresponds to AN015.

The ANSC0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

38.2.9 A/D Channel Select Register C1 (ADANSC1)

Address(es): S12AD.ADANSC1 0008 90D6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSC107	ANSC106	ANSC105	ANSC104	ANSC103	ANSC102	ANSC101	ANSC100
Value after 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	ANSC100	A/D Conversion Channel Select	Select whether the voltages on AN016 to AN023 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSC101			R/W
b2	ANSC102			R/W
b3	ANSC103			R/W
b4	ANSC104			R/W
b5	ANSC105			R/W
b6	ANSC106			R/W
b7	ANSC107			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADANSC1 register selects analog input channels for A/D conversion from among AN016 to AN023 in group C when group scan mode is selected. The S12AD.ADANSC1 register is not used in any scan mode other than group scan mode.

ANSC1n Bit (A/D Conversion Channel Select) (n = 00 to 07)

The ANSC1n bit selects analog input channels for A/D conversion from among AN016 to AN023 in group C when group scan mode is selected. The S12AD.ADANSC1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD.ADANSA0 and S12AD.ADANSA1 registers), the channels specified in group B (the S12AD.ADANSB0 and S12AD.ADANSB1 registers), and the channels corresponding to group A, selected with 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.

When A/D conversion is performed for the temperature sensor output or internal reference voltage for group C in group scan mode, set the register to 0000h. Do not select analog input channels.

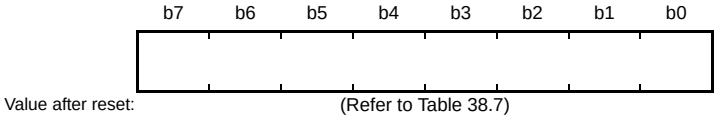
The ANSC100 bit corresponds to AN016 and the ANSC107 bit corresponds to AN023.

The ANSC1n bit should be set while the S12AD.ADCSR.ADST bit is 0.

38.2.10 A/D Channel Conversion Order Setting Register n (ADSCSn) (n = 0 to 23)

(1) S12AD.ADSCSn (n = 0 to 23)

Address(es): S12AD.ADSCS0 0008 91C0h, S12AD.ADSCS1 0008 91C1h, S12AD.ADSCS2 0008 91C2h, S12AD.ADSCS3 0008 91C3h, S12AD.ADSCS4 0008 91C4h, S12AD.ADSCS5 0008 91C5h, S12AD.ADSCS6 0008 91C6h, S12AD.ADSCS7 0008 91C7h, S12AD.ADSCS8 0008 91C8h, S12AD.ADSCS9 0008 91C9h, S12AD.ADSCS10 0008 91CAh, S12AD.ADSCS11 0008 91CBh, S12AD.ADSCS12 0008 91CCh, S12AD.ADSCS13 0008 91CDh, S12AD.ADSCS14 0008 91CEh, S12AD.ADSCS15 0008 91CFh, S12AD.ADSCS16 0008 91D0h, S12AD.ADSCS17 0008 91D1h, S12AD.ADSCS18 0008 91D2h, S12AD.ADSCS19 0008 91D3h, S12AD.ADSCS20 0008 91D4h, S12AD.ADSCS21 0008 91D5h, S12AD.ADSCS22 0008 91D6h, S12AD.ADSCS23 0008 91D7h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	Specify the order of channels for conversion by the 12-bit A/D converter. The channel with the number set in an ADSCSn register is the (n+1)th to be converted (refer to Table 38.7).	R/W

The ADSCSn register specifies the order of channels for conversion by the 12-bit A/D converter unit 0.
The order of conversion is set in the order AN000, AN001, ..., AN023 after a reset.
To change the order, change the values in the ADSCSn registers.
If a specified channel is not selected for conversion in the channel selection registers (ADANSA0, ADANSB0, and ADANSC0), A/D conversion of the corresponding channel does not proceed. Set the channels for conversion in order starting from the ADSCS0 register, and set all the channels other than those above in the rest of the registers. Do not set the same value to plural registers.

Table 38.7 Relationship Between the Value After Reset of the A/D Channel Conversion Order Setting Register n (ADSCSn) and Conversion Order

Register Symbol	Order of Conversion	Value After Reset	Channel Numbers that Can be Set:
ADSCS0	First	00h	00h to 17h (for AN000 to AN023)
ADSCS1	Second	01h	
ADSCS2	Third	02h	
ADSCS3	Fourth	03h	
ADSCS4	Fifth	04h	
ADSCS5	Sixth	05h	
ADSCS6	Seventh	06h	
ADSCS7	Eighth	07h	
ADSCS8	Ninth	08h	
ADSCS9	Tenth	09h	
ADSCS10	Eleventh	0Ah	
ADSCS11	Twelfth	0Bh	
ADSCS12	Thirteenth	0Ch	
ADSCS13	Fourteenth	0Dh	
ADSCS14	Fifteenth	0Eh	
ADSCS15	Sixteenth	0Fh	
ADSCS16	Seventeenth	10h	
ADSCS17	Eighteenth	11h	
ADSCS18	Nineteenth	12h	
ADSCS19	Twenty-th	13h	
ADSCS20	Twenty-first	14h	
ADSCS21	Twenty-second	15h	
ADSCS22	Twenty-third	16h	
ADSCS23	Twenty-fourth	17h	

38.2.11 A/D-Converted Value Addition/Average Function Channel Select Register 0 (ADADS0)

(1) S12AD.ADADS0

Address(es): S12AD.ADADS0 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADS015	ADS014	ADS013	ADS012	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	Set an A/D-converted value addition/average mode for AN000 to AN015. 0: A/D-converted value addition/average mode is disabled 1: A/D-converted value addition/average mode is enabled	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
b12	ADS012			R/W
b13	ADS013			R/W
b14	ADS014			R/W
b15	ADS015			R/W

The S12AD.ADADS0 register selects channels AN000 to AN015 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 (A/D-Converted Value Addition/Average Channel Select) (n = 00 to 15)

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 or the S12AD.ADCSR.DBLANS[4:0] bits and the S12AD.ADANSB0.ANSB0n bit and the S12AD.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 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.

An A/D conversion channel with addition/average mode disabled executes a regular one-time conversion and stores the value in the A/D data register.

The ADS0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

Figure 38.2 shows a scanning operation sequence in which both the S12AD.ADADS0.ADS002 and S12AD.ADADS0.ADS006 bits are set to 1.
It is assumed that addition mode is selected (S12AD.ADADC.AVEE bit = 0) in continuous scan mode (S12AD.ADCSR.ADCS[1:0] bits = 10b), the addition count is set to three times (S12AD.ADADC.ADC[2:0] bits = 011b), and channels AN000 to AN007 are selected (S12AD.ADANSA0 register = 00FFh). The conversion process begins with AN000. The AN002 conversion is performed successively four times (add three times), and the added (integrated) value is stored in A/D data register 2. After that the AN003 conversion is started. The AN006 conversion is performed successively four times and the added (integrated) value is stored in A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.

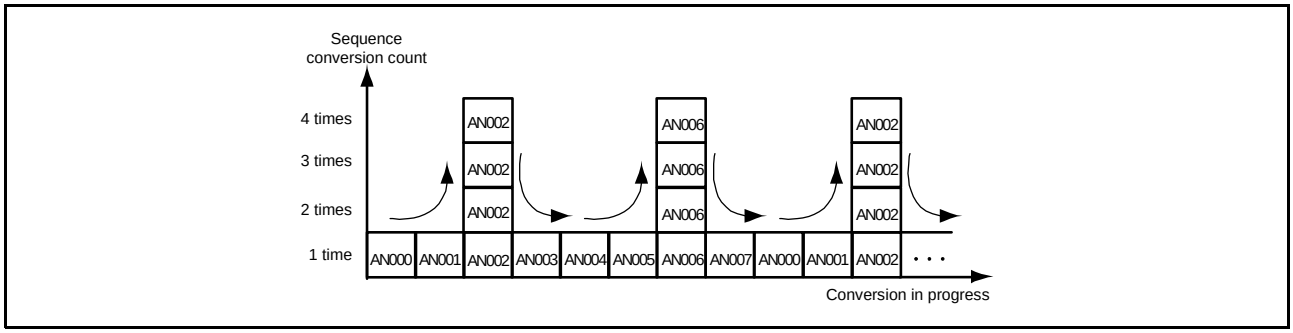


Figure 38.2 Scan Conversion Sequence with S12AD.ADADC.ADC[2:0] = 011b, S12AD.ADADC.AVEE = 0, S12AD.ADADS0.ADS002 = 1, and S12AD.ADADS0.ADS006 = 1

38.2.12 A/D-Converted Value Addition/Average Function Channel Select Register 1 (ADADS1)

Address(es): S12AD.ADADS1 0008 900Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADS107	ADS106	ADS105	ADS104	ADS103	ADS102	ADS101	ADS100
Value after 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	ADS100	A/D-Converted Value Addition/Average Channel Select	Set an A/D-converted value addition/average mode for AN016 to AN023. 0: A/D-converted value addition/average mode is disabled 1: A/D-converted value addition/average mode is enabled	R/W
b1	ADS101			R/W
b2	ADS102			R/W
b3	ADS103			R/W
b4	ADS104			R/W
b5	ADS105			R/W
b6	ADS106			R/W
b7	ADS107			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADADS1 register selects channels AN016 to AN023 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS1n Bit (A/D-Converted Value Addition/Average Channel Select) (n = 00 to 07)

When the ADS1n bit of the number that is the same as that of A/D-converted channel selected by the S12AD.ADANSA1.ANSA1n bit or the S12AD.ADCSR.DBLANS[4:0] bits, the S12AD.ADANSB1.ANSB1n bit, and the S12AD.ADANSC1.ANSC1n 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.

An A/D conversion channel with addition/average mode disabled executes a regular one-time conversion and stores the value in the A/D data register.

The ADS1n bit should be set while the S12AD.ADCSR.ADST bit is 0.

38.2.13 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch

	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] bits = 010b) nor 16-time conversion (ADADC.ADC[2:0] bits = 101b).

When the A/D-converted value addition/average modes are enabled for A/D conversion of the channels, temperature sensor output, or internal reference voltage, the number of addition and whether they are to be added or averaged is specified in the ADADC register.

ADC[2:0] Bits (Addition Count Select)

The number of addition from channels including those channels selected in double-trigger mode (by the ADCSR.DBLANS[4:0] bits), temperature sensor output, or internal reference voltage for which A/D conversion and addition or averaging is enabled, is selected by the setting of the ADC[2:0] bits.
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects whether the results of A/D conversion from channels, including those selected in double-trigger mode (by the DBLANS[4:0] bits), temperature sensor output, or internal reference voltage are to be averaged or added when A/D conversion and addition/average modes are enabled.
When average mode is selected by setting the AVEE bit to 1, do not set the addition count to one time (ADC[2:0] bits = 000b), three times (ADC[2:0] bits = 010b), or 16 times (ADC[2:0] bits = 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.

38.2.14 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	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 1/2 × AVCC for self-diagnosis. 1 1: Uses the voltage of AVCC for self-diagnosis.	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Right-justified is selected for the A/D data register format. 1: Left-justified is selected for the A/D data register format.	R/W

The ADCER register sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR register 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 DIAGLD bit.

Self-diagnosis should not be executed by setting the DIAGLD bit to 1 when the 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 (DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0 V, 1/2 × AVCC, and AVCC are converted in that 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 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 from among the internally generated voltages 0, 1/2 × AVCC, or AVCC is selected for conversion. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). The ADRD register 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.

When performing the self-diagnosis, A/D conversion of the temperature sensor output and internal reference voltage is prohibited regardless of scanning groups.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies right-justified or left-justified for the data to be stored in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADRD, ADCMPDR0, ADCMPDR1, ADWINLLB, or ADWINULB register.

The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, refer to section 38.2.1, A/D Data Registers y (ADDRy) (y = 0 to 23), 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 38.2.2, A/D Self-Diagnosis Data Register (ADRD), section 38.2.30, A/D Comparison Function Window A Lower Level Setting Register (ADCMPDR0), section 38.2.31, A/D Comparison Function Window A Upper Level Setting Register (ADCMPDR1), section 38.2.37, A/D Comparison Function Window B Lower Level Setting Register (ADWINLLB), and section 38.2.38, A/D Comparison Function Window B Upper Level Setting Register (ADWINULB).

38.2.15 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h

	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]	Group B A/D Conversion Start Trigger Select	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSTRGR register selects the A/D conversion start trigger.

TRSB[5:0] Bits (Group B A/D Conversion Start Trigger Select)

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 bit = 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 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 selecting a trigger from the MTU operated on a PCLKA cycle as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. Refer to section 38.3.7, Analog Input Sampling Time and Scan Conversion Time for details.

Table 38.8 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 performing scanning in group scan mode or double trigger mode, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.

- 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 selecting a trigger from the MTU operated on a PCLKA cycle as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. Refer to section 38.3.7, Analog Input Sampling Time and Scan Conversion

Time for details.

Table 38.9 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 38.8 Selection of A/D Trigger 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
MTU3	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
	TRGA0N or TRG0N	Compare match/input capture from MTU0.TGRA, or compare match from MTU0.TGRE	0	1	1	0	0	1
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
ELC	ELCTRG00N	A/D Startup source 0 from ELC	1	1	0	0	1	0
	ELCTRG01N	A/D Startup source 1 from ELC	1	1	0	0	1	1
	ELCTRG00N or ELCTRG01N	A/D Startup source 0 from ELC, or A/D Startup source 1 from ELC	1	1	1	0	1	0

Table 38.9 Selection of A/D Trigger 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
External pin	ADTRGn#	Trigger input pin	0	0	0	0	0	0
MTU3	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
	TRGA0N or TRG0N	Compare match/input capture from MTU0.TGRA, or compare match from MTU0.TGRE	0	1	1	0	0	1
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
ELC	ELCTRG00N	A/D Startup source 0 from ELC	1	1	0	0	1	0
	ELCTRG01N	A/D Startup source 1 from ELC	1	1	0	0	1	1
	ELCTRG00N or ELCTRG01N	A/D Startup source 0 from ELC, or A/D Startup source 1 from ELC	1	1	1	0	1	0

38.2.16 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD.ADEXICR 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	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 disabled. 1: Temperature sensor output A/D-converted value addition/average mode is enabled.	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 disabled. 1: Internal reference voltage A/D-converted value addition/average mode is enabled.	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: Temperature sensor output is not A/D-converted 1: Temperature sensor output is A/D-converted	R/W
b9	OCSA	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
b10	TSSB	Group B 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
b11	OCSB	Group B 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
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADEXICR register specifies the settings of A/D conversion of the temperature sensor output, internal reference voltage.

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[2:0] 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 the ADTSDR register 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 A/D conversion of the internal reference voltage is selected and the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is performed in sequence for the count set in the ADADC.ADC[2:0] bits (2, 3, 4, or 16 times). 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 the ADOCDR register. The OCSAD bit should be set while the ADCSR.ADST bit is 0.

TSSA Bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output for group A in single 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. When setting the TSSB or ADGCEXCR.TSSC bit to 1, set this bit to 0.

The temperature sensor output can be selected in single scan mode and group scan mode. Set the sampling time to at least 4 µs when A/D-converting the temperature sensor output. When setting the TSSA bit to 1, the temperature sensor is automatically activated. After activation, only start A/D conversion after waiting 200 µs for the sensor output to become stable. When A/D conversion is started, auto-discharging for 15 cycles of ADCLK proceeds before sampling.
A/D conversion of the temperature sensor outputs in groups B and C should be handled in the same way.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

The OCSA bit selects A/D conversion of the internal reference voltage for group A in single 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. When setting the OCSB or ADGCEXCR.OCSC bit to 1, set this bit to 0.
To perform A/D conversion of the internal reference voltage, set the OCSA bit to 1, wait for at least 400 ns, and then start A/D conversion. When A/D conversion is started, auto-discharging for 15 cycles of ADCLK proceeds before sampling.
When the disconnection detection assist function is also in use, the setting of the ADNDIS[4:0] bits is ignored.
A/D conversion of the internal reference voltages in groups B and C should be handled in the same way.

TSSB Bit (Group B Temperature Sensor Output A/D Conversion Select)

The TSSB 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 setting the TSSA or ADGCEXCR.TSSC bit to 1, set this bit to 0.
Refer to description of the TSSA bit for A/D-converting the temperature sensor output.

OCSB Bit (Group B Internal Reference Voltage A/D Conversion Select)

The OCSB 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 setting the OCSA or ADGCEXCR.OCSC bit to 1, set this bit to 0.
Refer to description of the OCSA bit for A/D-converting the internal reference voltage.

38.2.17 A/D Group C Extended Input Control Register (ADGCEXCR)

Address(es): S12AD.ADGCEXCR 0008 90D8h

	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)

The TSSC 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. When setting the ADEXICR.TSSA or ADEXICR.TSSB bit to 1, set this bit to 0.

Refer to description of the ADEXICR.TSSA bit for A/D-converting the temperature sensor output.

OCSC Bit (Group C Internal Reference Voltage A/D Conversion Select)

The OCSC 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. When setting the ADEXICR.OCSA or ADEXICR.OCSB bit to 1, set this bit to 0.

Refer to description of the ADEXICR.OCSA bit for A/D-converting the internal reference voltage.

38.2.18 A/D Group C Trigger Select Register (ADGCTRGR)

Address(es): S12AD.ADGCTRGR 0008 90D9h

	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: Group C scan end interrupt is disabled. 1: Group C scan end interrupt is enabled.	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

The ADGCTRGR register enables operation for group C and selects the A/D conversion start trigger. For details on group priority operation, refer to Table 38.13 and Table 38.14.

TRSC[5:0] Bits (Group C A/D Conversion Start Trigger Select)

The TRSC[5:0] 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 t_{SCAN}, A/D conversion by a trigger may have no effect.

When selecting a trigger from the MTU operated on a PCLKA cycle as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. Refer to section 38.3.7, Analog Input Sampling Time and Scan Conversion Time for details.

Table 38.10 lists the selection of A/D conversion start sources selected by the TRSC[5:0] bits for group C.

Table 38.10 Selection of A/D Trigger Sources by the TRSC[5:0] Bits (Group C)

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
MTU3	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
	TRGA0N or TRG0N	Compare match/input capture from MTU0.TGRA, or compare match from MTU0.TGRE	0	1	1	0	0	1
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
ELC	ELCTRG00N	A/D Startup source 0 from ELC	1	1	0	0	1	0
	ELCTRG01N	A/D Startup source 1 from ELC	1	1	0	0	1	1
	ELCTRG00N or ELCTRG01N	A/D Startup source 0 from ELC, or A/D Startup source 1 from ELC	1	1	1	0	1	0

GCADIE Bit (Group C Scan End Interrupt Enable)

The GCADIE bit enables or disables scan end interrupt generation for group C. A scan end interrupt for group C is represented by S12GCADI.

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.

38.2.19 A/D Sampling State Register n (ADSSTRn) (n = 0 to 15, L, 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, S12AD.ADSSTR8 0008 90E8h, S12AD.ADSSTR9 0008 90E9h, S12AD.ADSSTR10 0008 90EAh, S12AD.ADSSTR11 0008 90EBh, S12AD.ADSSTR12 0008 90ECh, S12AD.ADSSTR13 0008 90EDh, S12AD.ADSSTR14 0008 90EEh, S12AD.ADSSTR15 0008 90EFh, S12AD.ADSSTRL 0008 90DDh, S12AD.ADSSTRT 0008 90DEh, S12AD.ADSSTRO 0008 90DFh

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

The ADSSTRn register sets the sampling time for analog input.
Set the sampling time based on the number of clock cycles for ADCLK (A/D conversion clock). The initial value is 27 clock cycles. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0.
Specify the value for the register as a multiples of 3 in the range from 12 to 252 (clock cycles).
A sampling time can be determined by the following formula:

$$\text{Sampling time} = \text{ADSSTR} \times t_C(\text{ADCLK})$$

Table 38.11 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 38.3.7, Analog Input Sampling Time and Scan Conversion Time.

Table 38.11 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
	ADSSTR8 register	AN008
	ADSSTR9 register	AN009
	ADSSTR10 register	AN010
	ADSSTR11 register	AN011
	ADSSTR12 register	AN012
	ADSSTR13 register	AN013
	ADSSTR14 register	AN014
	ADSSTR15 register	AN015
	ADSSTRL register	AN016 to AN023
	ADSSTRT register	Temperature sensor output*1
	ADSSTRO register	Internal reference voltage*2

Note 1. When performing A/D conversion of the temperature sensor output, it is necessary to set the sampling time that meets the specification described in section 45., Electrical Characteristics.

Note 2. When performing A/D conversion of the internal reference voltage, the sampling time should be 4 µs or longer.

38.2.20 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADNDIS[3:0]	A/D Disconnection Detection Assist Setting	Specify the period for discharging or precharging as a number of ADCLK cycles. b3 b0 0 0 0 0: No charging (disconnection detection assist function is disabled.) 0 0 1 1: Charging period of 3 clock cycles 0 1 1 0: Charging period of 6 clock cycles 1 0 0 1: Charging period of 9 clock cycles 1 1 0 0: Charging period of 12 clock cycles 1 1 1 1: Charging period of 15 clock cycles Settings other than above are prohibited.	R/W
b4	ADNDIS[4]		0: Discharge 1: Precharge	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register sets the disconnection detection assist function.

ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)

The ADNDIS[4:0] 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 ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is disabled. Set the ADNDIS[3:0] bits to 0000b or a multiple of 3. A value other than those above cannot be set. The value set in the ADNDIS[3:0] bits is used to set the number of clock cycles that determines the precharging or discharging period.

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. Set the ADNDIS[3:0] bits to 0000b.

When A/D conversion of a temperature sensor output or internal reference voltage proceeds, auto-discharging is also executed beforehand. Therefore, the ADNDIS[4:0] bits are automatically fixed to 0Fh during conversion for a scan group that includes a temperature sensor output or internal reference voltage, and the corresponding analog input paths in the A/D converter are discharged. Sampling starts after discharging is completed.

After the A/D conversion of a temperature sensor output or internal reference voltage is finished, the ADNDIS[4:0] bits automatically return to the setting before the conversion.

38.2.21 A/D Event Link Control Register (ADELCCR)

Address(es): S12AD.ADELCCR 0008 907Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ELCC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ELCC[2:0]	Event Link Control	b2 b0 0 0 0: An event signal is generated when scanning for group A is completed. 0 0 1: An event signal is generated when scanning for group B is completed. 0 1 0: An event signal is generated when scanning for group A, group B, or group C is completed. 1 0 0: An event signal is generated when scanning for group C is completed. Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADELCR register is used to set an event signal generation condition at completion of scanning.

ELCC[2:0] Bits (Event Link Control)

The ELCC[2:0] bits are used to select an event signal generation condition at completion of scanning.

38.2.22 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h

	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	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for the group is not restarted after having been interrupted due to group priority control. 1: Scanning for the group is restarted after having been interrupted 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 *2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan is not continuously started. 1: Single scan for the lowest-priority group is continuously started.	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 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.

The ADGSPCR register is used to interrupted 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, refer to Table 38.13 and Table 38.14.

PGS Bit (Group Priority Control Setting)

The PGS 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 interrupted. 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 interrupted, 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 interrupted, 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 interrupted and scan for group A is started.
When setting the PGS bit to 0, clearing should be performed by software according to section 38.6.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 38.3.5.3, Operation under Group Priority Control.

GBRSCN Bit (Low-Priority Group Restart Setting)

The GBRSCN 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)

The LGRRS 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)

The GBRP 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.

38.2.23 A/D Comparison Function Control Register (ADCMPCR)

Address(es): S12AD.ADCMPCR 0008 9090h

	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 XOR 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 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 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 flag. The CMPAB[1:0] bits should be set when the ADCSR.ADST bit is 0.

CMPBE Bit (Comparison Window B Enable)

The CMPBE 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/A1/B0/B1/C0/C1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1, ADANSC0, and ADANSC1)
- 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)

The CMPAE 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/A1/B0/B1/C0/C1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1, ADANSC0, and ADANSC1)
- 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 and 1 (ADCMPANSR0 and ADCMPANSR1)
- Window A extended input select register (ADCMPANSER)

CMPBIE Bit (Comparison B Interrupt Enable)

Enables or disables comparison interrupts by a window B comparison condition match. A single comparison interrupt is represented by S12CMPBI.

WCMPE Bit (Window Function Setting)

The WCMPE 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 A Interrupt Enable)

Enables or disables comparison interrupts by a window A comparison condition match. A single comparison interrupt is represented by S12CMPAI.

38.2.24 A/D Comparison Function Window A Channel Select Register 0 (ADCMPANSR0)

(1) S12AD.ADCMPANSR0

Address(es): S12AD.ADCMPANSR0 0008 9094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPCHA015	CMPCHA014	CMPCHA013	CMPCHA012	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	These bits determine whether to include pins AN000 to AN015 as the targets for comparison window A. 0: Exclude from the targets for comparison window A. 1: Include 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
b12	CMPCHA012			R/W
b13	CMPCHA013			R/W
b14	CMPCHA014			R/W
b15	CMPCHA015			R/W

The S12AD.ADCMPANSR0 register selects analog inputs AN000 to AN015 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA0n Bit (Comparison Window A Channel Select) (n = 00 to 15)

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, the ADANSB0.ANSB0n bit, and the ADANSC0.ANSC0n bit is set to 1.

The CMPCHA0n bit should be set while the ADCSR.ADST bit is 0.

38.2.25 A/D Comparison Function Window A Channel Select Register 1 (ADCMPANSR1)

Address(es): S12AD.ADCMPANSR1 0008 9096h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPCHA107	CMPCHA106	CMPCHA105	CMPCHA104	CMPCHA103	CMPCHA102	CMPCHA101	CMPCHA100
Value after 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	CMPCHA100	Comparison Window A Channel Select	These bits determine whether to include pins AN016 to AN023 as the targets for comparison window A. 0: Exclude from the targets for comparison window A. 1: Include as the targets for comparison window A.	R/W
b1	CMPCHA101			R/W
b2	CMPCHA102			R/W
b3	CMPCHA103			R/W
b4	CMPCHA104			R/W
b5	CMPCHA105			R/W
b6	CMPCHA106			R/W
b7	CMPCHA107			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSR1 register selects analog inputs AN016 to AN023 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA1n Bit (Comparison Window A Channel Select) (n = 00 to 07)

The comparison function is enabled when the CMPCHA1n bit with the same index number as the A/D conversion channel selected by the ADANSA1.ANSA1n bit, the ADANSB1.ANSB1n bit, and the ADANSC1.ANSC1n bit is set to 1.

The CMPCHA1n bit should be set while the ADCSR.ADST bit is 0.

38.2.26 A/D Comparison Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): S12AD.ADCMPANSER 0008 9092h

	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

The ADCMPANSER 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 the 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 is enabled when the CMPSOC bit is set to 1 while the ADEXICR.OCSA, ADEXICR.OCSB, or ADGCEXCR.OCSC bit is 1. The CMPSOC bit should be set while the ADCSR.ADST bit is 0.

38.2.27 A/D Comparison Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

(1) S12AD.ADCMPLR0

Address(es): S12AD.ADCMPLR0 0008 9098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLCHA015	CMPLCHA014	CMPLCHA013	CMPLCHA012	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
b12	CMPLCHA012			R/W
b13	CMPLCHA013			R/W
b14	CMPLCHA014			R/W
b15	CMPLCHA015			R/W

The S12AD.ADCMPLR0 register specifies the conditions for comparison of the values in the S12AD.ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The S12AD.ADCMPLR0 register should be set while the S12AD.ADCSR.ADST bit is 0.

CMPLCHA0n Bit (Comparison Window A Comparison Condition Select) (n = 00 to 15)

The CMPLCHA0n bit specifies the comparison conditions of the channels (AN000 to AN015) 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 CMPLCHA015 bit corresponds to AN015.

When the result of comparison of each analog input matches with the pre-set condition, the ADCMPSR0.CMPSTCHA0n flag (n = 00 to 15) becomes 1 and a comparison interrupt (S12CMPAI) is generated. The conditions for comparison are shown in Figure 38.3.

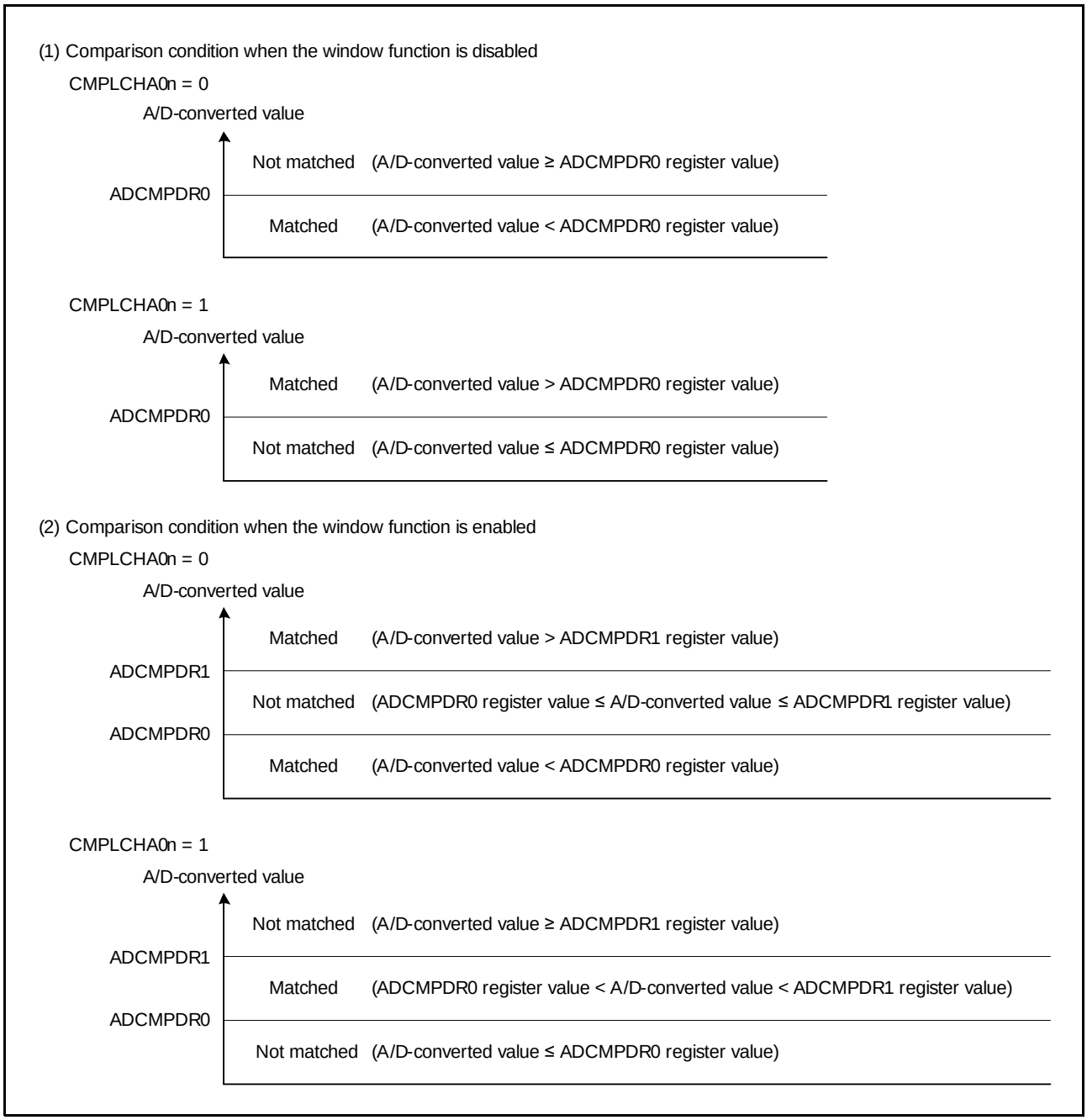


Figure 38.3 Details of Comparison Condition: Comparison Function Window A

38.2.28 A/D Comparison Function Window A Comparison Condition Setting Register 1 (ADCMPLR1)

Address(es): S12AD.ADCMPLR1 0008 909Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPLCHA107	CMPLCHA106	CMPLCHA105	CMPLCHA104	CMPLCHA103	CMPLCHA102	CMPLCHA101	CMPLCHA100
Value after 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	CMPLCHA100	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	CMPLCHA101			R/W
b2	CMPLCHA102			R/W
b3	CMPLCHA103			R/W
b4	CMPLCHA104			R/W
b5	CMPLCHA105			R/W
b6	CMPLCHA106			R/W
b7	CMPLCHA107			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLR1 register specifies the conditions for comparison of the values in the ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The ADCMPLR1 register should be set while the ADCSR.ADST bit is 0.

CMPLCHA1n Bit (Comparison Window A Comparison Condition Select) (n = 00 to 07)

The CMPLCHA1n bit specifies the comparison conditions of the channels (AN016 to AN023) targeted for the window A comparison conditions. The condition can be specified for each analog input for comparison. The CMPLCHA100 bit corresponds to AN016 and the CMPLCHA107 bit corresponds to AN023.

When the result of comparison of each analog input matches with the pre-set condition, the S12AD.ADCMPSR1.CMPSTCHA1n flag becomes 1 and a comparison interrupt (S12CMPAI) is generated. The conditions for comparison are shown in Figure 38.3.

38.2.29 A/D Comparison Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): S12AD.ADCMPLER 0008 9093h

	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 ADCMPDR1 register value < A/D-converted 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 ADCMPDR1 register value < A/D-converted 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

The ADCMPLER register specifies the conditions for comparison of the values in the ADCMPDR0/ADCMPDR1 register and the value 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)

The CMPLTS 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 becomes 1. A comparison interrupt (S12CMPAI) is generated. Figure 38.3 shows the conditions for comparison.

CMPLOC Bit (Comparison Window A Internal Reference Voltage Comparison Condition Select)

The CMPLOC 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 becomes 1. A comparison interrupt (S12CMPAI) is generated. Figure 38.3 shows the conditions for comparison.

38.2.30 A/D Comparison Function Window A Lower Level Setting Register (ADCMPDR0)

Address(es): S12AD.ADCMPDR0 0008 909Ch

	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 ADCMPDR0 register 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 (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/average count select register (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/Average Mode is Disabled

- Right-justified format
  The comparison level (lower) to be compared is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (lower) to be compared is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

- Right-justified format
  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.
- Left-justified format
  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.

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

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  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.

- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  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.

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.

38.2.31 A/D Comparison Function Window A Upper Level Setting Register (ADCMPDR1)

Address(es): S12AD.ADCMPDR1 0008 909Eh

	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 ADCMPDR1 register 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 (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/averaging time select register (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/Average Mode is Disabled

- Right-justified format
  The comparison level (upper) is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (upper) is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

- Right-justified format
  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.
- Left-justified format
  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.

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

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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 and 14.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)

The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 15 to 0.

- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set 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.

38.2.32 A/D Comparison Function Window A Channel Status Register 0 (ADCMPSR0)

(1) S12AD.ADCMPSR0

Address(es): S12AD.ADCMPSR0 0008 90A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPSTCHA015	CMPSTCHA014	CMPSTCHA013	CMPSTCHA012	CMPSTCHA011	CMPSTCHA010	CMPSTCHA009	CMPSTCHA008	CMPSTCHA007	CMPSTCHA006	CMPSTCHA005	CMPSTCHA004	CMPSTCHA003	CMPSTCHA002	CMPSTCHA001	CMPSTCHA000
Value after 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 AN015) for the window A comparison condition 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPSTCHA001			R/(W) *1
b2	CMPSTCHA002			R/(W) *1
b3	CMPSTCHA003			R/(W) *1
b4	CMPSTCHA004			R/(W) *1
b5	CMPSTCHA005			R/(W) *1
b6	CMPSTCHA006			R/(W) *1
b7	CMPSTCHA007			R/(W) *1
b8	CMPSTCHA008			R/(W) *1
b9	CMPSTCHA009			R/(W) *1
b10	CMPSTCHA010			R/(W) *1
b11	CMPSTCHA011			R/(W) *1
b12	CMPSTCHA012			R/(W) *1
b13	CMPSTCHA013			R/(W) *1
b14	CMPSTCHA014			R/(W) *1
b15	CMPSTCHA015			R/(W) *1

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

The S12AD.ADCMPSR0 register is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA0n Flag (Comparison Window A Flag) (n = 00 to 15)

The CMPSTCHA0n flag is a status flag that indicate the result of comparison of the channels (AN000 to AN015) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR0.CMPLCHA0n bit on completion of A/D conversion, this flag becomes 1.

When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI) is generated at the setting of the flag. The CMPSTCHA000 flag corresponds to AN000 and the CMPSTCHA015 flag corresponds to AN015.

1 cannot be written to the CMPSTCHA0n flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

38.2.33 A/D Comparison Function Window A Channel Status Register 1 (ADCMPSR1)

Address(es): S12AD.ADCMPSR1 0008 90A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPST CHA107	CMPST CHA106	CMPST CHA105	CMPST CHA104	CMPST CHA103	CMPST CHA102	CMPST CHA101	CMPST CHA100
Value after 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	CMPSTCHA100	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN016 to AN023) for the window A comparison condition 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPSTCHA101			R/(W) *1
b2	CMPSTCHA102			R/(W) *1
b3	CMPSTCHA103			R/(W) *1
b4	CMPSTCHA104			R/(W) *1
b5	CMPSTCHA105			R/(W) *1
b6	CMPSTCHA106			R/(W) *1
b7	CMPSTCHA107			R/(W) *1
b15 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 clear the flag after reading 1.

The ADCMPSR1 register is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA1n Flag (Comparison Window A Flag) (n = 00 to 07)

The CMPSTCHA1n flag is a status flag that indicate the result of comparison of the channels (AN016 to AN023) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR1.CMPLCHA1n bit on completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI) is generated upon the setting of the flag. The CMPSTCHA100 flag corresponds to AN016 and the CMPSTCHA107 flag corresponds to AN023.

1 cannot be written to the CMPSTCHA1n flag.

[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLR1.CMPLCHA1n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

38.2.34 A/D Comparison Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): S12AD.ADCMPSER 0008 90A4h

	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) *1
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) *1
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 clear the flag after reading 1.

The ADCMPSER register is a register that stores the result of comparison by the comparison window A function.

CMPFTS Flag (Comparison Window A Temperature Sensor Output Comparison Flag)

The CMPFTS flag 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 becomes 1. When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI) is generated upon the setting of the flag.
1 cannot be written to the CMPFTS flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and 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)

The CMPFOC flag 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 becomes 1. When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI) is generated upon the setting of the flag.
1 cannot be written to the CMPFOC flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLER.CMPLOC bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

38.2.35 A/D Comparison Function Window A/B Status Monitoring Register (ADWINMON)

Address(es): S12AD.ADWINMON 0008 908Ch

	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 Flag	This flag indicates the combination result. This flag is valid when both window A operation and window B operation are enabled. 0: The complex condition for window A/B is not satisfied. 1: The complex condition for window A/B is satisfied.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	MONCMPA	Comparison Result Monitor A Flag	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 Flag	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

The ADWINMON register can monitor the results of comparison and the combined result.

MONCOMB Flag (Combination Result Monitoring Flag)

The MONCOMB bit 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 bit = 1 and ADCMPCR.CMPBE bit = 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 bit = 0 or ADCMPCR.CMPBE bit = 0

MONCMPA Flag (Comparison Result Monitor A Flag)

The MONCMPA bit is a dedicated bit for reading 1 when the condition set in the ADCMPLR0, ADCMPLR1, or ADCMPLER register 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 bit = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When ADCMPCR.CMPAE bit = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is not satisfied.
- When ADCMPCR.CMPAE bit = 0 (the ADCMPCR.CMPAE bit is automatically cleared to 0).

MONCMPB Flag (Comparison Result Monitor B Flag)

The MONCMPB bit is a dedicated bit for reading 1 when the condition set in the ADCMPBNSR.CMPLB bit 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 bit = 1, the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When ADCMPCR.CMPBE bit = 1, the condition set in the ADCMPBNSR.CMPLB bit is not satisfied.
- When ADCMPCR.CMPBE bit = 0 (the ADCMPCR.CMPBE bit is automatically cleared to 0).

38.2.36 A/D Comparison Function Window B Channel Select Register (ADCMPBNSR)

(1) S12AD.ADCMPBNSR

Address(es): S12AD.ADCMPBNSR 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 1 0 1 1 0: AN022 0 1 0 1 1 1: AN023 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 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

The S12AD.ADCMPBNSR register is used to specify the comparison window B function.

CMPCHB[5:0] Bits (Comparison Window B Channel Select)

The CMPCHB[5:0] bits select the channel to be compared with the comparison window B from AN000 to AN023, temperature sensor, and internal reference voltage.

When the number of the A/D conversion channel selected in the ADANSAy.ANSAyn bit (y = 0 to 1, n = 00 to 15) and the 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)

The CMPLB bit specifies the condition for comparison of the channels for window B. When the result of comparison of

each analog input matches with the pre-set condition, the ADCMPBSR.CMPSTB flag becomes 1 and a comparison interrupt (S12CMPBI) is generated. The conditions for comparison are shown in Figure 38.4.

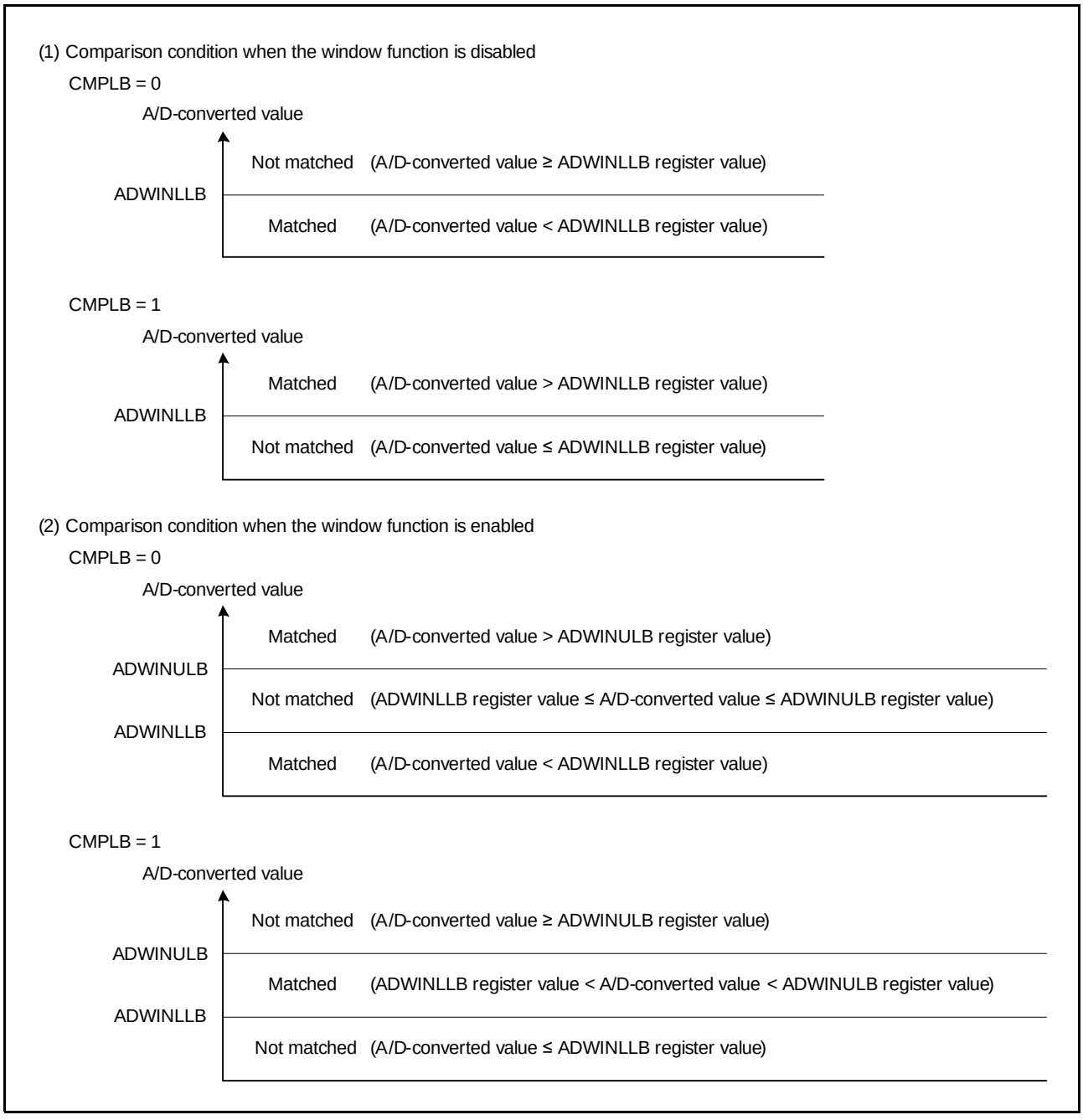


Figure 38.4 Details of Comparison Condition: Comparison Function Window B

38.2.37 A/D Comparison Function Window B Lower Level Setting Register (ADWINLLB)

Address(es): S12AD.ADWINLLB 0008 90A8h

	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 ADWINLLB register 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 ADWINLLB 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 register setting value ≥ ADWINLLB register 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 (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/averaging count select register (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/Average Mode is Disabled

- Right-justified format
  The comparison level (lower) to be compared is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (lower) to be compared is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

- Right-justified format
  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.
- Left-justified format
  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.

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

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  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.

- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  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.

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.

38.2.38 A/D Comparison Function Window B Upper Level Setting Register (ADWINULB)

Address(es): S12AD.ADWINULB 0008 90AAh

	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 ADWINULB register 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 register setting value ≥ ADWINLLB register setting value)] when setting.
The 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 (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/averaging count select register (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/Average Mode is Disabled

- Right-justified format
  The comparison level (upper) is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (upper) is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

- Right-justified format
  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.
- Left-justified format
  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.

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

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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 and 14.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)

The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 15 to 0.

- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  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.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set 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 ADWINULB 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 ADWINULB 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.

38.2.39 A/D Comparison Function Window B Channel Status Register (ADCMPBSR)

(1) S12AD.ADCMPBSR

Address(es): S12AD.ADCMPBSR 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) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD.ADCMPBSR register is used to store results of comparison of the comparison window B function.

CMPSTB Flag (Comparison Window B Flag)

The CMPSTB flag is a status flag that indicates the result of comparison of the channels for window B comparison conditions (AN000 to AN023, temperature sensor, internal reference voltage).

When the conversion result matches with the comparison condition that is set in the ADCMPBNSR.CMPLB bit upon completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPBIE 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 bit = 1 and the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

38.2.40 A/D Internal Reference Voltage Monitoring Circuit Enable Register (ADVMONCR)

Address(es): S12AD.ADVMONCR 0008 91E2h

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

Bit	Symbol	Bit Name	Description	R/W
b0	VDE	Voltage Monitoring Circuit Enable	0: Disables internal reference voltage monitoring circuit operation. 1: Enables internal reference voltage monitoring circuit operation.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADVMONCR register is used to enable the voltage monitoring circuit operation for the internal reference voltage. Set the register while the ADCSR.ADST bit is 0.

38.2.41 A/D Internal Reference Voltage Monitoring Circuit Output Enable Register (ADVMONO)

Address(es): S12AD.ADVMONO 0008 91E4h

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

Bit	Symbol	Bit Name	Description	R/W
b0	VDO	Voltage Monitoring Circuit Voltage Output Enable	0: Disables voltage output of internal reference voltage monitoring circuit. 1: Enables voltage output of internal reference voltage monitoring circuit.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADVMONO register is used to enable the voltage output of the voltage monitoring circuit for the internal reference voltage.
Set the register while the ADCSR.ADST bit is 0.

38.2.42 A/D Reference Voltage Control Register (ADVREFCR)

Address(es): S12AD.ADVREFCR 0008 8084h

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

Bit	Symbol	Bit Name	Description	R/W
b0	VREFSEL	Reference Voltage Select	0: AVCC0/AVSS0 are selected as the reference voltage. 1: VREFH0/VREFL0 are selected as the reference voltage.*1	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set the PJ6 and PJ7 pins to general input ports (PORTJ.PMR.B6 = 0, PORTJ.PMR.B7 = 0, PORTJ.PDR.B6 = 0, PORTJ.PDR.B7 = 0).

The ADVREFCR register specifies the reference voltages.
Set this register while the ADCSR.ADST bit is 0.

VREFSEL Bits (Reference Voltage Select)

This bit is used to select the reference voltage. Either AVCC0/AVSS0 or VREFH0/VREFL0 is selectable.
The VREFH0 and VREFL0 pins share pins with general I/O ports PJ6 and PJ7, respectively.
Select AVCC0/AVSS0 to secure more general I/O ports, and VREFH0/VREFL0 to use the A/D converter with the accuracy described in the electrical characteristics.

38.3 Operation

38.3.1 A/D Converter Initialization Flowchart

Initialize the A/D converter following the example of flowchart shown in Figure 38.5.

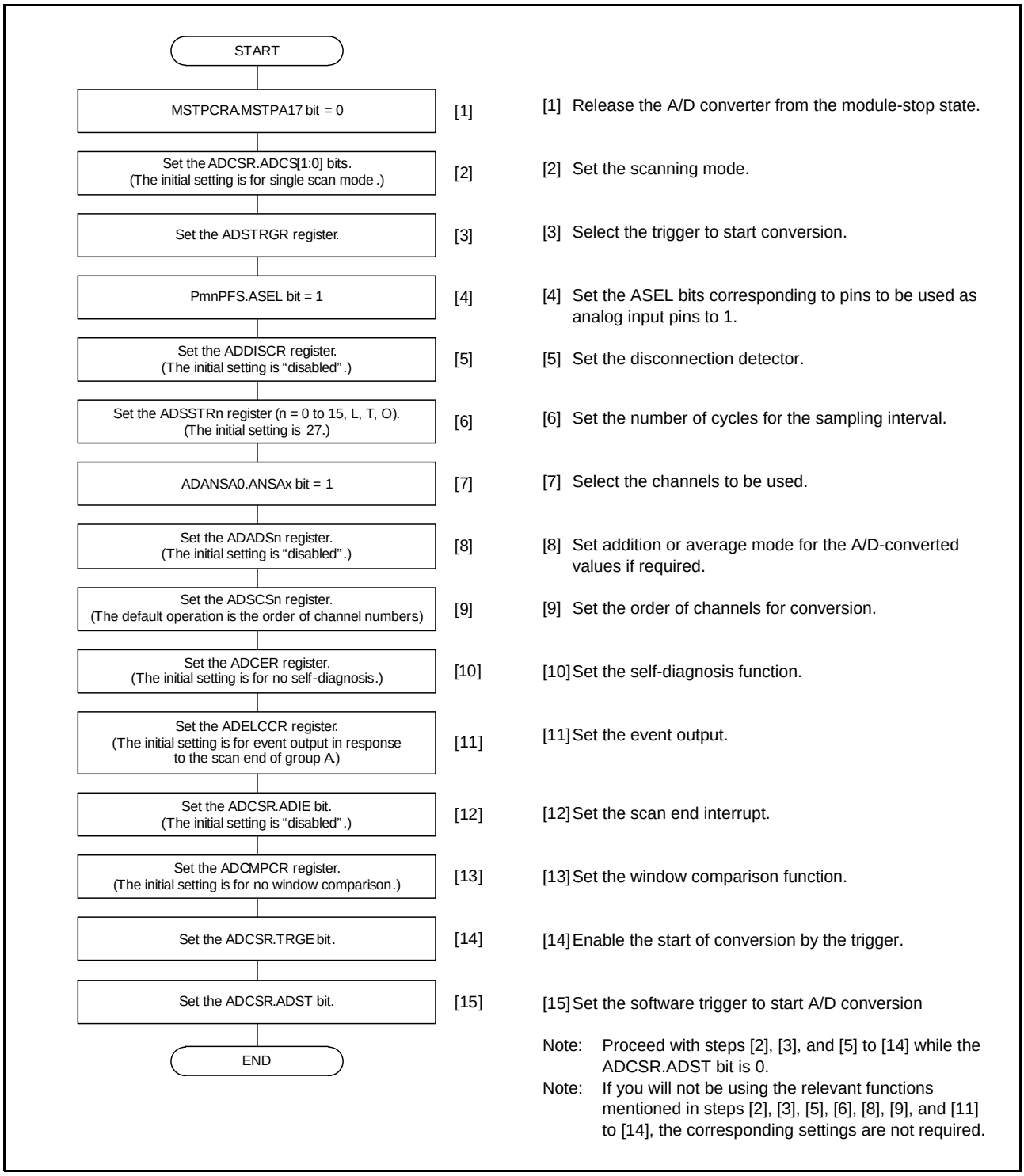


Figure 38.5 A/D Converter Initialization Flowchart

38.3.2 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 the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
In group scan mode, A/D conversion is performed for analog channels of groups A, B, and C selected by the ADANSA0 and ADANSA1 registers, the ADANSB0 and ADANSB1 registers, and the ADANSC0 and ADANSC1 registers, respectively, according to the order of conversion specified in the ADSCSn register.
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 the internal reference voltage can be selected; however, they cannot be selected in combination with an analog input of the channels in the same group.
Conversion of the temperature sensor output takes priority when the temperature sensor output and internal reference voltage are selected simultaneously.
When converting the temperature sensor output or the internal reference voltage, an auto-discharging is automatically performed.
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)
- TRGA0N or TRG0N (the ADSTRGR.TRSA[5:0] bits are set to 011001b)
- ELCTRG00N or ELCTRG01N, ELCTRG10N or ELCTRG11N, or ELCTRG20N or ELCTRG21N (the ADSTRGR.TRSA[5:0] bits are set to 111010b)

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.

38.3.3 Single Scan Mode

38.3.3.1 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 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register.
(Specified conversion order in Figure 38.6: AN004 → AN005 → AN006)
(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 (scan end interrupt is enabled).
(4) The ADCSR.ADST bit remains 1 (starting A/D conversion) 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.

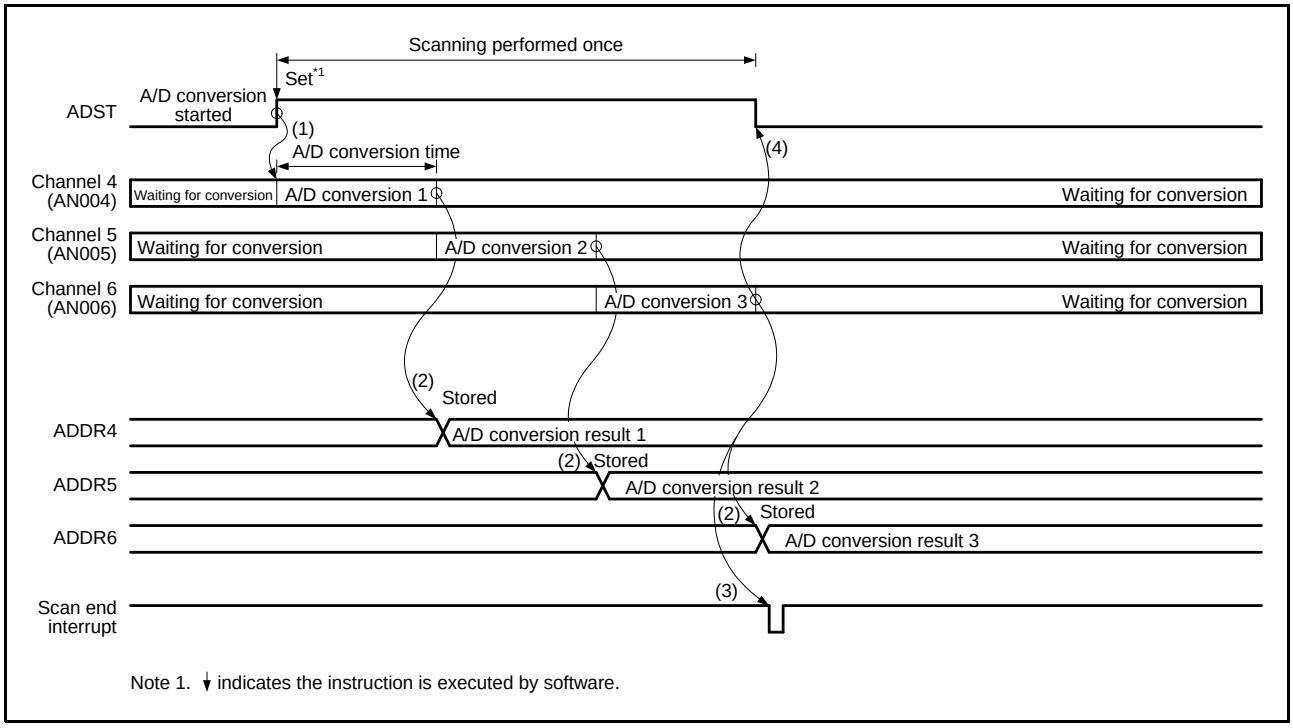
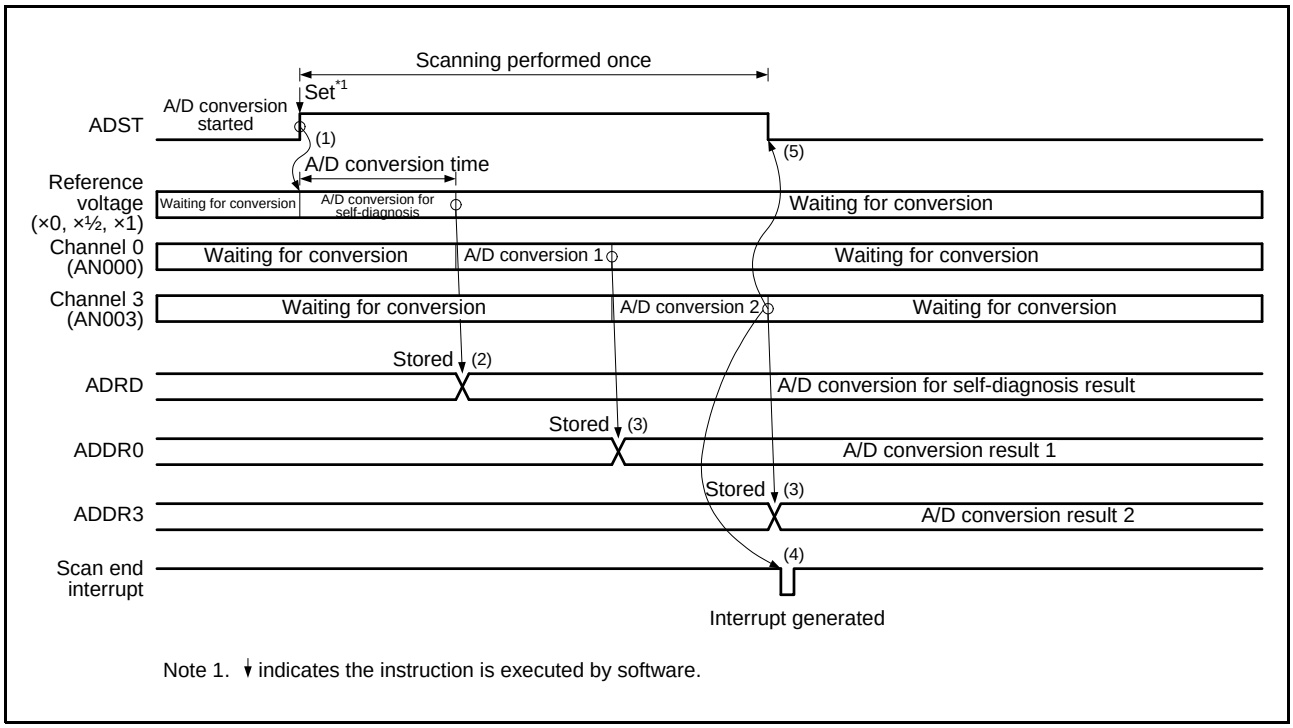


Figure 38.6 Example of Operation in Single Scan Mode (Basic Operation: AN004, AN005, AN006 Selected)

38.3.3.2 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 (starting A/D conversion) 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), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register.
(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 (scan end interrupt is enabled).
(5) The ADST bit remains 1 (starting A/D conversion) 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 38.7 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN003 Selected + Self-Diagnosis)

38.3.3.3 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 is performed only once for the temperature sensor output or for internal reference voltage as described below.
When the temperature sensor output and internal reference voltage are both selected, A/D conversion is performed for the temperature sensor output first and then for internal reference voltage.
Selecting the temperature sensor output and internal reference voltage simultaneously for analog input is prohibited.

(1) When the ADCSR.ADST bit becomes 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, auto-discharging and A/D conversion is performed in this order for the temperature sensor output.
(2) 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, auto-discharging and A/D conversion is performed in this order for the internal reference voltage.
(3) 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), a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(4) The ADCSR.ADST bit remains 1 (starting A/D conversion) 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.

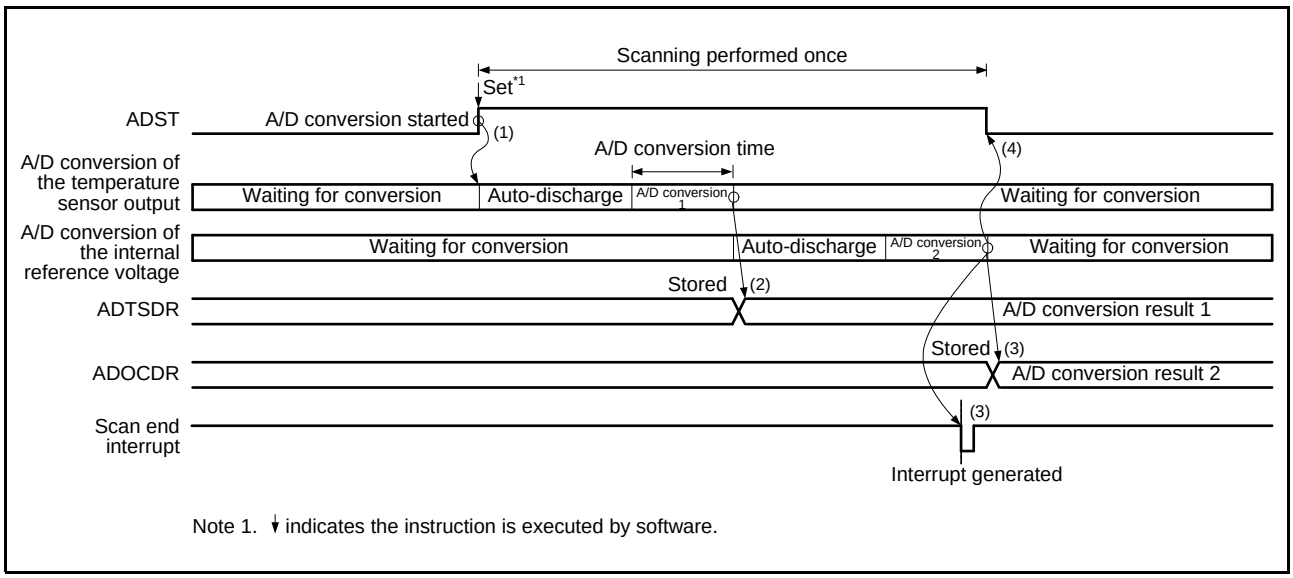


Figure 38.8 Example of Operation in Single Scan Mode (Basic Operation: Temperature Sensor Output and Internal Reference Voltage Selected)

38.3.3.4 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, ADGCEXCR.TSSC) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA, ADEXICR.OCSB, ADGCEXCR.OCSC) 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 and ADANSA1 registers 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 becomes 1 (starting A/D conversion) 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 (scan end interrupt is enabled).
(4) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) 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).
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit remains 1 (starting A/D conversion) 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.

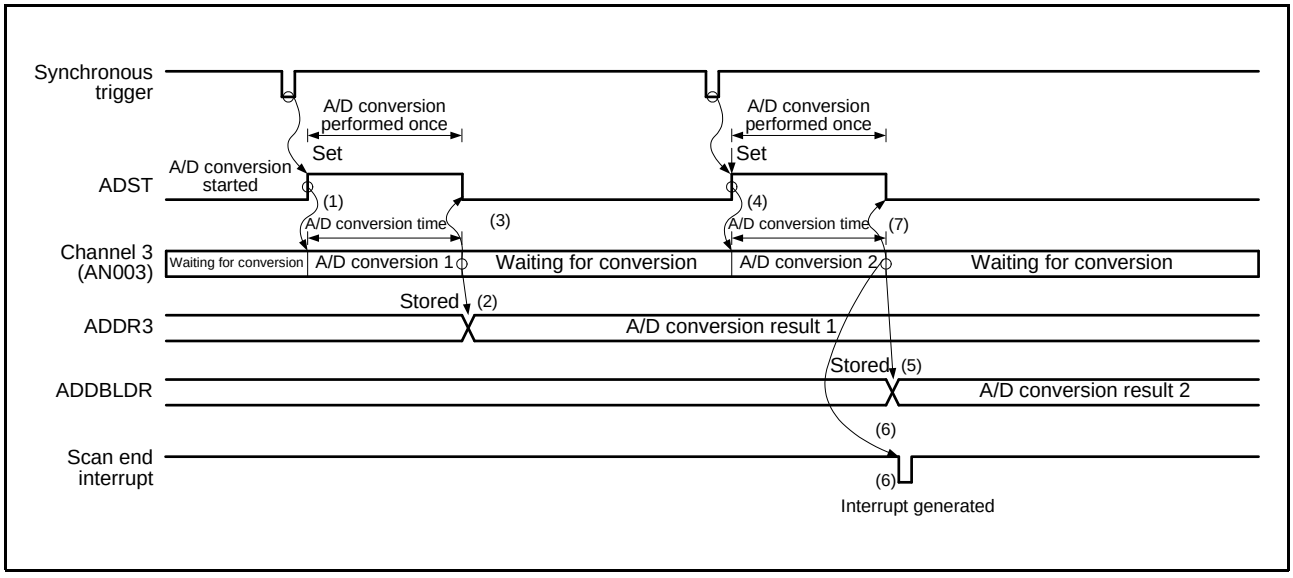


Figure 38.9 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

38.3.3.5 A/D Conversion in Extended Double Trigger Mode

When selecting the double-trigger mode in single-scanning mode, with TRG4AN or TRG4BN, TRG7AN or TRG7BN, TRGA0N or TRG0N, or ELCTRG00N or ELCTRG01N/ELCTRG10N or ELCTRG11N/ELCTRG20N or ELCTRG21N selected in the TRSA[5:0] bits of the A/D conversion start trigger select register (ADSTRGR), proceed with single scanning mode twice as follows.
Deselect self-diagnosis and set the temperature sensor A/D conversion select bit (ADEXICR.TSSA, ADEXICR.TSSB, ADGCEXCR.TSSC) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA, ADEXICR.OCSB, ADGCEXCR.OCSC) 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 and ADANSA1 registers 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 (starting A/D conversion) 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 (scan end interrupt is enabled).
(4) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) 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) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit remains 1 (starting A/D conversion) 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.

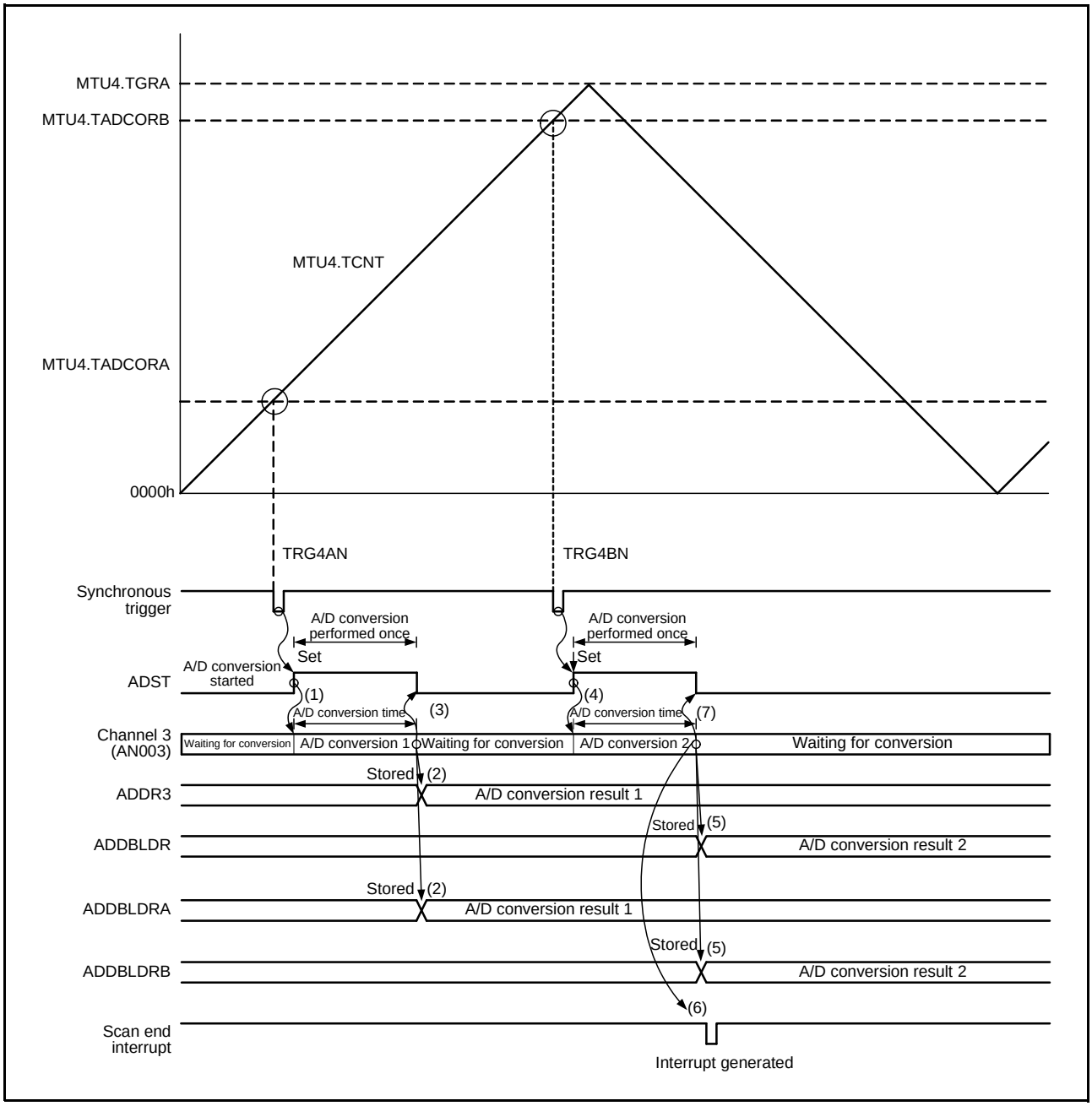


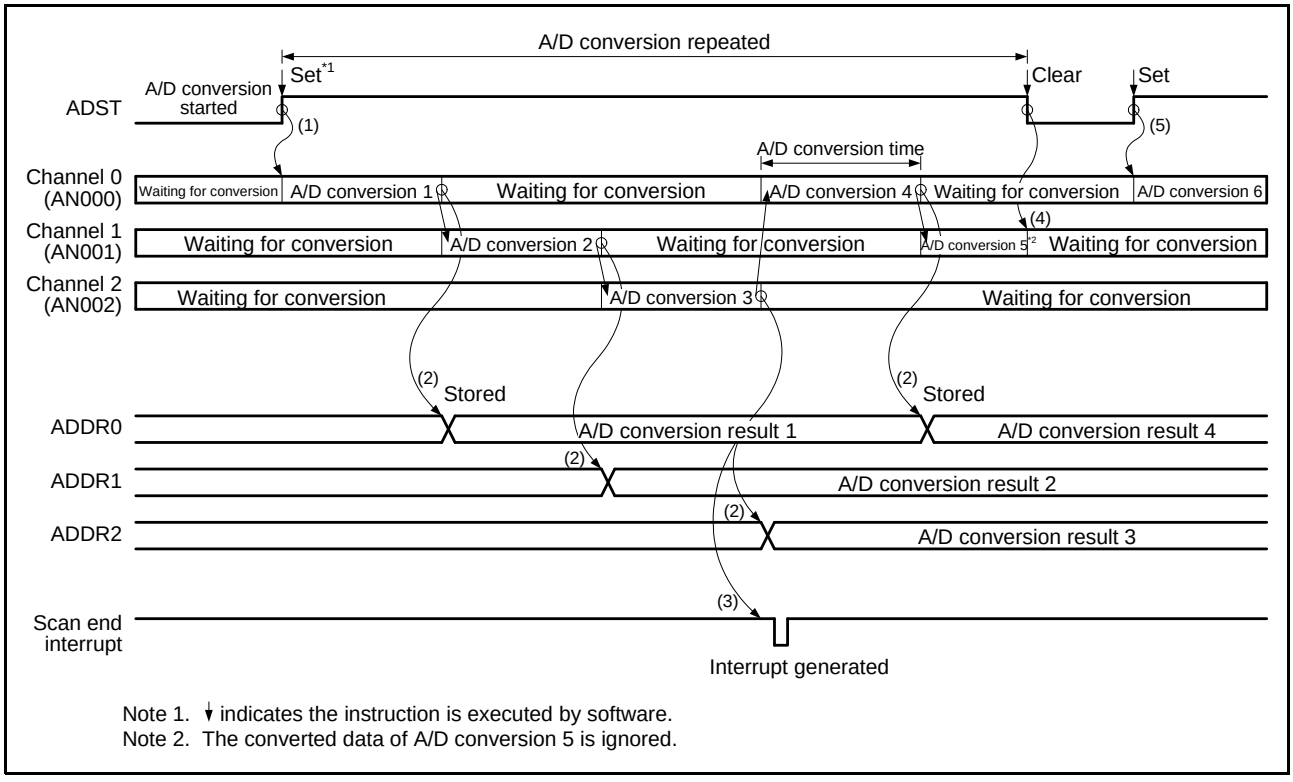
Figure 38.10 Example of Extended Operation in Double Trigger Mode (1)
(Duplication Selected for AN003, TRG4AN or TRG4BN Selected, First Trigger is TRG4AN)

38.3.4 Continuous Scan Mode

38.3.4.1 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.
In continuous scan mode, specify A/D conversion select bits for the temperature sensor and for the internal reference voltage to deselect respectively.

(1) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register. (Specified conversion order in Figure 38.11: AN000 → AN001 → AN002)
(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 (scan end interrupt is enabled).
The 12-bit A/D converter sequentially starts A/D conversion for the analog channels selected in the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register.
(4) The ADCSR.ADST bit is not automatically cleared and steps (2) and (3) are repeated as long as the bit remains 1 (starting A/D conversion). When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(5) When the ADCSR.ADST bit is later set to 1 (starting A/D conversion), A/D conversion starts again for the analog channels selected in the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register.



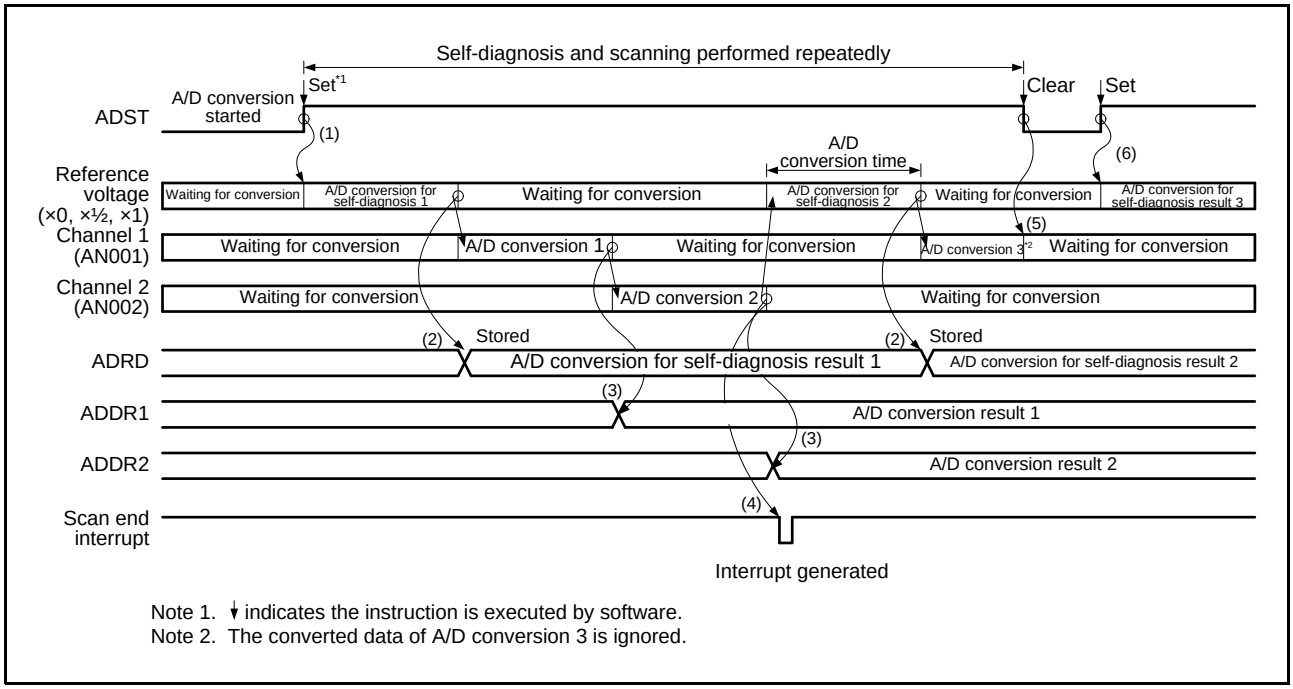
Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 38.11 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

38.3.4.2 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 (starting A/D conversion) 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, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register.
(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 (scan end interrupt is enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis, and, upon its completion, starts A/D conversion on analog channels selected by the ADANSA0 and ADANSA1 registers, according to the order of conversion specified in the ADSCSn register.
(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 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is later becomes 1 (starting A/D conversion), 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 38.12 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

38.3.5 Group Scan Mode

38.3.5.1 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, B, and C can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR register, and ADGCTRGR.TRSC[5:0] bits, 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 registers ADANSA0 and ADANSA1 and bits TSSA and OCSA in register ADEXICR for group A, registers ADANSB0 and ADANSB1 and bits TSSB and OCSB in register ADEXICR for group B, and registers ADANSC0 and ADANSC1 and bits TSSC and OCSC in register ADGCEXCR for group C.

When the temperature sensor output is to be scanned, set only one from among the TSSA, TSSB, and TSSC bits to 1. In the same way, when the internal reference voltage is to be scanned, set only one from among the OCSA, OCSB, and OCSC bits to 1. Discharging for a period of 15 ADCLK is performed for all the conversion channels in a scan group where the temperature sensor output or the internal reference voltage is set.

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 (scan end interrupt is 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 (group B scan end interrupt is 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 (group C scan end interrupt is enabled).

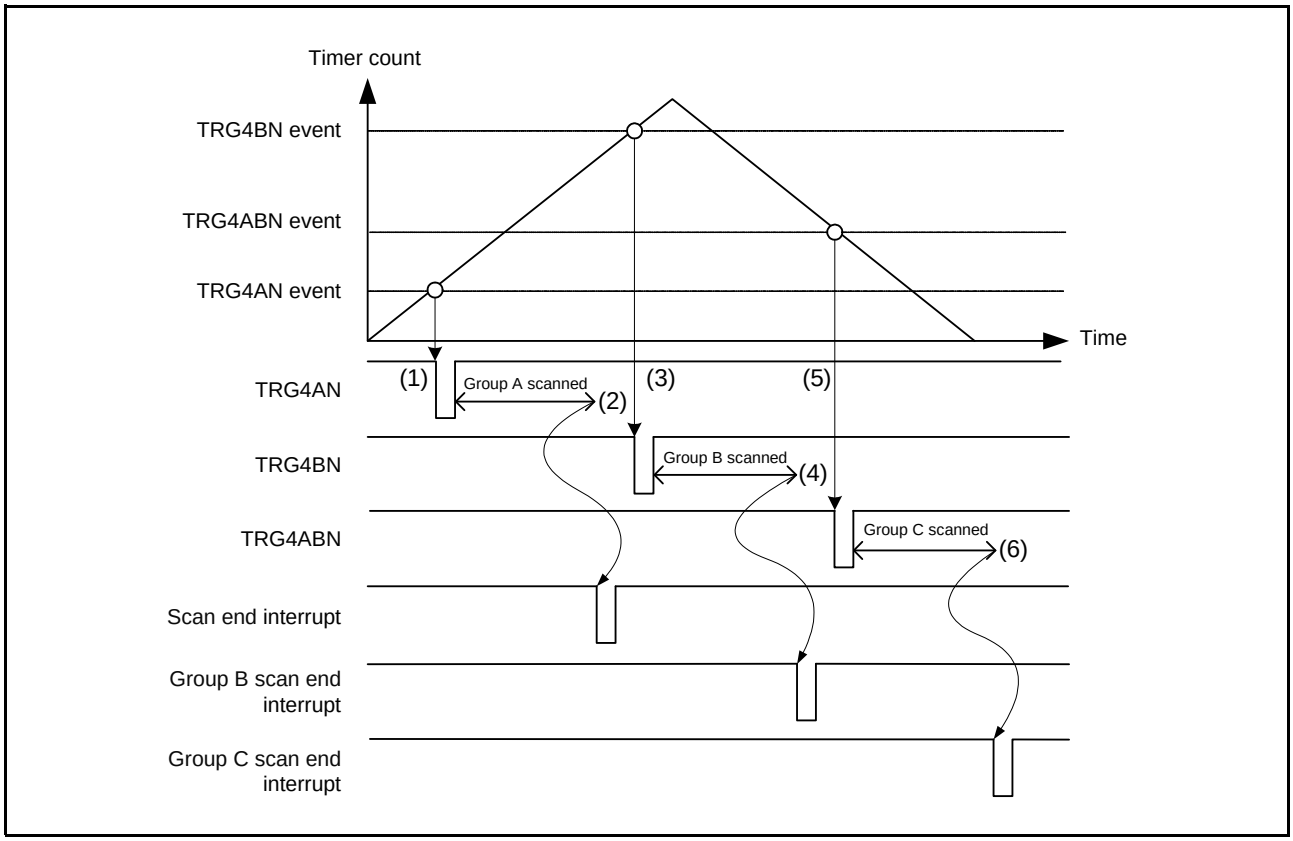


Figure 38.13 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from MTU Used)

The following describes operation using the temperature sensor output or the internal reference voltage (in group scan mode and not in group priority operation).

The operation is based on a setting: channel 0 to be A/D-converted for group A, the temperature sensor output and internal reference voltage to be A/D-converted for group B, and channel 1 to be A/D-converted for group C.

(1) When input of the group A trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), scan for group A starts.

(2) Start A/D conversion for channel 0, and on its completion, the A/D-converted value is stored in the corresponding A/D data register (ADDR0).

(3) A scan end interrupt is generated if the ADCSR.ADIE bit is set to 1 (scan end interrupt is enabled).

(4) The ADCSR.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.

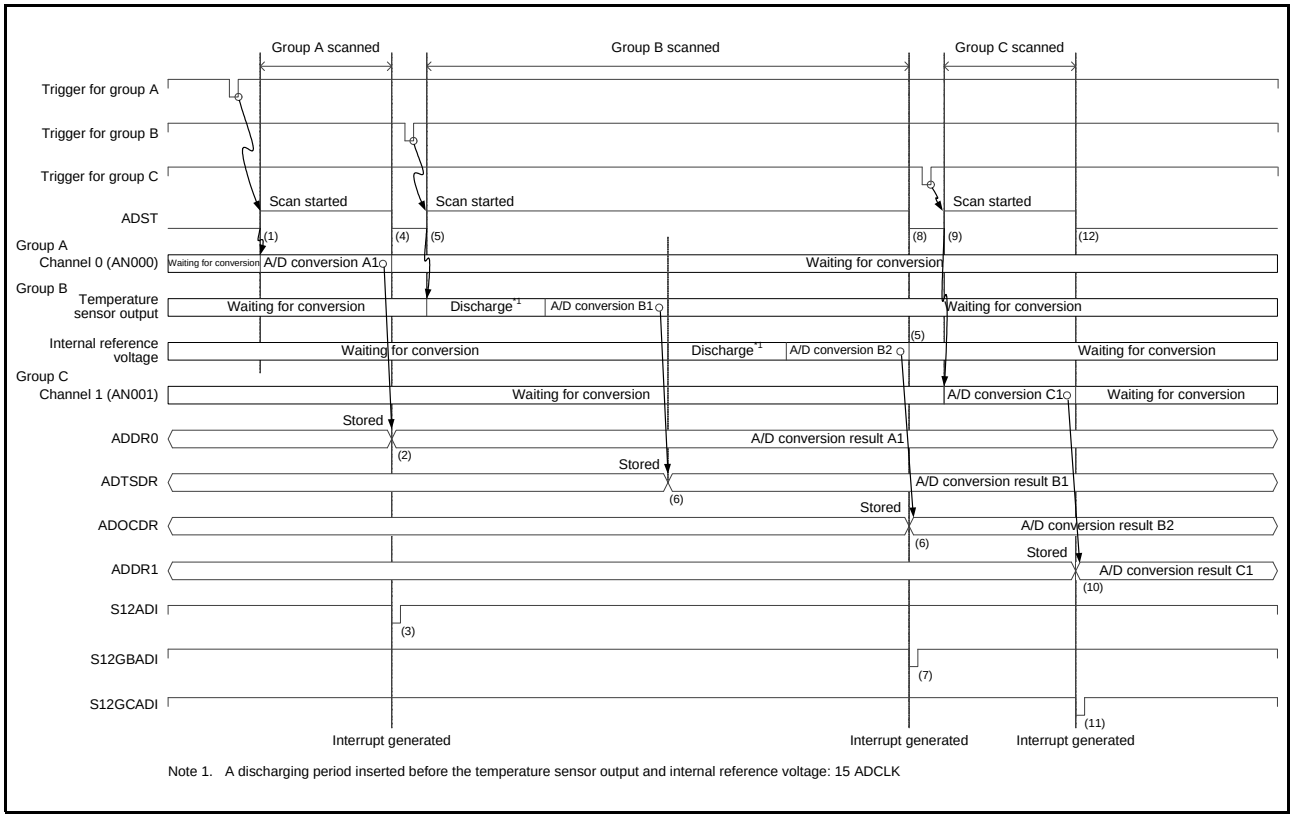
(5) When input of the group B trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), scan for group B starts.

(6) Before A/D-converting the temperature sensor output, 15 ADCLK is inserted as a discharging period. After a completion of the discharging period, A/D conversion for the temperature sensor output starts, and the A/D-converted value is stored in the corresponding A/D data register (ADTSDR) upon completion of A/D conversion. Next, before A/D-converting the internal reference voltage, 15 ADCLK is inserted as a discharging period. After a completion of the discharging period, A/D conversion for the internal reference voltage starts, and the A/D-converted value is stored in the corresponding A/D data register (ADOCDR) upon completion of A/D conversion.

(7) Group B scan end interrupt is output if the ADCSR.GBADIE bit is set to1 (group B scan end interrupt is enabled).

(8) The ADCSR.ADST bit is automatically cleared to 0 when scan for group B is completed, and the 12-bit A/D converter enters a wait state.

(9) When input of the group C trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), scan for group C starts.
(10) Start A/D conversion for channel 1, and on its completion, the A/D-converted value is stored in the corresponding A/D data register (ADDR1).
(11) Group C scan end interrupt is output if the ADGCTRGR.GCADIE bit is set to1 (group C scan end interrupt is enabled).
(12) The ADCSR.ADST bit is automatically cleared to 0 when scan for group C is completed, and the 12-bit A/D converter enters a wait state.



Note 1. A discharging period inserted before the temperature sensor output and internal reference voltage: 15 ADCLK

Figure 38.14 Operation When Using the Temperature Sensor Output and the Internal Reference Voltage (in Group Scan Mode and Not in Group Priority Operation)

38.3.5.2 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, B, and C can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR register, and ADGCTRGR.TRSC[5:0] bits, 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 TRG4AN or TRG4BN, TRG7AN or TRG7BN, TRGA0N or TRG0N, or ELCTRG00N or ELCTRG01N/ELCTRG10N or ELCTRG11N/ELCTRG20N or ELCTRG21N is selected as the synchronous trigger of group A by the ADSTRGR.TRSA[5:0] bits, operation is performed in extended double trigger mode.
The channels to be scanned are selected using bits ADCSR.DBLANS[4:0] for group A, registers ADANSB0 and ADANSB1 for group B, and registers ADANSC0 and ADANSC1 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 (group C scan end interrupt is 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 (group B scan end interrupt is 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 ADCSR.ADIE bit setting.
(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 the ADDBLDR register a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is 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 end interrupt is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is 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 end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).

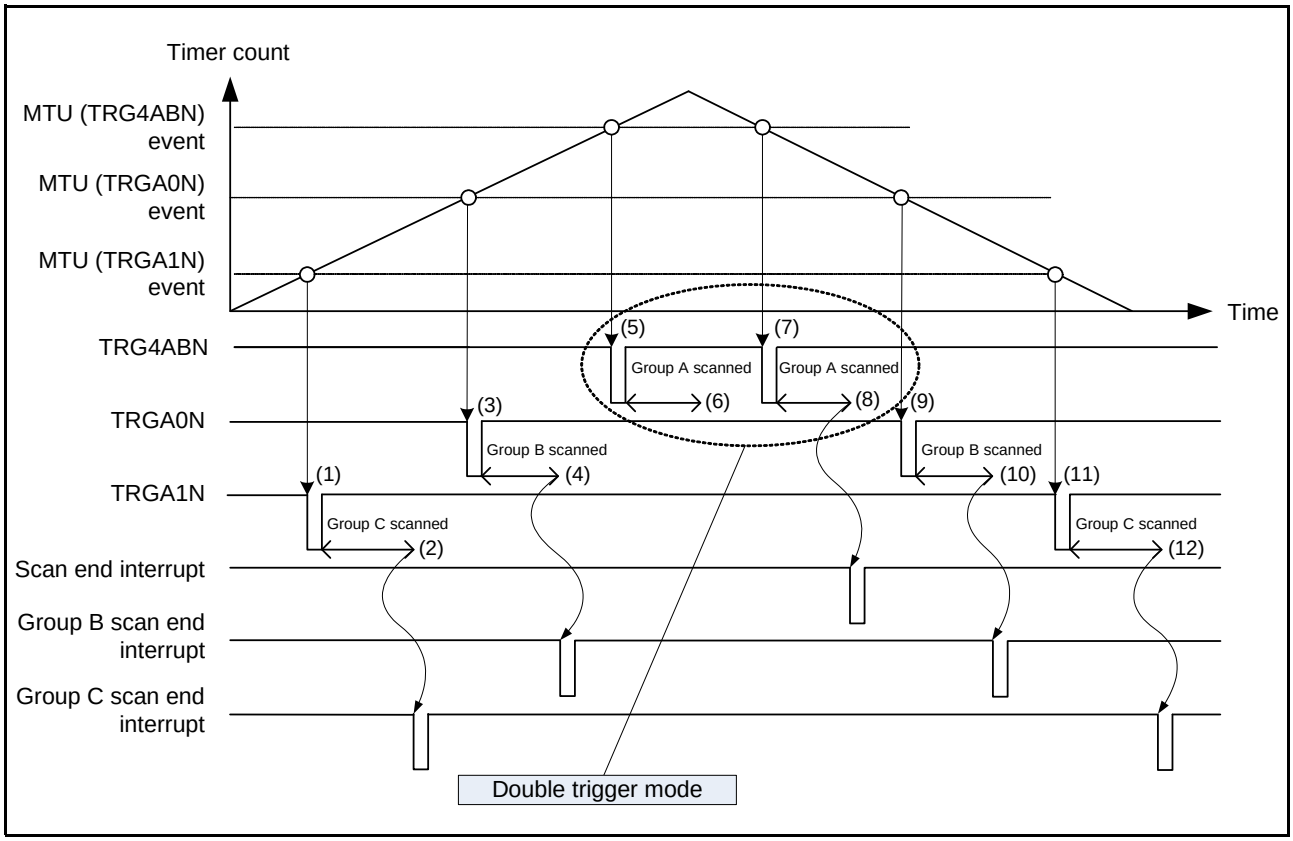


Figure 38.15 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from MTU Used)

38.3.5.3 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 38.16. 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 interrupted 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 interrupted.

However, when self-diagnosis is used, the converter restarts scanning from the channel on which scanning is interrupted after self-diagnosis is completed.

Table 38.12 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, scanning operations for the group of the lowest priority are continuously performed in single scan mode.

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 and ADANSA1 registers, and the ADEXICR.TSSA and ADEXICR.OCSA bits; select the group B channels by using the ADANSB0 and ADANSB1 registers, and the ADEXICR.TSSB and ADEXICR.OCSB bits; and select the group C by using the ADANSC0 and ADANSC1 registers, and the ADGCEXCR.TSSC and ADGCEXCR.OCSC bits respectively.

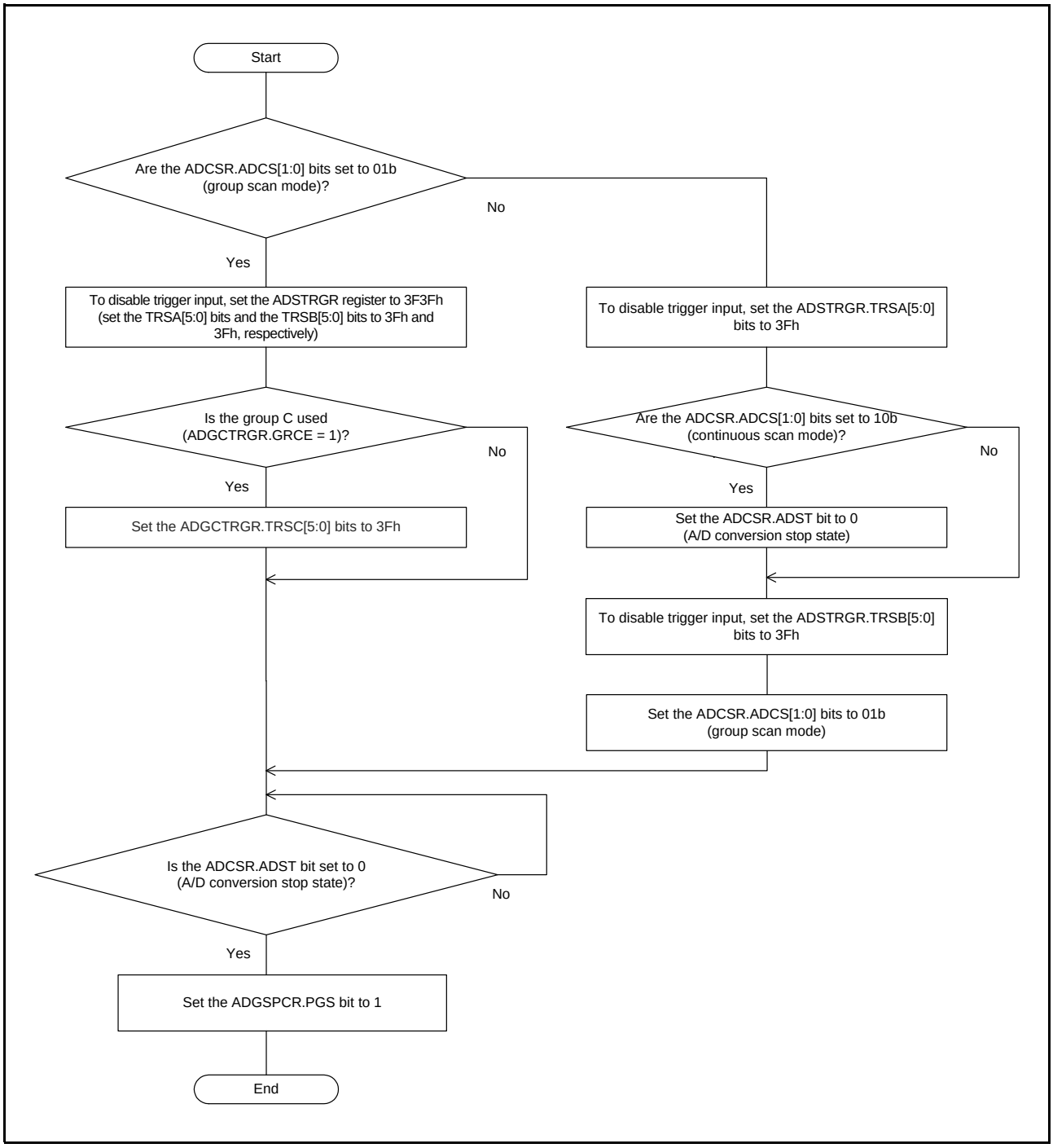


Figure 38.16 Flow of Setting the ADGSPCR.PGS Bit

Table 38.12 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 interrupted and conversion for group A starts.	• A/D conversion in progress for group B is interrupted 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 interrupted and conversion for group A starts.	• A/D conversion in progress for group C is interrupted 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 interrupted and conversion for group B starts.	• A/D conversion in progress for group C is interrupted 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 38.13 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 a scan for group B is interrupted, upon completion of a scan for group A, a scan for group B starts again from the beginning according to the order specified in the ADANSB0 register.
1	1	0	Group priority operation for two groups (groups A and B) • After group B scan is interrupted, group B scan is restarted from the channel on which scan was interrupted*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 interrupted, 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 interrupted, single scan is restarted from the channel on which scan was interrupted*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 interrupted channel is started after self-diagnosis.

Table 38.14 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 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 interrupted, scan is restarted from the head of the channel selected with the ADANSC0 and ADANSC1 registers 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 interrupted, scan is restarted from the head of the channel selected with the ADANSB0 and ADANSB1 registers after group A scan is completed. • After group C scan is interrupted, scan is restarted from the head of the channel selected with the ADANSC0 and ADANSC1 registers 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 interrupted, scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSB0 and ADANSB1 registers after scan for group A is completed. • After group C scan is interrupted, scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSC0 and ADANSC1 registers 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 interrupted, scan is restarted from the head of the channel selected with the ADANSB0 and ADANSB1 registers after group A scan is completed. • Single scan for group C is started continuously without start trigger input. After group C scan is interrupted, single scan is restarted from the head of the channel selected with the ADANSC0 and ADANSC1 registers 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 interrupted, once scanning for group A completes, scanning is resumed starting from the channel on which scanning was interrupted*1, among those channels selected with the ADANSB0 and ADANSB1 registers. • Single scan for group C is started continuously without start trigger input. After group C scan is interrupted, single scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSC0 and ADANSC1 registers 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 interrupted 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: Input of the group A trigger signal during a group B scan, with the rescan setting

(1) When input of the group B trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. (Specified conversion order in Figure 38.15: AN000 → AN001 → AN002 → AN003)

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(3) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.

(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 (scan end interrupt is enabled).

(6) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted.

(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 (group B scan end interrupt is 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.

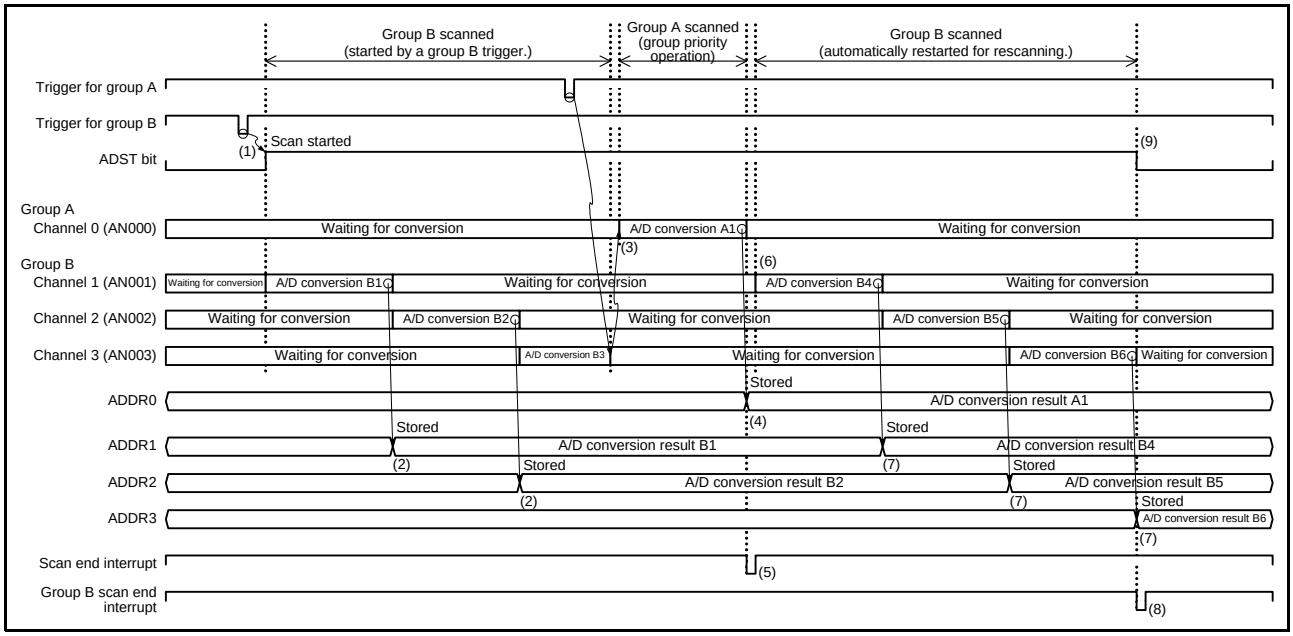


Figure 38.17 Example 1 of Group Priority Operation: Input of Group A Trigger Signal During Group B Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 2: Input of the group A trigger signal during a group B rescan, with the rescan setting

Figure 38.18 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.

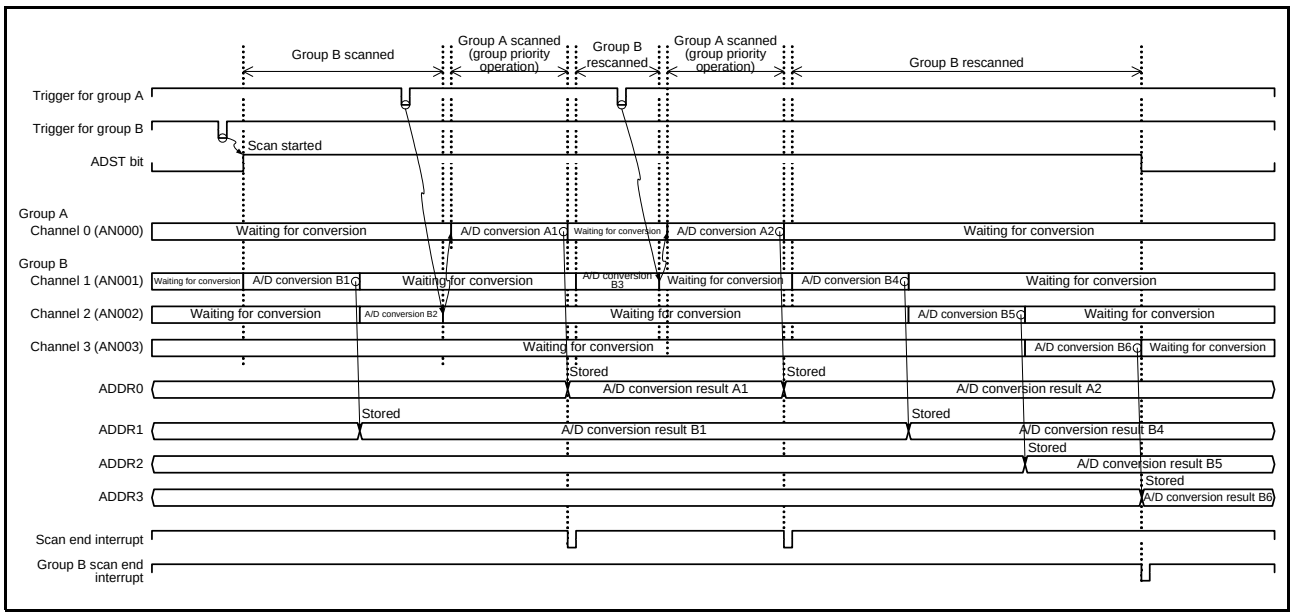


Figure 38.18 Example 2 of Group Priority Operation: Input of Group A Trigger Signal During Group B Rescan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 3: Input of the group B trigger signal during a group A scan, with the 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 interrupted 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 input of the group A trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), a group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(2) When a group B trigger is input during a group A scanning, a group B becomes ready for a scan.
(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) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(5) After the group A scan ends, the ADCSR.ADST bit retains the value as 1, and the group B scan is started for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register.
(When a group A trigger is input during the group B scanning, the group A scan is started in the same way as in the example 1, and the group B scan is restarted after the group A scan ends.)
(6) 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).
(7) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is 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.

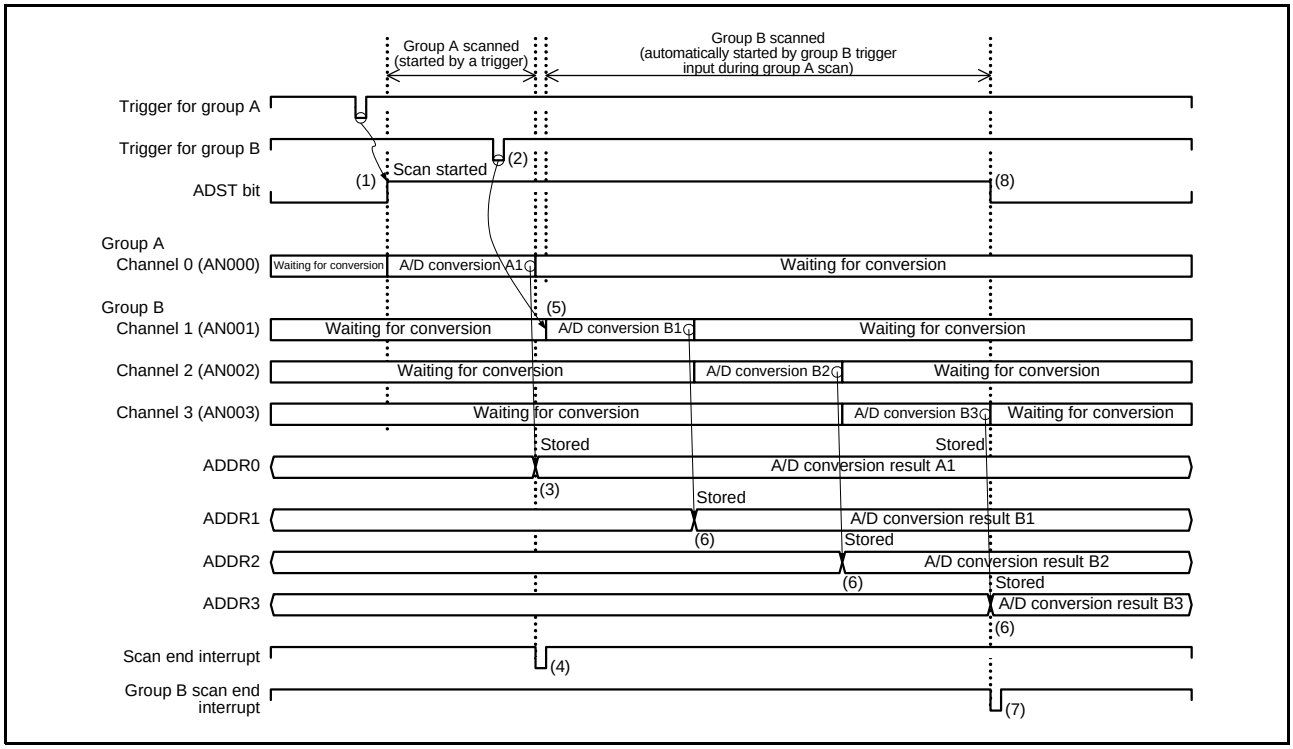


Figure 38.19 Example 3 of Group Priority Operation: Input of Group B Trigger Signal During Group A Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

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 Input of the group A trigger signal during a group B scan, without the rescan setting

(1) When input of the group B trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), a group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.

(4) 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).

(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).

(6) The ADCSR.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.

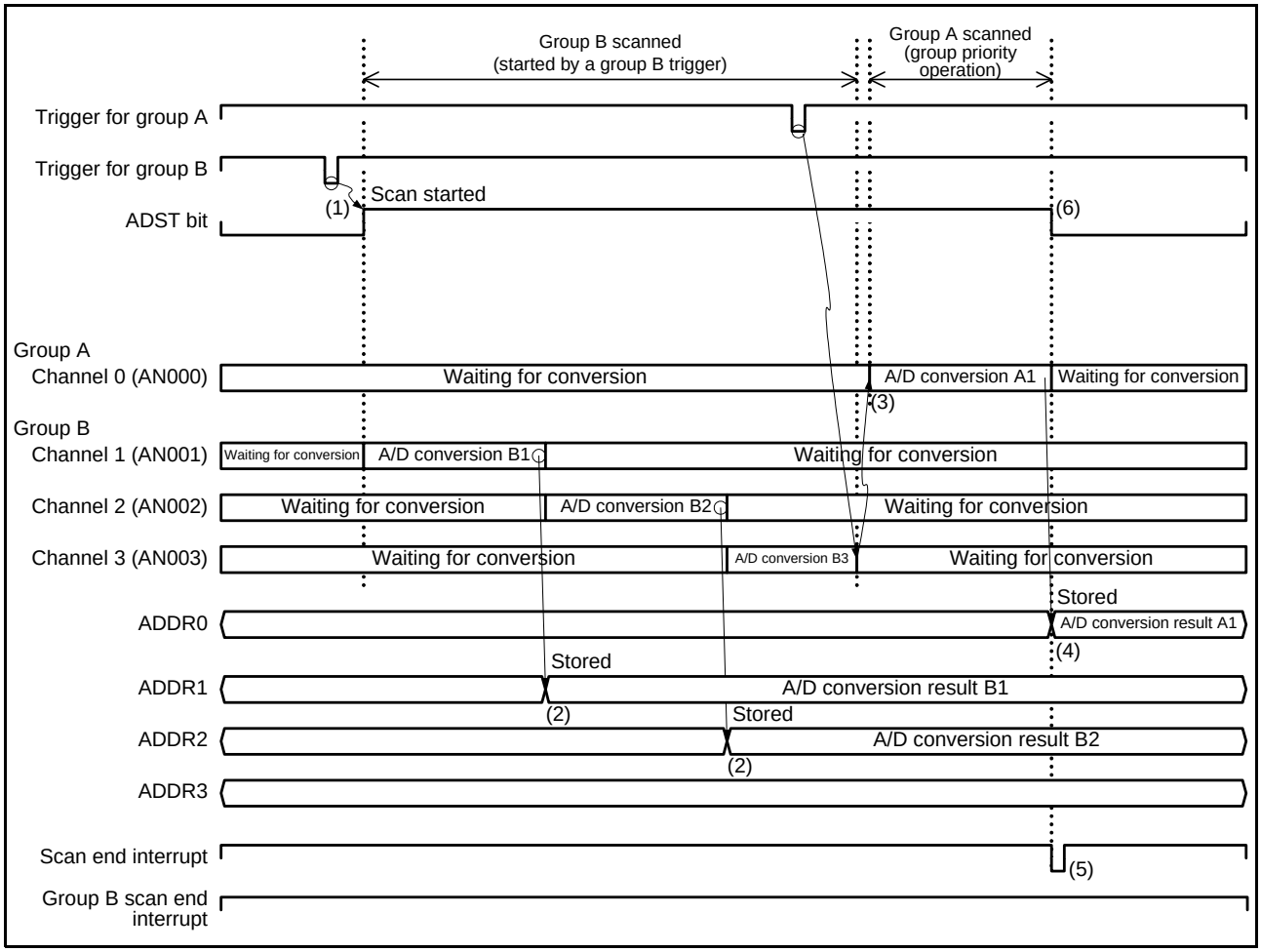


Figure 38.20 Example 4 of Group Priority Operation: Input of Group A Trigger Signal During Group B Scan, Without the Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

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 the ADGSPCR.GBRP bit is set to 1, the ADCSR.ADST bit is set to 1 (starting A/D conversion), and a group B scan is started for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(4) 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).
(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(6) When the ADGSPCR.GBRP bit is set to 1 (selecting continuous single scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted.
(7) 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).
(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(9) When the ADGSPCR.GBRP bit is set to 1 (selecting continuous single scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted.

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 38.6.2, Notes on Stopping A/D Conversion.

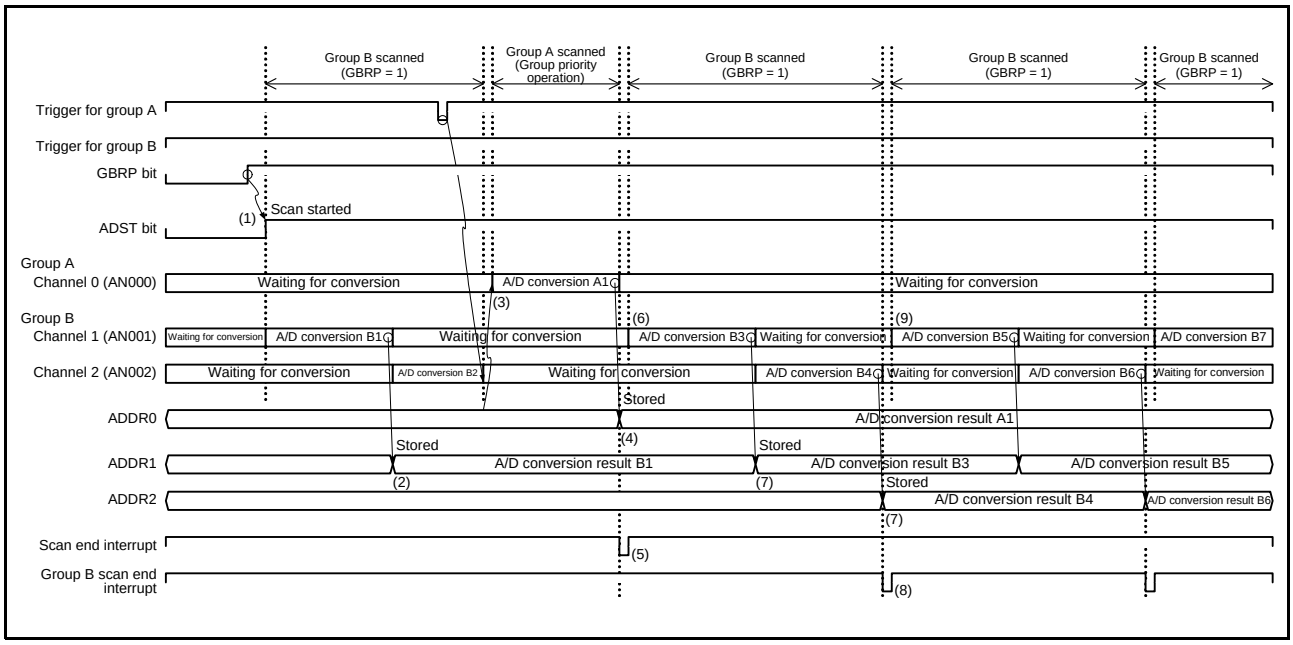


Figure 38.21 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group B (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 0)

(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 the rescan setting

(1) When input of the group C trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), the group C scan starts for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When a group B trigger is input during the group C scanning, the ADCSR.ADST bit retains the value as 1, the group C scan is interrupted, and the group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(4) 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).
(5) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(6) 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).
(7) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(8) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group B scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(9) 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).
(10) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(11) When the ADGSPCR.GBRSCN bit is set to 1, the ADCSR.ADST bit retains the value as 1, the group C scan for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group C scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(12) 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).
(13) A group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).
(14) 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.

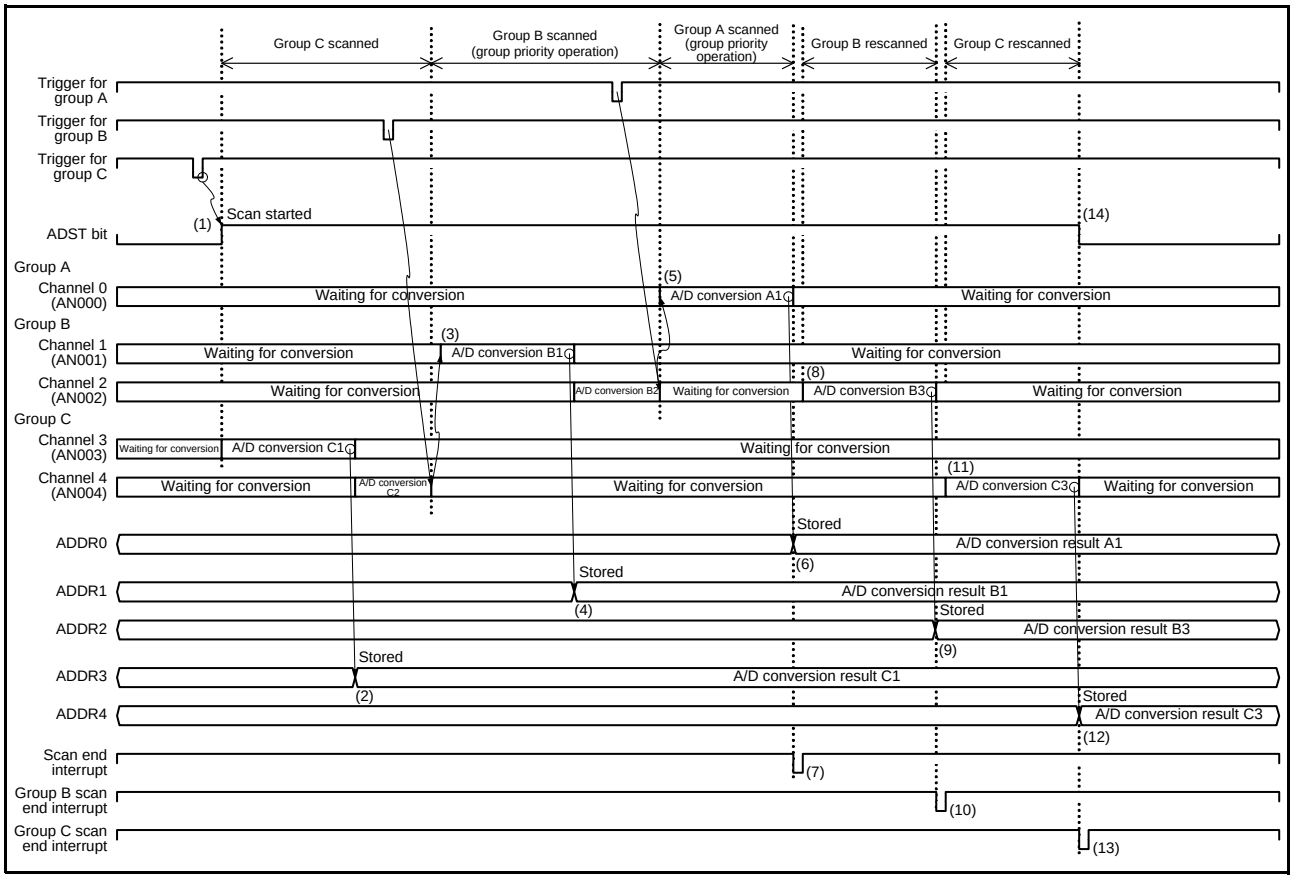


Figure 38.22 Example 1 of Group Priority Operation: Priority Group Trigger Input During Low-Priority Group Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 2: Priority group trigger input during low-priority group rescan, with the rescan setting

Figure 38.23 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 interrupted.

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.

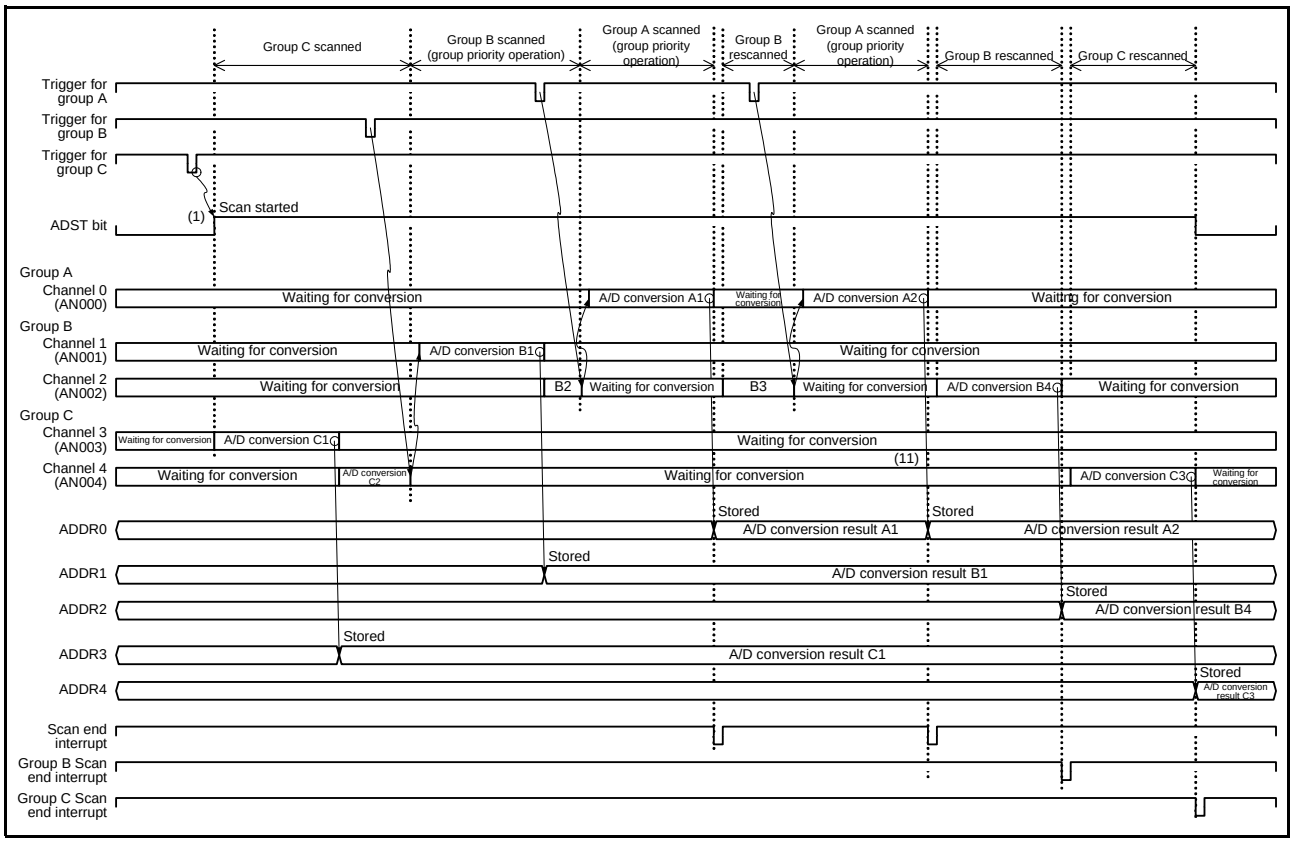


Figure 38.23 Example 2 of Group Priority Operation: Priority Group Trigger Input During Low-Priority Group Rescan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 3: Low-priority group trigger input during priority group scan, with the 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 interrupted 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 input of the group A trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), a group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(2) If a group B trigger is input during scan for group A, scan for group B can be started.
(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) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(5) After the group A scan ends, the ADCSR.ADST bit retains the value as 1, and the group B scan is started for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. At this time, the group B scan is restarted from the channel that the A/D conversion was suspended if the ADGSPCR.LGRRS bit is set to 1.
(When a group A trigger is input during the group B scanning, the group A scan is started in the same way as in the example 1, and the group B scan is restarted after the group A scan ends.)
(6) If a group C trigger is input during scan for group B, scan for group C can be started.
(7) 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).
(8) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(9) After scan for group B is completed, scan for the ANx channels of group C selected in the ADANSC0 and ADANSC1 registers, starts from the channel with the smallest number x 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 interrupted.
(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) 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).
(11) After scan for group C is completed, a group C scan end interrupt request is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is 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.

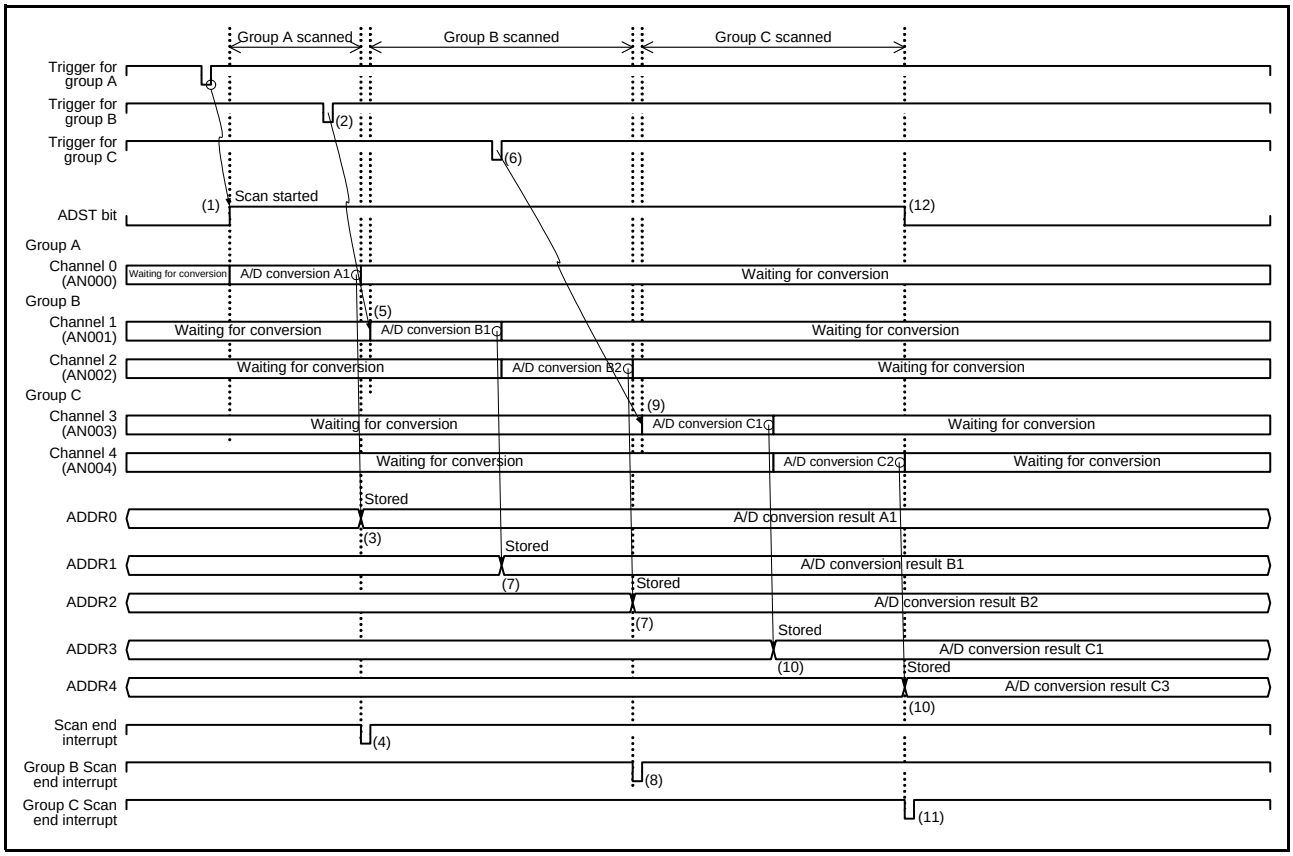


Figure 38.24 Example 3 of Group Priority Operation: Low-Priority Group Trigger Input During Priority Group Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 1)

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 the rescan setting

(1) When input of the group C trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), the group C scan starts for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When a group B trigger is input during the group C scanning, the ADCSR.ADST bit retains the value as 1, the group C scan is interrupted, and the group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.

(4) 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).

(5) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion

result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.

(6) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).

(7) The ADCSR.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.

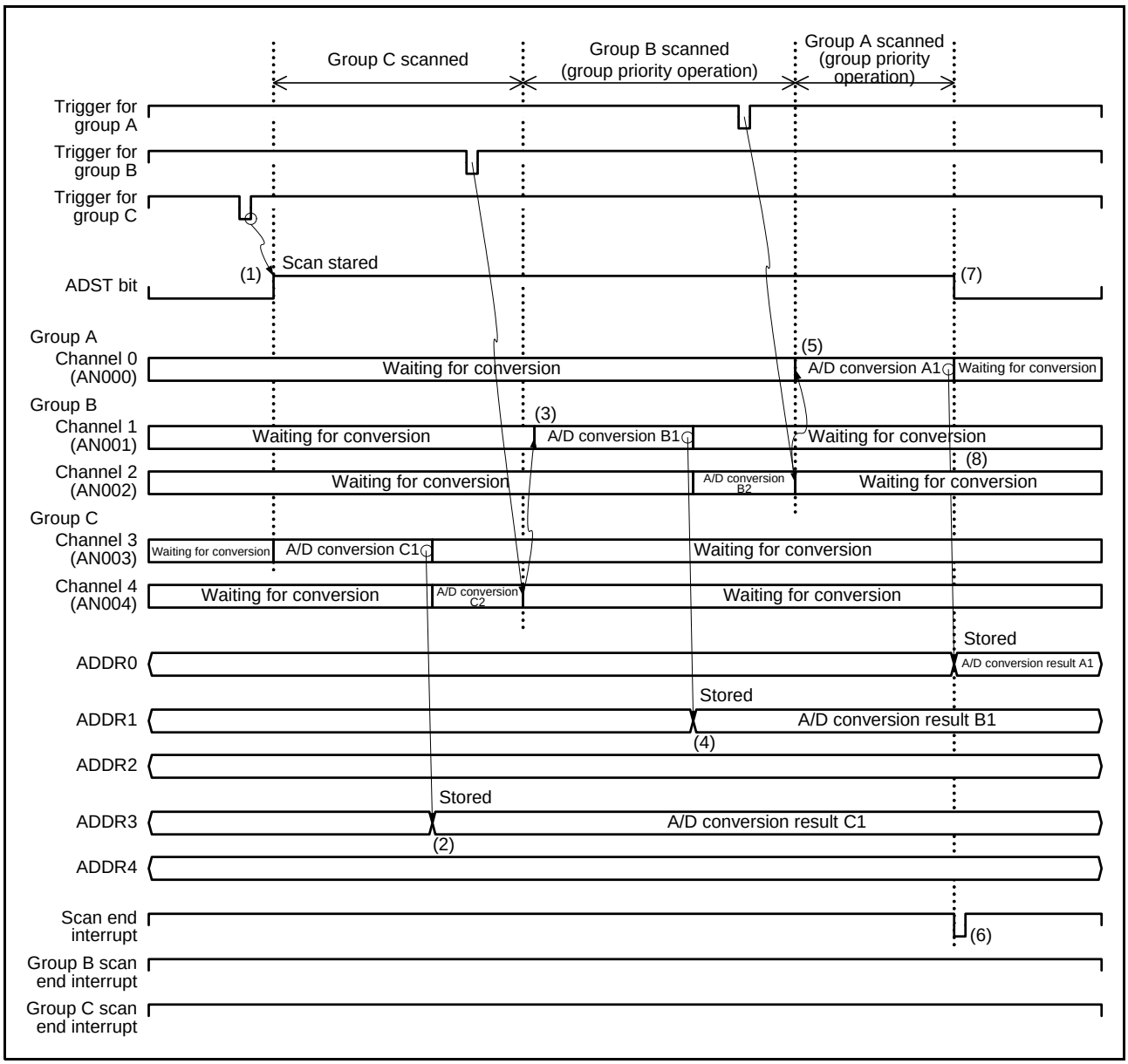


Figure 38.25 Example 4 of Group Priority Operation: Priority Group Trigger Input During Low-Priority Group Scan, Without the Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

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 0, 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 the ADGSPCR.GBRP bit is set to 1, the ADCSR.ADST bit is set to 1 (starting A/D conversion), and a group C scan is started for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When a group B trigger is input during the group C scanning, the ADCSR.ADST bit retains the value as 1, the group C scan is interrupted, and the group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(4) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(5) 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).
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group B scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(8) 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).
(9) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(10) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group C scan for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group C scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(11) 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).
(12) A group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).
(13) When the ADGSPCR.GBRP bit is set to 1 (selecting continuous single scanning), the ADCSR.ADST bit retains the value as 1, and the group C scan for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register is restarted.

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 38.6.2, Notes on Stopping A/D Conversion.

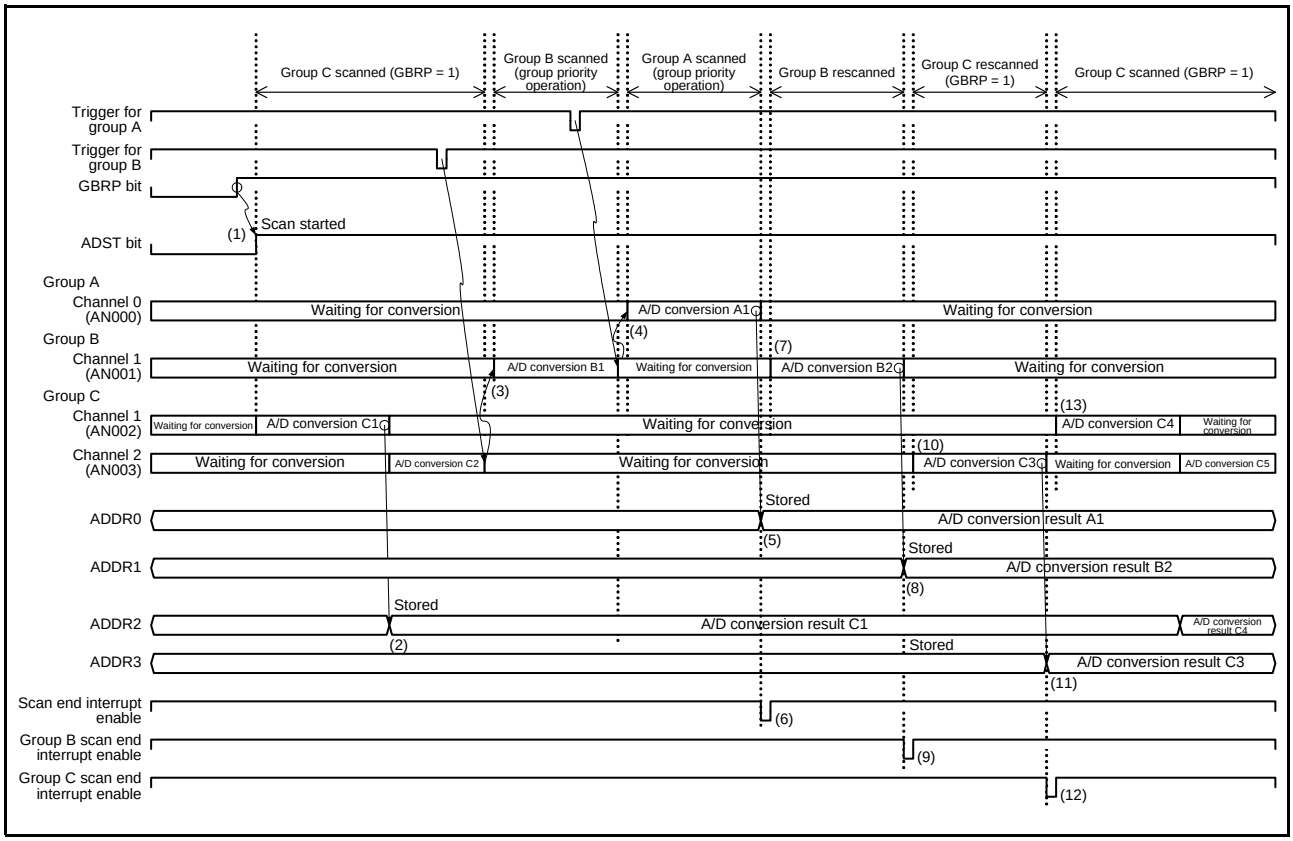


Figure 38.26 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group C (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1)

38.3.6 Comparison Function (Window A and Window B)

38.3.6.1 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 (starting A/D conversion) 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.

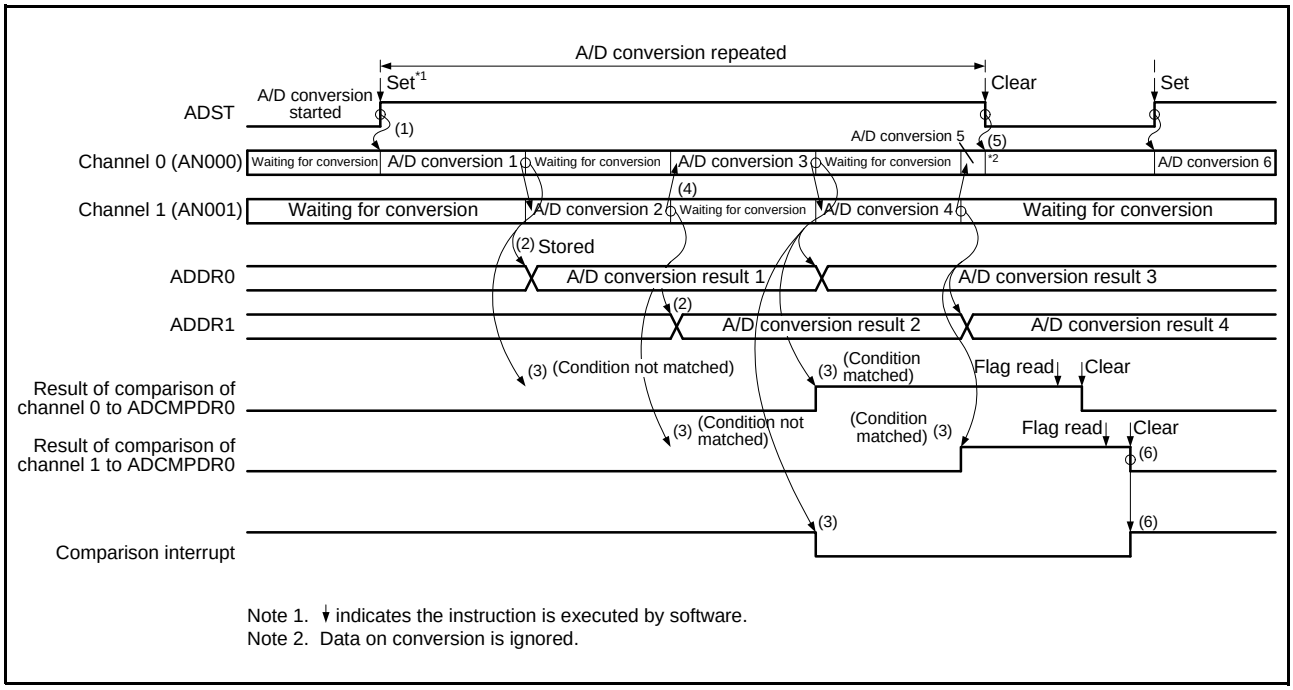
(2) Each time A/D conversion of a single channels is completed, the A/D-converted value is stored in the corresponding A/D data register (ADDRy, ADTSDR, or ADOCDR). When the ADCMPCR.CMPAE bit = 1 and window A is selected in the ADCMPANSRy and ADCMPANSER registers, the A/D-converted value is compared with the setting values of the ADCMPDR0 and ADCMPDR1 registers.
When the ADCMPCR.CMPBE bit = 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) For window A, upon a comparison match with the conditions specified in the ADCMPLR0, ADCMPLR1, and ADCMPLER registers, the flags of comparison window A (ADCMPSR0.CMPSTCHA0n, ADCMPSR1.CMPSTCHA1n, ADCMPSER.CMPFTS, and ADCMPSER.CMPFOC) become 1.
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) becomes 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 bit to 0 (stopping A/D conversion) 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 38.27 Example of Operation of Comparison (AN000 and AN001 for Targets for Comparison)

38.3.6.2 Restrictions on Comparison Function

The Comparison function has the following restrictions.

1. Use of self-diagnosis and double trigger mode is prohibited.
   (ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB are not targeted for the comparison function)
2. When using a matching or unmatching event, set to single scan mode.
3. When temperature sensor or internal reference voltage is selected for window A, operation of window B is prohibited.
4. When temperature sensor or internal reference voltage is selected for window B, operation of window A is prohibited.
5. The same channel cannot be set for window A and window B.
6. Satisfy the condition of [reference value (high side) ≥ reference value (low side)].

38.3.7 Analog Input Sampling Time and Scan Conversion Time

Figure 38.28 shows the scan conversion timing in single scan mode, in which scan conversion is started by software or a synchronous trigger. Figure 38.29 shows the scan conversion timing in single scan mode, in which scan conversion is started 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, auto-discharging processing time when A/D converting the temperature sensor output and the internal reference voltage (t_{ADIS}), 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}) is at 24 clock cycles of ADCLK. Table 38.15 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 $+ t_{ED}$

t_{SCAN} (when converting temperature sensor output and internal reference voltage) $= t_D + (t_{ADIS} \times m) + (t_{CONV} \times m) + t_{ED}$*4, *5, *6

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, $t_{DIS} = 0$. When A/D-converting the temperature sensor output and the internal reference voltage, the value is fixed to 0Fh indicating 15 ADCLK.
Note 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.
Note 3. Although the total t_{CONV} can be expressed as $t_{CONV} \times n$ when the sampling times (t_{SPL}) of the selected channels are the same, it is basically expressed as the sum of the sampling times (t_{SPL}) for each of the channels and times for conversion by successive approximation (t_{SAM}) for each of the channels.
Note 4. Set m to 2 to perform A/D conversion both the temperature sensor output and internal reference voltage, or set m to 1 to perform A/D conversion either of them.
Note 5. When performing A/D conversion of the temperature sensor output or internal reference voltage, t_{DIAG} is 0, since A/D conversion of the self-diagnosis and analog channels cannot be selected in these cases.
Note 6. If the self-diagnosis and analog channels are in one group and the temperature sensor output and internal reference are in another, A/D conversion for all of them can be performed for a group scan.

Table 38.15 Times for Conversion During Scanning (in Numbers of Cycles of ADCLK and PCLKB)

Item			Symbol		Type/Conditions: Synchronous Trigger*3	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 started after low-priority group B is stopped due to an A/D conversion start trigger of the priority group.)	t_D		2 PCLKB + 6 ADCLK (5 PCLKB + 3 ADCLK)*4	—	—	Cycle
		The low-priority group is not to be stopped. (Started by an A/D conversion start trigger of the priority group.)			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	Other than above				2 PCLKB + 4 ADCLK	4 PCLKB + 4 ADCLK	4 ADCLK	
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK*5			
Auto-discharging processing time (Time required at the conversion of the temperature sensor output and the internal reference voltage)			t_{ADIS}		15 ADCLK			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 1Bh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	24 ADCLK			
	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_{DED}		2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 15, L, T, O) (initial value = 1Bh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	24 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK (2 PCLKB + 2 ADCLK)*4			

Note 1. For t_D, t_{DIAG}, t_{CONV}, and t_{ED}, refer to Figure 38.28 and Figure 38.29.
Note 2. This is the maximum time required from software writing or trigger input to start A/D conversion.
Note 3. Time consumed on the paths from timer output to trigger input is not included.
Note 4. Maximum time for when ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2)
Note 5. When A/D-converting the temperature sensor output and the internal reference voltage, the value is fixed to 0Fh indicating 15 ADCLK.

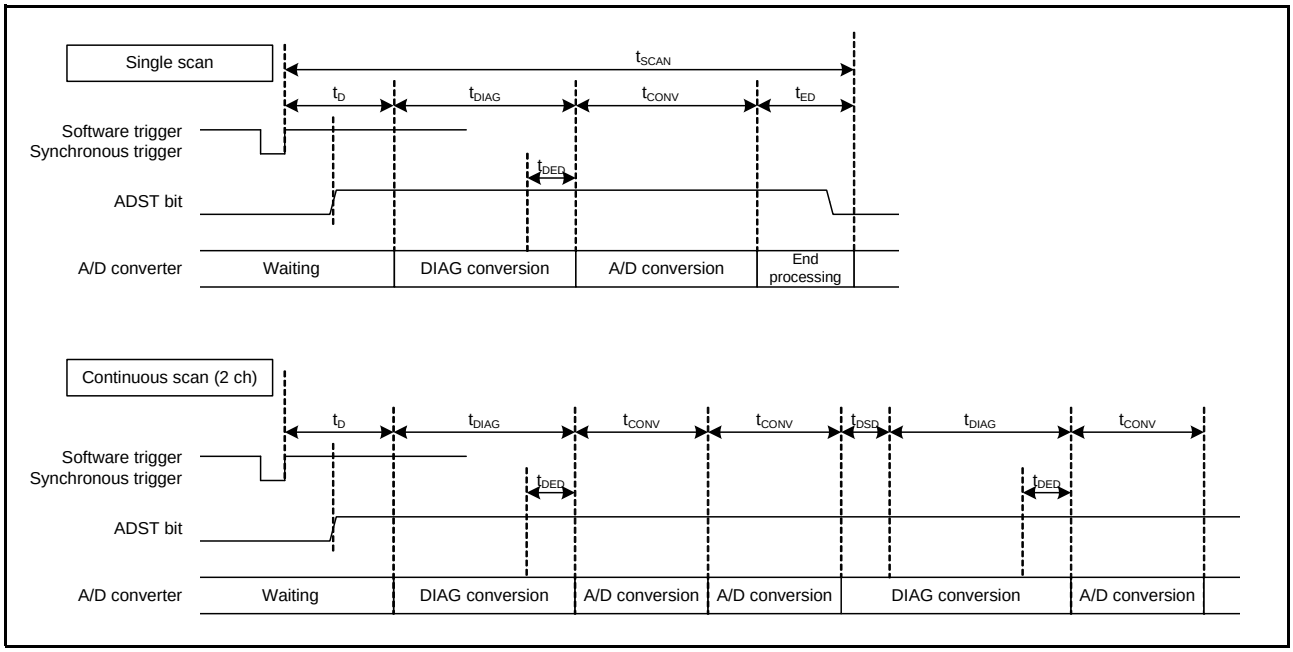


Figure 38.28 Scan Conversion Timing (Started by Software or Synchronous Trigger)

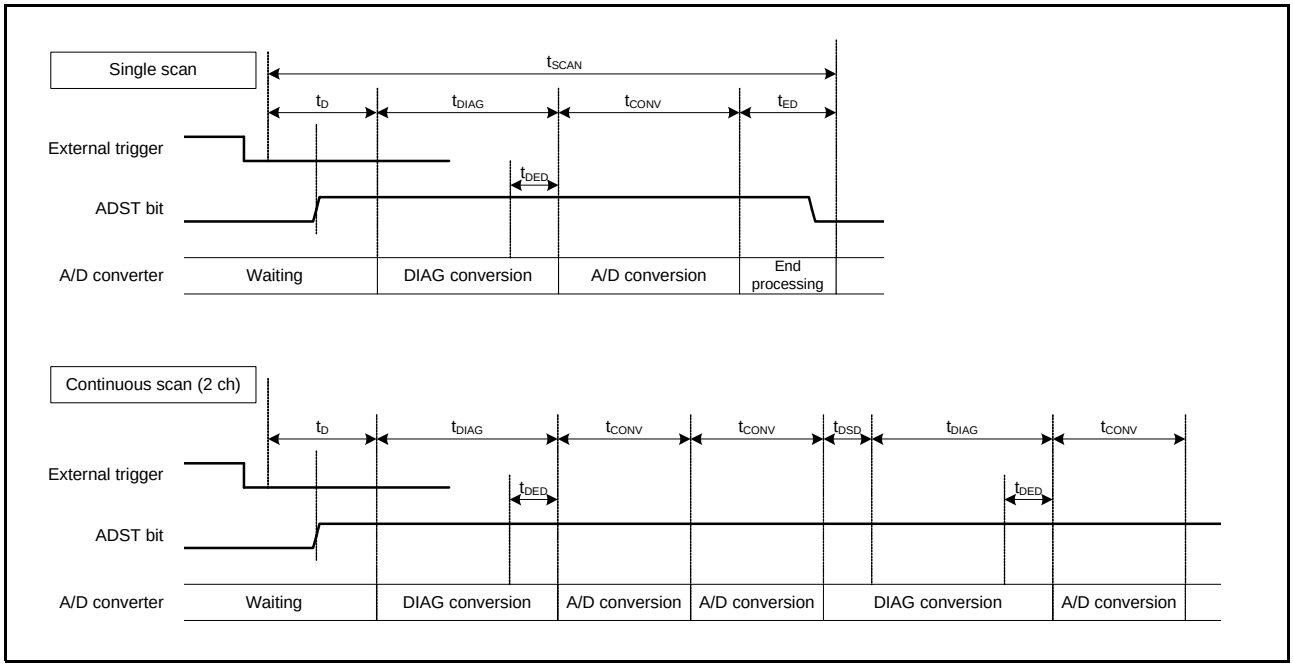


Figure 38.29 Scan Conversion Timing (Started by Software or Asynchronous Trigger)

38.3.7.1 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 38.30 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 38.30.
The timing for consecutive single scans is the same for group B and group C.

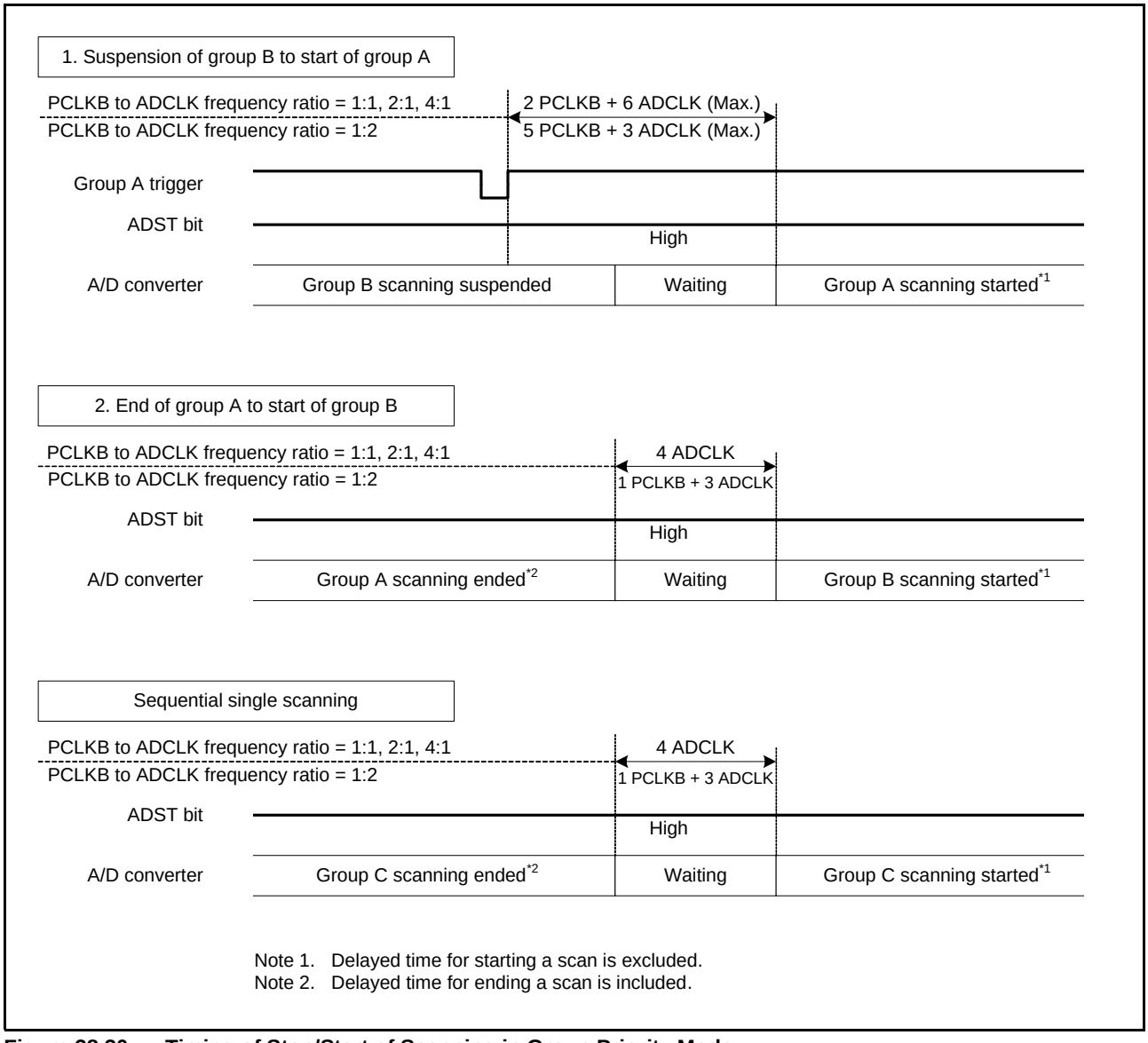


Figure 38.30 Timing of Stop/Start of Scanning in Group Priority Mode

38.3.8 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, and ADDBLDRB) to 0000h when the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, and 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, and 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 register value. Furthermore, if this ADDRy register 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, the ADDRy register is automatically cleared to 0000h. After that, if the A/D conversion result (0222h) cannot be transferred to the ADDRy register for some reason, the cleared data (0000h) remains as the ADDRy register value. If this ADDRy register 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 register update failure can be determined by simply checking that the read data value is 0000h.

38.3.9 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 S12AD2 only), or internal reference voltage (for S12AD2 only) is selected.

38.3.10 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 38.31 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 38.32 shows an example of disconnection detection when precharge is selected. Figure 38.33 shows an example of disconnection detection when discharge is selected.

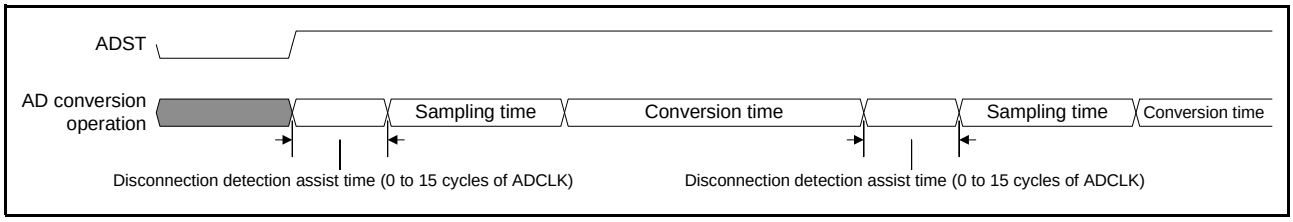
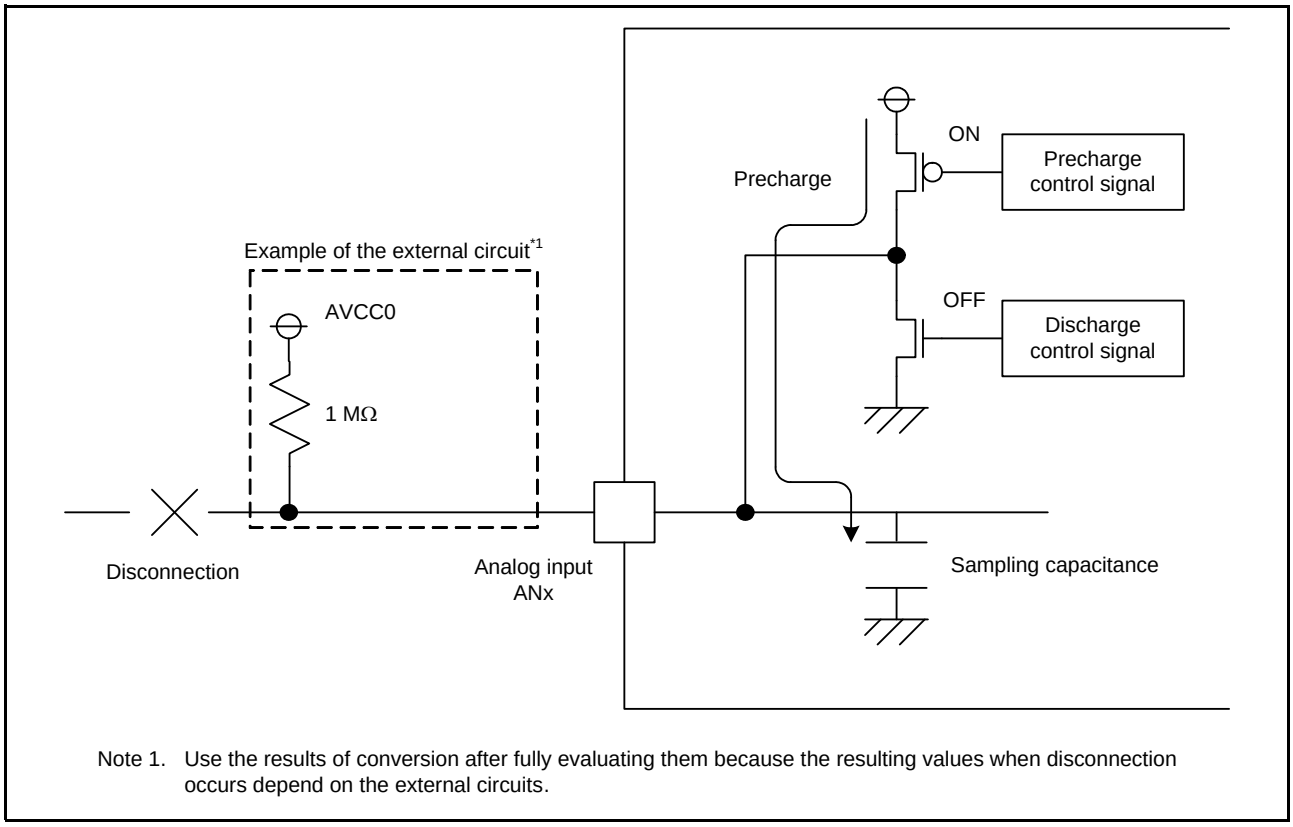
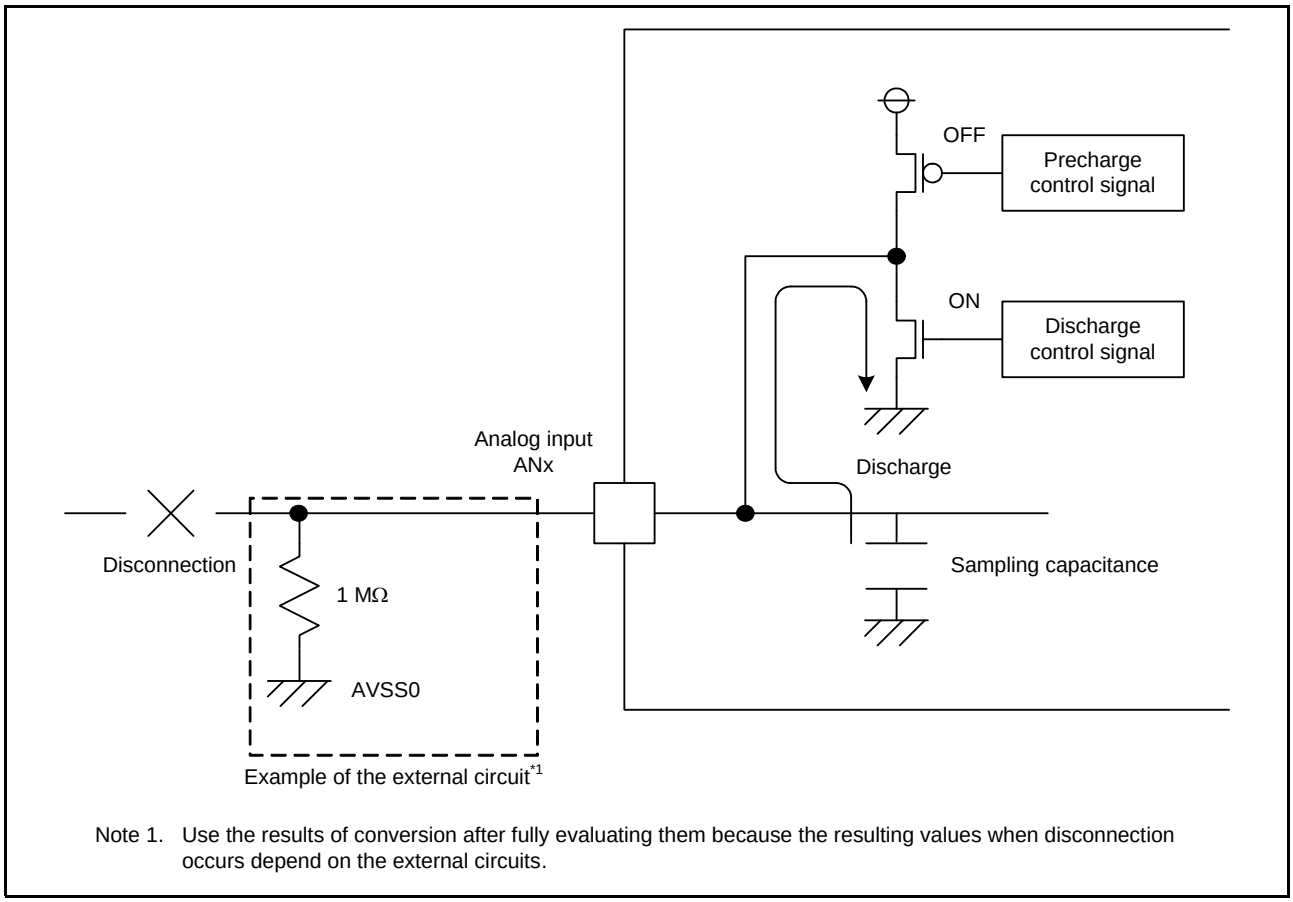


Figure 38.31 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. Use the results of conversion after fully evaluating them because the resulting values when disconnection occurs depend on the external circuits.

Figure 38.32 Example of Disconnection Detection When Precharge is Selected



Note 1. Use the results of conversion after fully evaluating them because the resulting values when disconnection occurs depend on the external circuits.

Figure 38.33 Example of Disconnection Detection When Discharge is Selected

38.3.11 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 38.34 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 38.6.3, A/D Conversion Restarting Timing and Termination Timing. An asynchronous trigger cannot be selected for group B or group C in group scan mode.

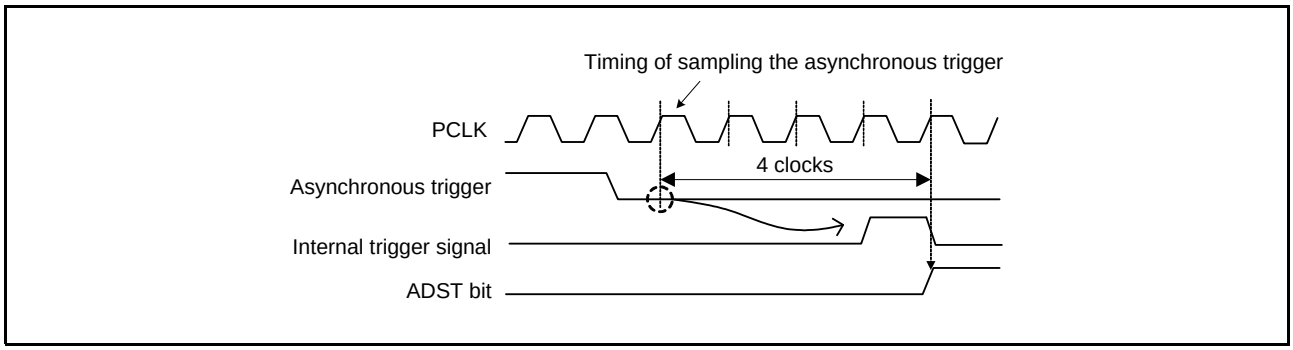


Figure 38.34 Timing of the Asynchronous Trigger Input

38.3.12 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.

38.3.13 Conversion Function in the Order of Arbitrary Channel Number

Conversion function in the order of arbitrary channel number is used to convert analog channels in the 12-bit A/D converter in the order set by the ADSCSn register.

Setting the same channel in the ADSCSn register is prohibited. Channels whose order can be set with this function are analog channels only.

The conversion order for the self-diagnosis, temperature sensor, and internal reference voltage cannot be changed.

Table 38.16 Example of Conversion Order When the ADSCSn Register (n = 0 to 4) is Set to the Initial Value in the Scan Group A

ADSCSn Setting Value		A/D Conversion Order			ADANSA0 Setting Value
Symbol	Initial Value	Physical Channel	Priority	Conversion Order	Channel Select Bit
ADSCS0	00h	AN000	High	1	ANSA0[0] =1
ADSCS1	01h	AN001	↓	2	ANSA0[1] =1
ADSCS2	02h	AN002		3	ANSA0[2] =1
ADSCS3	03h	AN003		4	ANSA0[3] =1
ADSCS4	04h	AN004	Low	5	ANSA0[4] =1

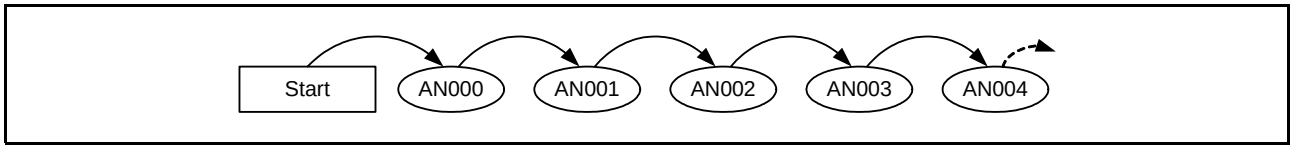


Figure 38.35 Example of Conversion Order (Setting Based on Table 38.16)

Table 38.17 Example of Conversion Order When a Setting Value of the ADSCSn Register (n = 0 to 4) is Modified in the Scan Group A

ADSCSn Setting Value		A/D Conversion Order			ADANSA0 Setting Value
Symbol	Value Set by User	Physical Channel	Priority	Conversion Order	Channel Select Bit
ADSCS0	04h	AN004	High	1	ANSA0[4] =1
ADSCS1	02h	AN002	↓	2	ANSA0[2] =1
ADSCS2	00h	AN000		3	ANSA0[0] =1
ADSCS3	01h	AN001		4	ANSA0[1] =1
ADSCS4	03h	AN003	Low	5	ANSA0[3] =1

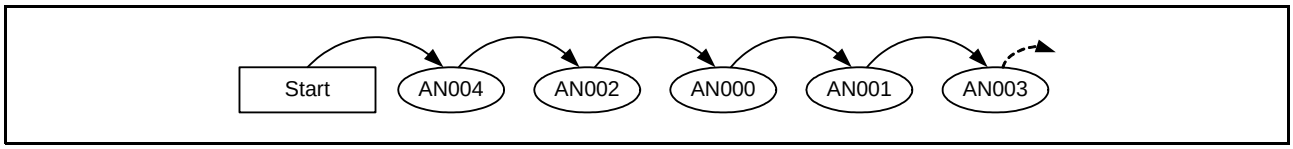


Figure 38.36 Example of Conversion Order (Setting Based on Table 38.17)

Table 38.18 Example of Conversion Order When a Setting Value of the ADSCSn Register (n = 0 to 4) is Modified in the Scan Group A (When the ANSA0[1] Bit is Set to Deselect.)

ADSCSn Setting Value		A/D Conversion Order			ADANSA0 Setting Value
Symbol	Value Set by User	Physical Channel	Priority	Conversion Order	Channel Select Bit
ADSCS0	04h	AN004	High	1	ANSA0[4] =1
ADSCS1	02h	AN002	↓	2	ANSA0[2] =1
ADSCS2	00h	AN000		3	ANSA0[0] =1
ADSCS3	01h	AN001		4	ANSA0[1] =0
ADSCS4	03h	AN003	Low	5	ANSA0[3] =1

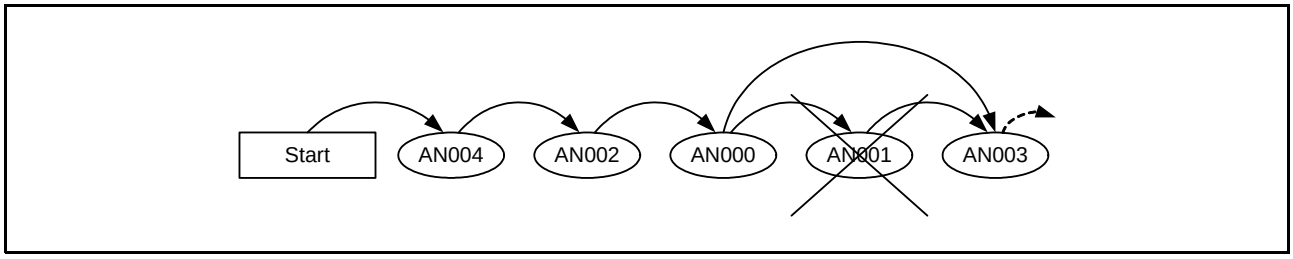


Figure 38.37 Example of Conversion Order (Setting Based on Table 38.18)

38.3.14 Internal Reference Voltage Monitoring Function

The internal reference voltage monitoring function is used to output the internal reference voltage to the A/D converter. Set the A/D internal reference voltage monitoring circuit enable register (ADVMONCR) and the A/D internal reference voltage monitoring circuit output enable register (ADVMONO), and enable the internal reference voltage A/D select bit (OCSA/B/C) to perform A/D conversion of the internal reference voltage.

Example of the operation is shown below.

(1) Set the VDE bit in the A/D internal reference voltage monitoring circuit enable register (ADVMONCR) to 1.
(2) Set the VDO bit in the A/D internal reference voltage monitoring circuit output enable register (ADVMONO) to 1.
(3) Select the internal reference voltage either by the internal reference voltage A/D conversion select bit (OCSA) in the A/D conversion extended input control register, the group B internal reference voltage A/D conversion select bit (OCSB), or the group C internal reference voltage A/D conversion select bit (OCSC) in the A/D group C extended input control register (ADGCEXCR).
(4) After inserting the waiting period of 500 ns, the ADCSR.ADST bit becomes 1 (starting A/D conversion) either by software, synchronous trigger (MTU, TMR, and ELC), or asynchronous trigger, and then, auto-discharging of the internal reference voltage is performed and A/D conversion of the internal reference voltage starts.
(5) When A/D conversion of the internal reference voltage is completed, A/D conversion result is stored to the corresponding A/D internal reference voltage data register (ADOCDR).
(6) S12ADI interrupt request is generated (pulse output and level output) if the ADCSR.ADST bit is set to 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit is automatically cleared upon completion of A/D conversion, and the 12-bit A/D converter enters a wait state.
(8) Set the VDO bit in the A/D internal reference voltage monitoring circuit output enable register (ADVMONO) to 0.
(9) Set the VDE bit in the A/D internal reference voltage monitoring circuit enable register (ADVMONCR) to 0.

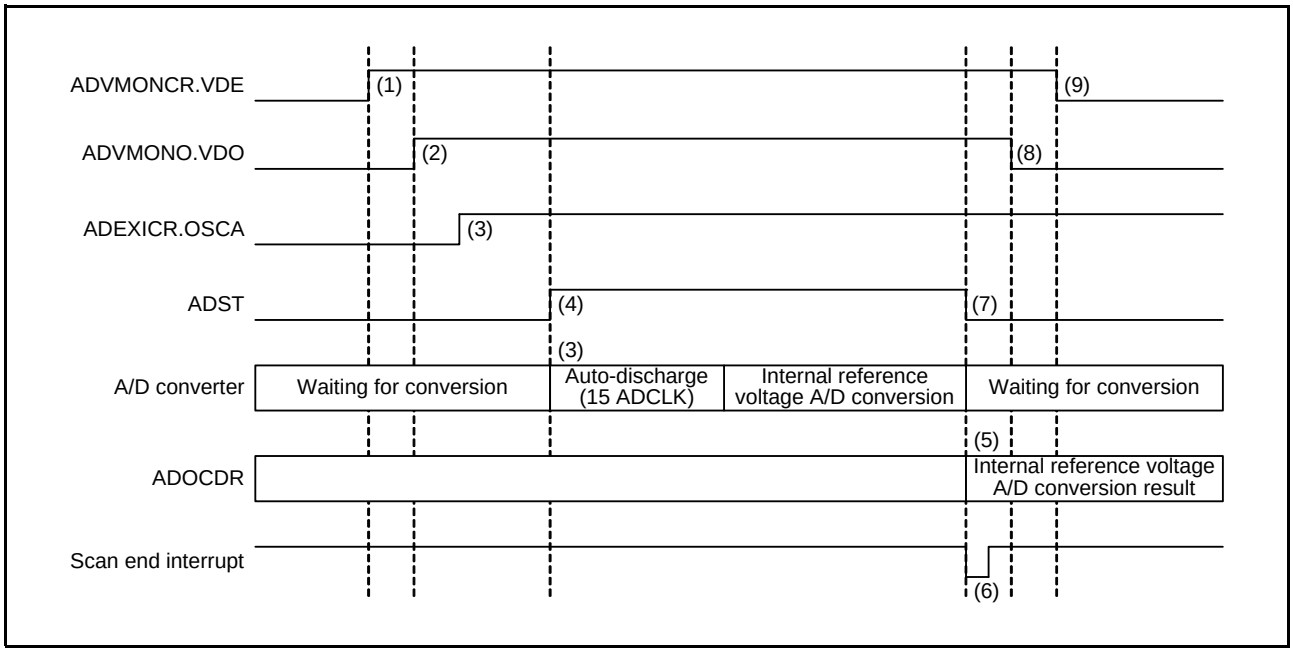


Figure 38.38 Example of Internal Reference Voltage Monitor Output A/D Conversion

38.4 Interrupt Sources and DTC/DMA Transfer Requests

38.4.1 Interrupt Requests

The 12-bit A/D converter can generate scan end interrupt requests S12ADI, S12GBADI, and S12GCADI to the CPU. The module also can generate the S12CMPAI or S12CMPBI interrupt which is the request of interrupting 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 interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a S12GBADI 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/DMA transfer can be triggered when an S12ADI, S12GBADI, or S12GCADI interrupt is generated. Using an S12ADI, S12GBADI, or S12GCADI 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, refer to section 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa).

38.4.2 Scan Complete Event Output to ELC

The ELC can set up linked operation of a module specified in advance by using the S12ADI interrupt request signal as an event signal.

The S12GBADI interrupt, S12CMPAI interrupt, and S12CMPBI 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.

38.5 Allowable Impedance of Signal Source

Figure 38.39 shows an equivalent circuit of an analog input pin and an external sensor.
To perform A/D conversion accurately, the internal capacitor (Cs) must be fully charged within the sampling time. If the impedance (R0) of the signal source is high and it takes time to charge Cs, extend the sampling time with the ADSSTRn register. Conversely, if R0 is small, the sampling time can be shortened. Refer to the electrical characteristics for the permissible signal source impedance under various operating conditions.
When converting only a single pin input in single scan mode, the influence of R0 can be ignored because the input load becomes practically only the internal input resistor (Rs) by connecting an external high-capacity capacitor (C). However, because a low-pass filter is formed by R0 and C, it may not possible to follow the analog signal that changes at high speed. Insert a low-impedance buffer when converting high speed analog signals or when converting multiple pins in scan mode.

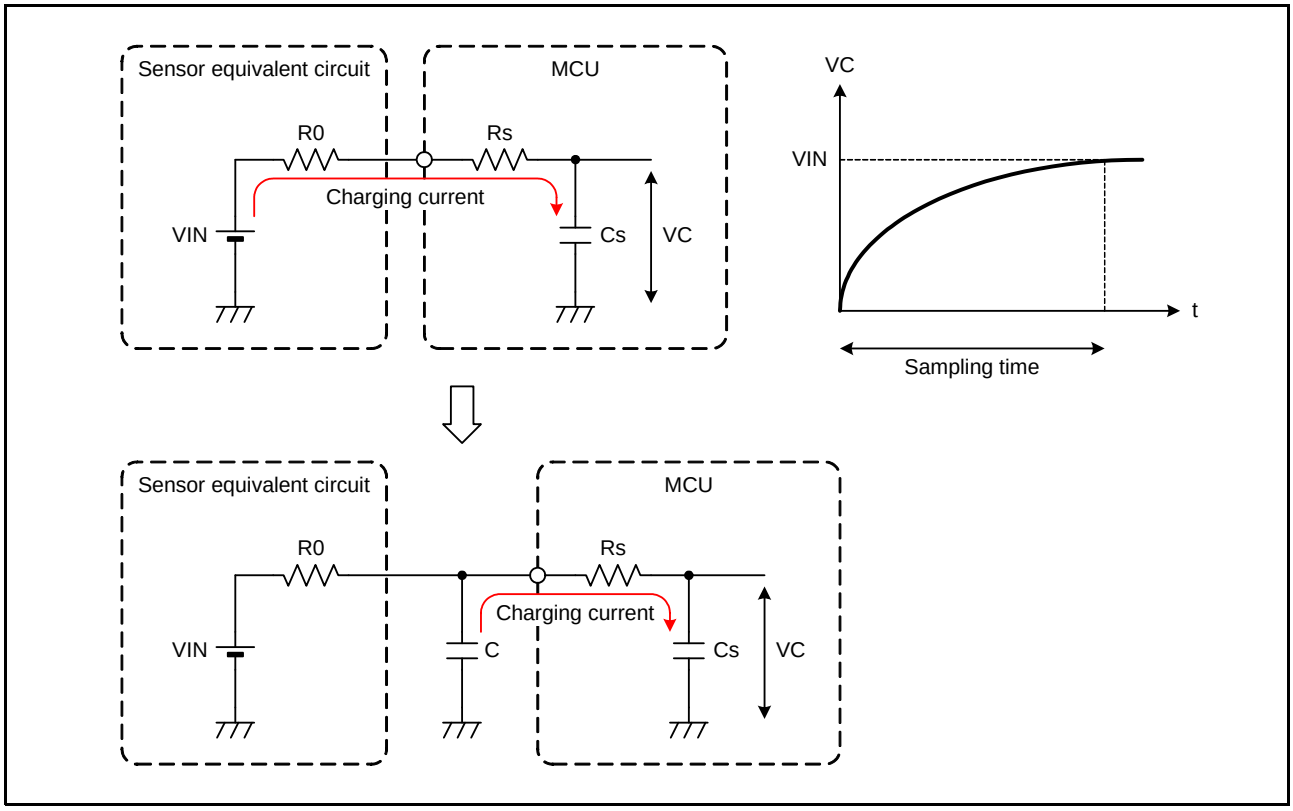


Figure 38.39 Equivalent Circuit of Analog Input Pin and External Sensor

38.6 Usage Notes

38.6.1 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.

38.6.2 Notes on Stopping A/D Conversion

38.6.2.1 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 38.40.

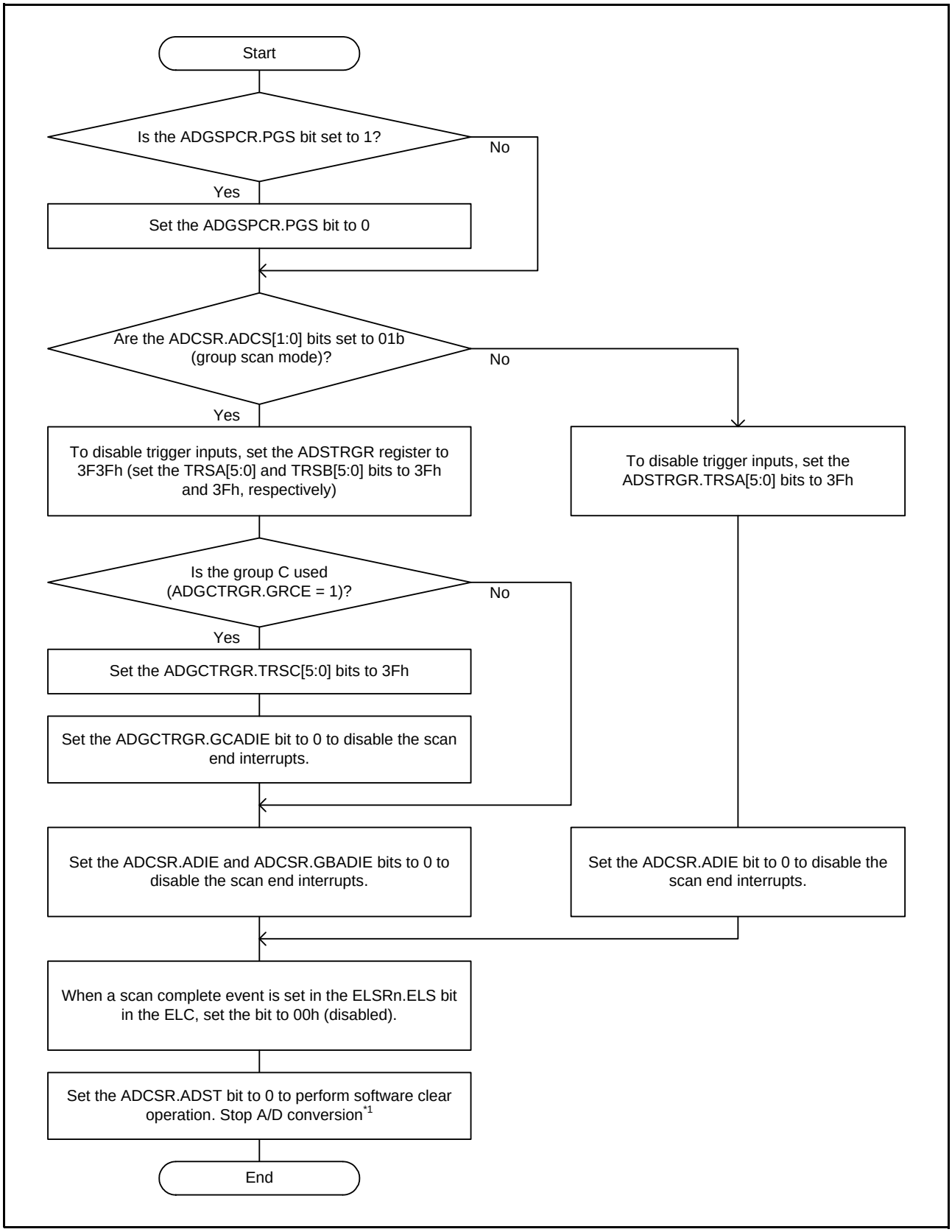


Figure 38.40 Procedure for Clear Operation by Software Through the ADCSR.ADST Bit

38.6.2.2 Notes on Modes and Status Flags

The following values should be initialized or reset as described below: the voltage status used in self-diagnosis, the odd-even determination for the double-trigger mode, the monitor flags for the comparison function.

- Re-set 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 ADCSR.DBLE bit from 0 to 1.
- Set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0, which clears the monitor flags for the comparison function (MONCMPA, MONCMPB, MONCOMB).

38.6.3 A/D Conversion Restarting Timing and Termination Timing

Starting operation of the analog blocks of the 12-bit A/D converter after setting the ADCSR.ADST bit to 1 takes up to six cycles of ADCLK. Stopping operation of the analog blocks of the 12-bit A/D converter by setting the ADCSR.ADST bit to 0 for a forced stop of the converter takes up to two cycles of ADCLK.
A period from forcible termination to idling of analog blocks takes a maximum of 1 PCLKB + 1 ADCLK cycles when PCLKB to ADCLK frequency ratio is 1:2 (ADCLK is faster).

38.6.4 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.

38.6.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register. The 12-bit A/D converter is stopped after released from the reset state. 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. To stop the operation of the 12-bit A/D converter using module stop function, set the ADVMONCR.VDE and ADVMONO.VDO bits to 0.
For details, refer to section 11, Low Power Consumption.

38.6.6 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 38.40 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.

38.6.7 Notes on Canceling Software Standby Mode

After releasing the A/D converter from software standby, wait for the oscillation stabilization flag for the main clock or PLL clock to become 1, wait for at least a further 1 µs after that, and then start A/D conversion.
For details, refer to section 11, Low Power Consumption.

38.6.8 Notes on Using an External Bus

A/D conversion during access to the external bus may reduce the precision.
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.

38.6.9 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs).
This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs/Rp

38.6.10 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
  Set the voltage within the range from AVSS0 to AVCC0 when applying voltages to analog pins AN000 to AN023.
- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)
  The following condition should be satisfied: AVSS0 = VREFL0 = VSS. When performing A/D conversion of analog input pins AN016 to AN023, the following condition should be satisfied: AVCC0 = VREFH0 = VCC.
  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 38.41, and connection should be made so that the following condition is satisfied at the supply side: AVSS0 = VREFL0 = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied:
  AVCC0 = VCC and AVSS0 = VSS

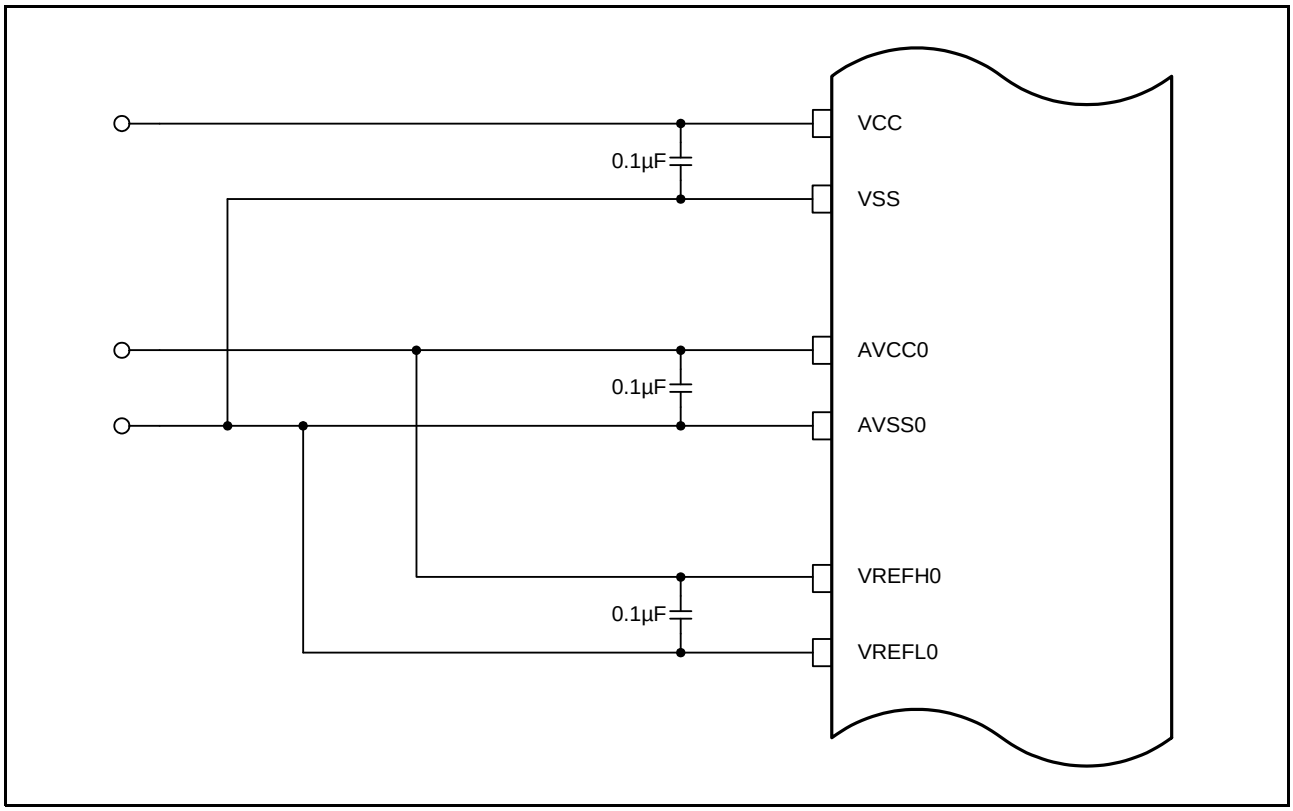


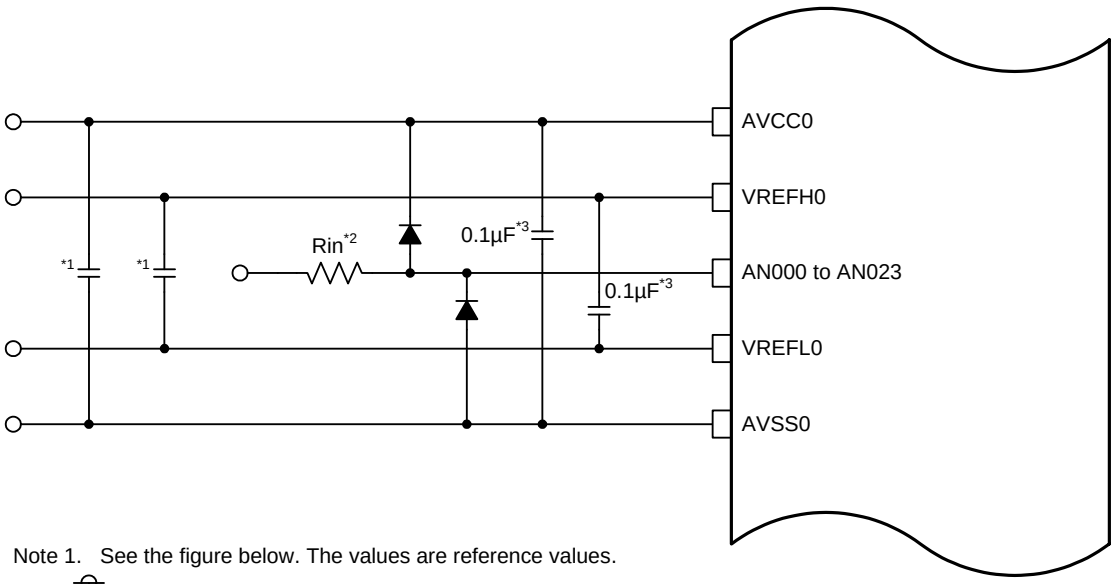
Figure 38.41 Example for Connecting Power Supply Pins

38.6.11 Notes on Board Design

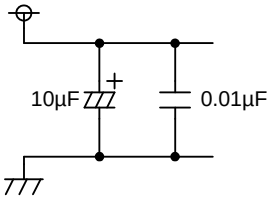
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN023), reference power supply pin (VREFH0), reference ground pin (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

38.6.12 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN023) from being destroyed by abnormal voltage such as excessive surge, capacitors should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and protection circuits should be connected to protect the above analog input pins as shown Figure 38.42.



Note 1. See the figure below. The values are reference values.



Note2. Rin: Signal source impedance
Note3. Place the capacitors to be placed between the power supply pins AVCC0 and AVSS0 and between VREFH0 and VREFL0 as close to the pins as possible to improve the precision of A/D conversion.

Figure 38.42 Sample Protection Circuit for Analog Inputs

39. 12-Bit D/A Converter (R12DAb)

39.1 Overview

This MCU includes two channels of 12-bit D/A converter.
Table 39.1 lists the specifications of the 12-bit D/A converter and Figure 39.1 shows a block diagram of the 12-bit D/A converter.

Table 39.1 Specifications of 12-Bit D/A Converter

Item	Specifications
Resolution	12 bits
Output channels	Two channels
Measure against mutual interference between analog modules	Measure against interference between D/A and A/D conversion D/A converted data update timing is controlled by the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter. Therefore, the degradation of A/D conversion accuracy due to interference is reduced by controlling the timing in which the 12-bit D/A converter inrush current occurs, with the enable signal.
Low power consumption function	Module stop state can be set.
Event link function (input)	DA0 conversion can be started when an event signal is input.
Destination Selection	Outputs to the external pin and to the comparator C are separately controllable.

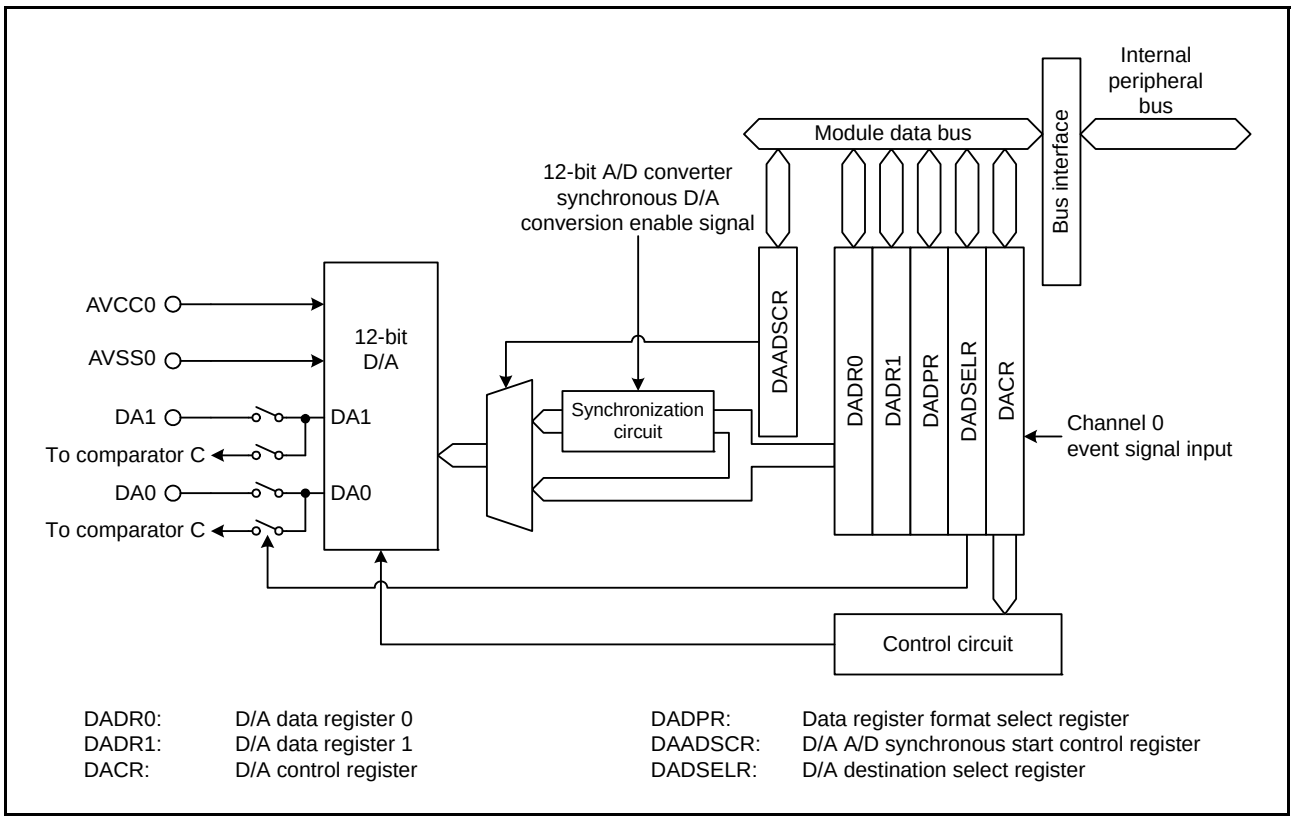


Figure 39.1 Block Diagram of 12-Bit D/A Converter

Table 39.2 lists the pin configuration of the 12-bit D/A converter.

Table 39.2 Pin Configuration of 12-Bit D/A Converter

Pin Name	I/O	Function
AVCC0	Input	Analog power supply pin
AVSS0	Input	Analog ground pin
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

39.2 Register Descriptions

39.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DA.DADR0 0008 8040h, DA.DADR1 0008 8042h

• DADPR.DPSEL bit = 0 (data is right-justified)

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

• DADPR.DPSEL bit = 1 (data is left-justified)

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

The DADRm register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output from the D/A converter.

12-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits “—” are read as 0. The write value should be 0.

39.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 8044h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	DAE	D/A Enable*1	0: D/A conversion of channels 0 and 1 is controlled individually. 1: D/A conversion of channels 0 and 1 is enabled collectively.	R/W
b6	DAOE0	D/A Output Enable 0	0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.	R/W
b7	DAOE1	D/A Output Enable 1	0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.	R/W

Note 1. This bit controls D/A conversion in combination with the DAOEm bit (m = 0, 1). The DAOEm bit controls output of the results of conversion. For details, see Table 39.3.

Table 39.3 Controls of D/A Conversion

b5	b7	b6	
DAE	**DAOE1**	**DAOE0**	**Description**
0	0	0	D/A conversion and analog output pins (DA0, DA1) are disabled.*1
		1	D/A conversion of channel 0 is enabled. D/A conversion of channel 1 is disabled. Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1
	1	0	D/A conversion of channel 0 is disabled. D/A conversion of channel 1 is enabled. Analog output of channel 0 (DA0) is disabled.*1 Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled.
1	x	x	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled collectively.

x: Don't care

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

This register should be set when the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled) while the 12-bit A/D converter is halted (the ADCSR.ADST bit is 0). At that time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.
Select the destination for each channel by using the DADSELR register before setting this register.

DAE Bit (D/A Enable)

The DAE bit controls D/A conversion and analog output in combination with the DAOEm bit (m = 0, 1).
When the measure against an interference between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAE bit while the ADCSR.ADST bit in the 12-bit A/D converter is 0. At that time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.

DAOEm Bit (D/A Output Enable m) (m = 0, 1)

The DAOEm bit controls D/A conversion and analog output in combination with the DAE bit.

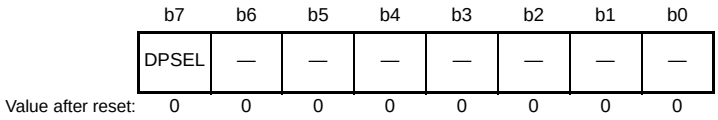
When both the DAOEm bit and DAE bit are 0, D/A conversion of channel m is not done and no conversion result is output.

When the measure against an interference between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAOEi bit while the ADCSR.ADST bit in the 12-bit A/D converter is 0. At that time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.

The event link function can be used to set the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by setting the ELSR16 register of the ELC occurs, and output of the D/A conversion results starts.

39.2.3 Data Register Format Select Register (DADPR)

Address(es): DA.DADPR 0008 8045h



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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	Format Select	0: Data is right-justified. 1: Data is left-justified.	R/W

39.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DA.DAADSCR 0008 8046h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: 12-bit D/A converter operation does not synchronize with 12-bit A/D converter operation. (measure against interference between D/A and A/D conversion is disabled) 1: 12-bit D/A converter operation synchronizes with 12-bit A/D converter operation. (measure against interference between D/A and A/D conversion is enabled)	R/W

As a measure against interference between D/A and A/D conversion, the DAADSCR register selects whether or not the timing for starting 12-bit D/A conversion is synchronized with the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter.
This register should be set while the 12-bit A/D converter is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 12-bit A/D converter trigger).

DAADST Bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value (m = 0, 1) to be converted into analog data at any time.
Setting the DAADST bit to 1 allows synchronous D/A conversion with the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter. Therefore, even if the DADRm register value is modified, D/A conversion does not start until the 12-bit A/D converter completes A/D conversion.
Set this bit while the ADCSR.ADST bit is set to 0. At this time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.
The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR16 register of the ELC. The setting of the DAADST bit is common to channels 0 and 1 of the 12-bit D/A converter.

39.2.5 D/A Destination Select Register (DADSELR)

Address(es): DA.DADSELR 0008 8049h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OUTREF1	OUTREF0	OUTDA1	OUTDA0
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	OUTDA0	DA0 pin output enable	0: Does not output the channel 0 analog voltage from the DA0 pin 1: Output the channel 0 analog voltage from the DA0 pin	R/W
b1	OUTDA1	DA1 pin output enable	0: Does not output the channel 1 analog voltage from the DA1 pin 1: Output the channel 1 analog voltage from the DA1 pin	R/W
b2	OUTREF0	Reference voltage 0 output enable	0: Does not use the channel 0 analog voltage as a reference voltage for the comparator C 1: Use the channel 0 analog voltage as a reference voltage for the comparator C	R/W
b3	OUTREF1	Reference voltage 1 output enable	0: Does not use the channel 1 analog voltage as a reference voltage for the comparator C 1: Use the channel 1 analog voltage as a reference voltage for the comparator C	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

The DADSELER register is used to set the destinations of the analog voltage generated by the D/A converter. Select such destinations with this register before the output is enabled with the DACR register.

39.3 Operation

The 12-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DACR.DAOEm bit (m = 0, 1) is set to 1, D/A converter is enabled and the conversion result is output. An operation example of D/A conversion on channel 0 is shown below. Figure 39.2 shows the timing of this operation.

(1) Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
(2) Select the destination of the D/A converter analog voltage by using the DADSELR register.
(3) Set the DACR.DAOE0 bit to 1 to start D/A conversion. The DA0 output settles to the voltage corresponding to the setting value after the conversion time tDCONV has elapsed. The DA0 output voltage is held at this level until the DADR0 register is updated or the DAOE0 bit is set to 0. The output voltage (reference) is expressed by the following formula:

$$\frac{\text{Value of DADRm register}}{4096} \times \text{AVCC0}$$

(4) When the DADR0 register is updated, the conversion starts. The DA0 output settles at the new output voltage after the conversion time tDCONV has elapsed.
When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), it takes a maximum of one A/D conversion time for D/A conversion to start. When ADCLK is faster than the peripheral module clock, it may take longer than one A/D conversion time.
(5) When the DAOE0 bit is set to 0, analog output is disabled.

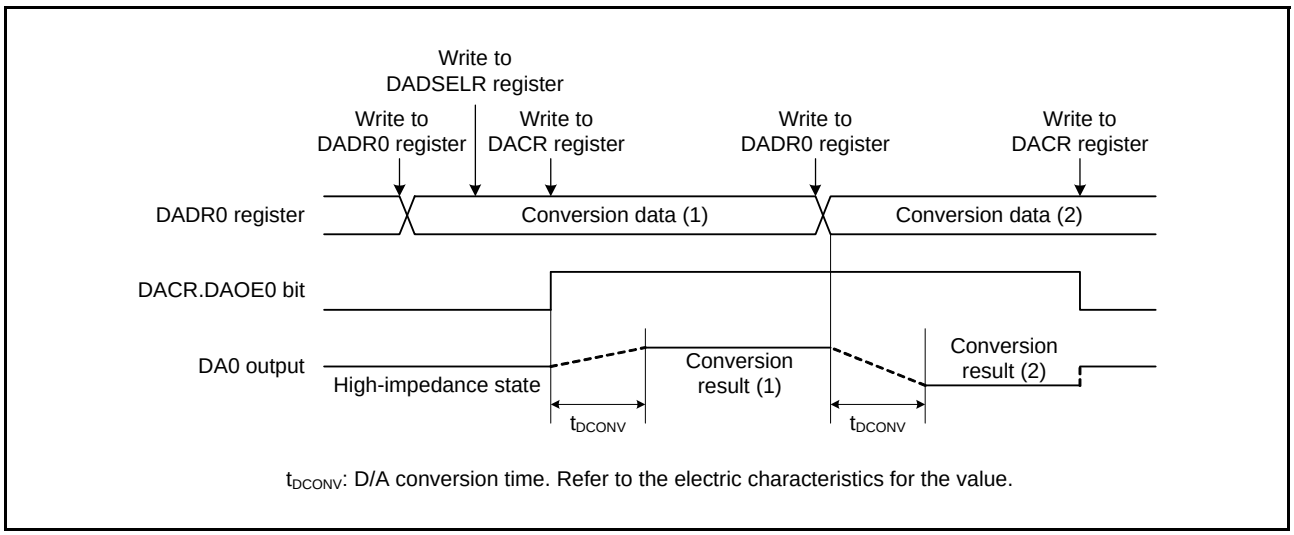


Figure 39.2 Example of 12-Bit D/A Converter Operation

39.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, an inrush current occurs to the 12-bit D/A converter. Since the same analog power supply is shared by the 12-bit D/A converter and 12-bit A/D converter, the inrush current may interfere with the proper operation of the 12-bit A/D converter.

With the DAADSCR.DAADST bit being 1, even if the DADRm register data (m = 0, 1) is modified during 12-bit A/D converter operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. It takes a maximum of one A/D conversion time for the DADRm register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADRm register value does not correspond to the analog output value.

When this function is enabled, it is impossible to check by any software means whether the DADRm register value has been D/A converted or not.

Even with the DAADSCR.DAADST bit being 1, when the DADRm register data is modified while the 12-bit A/D converter is halted, D/A conversion starts in one PCLKB cycle.

Figure 39.3 shows an example of channel 0 D/A conversion, in which the 12-bit D/A converter operates synchronously with the 12-bit A/D converter.

(1) Confirm that the 12-bit A/D converter is halted. Set the DAADSCR.DAADST bit to 1.
(2) Confirm that the 12-bit A/D converter is halted. Set the DACR.DAOE0 bit to 1.
(3) Set the DADR0 register. When ADCLK is faster than the peripheral module clock, it may take longer than one A/D conversion time.

- If the 12-bit A/D conversion is halted (ADCSR.ADST bit = 0) when the DADR0 register is modified, D/A conversion starts in one PCLKB cycle.
- If the 12-bit A/D conversion is in progress (ADCSR.ADST bit = 1) when the DADR0 register is modified, D/A conversion starts upon A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update may not be converted.

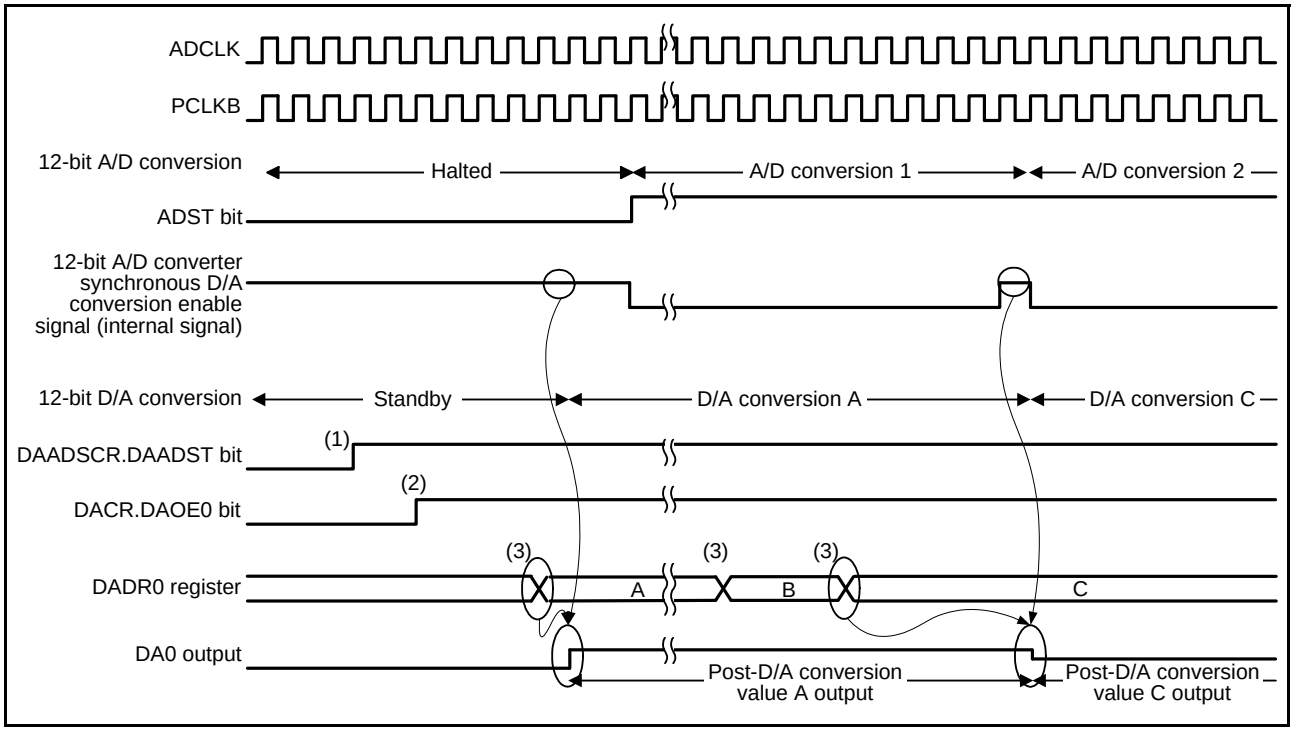


Figure 39.3 Example of Conversion When the 12-Bit D/A Converter is Synchronized with the 12-Bit A/D Converter

When ADCLK is faster than PCLKB, the 12-bit D/A converter may not be able to capture a 12-bit A/D converter synchronous D/A conversion enable signal for one ADCLK cycle which is output between A/D conversion 1 and A/D conversion 2. Figure 39.4 shows example when the 12-bit D/A converter cannot capture the 12-bit A/D converter synchronous D/A conversion enable signal. In this case, the DA0 output is held at the level of the post-D/A conversion value A.

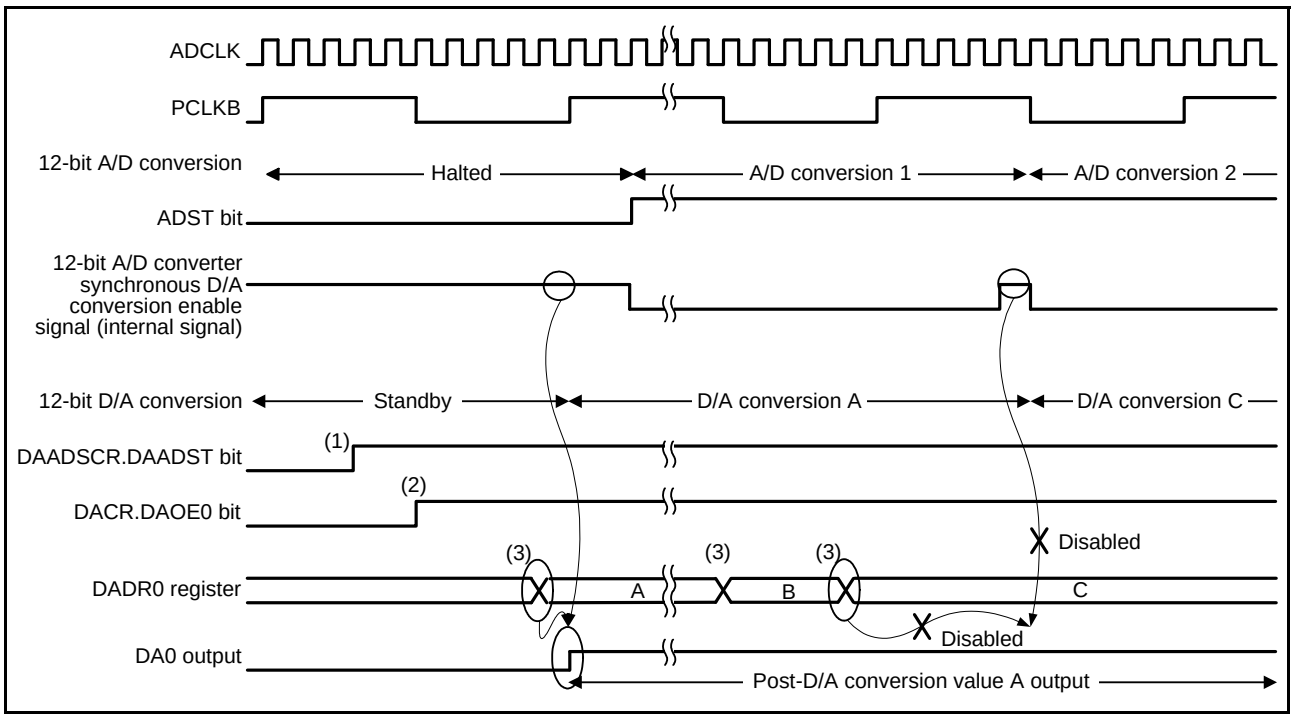


Figure 39.4 Example When the 12-Bit D/A Converter Cannot Capture the 12-Bit A/D Converter Synchronous D/A Conversion Enable Signal

39.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

(1) Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
(2) Select the destination of the D/A converter analog voltage by using the DADSELR register.
(3) Set the bit value of the ELSR16 setting event signal to link the ELSR16 register of the ELC.
(4) Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
(5) Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
(6) Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop event link operation of 12-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

39.5 Usage Notes on Event Link Operation

(1) When the event link function is used, set the DACR.DAE bit to 0.
(2) When the event specified by the ELSR16 register is generated while the write cycle is performed to the DACR.DAOE0 bit, the write cycle is stopped, and the setting to 1 by the generated event takes precedence.
(3) Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 as the measure against an interfere between D/A and A/D conversions.

39.6 Usage Notes

39.6.1 Module Stop Function Setting

Operation of the 12-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

39.6.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

39.6.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

39.6.4 Note on Entering Deep Software Standby Mode

When the MCU enters deep software standby mode with D/A conversion enabled, the outputs of the D/A converter are placed in a high impedance state.

39.6.5 Note on Usage When Measure against Interference between D/A and A/D Conversion is Enabled

When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), do not place the 12-bit A/D converter in the module stop state. It may halt D/A conversion in addition to A/D conversion.

39.6.6 Note on Using the D/A Converter Analog Voltage as a Reference Voltage for the Comparator C

When setting the DADSELR.OUTREFn bit (n = 0, 1) to 1, refer to section 41.4.4, Setting the D/A Converter.

39.6.7 Note on Enabling the Outputs Both to the DAn pin (n = 0, 1) and Comparator C

When the DADSELR.OUTDAn and OUTREFn bits are set to 1 simultaneously, the characteristics specified in section 45.6, D/A Conversion Characteristics may not be satisfied due to the influence of the load connected to the DAn pin. A full evaluation is recommended before use when both outputs are to be enabled.

40. Temperature Sensor (TEMPS)

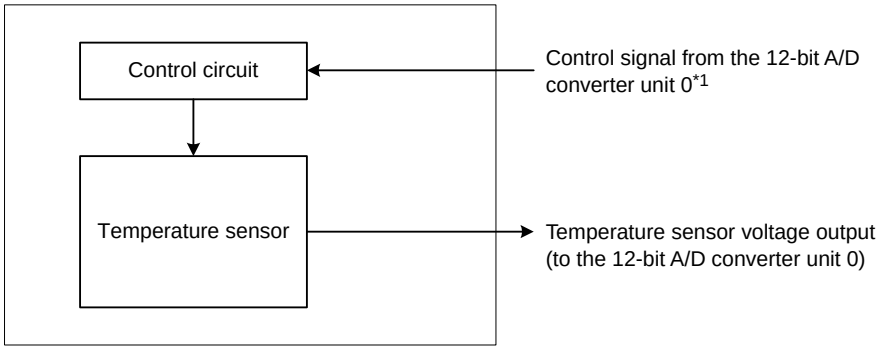
40.1 Overview

This MCU includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The 12-bit A/D converter unit 0 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 40.1 lists the specifications of the temperature sensor, and Figure 40.1 shows a block diagram of the temperature sensor.

Table 40.1 Specifications of Temperature Sensor

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter unit 0.
Temperature Sensor Calibration Data	Reference data measured for each chip at factory shipment is stored.



Note 1. When any of TSSA or TSSB bit in the ADEXICR register, or ADGCEXCR.TSSC bit for 12-bit A/D converter is set to 1, the temperature sensor is activated.

Figure 40.1 Block Diagram of Temperature Sensor

40.2 Register Descriptions

40.2.1 Temperature Sensor Calibration Data Register (TSCDR)

Address(es): TEMPS.TSCDR 007F B17Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each chip											

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 0 to convert the voltage output by the temperature sensor under the condition of Ta = Tj = 128°C and AVCC0 = 5 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 = 5 × value of the TSCDR register/4096 [V]

Note that the AVCC0 voltage does not affect voltage V1.

40.3 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 0. The temperature around the MCU can be obtained by converting the value into the temperature.

40.3.1 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 0 to measure the voltage V1 output by the temperature sensor at temperature T1. Again, using the 12-bit A/D converter unit 0, 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 45.49 of section 45, Electrical Characteristics, use the 12-bit A/D converter unit 0 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 45.49 ÷ 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 AVCC0 = 5 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},

$V1 = 5 \times CAL_{128}/4096$ [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 AVCC0 = 5 V (V)
Slope: Temperature slope given in Table 45.49 ÷ 1000 (V/°C)

Error in the measured temperature (the range of variation is 3σ) is shown in Figure 40.2.

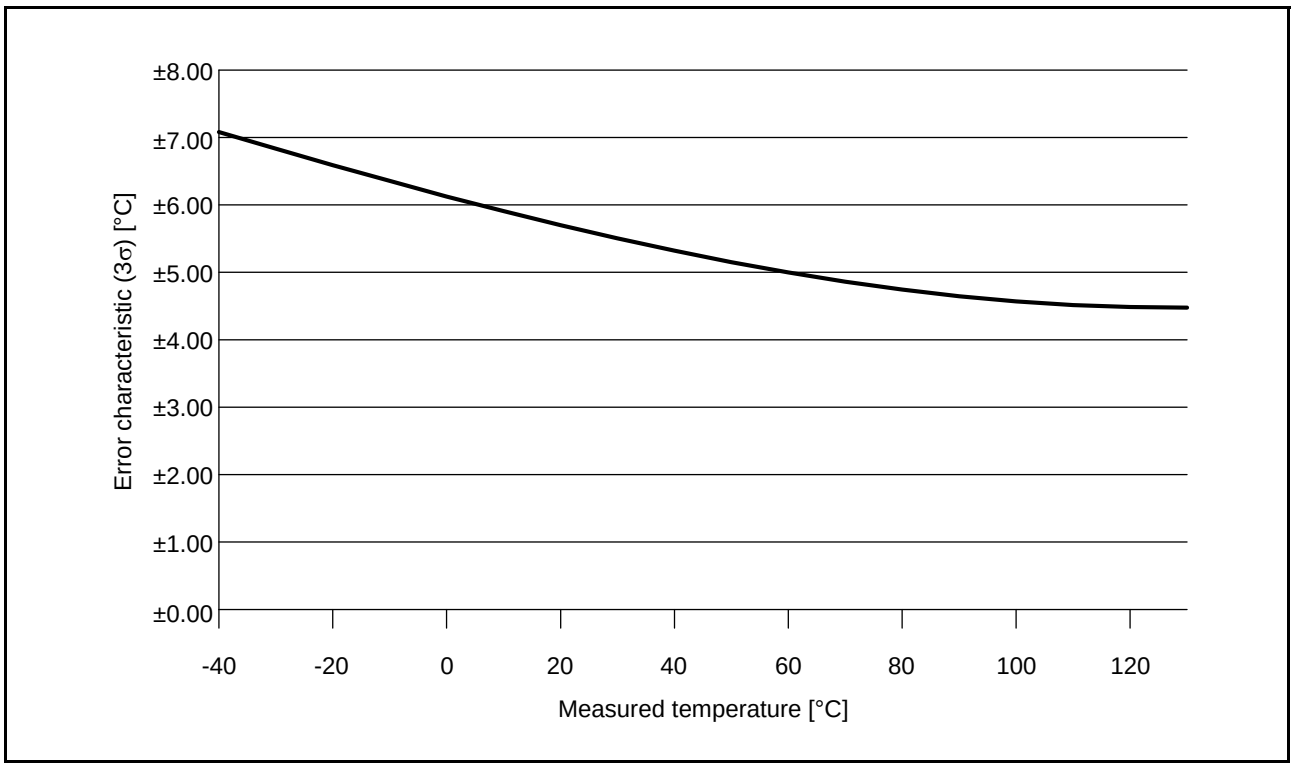


Figure 40.2 Error in the Measured Temperature

40.3.2 Setting of 12-Bit A/D Converter Unit 0

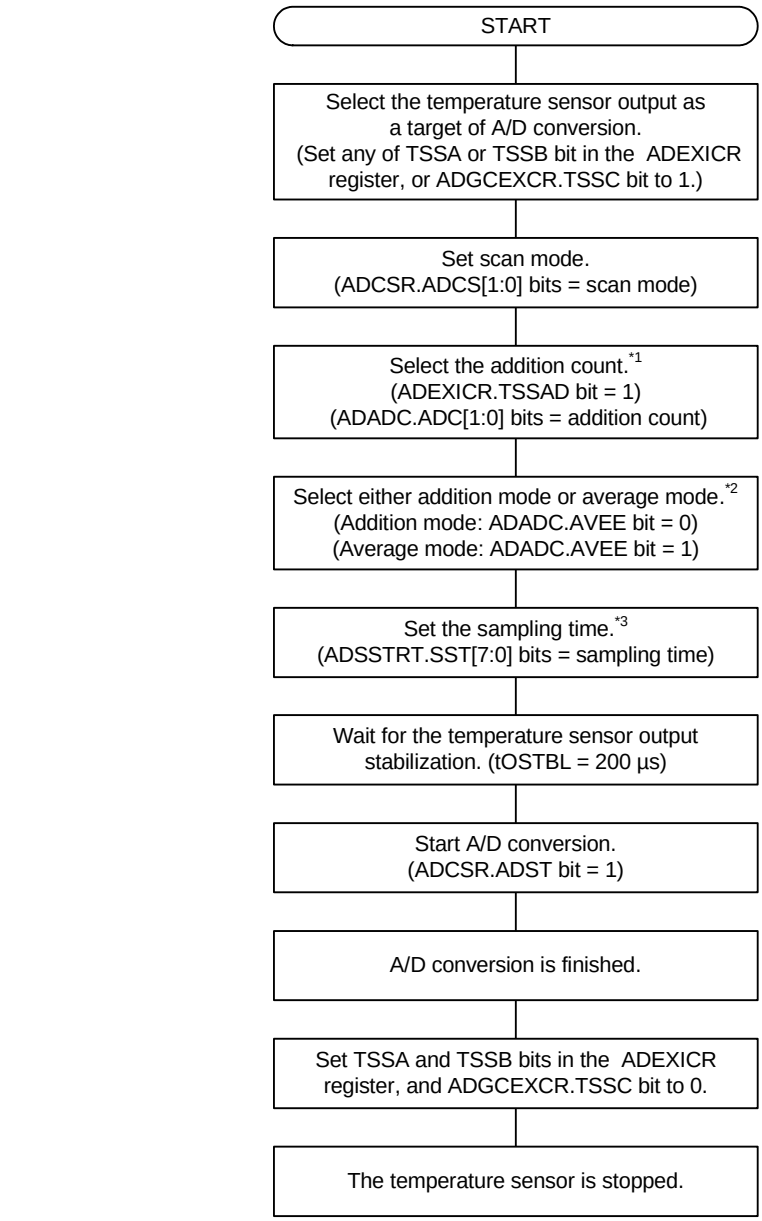
For A/D conversion of temperature sensor output voltages, 12-bit A/D converter unit 0 registers should be set as follows.

- 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, or ADGCEXCR.TSSC) 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]).
  Set the bits to the single scan mode or the group scan mode.
- 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 0
  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 27 cycles of ADCLK. To change the setting of sampling time from 27 cycles of ADCLK, set the A/D sampling state register T (ADSSTRT) while the ADCSR.ADST bit is 0. Refer to section 38.2.19, A/D Sampling State Register n (ADSSTRn) (n = 0 to 15, L, T, O) for the setting range of the sampling time. Set the sampling time to satisfy the specifications described in section 45, Electrical Characteristics.

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 40.3.3, Procedure for Using the Temperature Sensor.

40.3.3 Procedure for Using the Temperature Sensor

Figure 40.3 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 27 cycles of ADCLK. If the sampling time is not to be changed, this setting is not required.

Figure 40.3 Procedure for Using the Temperature Sensor

40.3.4 Timing of A/D Conversion of Temperature Sensor Output

Figure 40.4 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 40.2.
When setting the ADST bit to 1 to perform the A/D conversion of the temperature sensor, auto-discharge is performed before sampling. The A/D conversion time of the temperature sensor includes the time for the auto discharge.

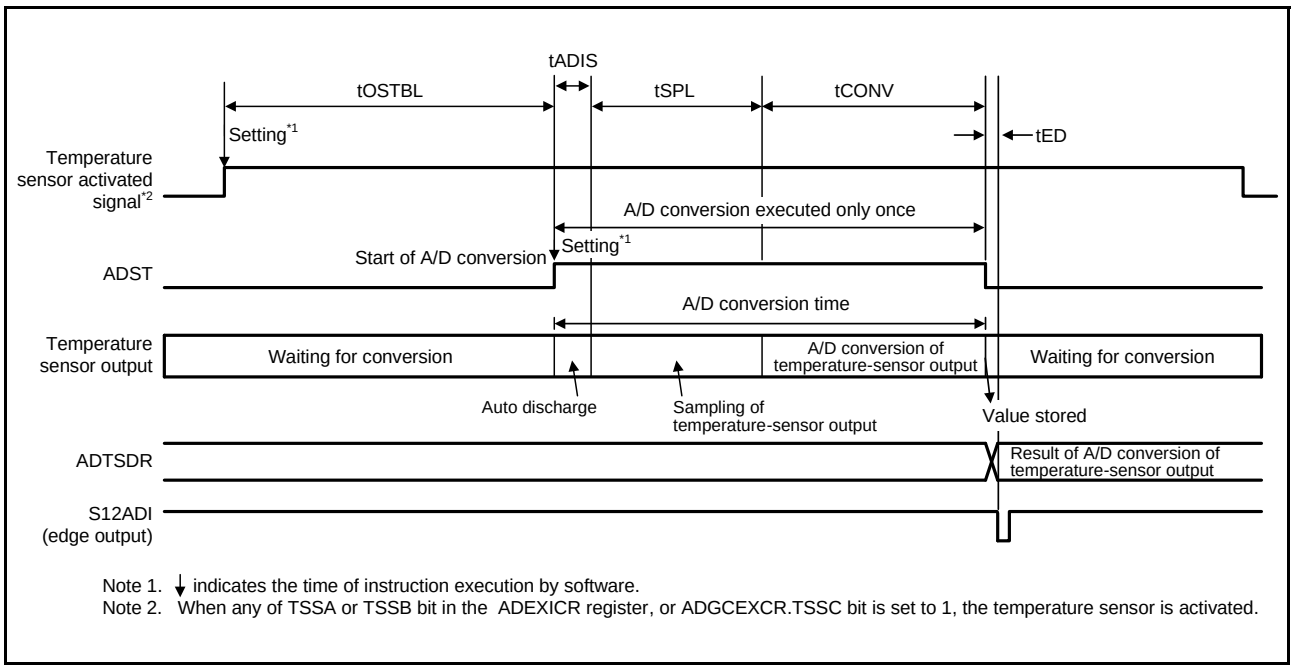


Figure 40.4 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

Table 40.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Temperature-sensor output stabilization wait time	tOSTBL	200 µs (min)
Auto discharge time	tADIS	15 × tC(ADCLK)
12-bit A/D converter unit 0 input sampling time	tSPL	ADSSTRT setting × tC(ADCLK)
A/D conversion time	tCONV	Refer to Table 38.15, Times for Conversion During Scanning (in Numbers of Cycles of ADCLK and PCLKB) in section 38.3.7, Analog Input Sampling Time and Scan Conversion Time.
Scan conversion end delay time	tED	Refer to Table 38.15, Times for Conversion During Scanning (in Numbers of Cycles of ADCLK and PCLKB) in section 38.3.7, Analog Input Sampling Time and Scan Conversion Time.

40.4 Usage Note

40.4.1 Activating the Temperature Sensor

The temperature sensor is controlled by the registers in the 12-bit A/D converter unit 0 (S12AD). To activate the temperature sensor, set any of TSSA or TSSB bit in the ADEXICR register, or ADGCEXCR.TSSC bit to 1 after releasing the S12AD from the module stop state. For details, refer to section 38, 12-Bit A/D Converter (S12ADH).

41. Comparator C (CMPC)

41.1 Overview

Comparator C compares a reference input voltage to an analog input voltage.
The comparison result can be read by software and output externally, and an interrupt request can be generated upon any changes to the comparison result.
The reference input voltage of comparator C can be selected from the input from the CVREFC0 to CVREFC3 pins, or the output from the on-chip D/A converter 0 or the on-chip D/A converter 1.
Table 41.1 lists the specification of comparator C, Figure 41.1 shows a block diagram of comparator C, and Table 41.2 shows a comparator C pin configuration.
In this section, “PCLK” is used to refer to PCLKB.

Table 41.1 Comparator C Specifications

Item	Specification
Number of channels	Four (comparator C0 to comparator C3)
Analog input voltages	Input voltage from the CMPCn0 pin (n = channel number)
Reference input voltage	Input voltage from the CVREFC0 to CVREFC3 pins, or output voltage from on-chip D/A converter 0 or on-chip D/A converter 1
Comparison result	The comparison result can be output externally.
Digital filter function	• One of three sampling periods can be selected. • The filter function can also be disabled. • A noise-filtered signal can be used to generate the interrupt request output and event output to the ELC, and the signal can be used to read the comparison result via registers.
Interrupt request	• An interrupt request is generated upon detecting a valid edge of the comparison result. • A valid edge can be selected from a rising or a falling edge or both edges.
Low power consumption function	Module stop state can be set.

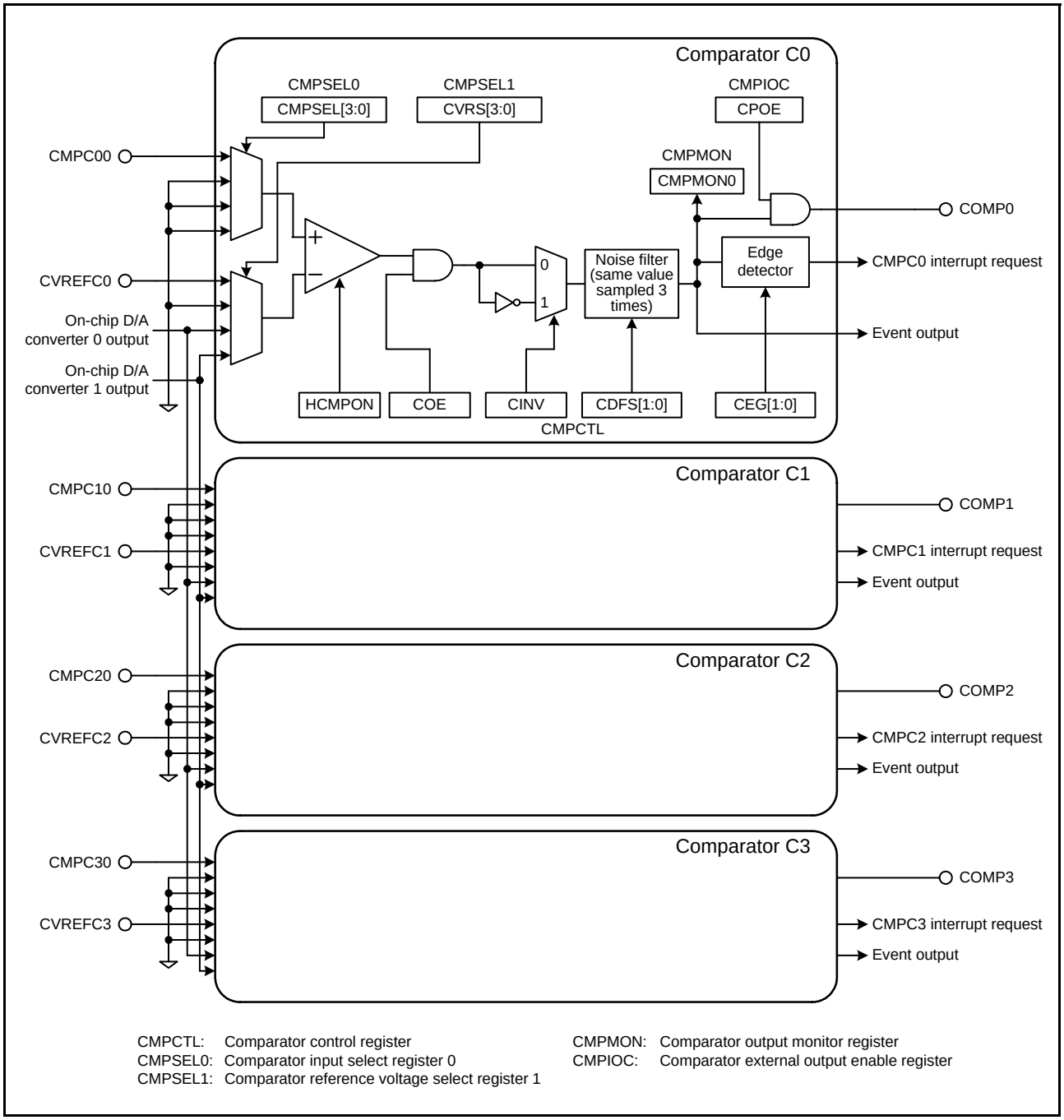


Figure 41.1 Block Diagram of Comparator C

Table 41.2 Comparator C Pin Configuration

Pin Name	I/O	Function
CMPC00	Input	Comparator C0 analog input pin
CMPC10	Input	Comparator C1 analog input pin
CMPC20	Input	Comparator C2 analog input pin
CMPC30	Input	Comparator C3 analog input pin
CVREFC0	Input	Comparator C0 reference input voltage pin
CVREFC1	Input	Comparator C1 reference input voltage pin
CVREFC2	Input	Comparator C2 reference input voltage pin
CVREFC3	Input	Comparator C3 reference input voltage pin
COMP0	Output	Comparator C0 comparison result output pin
COMP1	Output	Comparator C1 comparison result output pin
COMP2	Output	Comparator C2 comparison result output pin
COMP3	Output	Comparator C3 comparison result output pin

41.2 Register Descriptions

41.2.1 Comparator Control Register (CMPCTL)

Address(es): CMPC0.CMPCTL 000A 0C80h, CMPC1.CMPCTL 000A 0CA0h, CMPC2.CMPCTL 000A 0CC0h, CMPC3.CMPCTL 000A 0CE0h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG[1:0]		—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CINV	Comparator Output Polarity Select*1, *4	0: Comparator output not inverted 1: Comparator output inverted	R/W
b1	COE	Comparator Output Enable	0: Comparator output disabled (the output signal is fixed to 0) 1: Comparator output enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CEG[1:0]	Comparator Edge Interrupt Detection Select	b4 b3 0 0: Interrupt request is not generated. 0 1: Rising edge 1 0: Falling edge 1 1: Both edges	R/W
b6, b5	CDFS[1:0]	Noise Filter Sampling Select*1, *2, *4	b6 b5 0 0: Noise filter not used 0 1: Sampling frequency is PCLK/8. 1 0: Sampling frequency is PCLK/16. 1 1: Sampling frequency is PCLK/32.	R/W
b7	HCMPON	Comparator Operation Enable*3	0: Operation stopped (the output signal is fixed to 0) 1: Operation enabled (input to the comparator pins is enabled)	R/W

Note: Set this register before setting registers in the ELC when using event output.
Note 1. Rewrite the CDFS[1:0] and CINV bits only after disabling the comparator output (COE bit = 0).
Note 2. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), allow four sampling times to elapse until the filter output is updated, and then use the CMPCn interrupt request output and an event output.
Note 3. The operation stabilization wait time is required after enabling comparator operation (HCMPON bit = 1). As for the value, refer to section 45, Electrical Characteristics.
Note 4. Rewriting the CINV bit or CDFS[1:0] bits may generate a CMPCn interrupt request and event output. Before changing these bits, set the registers in the ELC so that the comparator output is not linked. After changing these bits, also set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

CEG[1:0] Bits (Comparator Edge Interrupt Detection Select)

These bits select which edge of comparator output signal is used to generate an interrupt request.

The valid edge is set for the signal after the comparator polarity is selected by the CINV bit and the filter is selected by CDFS[1:0] bits.

41.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): CMPC0.CMPSEL0 000A 0C84h, CMPC1.CMPSEL0 000A 0CA4h, CMPC2.CMPSEL0 000A 0CC4h, CMPC3.CMPSEL0 000A 0CE4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMPSEL[3:0]	Comparator Input Select*1, *2	• Comparator C0 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC00 selected Settings other than above are prohibited. • Comparator C1 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC10 selected Settings other than above are prohibited. • Comparator C2 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC20 selected Settings other than above are prohibited. • Comparator C3 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC30 selected 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

Note 1. Rewrite the CMPSEL[3:0] bits in the following procedure. Writing a value other than 0000b while the value of these bits is not 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CMPSEL[3:0] bits to 0000b.
(3) Set a new value to the CMPSEL[3:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection. As for the value, refer to section 45, Electrical Characteristics.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

Note 2. When using the event output, note that writing to these bits after the corresponding register in the ELC is set may trigger the output of an event signal.

41.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): CMPC0.CMPSEL1 000A 0C88h, CMPC1.CMPSEL1 000A 0CA8h, CMPC2.CMPSEL1 000A 0CC8h, CMPC3.CMPSEL1 000A 0CE8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CVRS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CVRS[3:0]	Reference Input Voltage Select*1, *2, *3	b3 b0 0 0 0 0: No input 0 0 0 1: On-chip D/A converter 1 output voltage selected as reference input voltage 0 0 1 0: On-chip D/A converter 0 output voltage selected as reference input voltage 1 0 0 0: Input voltage to the CVREFCn pin (n = 0 to 3) selected as reference input voltage 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

Note 1. When the on-chip D/A converter output voltage is used, set the D/A converter for generating comparator C reference voltage before enabling comparator operation (CMPCTL.HCMPON bit = 1). For details on setting the D/A converter, refer to section 39, 12-Bit D/A Converter (R12DAb).

Note 2. Rewrite the CVRS[3:0] bits in the following procedure. Be sure to set the CVRS[3:0] bits to 0000b before changing the set value. Direct rewriting from 0001b to 0010b or 0010b to 0001b, for example, will be ignored even if it is tried.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CVRS[3:0] bits to 0000b.
(3) Set a new value to the CVRS[3:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection. As for the value, refer to section 45, Electrical Characteristics.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

Note 3. When using the event output, note that writing to these bits after the corresponding register in the ELC is set may trigger the output of an event signal.

41.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): CMPC0.CMPMON 000A 0C8Ch, CMPC1.CMPMON 000A 0CACh, CMPC2.CMPMON 000A 0CCCh, CMPC3.CMPMON 000A 0CECh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPMON0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPMON0	Comparator Output Monitor Flag *1	0: Comparator output is 0. 1: Comparator output is 1.	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When comparator operation is enabled (CMPCTL.HCMPON and COE bits are 1) while the noise filter is disabled (CMPCTL.CDFS[1:0] bits are 00b), read the CMPMON0 bit twice and use the value only when the results match.

41.2.5 Comparator External Output Enable Register (CMPIOC)

Address(es): CMPC0.CMPIOC 000A 0C90h, CMPC1.CMPIOC 000A 0CB0h, CMPC2.CMPIOC 000A 0CD0h, CMPC3.CMPIOC 000A 0CF0h

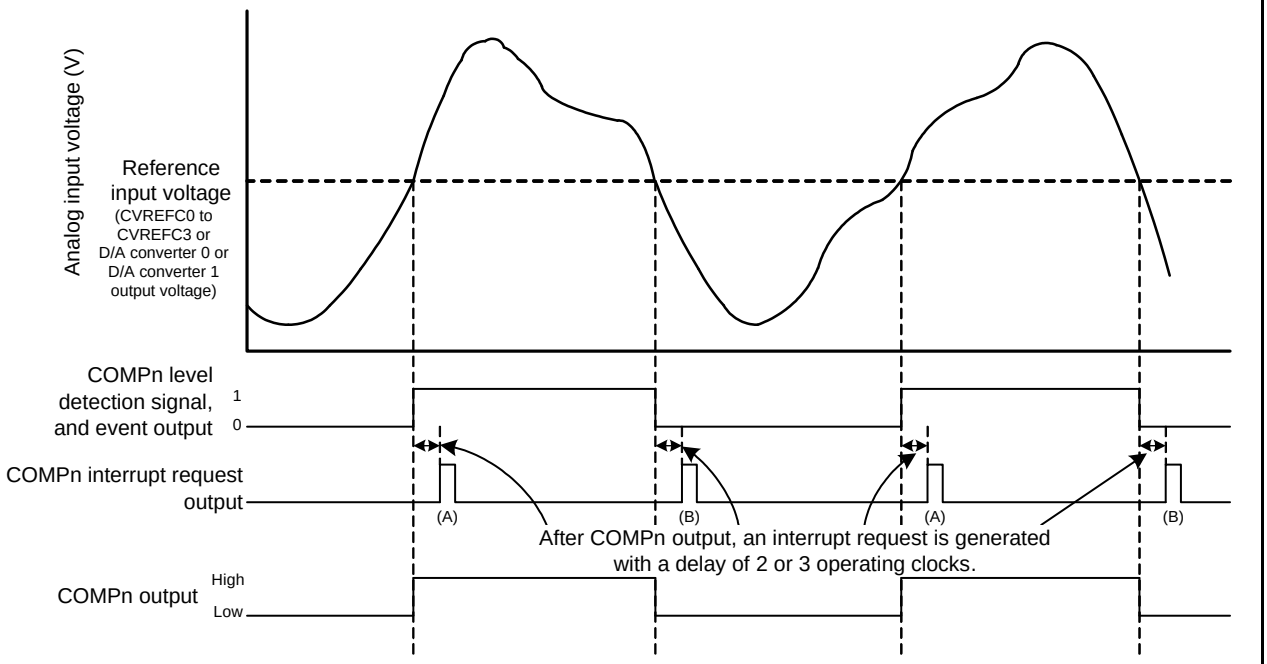
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPOE	External Pin Output Enable	Comparison result by the comparator is output to an external pin. 0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

41.3 Operation

41.3.1 Comparator Operation Example

Figure 41.2 shows an operation example of the comparator. The COMPn level detection signal (n = 0 to 3) becomes high when the analog input voltage is higher than the reference input voltage, and the COMPn level detection signal becomes low when the analog input voltage is lower than the reference input voltage (when the CMPCTL.CINV bit is 0). The COMPn level detection signal can also be used as event output signal. When the CPOE bit in the corresponding CMPIOC register is 1, the signal is output from the COMPn pin. Interrupt request is output in response to changes in the comparator output.



Note: The above diagram applies when the CMPIOC.CPOE bit is 1 (pin output enabled), the CMPCTL.CDFS[1:0] bits are 00b (filter not used), the CMPCTL.CEG[1:0] bits are 11b (both edges selected), and the CMPCTL.CINV bit is 0 (comparator output not inverted).

Figure 41.2 Comparator Operation Example (n = 0 to 3)

41.3.2 Noise Filter

Comparator C contains a noise filter. The sampling clock can be selected by the CMPCTL.CDFS[1:0] bits.
The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, that value is determined as the noise filter output at the next sampling clock.
Figure 41.3 shows the configuration of the noise filter and edge detector and Figure 41.4 shows an example of the comparator noise filter and interrupt operation.

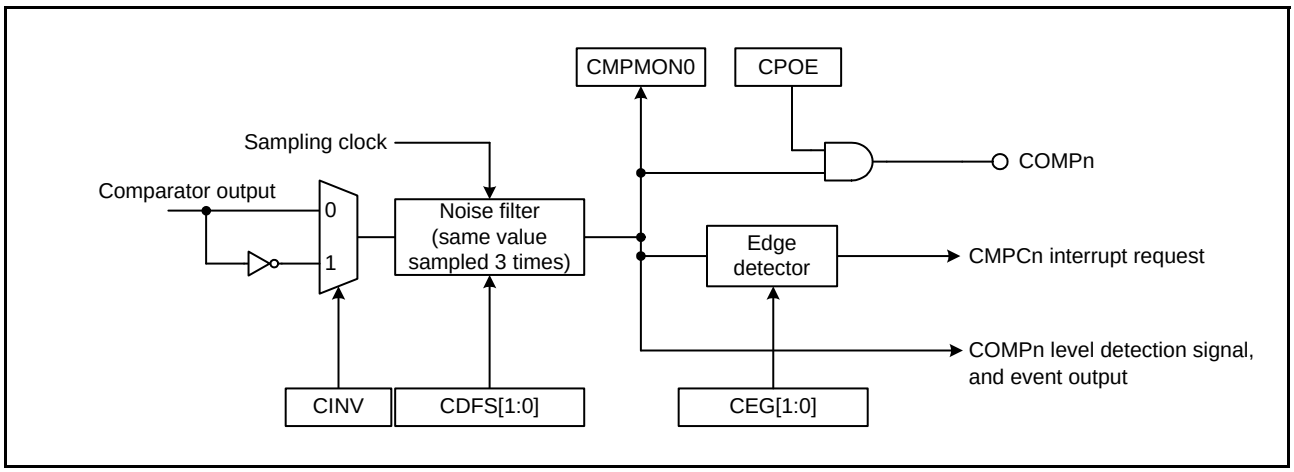
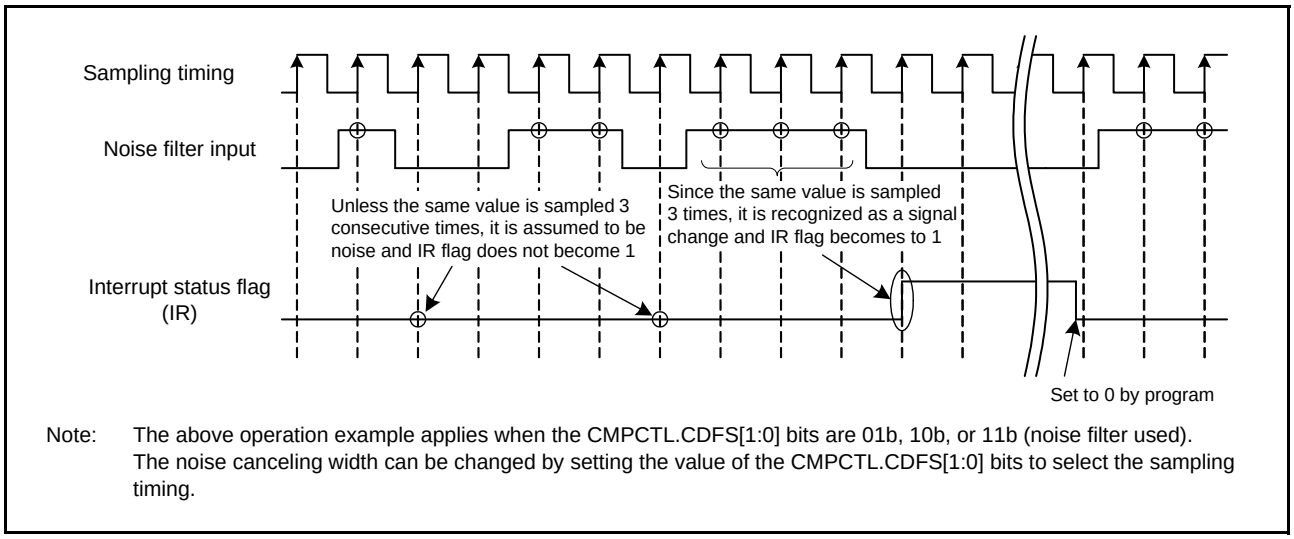


Figure 41.3 Noise Filter and Edge Detector Configuration (n = 0 to 3)



Note: The above operation example applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).
The noise canceling width can be changed by setting the value of the CMPCTL.CDFS[1:0] bits to select the sampling timing.

Figure 41.4 Noise Filter and Interrupt Operation Example

41.3.3 Interrupts

Comparator C generates an interrupt request upon detecting any changes in the comparison result.
When using the CMPCn interrupt, set at least one of bits CMPCTL.CEG[1:0] to 1 (to a value other than 00b (interrupt request is not generated)).
To use the CMPCn interrupt, use the following setting procedure. Note that steps (1), (2), and (3) can be set in any order.

(1) When using the on-chip D/A converter output voltage as the reference input voltage, set the D/A converter for generating comparator C reference voltage and enable operation.
(2) Set the CMPSEL0 or CMPSEL1 register to set the input of the comparator.
(3) Set the CMPCTL.CINV and CDFS[1:0] bits to select inversion or non-inversion processing and the sampling timing of the noise filter.
(4) Enable the edge for interrupt detection (set the CMPCTL.CEG[1:0] bits to a value other than 00b).
(5) Enable input of the comparator (set the CMPCTL.HCMPON bit to 1) and wait for the time until the comparator operation is stabilized. As for the value, refer to section 45, Electrical Characteristics.
(6) Enable output of the comparator (set the CMPCTL.COE bit to 1).

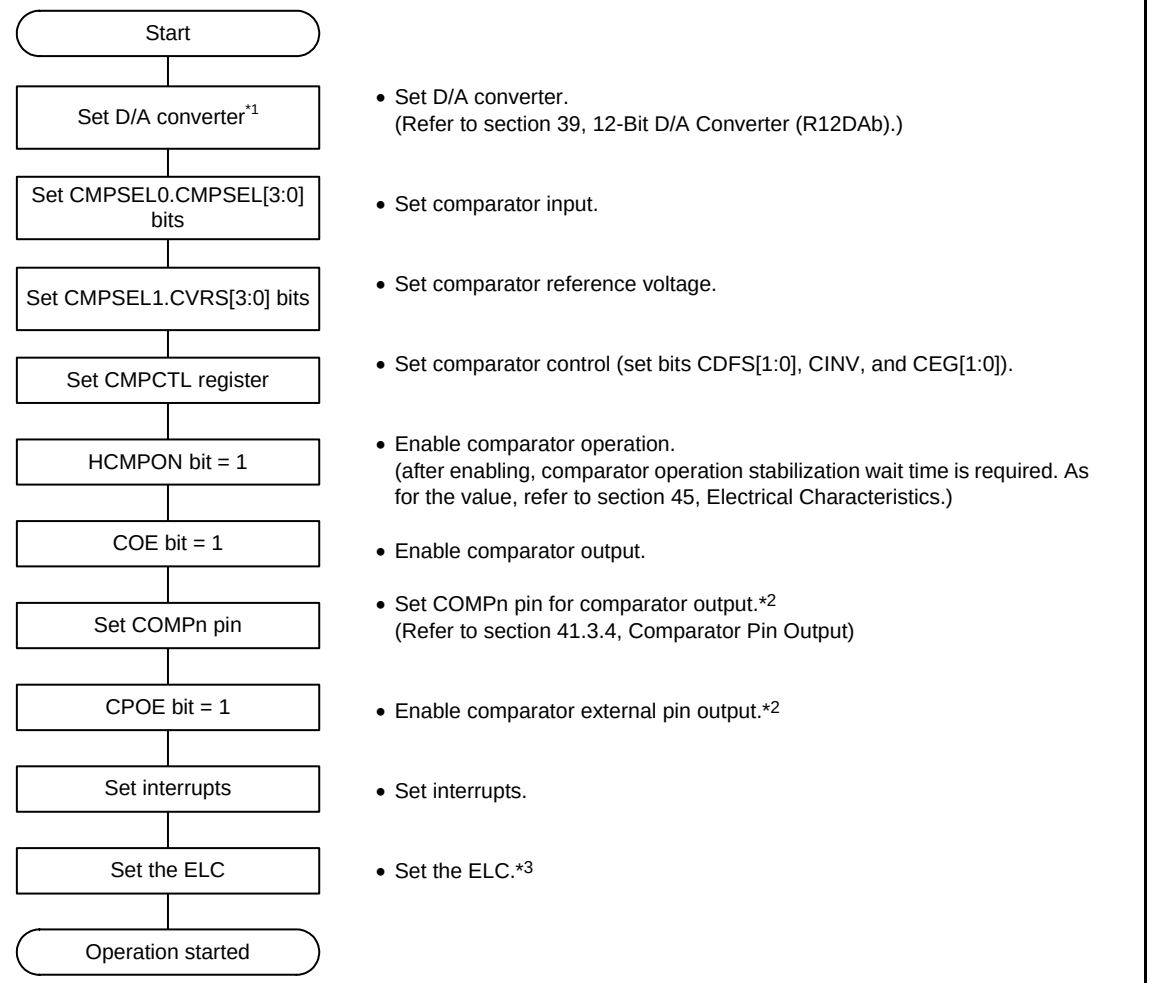
41.3.4 Comparator Pin Output

The comparison results can be output to the COMPn pins (n = 0 to 3). The CMPCTL.CINV bit can be used to set the output polarity (non-inverted output or inverted output), and the CMPIOC.CPOE bit can be used to enable or disable the output.
To output the comparison result to the external pin COMPn, use the following setting procedure. Note that the ports are set to input after reset.

(1) Execute steps (1) to (3) and steps (5) and (6) shown in section 41.3.3, Interrupts.
(2) Output the comparison result by the comparator to an external pin (set the CMPIOC.CPOE bit to 1).
(3) Set the port mode register and the pin function control register corresponding to each comparator output pin.

41.3.5 Comparator Setting Flowchart

Figure 41.5 shows the flowchart for setting the comparator-related registers.



Note 1. This is not required when the D/A converter is not used as a reference input voltage.
Note 2. This is not required when the comparison result is not output to the external pin.
Note 3. This setting is not required when comparator output is not used in the ELC.

Figure 41.5 Comparator Operation Setting Flowchart (n = 0 to 3)

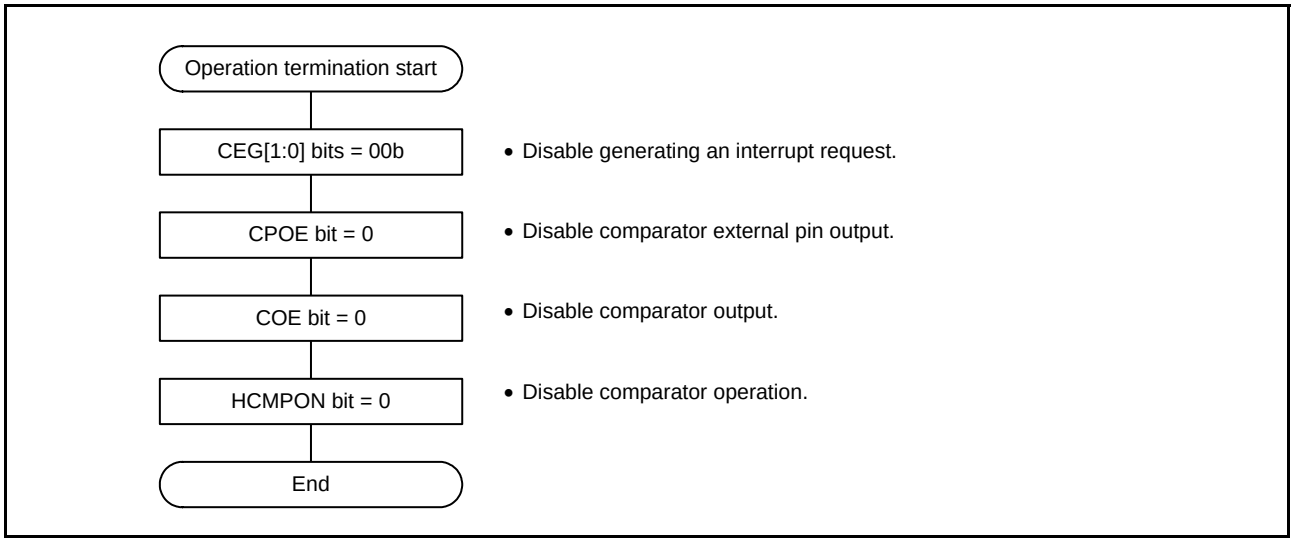


Figure 41.6 Comparator Operation Termination Flowchart

41.4 Usage Notes

41.4.1 Module Stop Function Setting

Operation of comparator C can be disabled or enabled using module stop control register B (MSTPCRB). After the reset, comparator C is halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

41.4.2 Comparator C Operation in Module Stop State

When the module stop state is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in the module stop state, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

41.4.3 Comparator C Operation in Software Standby Mode

When software standby mode is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in software standby mode, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

41.4.4 Setting the D/A Converter

Set the D/A converter, set the output to the comparator C using the D/A destination select register (DADSELR), and wait for the D/A converter output settling time. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

42. Data Operation Circuit (DOCA)

42.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16- or 32-bit values.

Table 42.1 lists the specifications of the DOC and Figure 42.1 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 42.1 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.

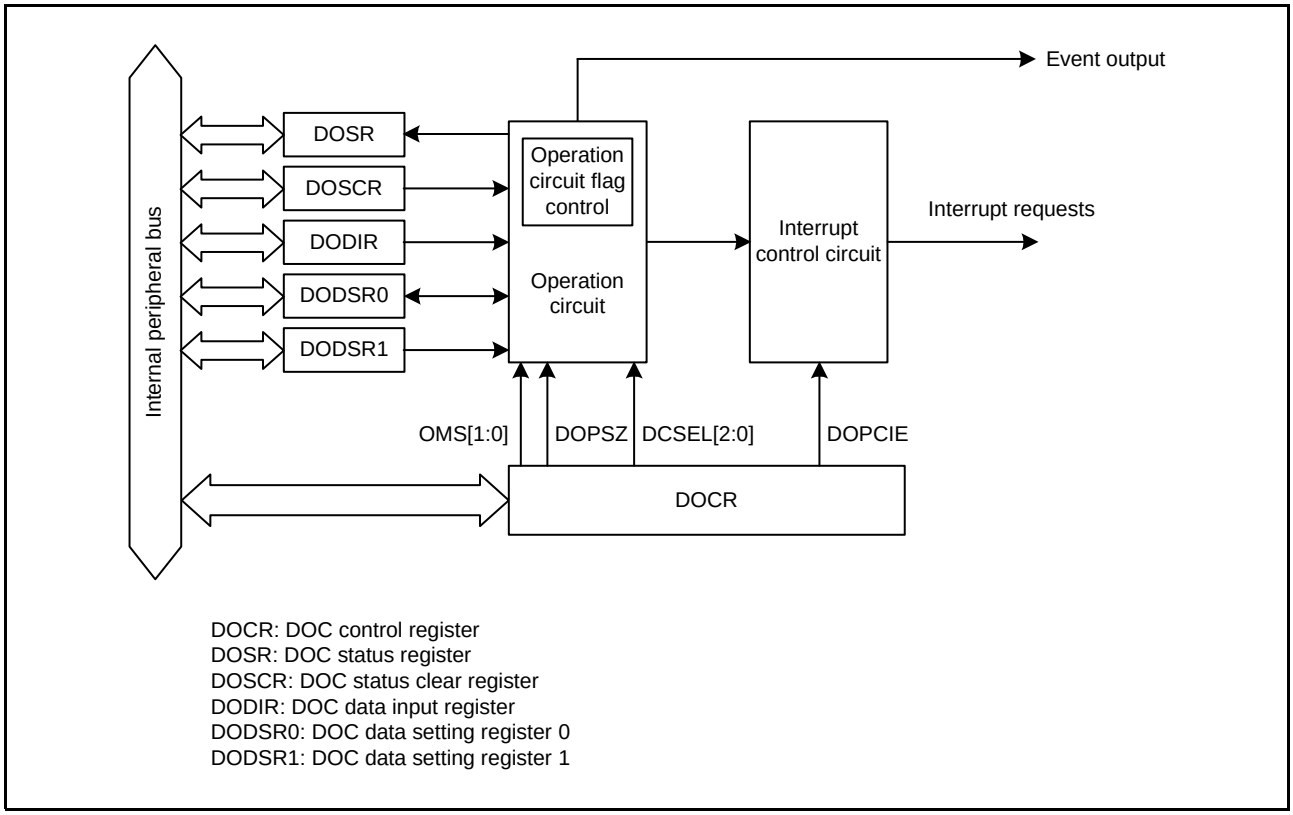
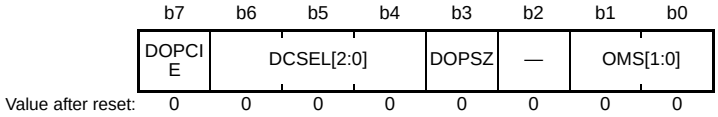


Figure 42.1 DOC Block Diagram

42.2 Register Descriptions

42.2.1 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.

42.2.2 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

42.2.3 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.

42.2.4 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.

42.2.5 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.

42.2.6 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.

42.3 Operation

42.3.1 Data Comparison Mode

Figure 42.2 to Figure 42.7 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.

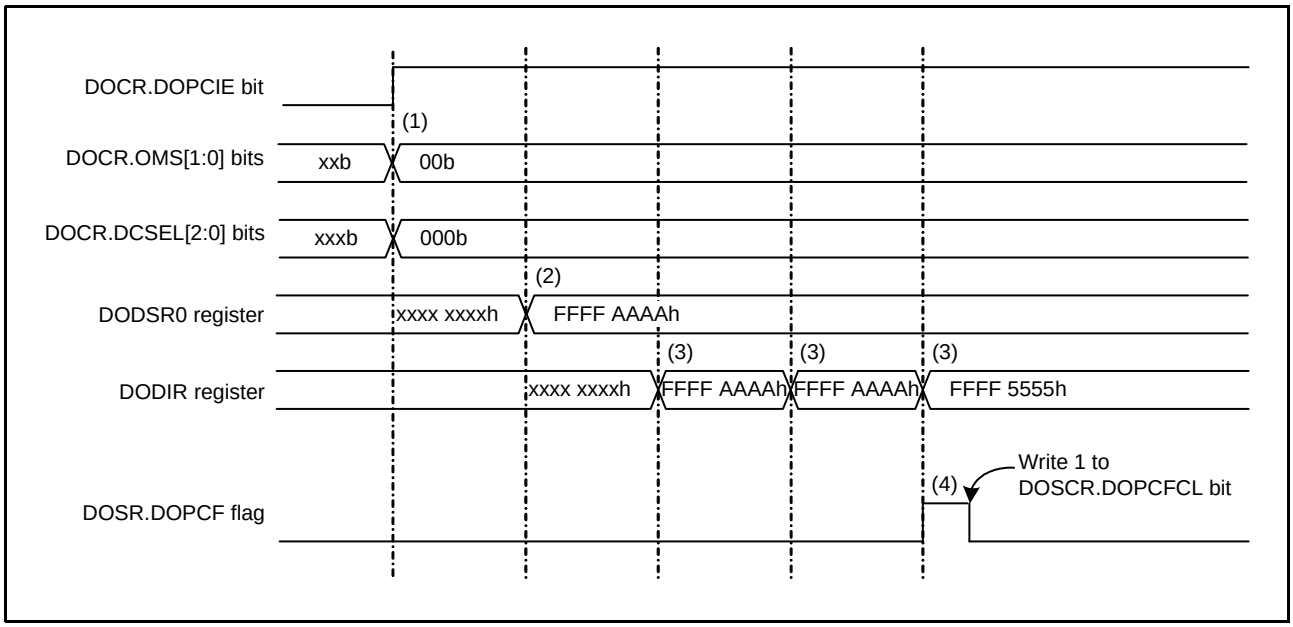


Figure 42.2 Example of the Operation of ‘Not Equal to’ as the Detection Condition in Data Comparison Mode

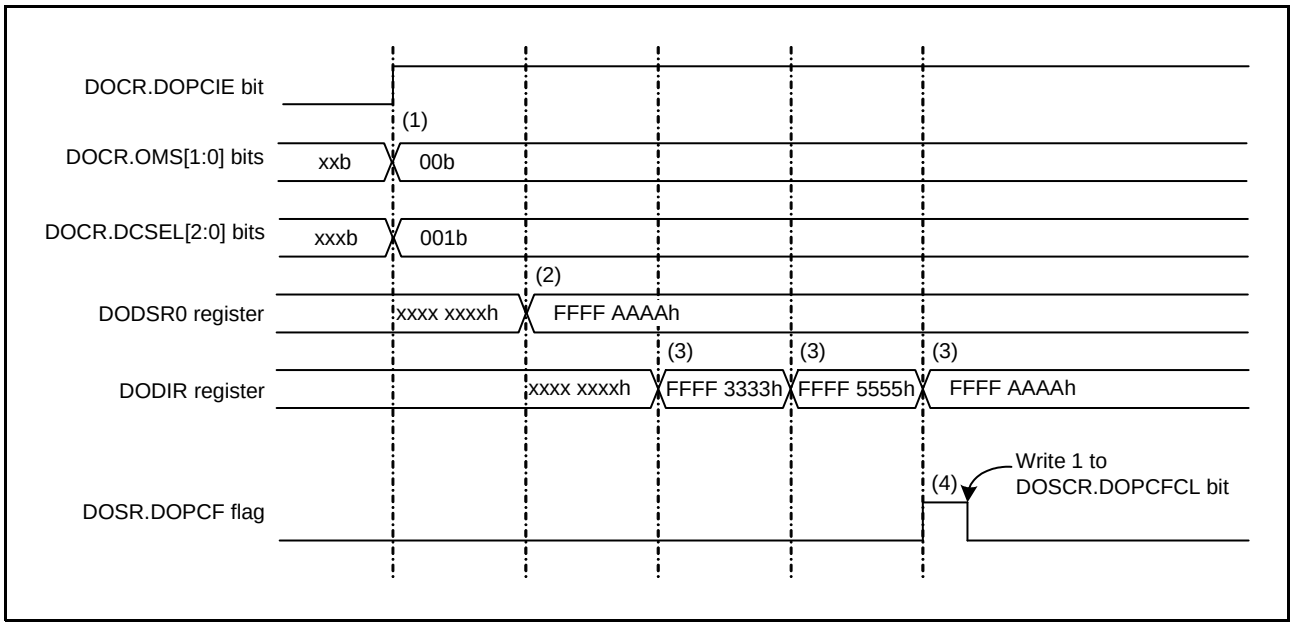


Figure 42.3 Example of the Operation of ‘Equal to’ as the Detection Condition in Data Comparison Mode

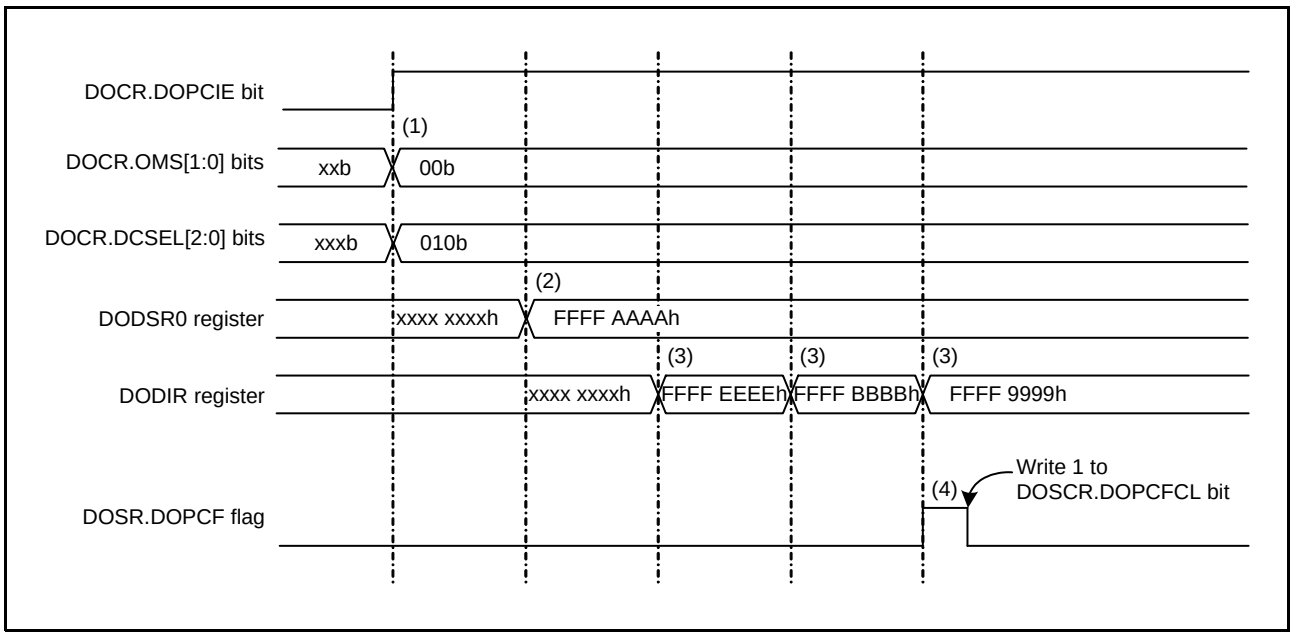


Figure 42.4 Example of the Operation of ‘Less Than’ as the Detection Condition in Data Comparison Mode

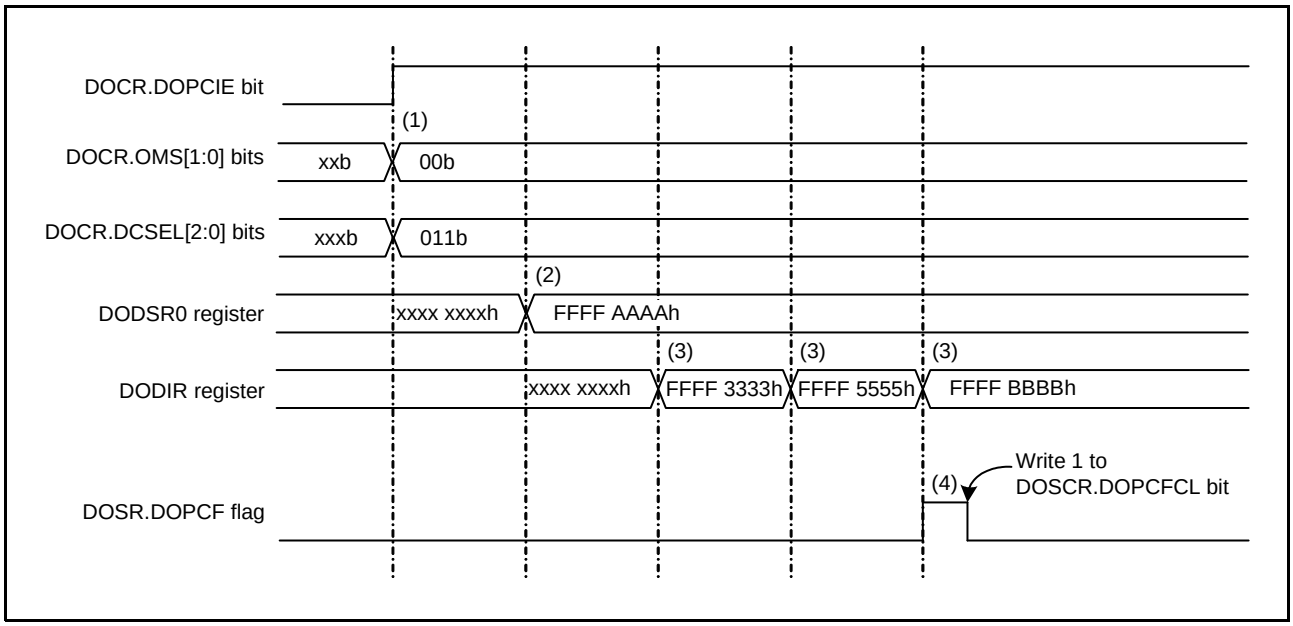


Figure 42.5 Example of the Operation of 'Greater Than' as the Detection Condition in Data Comparison Mode

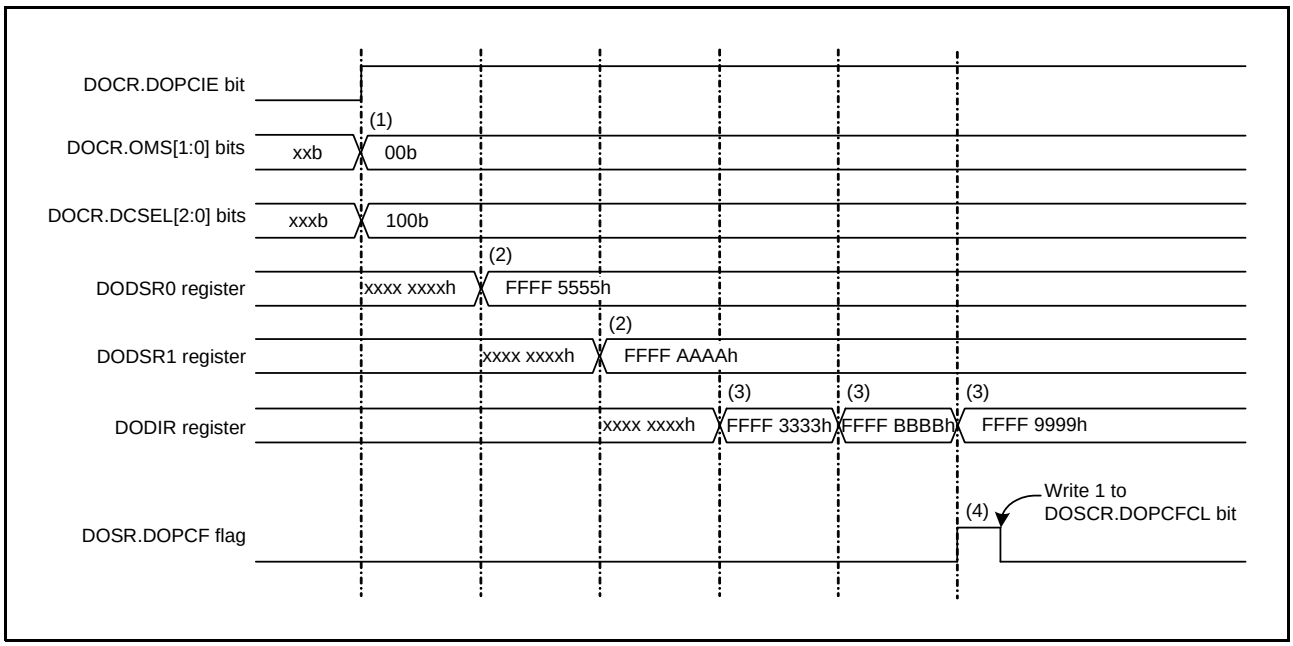


Figure 42.6 Example of the Operation of 'Within the Range' as the Detection Condition in Data Comparison Mode

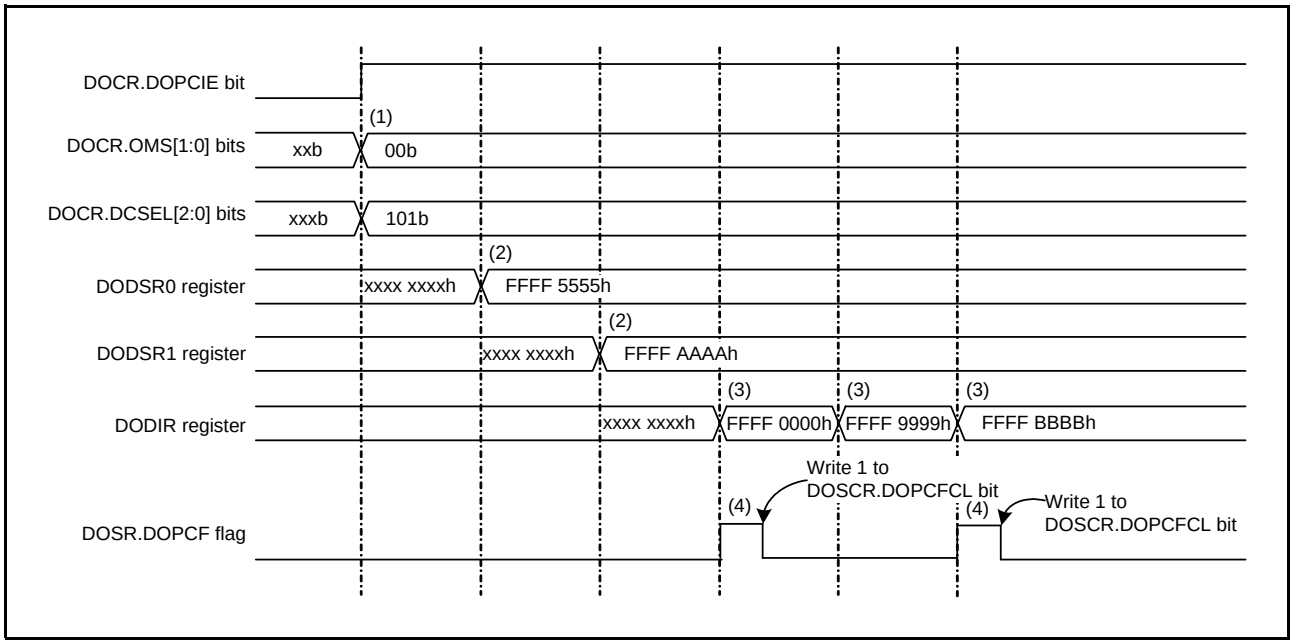


Figure 42.7 Example of the Operation of 'Beyond the Range' as the Detection Condition in Data Comparison Mode

42.3.2 Data Addition Mode

Figure 42.8 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.

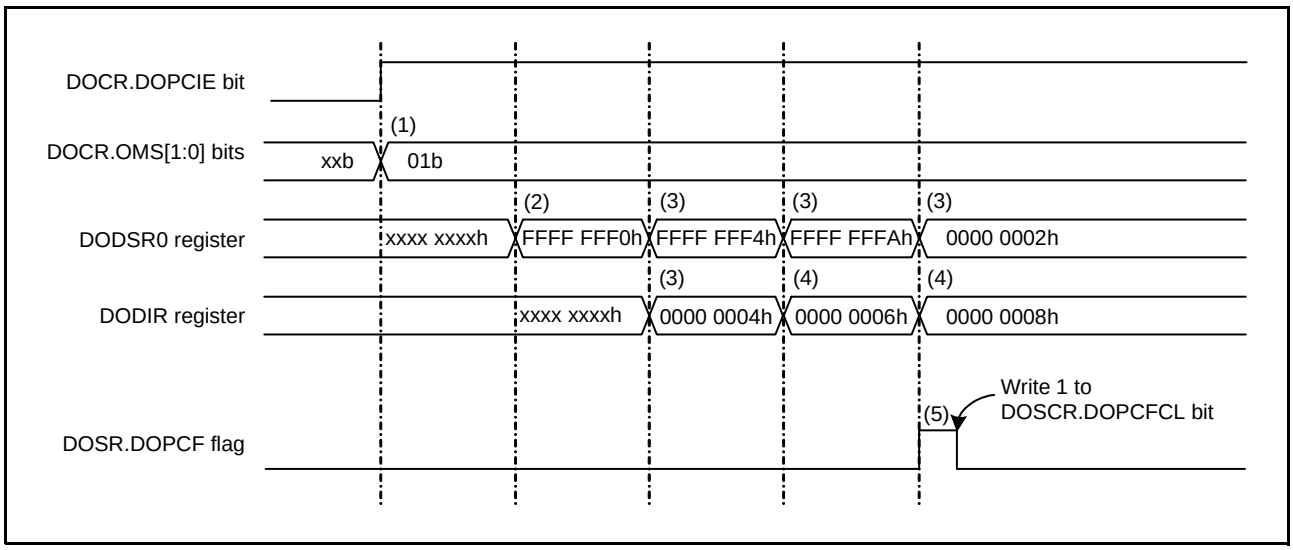


Figure 42.8 Example of Operation in Data Addition Mode

42.3.3 Data Subtraction Mode

Figure 42.9 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.

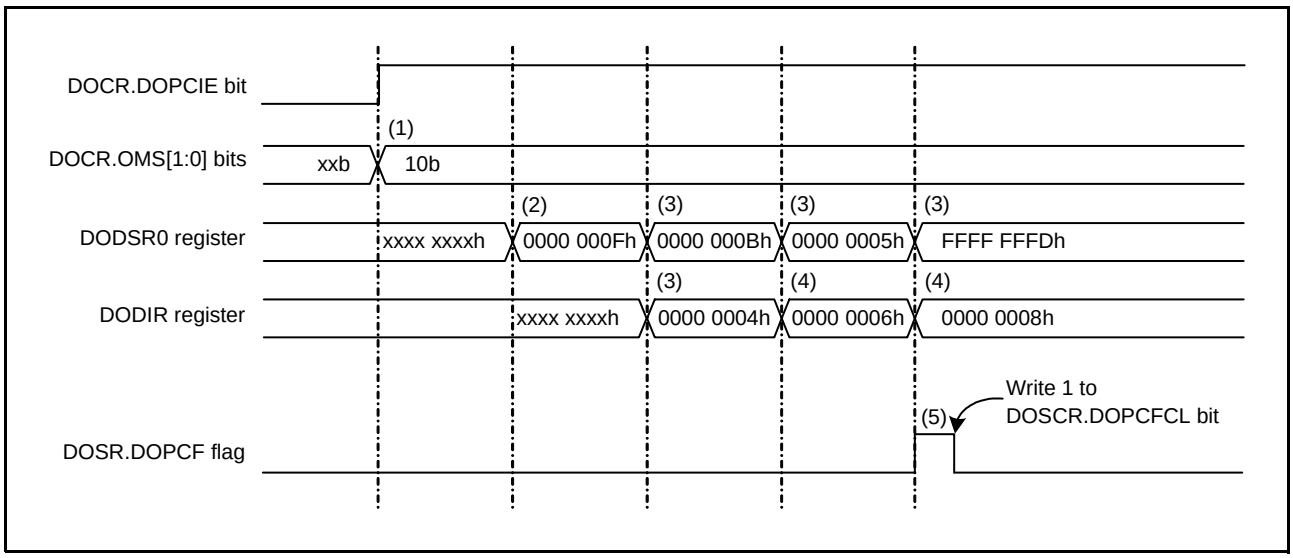


Figure 42.9 Example of Operation in Data Subtraction Mode

42.4 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 42.2 lists the details of the interrupt request.

Table 42.2 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).

42.5 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).

42.5.1 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.

42.6 Usage Note

42.6.1 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.

43. RAM

This MCU has a 128-Kbyte high-speed static RAM (RAM), which runs at the frequency of 120 MHz in the no-wait state.

43.1 Overview

Table 43.1 lists the specifications of the RAM.

Table 43.1 Specifications of RAM

Item	RAM
Capacity	128 Kbytes
Address	0000 0000h to 0001 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 3.2.4, System Control Register 1 (SYSCR1).

43.2 Register Descriptions

43.2.1 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.

43.2.2 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.

43.2.3 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.

43.2.4 RAM Protection Register (RAMPRCR)

Address(es): 0008 1204h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							RAMPRCR
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.

43.3 Operation

43.3.1 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.

43.3.2 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.

43.3.3 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 14, Interrupt Controller (ICUF).

Table 43.2 RAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
RAMERR	RAM error	Not possible	Not possible

43.4 Usage Notes

43.4.1 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.

43.4.2 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.

43.4.3 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).

44. 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).

44.1 Overview

Table 44.1 lists the specifications of the code flash memory/data flash memory, and Figure 44.1 is a block diagram of the flash memory related modules.
The I/O pins used in boot mode, refer to Table 44.18.
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, refer to Figure 44.2, and for the configuration of the data flash memory, refer to Figure 44.3.

Table 44.1 Specifications of Code Flash Memory and Data Flash Memory

Item	Code Flash Memory	Data Flash Memory
Memory capacity	• User area: 1 Mbyte max. • User boot area: 32 Kbytes	Data area: 32 Kbytes
Read cycle	One cycle	A read operation takes eight cycles of FCLK in 16- or 8-bit access
Value after erasure	FFh	Undefined
Programming/erasing method	• Programming and erasing the code flash memory/data flash memory is handled by the FACI commands specified in the FACI command issuing area (007E 0000h). • Programming/erasure through transfer by a flash-memory programmer via a serial interface (serial programming) • Programming/erasure of flash memory by a user program (self-programming)	
Security function	Protects against illicit tampering with or reading out of data in flash memory	
Protection function	Protects against erroneous rewriting of the flash memory	
Trusted memory (TM) function	Protects against illicit reading of blocks 8 and 9 in the code flash memory	
Background operation (BGO)	The user area can be read while the data area is being programmed or erased.	
Units of programming and erasure	• Units of programming for the user area or user boot area: 256 bytes • Units of erasure for the user area: Block units	• Unit of programming for the data area: 4 bytes • Unit of erasure for the data area: Block units
Other functions	Interrupts can be accepted during self-programming.	
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. • The user boot area can also be programmed or erased. Programming/erasure in boot mode (FINE interface) • FINE is used. Programming/erasure in user boot mode • Able to create original boot programs of the user's making. Programming/erasure in single-chip 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 by Parallel Programmer)	Programming and erasure of the user area and user boot area by using a parallel programmer is possible.	Programming or erasure of the data area by a parallel programmer is not possible.
Unique ID	A 12-byte ID code provided for each MCU	

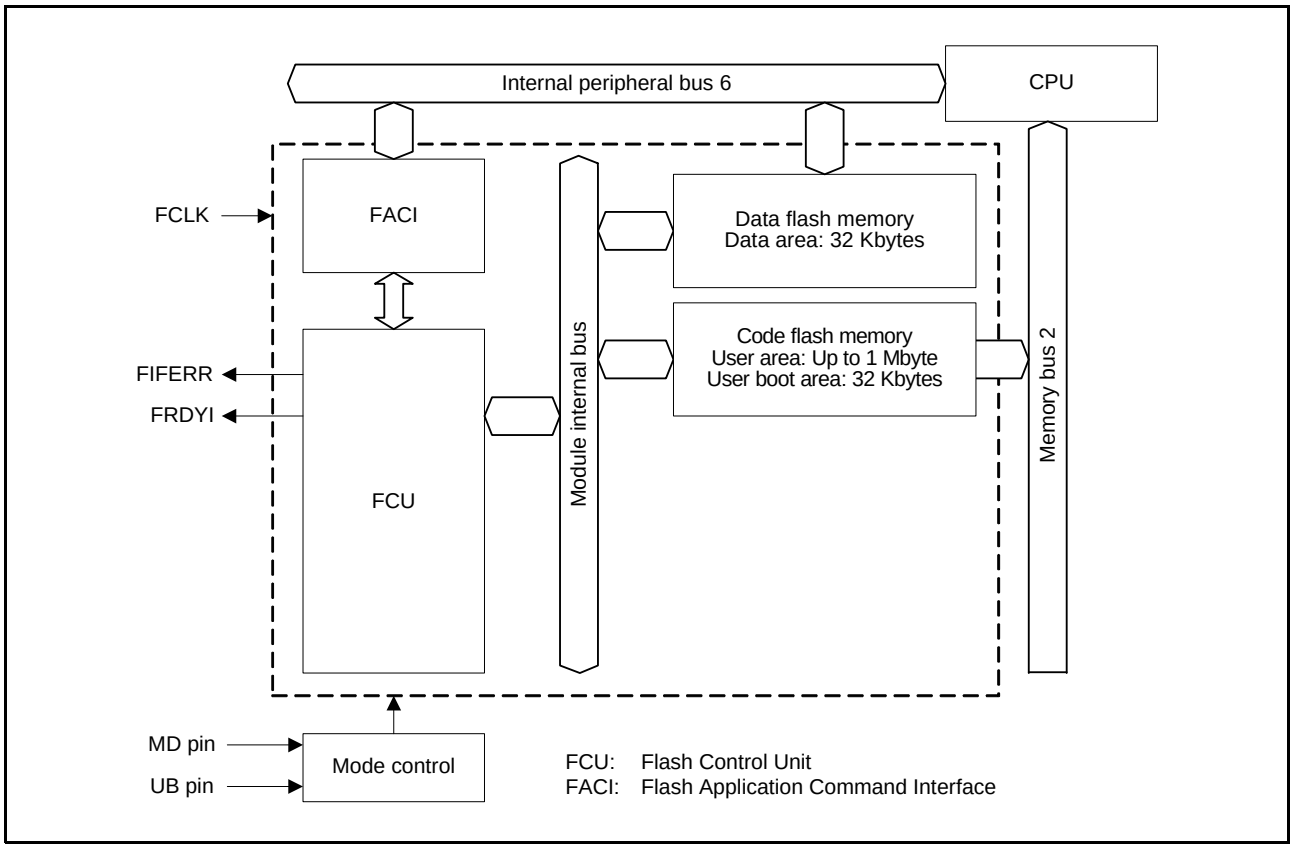


Figure 44.1 Block Diagram of Flash Memory Related Modules

44.2 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 44.2 summarizes information on all of these areas.

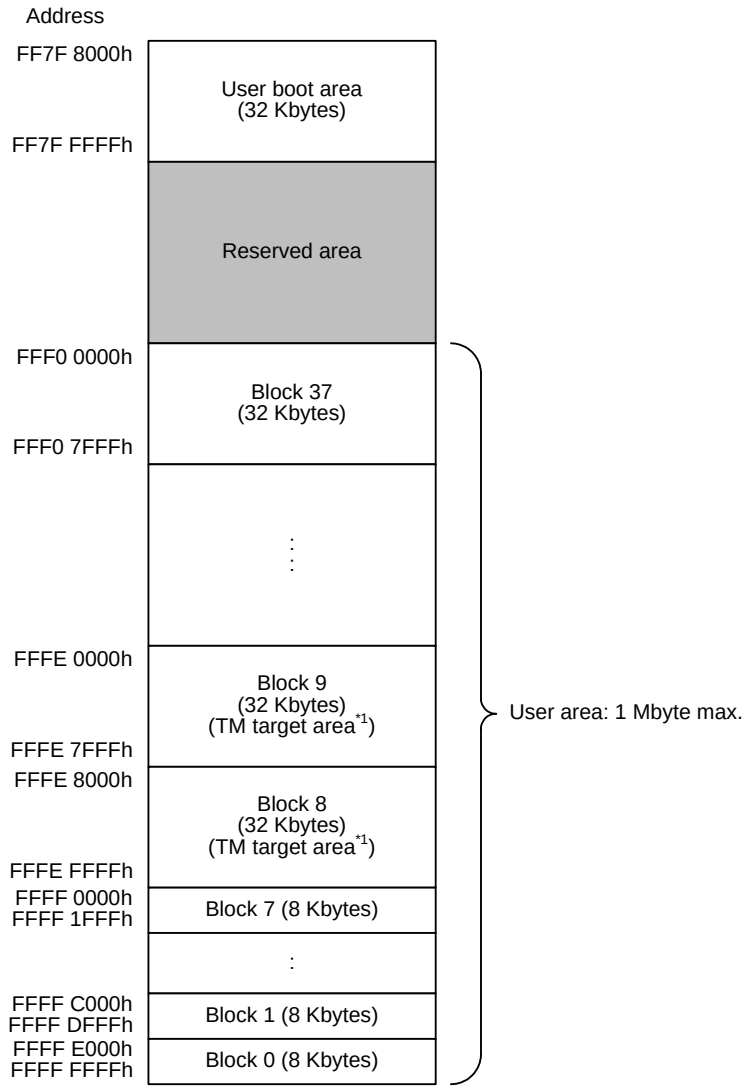
Table 44.2 Information on the Hardware Interface Area

Area	Address	Capacity
Area containing the various registers of the hardware	Refer to section 44.4, Register Descriptions.	Refer to section 44.4, Register Descriptions.
FACI command-issuing area	007E 0000h	4 bytes
Configuration setting area	0012 0040h to 0012 007Fh	64 bytes

44.3 Structure of Memory

Figure 44.2 illustrates the mapping of the code flash memory. The user area of the code flash memory in this MCU is divided into 8- and 32-Kbyte blocks, which serve as the units of erasure. When the TM function is enabled, blocks 8 and 9 are the TM target areas. A 32-Kbyte user boot area is also incorporated as a single block. The user area and user boot area are available as areas for storing the user program.

Furthermore, a 32-Kbyte user boot area, which is protected against programming by self-programming, is incorporated as a single block. This area is thus available as an area for storing boot programs, etc., for which rewriting while the user program is running has to be prohibited.



Code Flash Memory Capacity	Address	Number of Blocks
1 Mbyte	FFF0 0000h to FFFF FFFFh	0 to 37
512 Kbytes	FFF8 0000h to FFFF FFFFh	0 to 21

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 44.2 Mapping of the Code Flash Memory

The data area of the data flash memory in this MCU is divided into 64-byte blocks, with each being a unit for erasure. Figure 44.3 shows the mapping of the data flash memory.

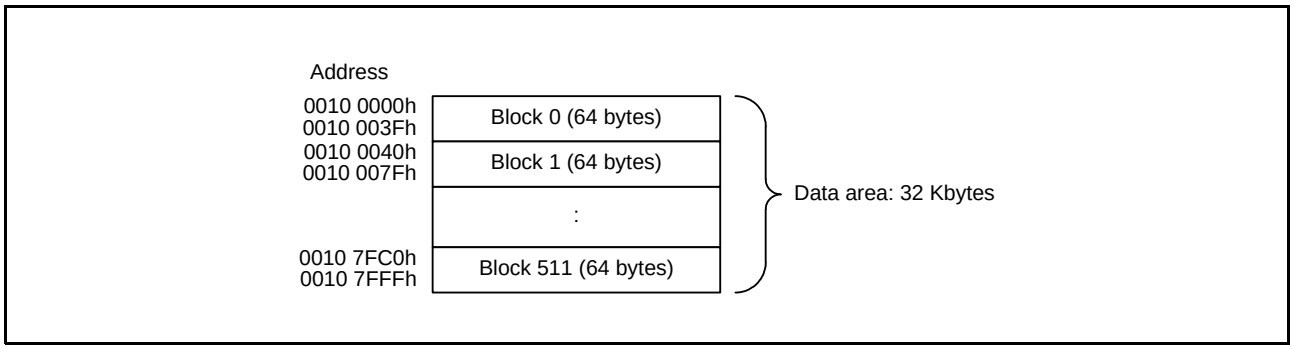


Figure 44.3 Mapping of the Data Flash Memory

44.4 Register Descriptions

44.4.1 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 Enable	b1 b0 0 0: Programming/erasure, programming/reading of lock bits, and blank checking are disabled 0 1: Programming/erasure, programming/reading of lock bits, and blank checking are enabled 1 0: Programming/erasure, programming/reading of lock bits, and blank checking are disabled 1 1: Programming/erasure, programming/reading of lock bits, and blank checking are disabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Programming and erasure of the flash memory, programming and reading of lock bits, and blank checking are enabled or disabled by hardware.

This register 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 deep software standby and software standby modes.

44.4.2 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 (refer to section 44.5.3.2, Error Protection). To release it from the command-locked state, a status clear command or forced

stop command must be issued by the FACI after clearing the CFAE and DFAE flags in the FASTAT register to 0.

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]

- Refer to Table 44.10, Error Protection Type

[Clearing Condition]

- When 0 is written after reading of 1

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 44.10, 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 while the CFAE or DFAE flag in the FASTAT register is 0

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]

- Refer to Table 44.10, Error Protection Type

[Clearing Condition]

- When 0 is written after reading of 1

44.4.3 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	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	After a reset is released, write 0 to this bit.	R/W
b2, b1	—	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.

44.4.4 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

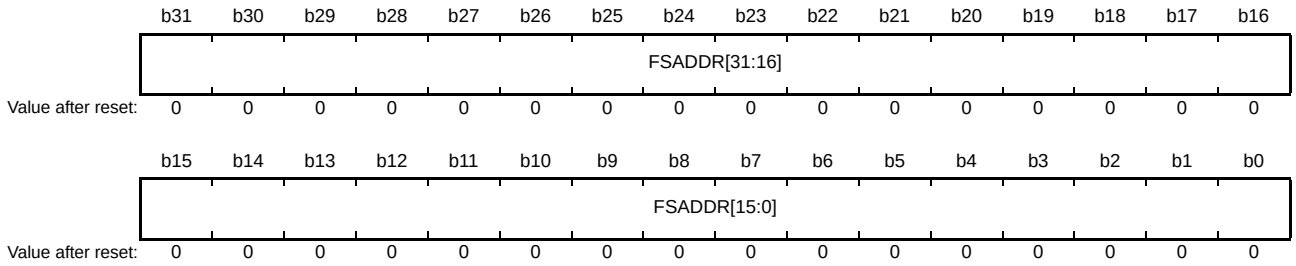
This register enables or disables generation of a flash ready (FRDY) interrupt request.

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.

44.4.5 FACI Command Processing Start Address Register (FSADDR)

Address(es): FLASH.FSADDR 007F E030h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	FSADDR[31:0]	FACI Command Processing Start Address	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, blank checking, configuration setting, lock-bit programming, or lock-bit reading 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 (FACI Command Processing Start Address)

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 19 are ignored in FACI command processing for the data flash memory. Bits that do not reach the address boundaries are also ignored. Table 44.3 shows the address boundary for each command.

Table 44.3 Address Boundary for Each of the Commands

Command	Address Boundary
Programming (code flash memory)	256-byte
Programming (data flash memory)	4-byte
Block erase (code flash memory)	8-Kbyte or 32-Kbyte
Block erase (data flash memory)	64-byte
Blank check	4-byte
Configuration setting	16-byte
Lock-bit programming	8-Kbyte or 32-Kbyte
Lock-bit reading	8-Kbyte or 32-Kbyte

Refer to Table 44.17, Address Used by Configuration Set Command for the start address of the configuration setting area.

44.4.6 FACI Command Processing 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]	FACI Command Processing End Address Register	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 blank checking command handling. When the FBCCNT.BCDIR bit is 0, the setting value of the FSADDR register must be the setting of the FEADDR register or lower. When the FBCCNT.BCDIR bit is 1, the value of the FSADDR register must be at least that of the FEADDR 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 (refer to section 44.5.3.2, 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 (FACI Command Processing End Address Register)

These bits are used to specify the end address for handling of the blank checking command. In command processing, bits 31 to 19, 1, and 0 are ignored.

44.4.7 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	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 Command 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, P/E suspend, P/E resume, forced stop, blank check, configuration setting, lock-bit program, or lock-bit read command processing is in progress. 1: None of the above is in progress.	R
b31 to b16	—	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]

- Refer to Table 44.10, 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 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 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]

- Refer to Table 44.10, 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]

- Refer to Table 44.10, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

ILGLERR Flag (Illegal Error Command 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]

- Refer to Table 44.10, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command while the DFAE and CFAE flag in the FASTAT register is 0

If the flash sequencer completes processing of a status clear or forced stop command while the CFAE or DFAE flag in the FASTAT register is 1, this flag is set to 1. This flag is temporarily set to 0 during processing of a forced stop command, and is re-set to 1 when the CFAE or DFAE flag is detected as 1 upon completion of command processing.

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

44.4.8 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 enabled only when AAh 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 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 flag 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[7:0] 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[7:0] 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.

44.4.9 Flash Protection Register (FPROTR)

Address(es): FLASH.FPROTR 007F E088h

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

Bit	Symbol	Bit Name	Description	R/W
b0	FPROTCN	Lock Bit Protection Cancel	0: Enables protection by the lock bits. 1: Disables protection by the lock bits.	R/W*1
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*2

Note 1. Writing to this bit is possible only when 55h is written to the KEY[7:0] bits in 16-bit units.
Note 2. Written values are not retained by these bits. These bits are read as 0.

This register enables or disables protection by the lock bits of the code flash memory against programming and erasure. The register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FPROTCN Bit (Lock Bit Protection Cancel)

This bit enables or disables protection by the lock bits of the code flash memory against programming and erasure.
[Setting Condition]

- When 1 is written to the FPROTCN bit while writing to the FPROTR register is enabled and the value of the FENTRYR register is other than 0000h

[Clearing Conditions]

- When the FPROTR register is written in 8-bit units
- When a value other than 55h specified for the KEY[7:0] bits and the FPROTR register is written in 16-bit units
- When 0 is written to the FPROTRCN bit while writing to the FPROTR register is enabled
- When the value of FENTRYR is 0000h

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the FPROTCN bit.

44.4.10 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, FPROTR, FCPSR, FSADDR, FENTRYR, and FBCCNT flash sequencer set-up registers keep their current values. 1: The FEADDR, FPROTR, 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[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 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
- FPROTR
- FCPSR
- FSADDR
- FENTRYR
- FBCCNT

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the SUINIT bit.

44.4.11 Lock Bit Status Register (FLKSTAT)

Address(es): FLASH.FLKSTAT 007F E090h

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

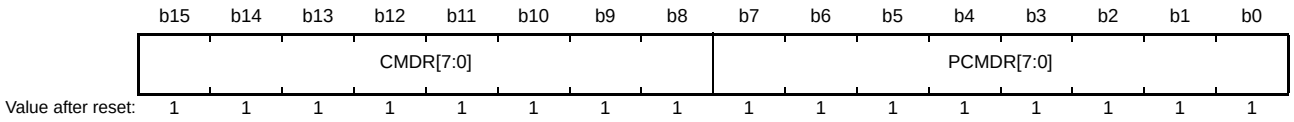
Bit	Symbol	Bit Name	Description	R/W
b0	FLOCKST	Lock Bit Status Flag	0: Protected state 1: Non-protected state	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FLOCKST Flag (Lock Bit Status Flag)

This flag reflects the value of the lock bit as read by executing a lock-bit read command. When the FSTATR.FRDY flag becomes 1 after the lock-bit read command is issued, the value of the target lock bit is stored in the FLOCKST flag. The value of the FLOCKST flag is retained until the next lock-bit command is completed.

44.4.12 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 44.4 States of FCMDR after Receiving Commands

Command	CMDR	PCMDR
Programming	E8h	Previous command
Block erase	D0h	20h
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
Lock-bit programming	D0h	77h
Lock-bit read	D0h	71h

44.4.13 Flash P/E Status Register (FPESTAT)

Address(es): FLASH.FPESTAT 007F E0C0h

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PEERRST[7:0]	P/E Error Status Flag	00h: No error 01h: Programming error due to an area protected by its lock bit 02h: Programming error for reasons other than lock-bit protection 11h: Erasure error due to an area protected by its lock bit 12h: Erasure error for reasons other than lock-bit protection	R
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the result of programming or erasure of the flash memory.

PEERRST[7:0] Flags (P/E Error Status Flag)

These flags indicate the source of an error that occurred during processing for the programming or erasure of the code flash memory or data flash memory. The value of these flags is only valid if the ERSERR or PRGERR flag in the FSTATR register is 1 when the FSTATR.FRDY flag becomes 1. When the ERSERR and PRGERR flags are 0, these flags retain their value to indicate the source of the last error to have occurred.

44.4.14 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.

44.4.15 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.
At the point where the FSTATR.FRDY flag is set to 1, the valid data is stored in the BCST flag.

44.4.16 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[18:16]		
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
b18 to b0	PSADR[18:0]	Programmed Area Start Address	The address of the first programmed area	R
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the address of the first programmed area to be found in processing of a blank check command.

PSADR[18:0] Bits (Programmed Area Start Address)

These bits indicate the address of the first programmed area to be found in processing of a blank check command. The address is an offset from the address where the data flash memory starts. The setting of these bits is valid only while the FBCSTAT.BCST flag is 1 and when the FSTATR.FRDY flag becomes 1. When the FBCSTAT.BCST flag is 0, the PSADR[18:0] bits holds the address detected by the previous check.

44.4.17 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 (refer to section 44.6.6.5, P/E Suspend Command). This bit should be set before issuing a block erase command.

44.4.18 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 enabled 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.

44.4.19 Unique ID Register n (UIDRn) (n = 0 to 2)

Address(es): FLASH.UIDR0 007F B174h, FLASH.UIDR1 007F B1E4h, FLASH.UIDR2 007F B1E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each chip															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip															

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 12-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

44.5 Overview of Functions

44.5.1 Methods of Programming and Erasure

By using a 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. Blocks 8 and 9 of the code flash memory can also be protected against reading by using the TM function.
Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe 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 44.5 lists the overview of the methods of programming and the corresponding operating modes.

Table 44.5 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	A user program written to the code flash memory in advance by serial or off-board programming may, when executed, rewrite the flash memory. The TM function can be in enabled at this time. BGO can be used for rewriting to the data area. A program to handle rewriting must be transferred to the internal RAM or external memory in advance of execution when BGO is not available.	Single-chip mode User boot mode

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 44.6.

Table 44.6 Differences between Modes

Item	Single-chip Mode	User Boot Mode	Boot Mode (SCI or FINE Interface)
Programmable and erasable area	• User area • Data flash memory • Option-setting memory (programming only)	• User area • Data flash memory • Option-setting memory (programming only)	• User area • Data flash memory • User boot area • Option-setting memory
Boot program at a reset	User area program	User boot area program	Boot program

Table 44.7 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, refer to section 7.2.1, Serial Programmer Command Control Register (SPCC), in section 7, Option-Setting Memory (OFSM).

Table 44.7 List of Basic Functions

Function	Functional Overview	Support Status	
		Serial Programming	Self-programming
Blank check	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that programming to memory has not proceeded after erasure.	Supported	Supported
Block erasure	Erases the memory contents in the specified block.	Supported	Supported
Programming	Write to the specified address.	Supported	Supported
Verify/Checksum	Compares the data read from flash memory and the data transferred from a flash memory programmer.	Supported	Not supported (read by user program is possible)
Read	Read data programmed to flash memory.	Supported	Supported
Setting of control or ID codes	Sets the OSIS register.	Supported	Supported
Setting the security functions	Sets the SPCC register and enables the following functions. • Prohibition of on-chip debugger connection • ID code protection for serial programmers • Prohibition of serial programmer connection • Prohibition of block erase commands • Prohibition of programming commands • Prohibition of read commands	Supported	Supported with conditions (Only switching the configuration from disabled to enabled is possible)
Setting the protection function	Configures lock bits for each block in code flash memory.	Supported	Supported
Configuration clearing	Erases the configuration setting area. Disables the TM function.	Supported	Not supported
Setting the TM function	Sets the TM function.	Supported	Supported with conditions (Only switching the configuration from disabled to enabled is possible)

44.5.2 Security Functions

The flash memory supports various security functions.
The security function includes ID code protection for on-chip debuggers, prohibition of on-chip debugger connection, ID code protection for serial programmers, prohibition of serial programmer connection, prohibition of commands (for block erasure, programming, and reading), and ROM code protection.
In serial programming, ID code protection for on-chip debuggers, prohibition of on-chip debugger connection, ID code protection for serial programmers, prohibition of serial programmer connection, and prohibition of commands (for block erasure, programming, and reading), can be used. In off-board programming, ROM code protection can be used.
Table 44.8 lists the security functions supported by the flash memory, and Table 44.9 lists the operations with security settings.

Table 44.8 Lists of Security Functions

Function	Description
ID code protection for on-chip debuggers	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.
ID code protection for serial programmers	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.
Prohibition of block erase commands	Block erase commands, area erasure command, and configuration clearing command at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erase commands are prohibited, changing a security setting from “disabled” to “enabled” is not possible.
Prohibition of programming commands	Programming commands are prohibited at the time of serial programming, and a condition applies to the execution of block erasure commands and area erasure command. Only through execution of the configuration clearing command can the prohibition be lifted.
Prohibition of read commands	Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.
ROM code protection	This function is to prohibit reading, programming, and erasure of the flash memory when a parallel programmer is in use.

Table 44.9 Operations with Security Settings

Function	Programming/Erasing/Reading with Respective Security Settings		Notes for Security Setting	
	Serial Programming	**Self-Programming**	**Serial Programming**	**Self-Programming**
Prohibition of on-chip debugger connection	Block erase commands: Supported Programming commands: Supported Read commands: Supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	The configuration clearing command can initialize the setting for prohibition.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
ID code protection for serial programmers	(When the ID codes do not match) Block erase commands: Not supported Programming commands: Not supported Read commands: Not supported (When the ID codes match) Block erase commands: Supported Programming commands: Supported Read commands: Supported	(Judgment of the ID codes is not performed.) Block erase commands: Supported Programming commands: Supported Read commands: Supported	The configuration clearing command can initialize the setting for prohibition. The setting for prohibition of block erase commands is not available. The setting for prohibition of programming commands is not available. The setting for prohibition of read commands is not available.	(Judgment of the ID codes is not performed.)
Prohibition of serial programmer connection	Block erase commands: Not supported Programming commands: Not supported Read commands: Not supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of block erase commands	Block erase commands: Not supported Programming commands: Supported Read commands: Supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible. The setting for ID code protection to be effective for serial programming is not available.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of programming commands	Block erase commands: Supported with conditions*1 Programming commands: Not supported Read commands: Supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	The configuration clearing command can initialize the setting for prohibition. The setting for ID code protection to be effective for serial programming is not available.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of read commands	Block erase commands: Supported Programming commands: Supported Read commands: Not supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported		

Note 1. For details, refer to section 44.9.45, Flow for Erasure when Programming Commands are Prohibited.

44.5.3 Protection Function

44.5.3.1 Software Protection

Software protection disables programming and erasure of the user area through the settings of control registers and lock bit settings in the user area. If an attempt is made to issue an FACI command against software protection, the flash sequencer enters the command-locked state.

(1) Protection through FWEPROR Register

Programming cannot proceed in any mode unless the FWEPROR.FLWE[1:0] bits are set to 01b.

(2) Protection through FENTRYR Register

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.

(3) Protection through Lock Bit

Each block in the user area includes a lock bit. When the FPROTR.FPROTCN bit is 0, blocks whose lock bit is set to 0 are disabled from being programmed/erased. To program or erase blocks whose lock bit is set to 0, set the FPROTR.FPROTCN bit to 1. When the lock bit protection is violated and programming, block erase or lock-bit programming command is issued for the user area, the flash sequencer enters the command-locked state.

44.5.3.2 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 while the CFAE and DFAE flags in the FASTAT register are 0. 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 FRDY flag.
Generation of a flash access error (FIFERR) interrupt detects malfunction. An FIFERR interrupt is generated under any of the following conditions.

- When violation in access to the data flash memory occurred (the FASTAT.DFAE flag is 1) while the FAEINT.DFAEIE bit is 1
- 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 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 FSTATR.ILGLERR flag becomes 1 and the other flags retain the values set due to previous error detection.
Table 44.10 shows error protection types and status bit values after error detection.

Table 44.10 Error Protection Type

Error Type	Description	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
FENTRYR setting error	The FENTRYR register setting is AA81h	1	0	0	0	0	0
	The FENTRYR register value at suspension disagrees with that at resumption	1	0	0	0	0	0
Illegal command error	Access with an undefined code was attempted in the first access by the FACI command	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	0	0
	The value “N” (refer to Table 44.16) specified in the second write access of an FACI command in the programming or configuration setting command is wrong.	1	0	0	0	0	0
	After 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 register > that in FEADDR register • When the FBCCNT.BCDIR bit is 1, and the value in FEADDR register > that in FSADDR register • The setting range of bits b18 to b0 in FEADDR register is between 0 8000h and 7 FFFFh	1	0	0	0	0	0
	An FACI command not acceptable in each mode has been issued (refer to Table 44.13)	1	0	0	0	0	0
	An FACI command has been issued when command acceptance conditions are not satisfied (refer to Table 44.14)	1	0/1	0/1	0/1	0/1	0/1
Erasure error	An error occurs during erasure	0	1	0	0	0	0
	A block erase command has been issued against lock bit protection.	0	1	0	0	0	0
Programming error	An error occurs during programming	0	0	1	0	0	0
	A programming or lock-bit programming command has been issued against lock bit protection.	0	0	1	0	0	0
Code flash memory access violation	After a programming command, block erase command, lock-bit programming command, or lock-bit reading 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 register is within the range from 00 0000h to EF FFFFh	1	0	0	0	1	0
Data flash memory access violation	After a programming command or block erase command has been issued under the following settings in data flash memory P/E mode • The setting value of bits 18 to 0 in FSADDR register is within the range from 0 8000h to 7 FFFFh	1	0	0	0	0	1
	After a configuration setting command has been issued under the following settings in data flash memory P/E mode • The setting value of bits 18 to 0 in FSADDR register is within the range from 0 0000h to 0 003Fh, or from 0 0100h to 7 FFFFh	1	0	0	0	0	1
Others	The FACI command-issuing area has been accessed in read mode	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	1	0	0	0	0	0
Flash write erase protection error	A flash memory write protection error has been detected by the setting of the FWEPROR register*1 during command processing by the flash sequencer.	0	0/1	0/1	1	0	0
Configuration set ECC error	A 2-bit error has been detected when the configuration setting value is read.	0	0	0	0	0	0
Programming parameter ECC error	A 2-bit error has been detected when the programming parameter table is read.	0	0	0	0	0	0

Note 1. For details on the FWEPROR register, refer to section 44.4.1, Flash P/E Protect Register (FWEPROR).

44.5.3.3 Boot Program Protection

(1) User Boot Protection

The user boot area can only be written in boot mode (SCI interface). Since this area is usually write-protected in normal operating mode and user boot mode, it can be used for the safe storage of programs such as a boot program.

44.5.4 Suspend Function

Reading from the user area/data area without using the BGO function is not possible during programming or erasure. When a P/E suspend command is issued to suspend the programming or erasure of the user area/data area, reading from the user area/data area 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.

44.5.5 Trusted Memory

This MCU has a trusted memory (hereafter called TM) facility to prevent the reading of blocks 8 and 9 in the user area 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 44.11 lists the specifications of the TM function, Table 44.12 lists access restrictions within the TM target area when TM function is enabled, and Figure 44.4 shows the cases where the CPU is able to operate in relation to the TM target area.

Table 44.11 TM Specifications

Item	Description
TM target area	Blocks 8 and 9 in the user area (64 Kbytes in total)
Access restrictions when TM is enabled	Refer to Table 44.12, Restrictions on Access to the TM Target Area while TM Function is Enabled.
How to run program code when the TM function is enabled	When TM is enabled, starting to run program code in an area being handled as TM is only possible with a branch instruction from program code outside the areas being handled as TM.
Interrupt processing during the execution of program code in an area being handled as TM while the TM function is enabled	Both the acceptance of requests for interrupt processing and return from interrupt processing are possible.
Security function	Enabling the TM function restricts access to program code in the areas for handling as TM to instruction fetching only
Protection functions	• Restrictions on 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 target area are erased. • Once enabled, the TM function prevents further writing to the areas being handled as TM target area

Note 1. Access to data in operations that include the borders of the TM target areas is also not allowed.

Table 44.12 Restrictions on Access to the TM Target Area while TM Function is Enabled

Type of Access	CPU	DMAC/DTC
Instruction fetching	Yes	—
Access to data*1	No	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, refer to the manual of the given OCD.
Note 1. Place data to which access will be required in areas other than the TM target areas.

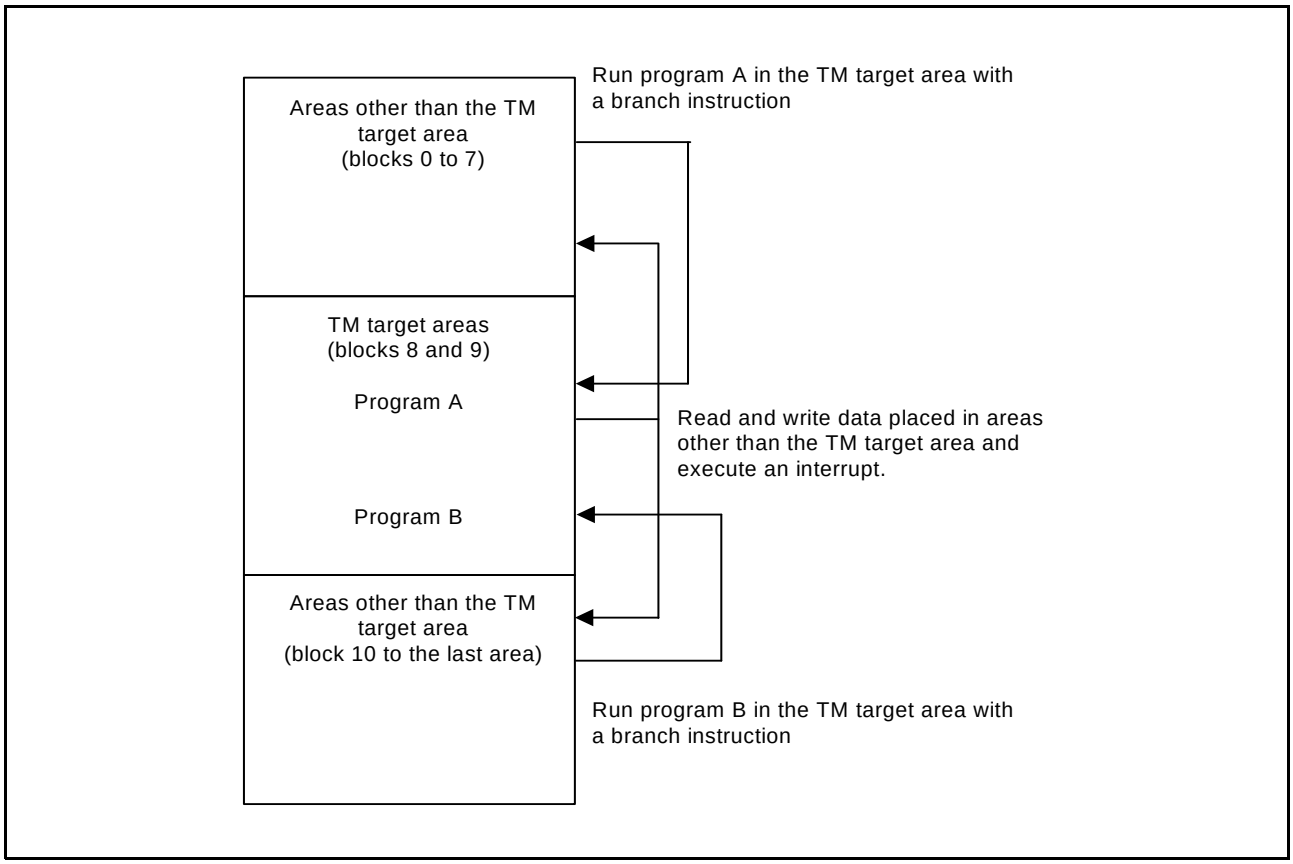


Figure 44.4 Cases where CPU is Allowed to Operate in Relation to the TM Target Area when the TM Function is Enabled

44.5.5.1 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.

44.5.5.2 How to Enable the TM Function

(1) By Self-Programming

After writing to blocks 8 and 9 of the user area, i.e. the target areas for TM, use the configuration setting command of the FACI to enable the TM function.
Figure 44.5 is a flowchart of the procedure for enabling the TM function by self-programming.

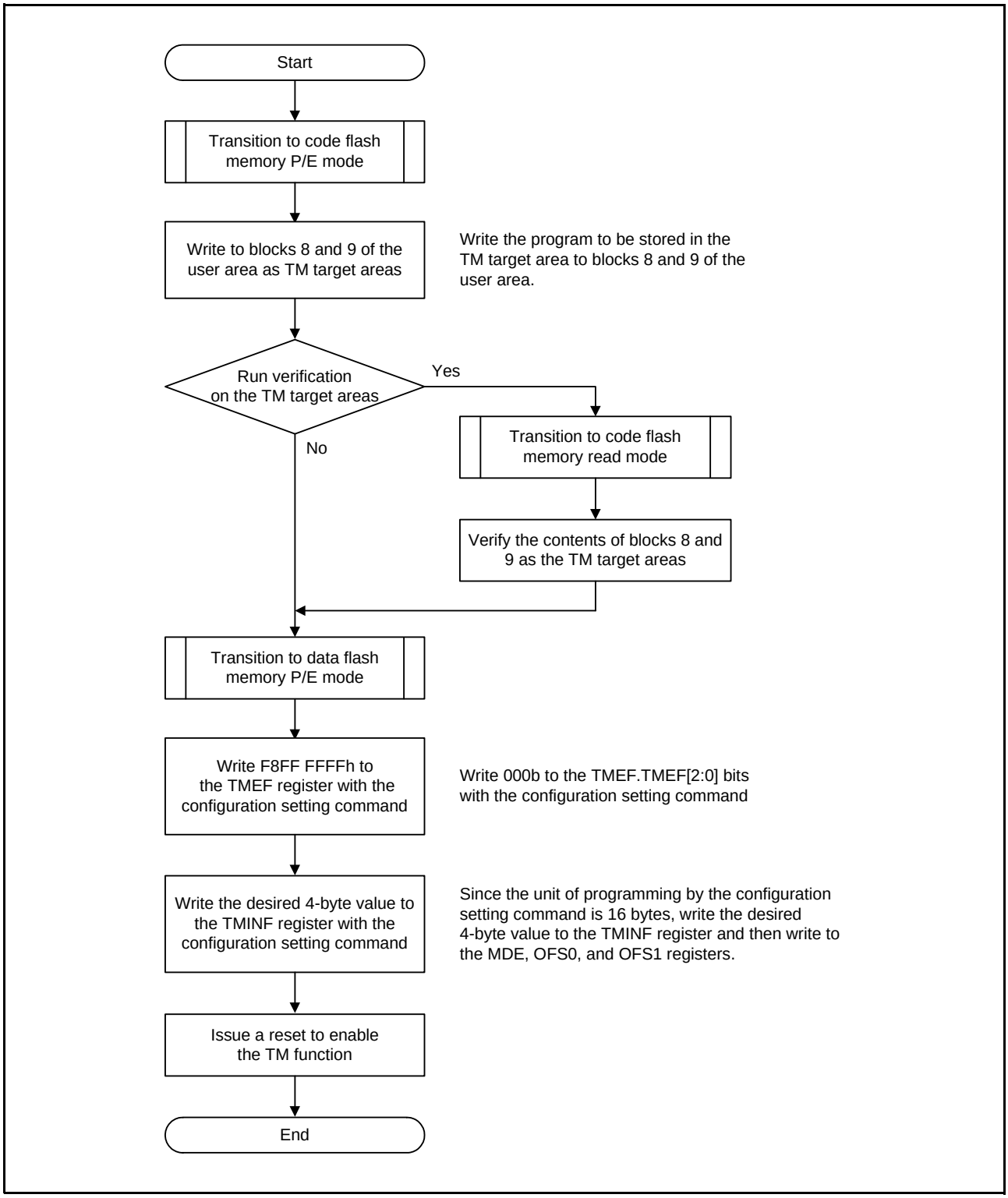


Figure 44.5 Flowchart for Enabling the TM Function by Self-Programming

(2) By Using Boot Mode

In boot mode, use the TM setting command among the boot commands to enable the TM function after writing to blocks 8 and 9 of the user area.

For the TM setting command among the boot commands, refer to section 44.9.38, TM Setting Command.

Figure 44.6 is a flowchart of enabling the TM function in boot mode.

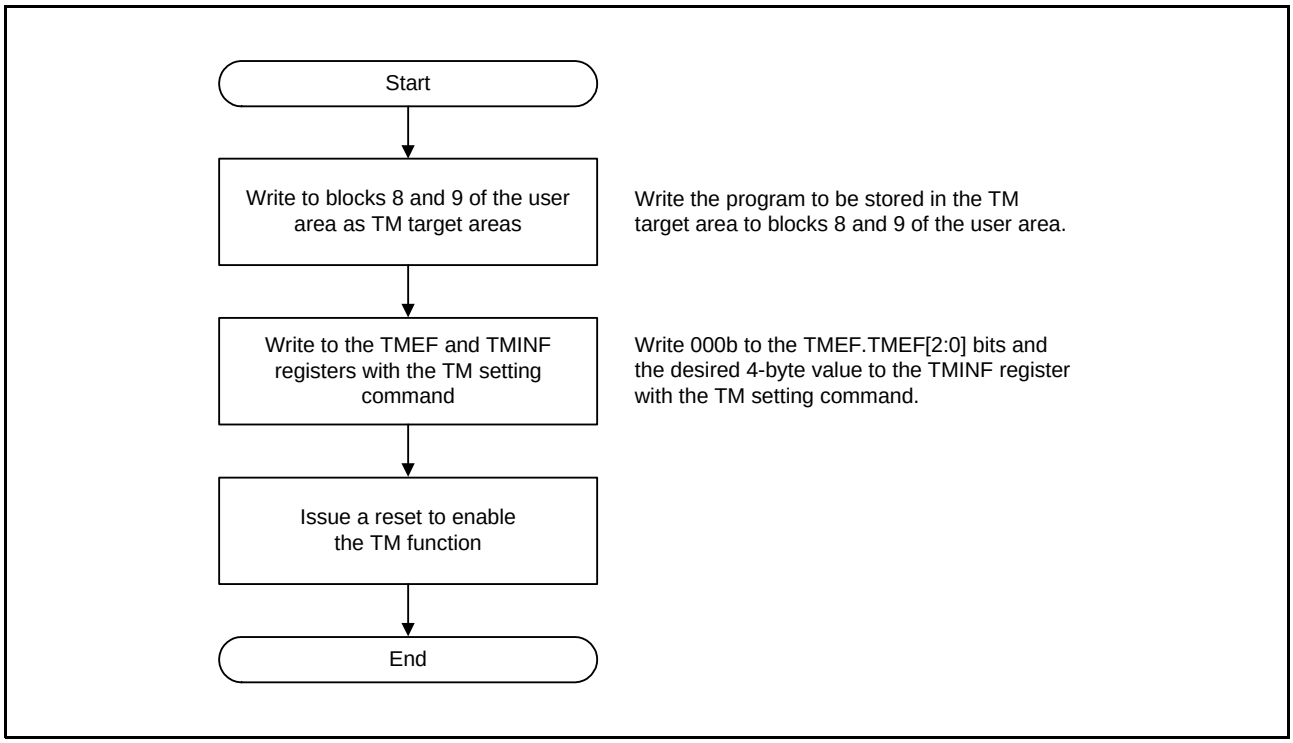
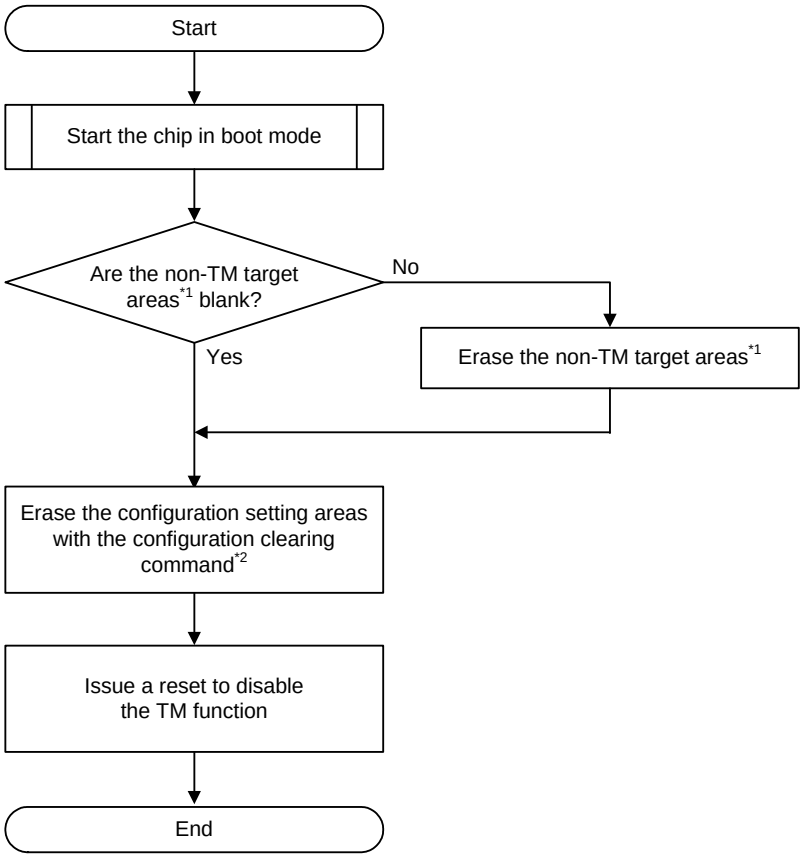


Figure 44.6 Flowchart for Enabling the TM Function by Using Boot Mode

44.5.5.3 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 44.7 is a flowchart of disabling the TM function in boot mode.



Note 1. Non-TM target areas are the data flash memory, user boot area, and user area other than blocks 8 and 9 of the user area.
Note 2. Issuing the configuration clearing command leads to erasure of the configuration setting areas, including the TMEF and TMINF registers, as well as the TM target areas (blocks 8 and 9 of the code flash memory).

Figure 44.7 Flowchart for Disabling the TM Function in Boot Mode

44.5.5.4 Notes on Enabling the TM Function

(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 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.

(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 need to write further data, follow the procedure described in section 44.5.5.3, How to Disable the TM Function; and follow the procedure described in section 44.5.5.2, How to Enable the TM Function, to program blocks 8 and 9 of the code flash memory and to re-enable the TM function.
To erase the user area, user boot area, and data area, refer to section 44.9.45, Flow for Erasure when Programming Commands are Prohibited.

(3) Executing the Configuration Clearing Command

Follow the procedure in section 44.5.5.3, How to Disable the TM Function, before issuing the configuration clearing command.

(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.

(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.

(6) Effect on the Setting for Command Protection

Executing the configuration clearing command to disable the TM function also initializes the command protected state at the same time.

(7) Conditions for Correct Operation of the TM Function

The TM function operates normally under the conditions prescribed in section 45, Electrical Characteristics.

44.6 Flash Sequencer

44.6.1 Operating Modes of the Flash Sequencer

The flash sequencer has three operating modes as shown in Figure 44.8. 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, the code flash memory and data flash memory is not readable.
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.

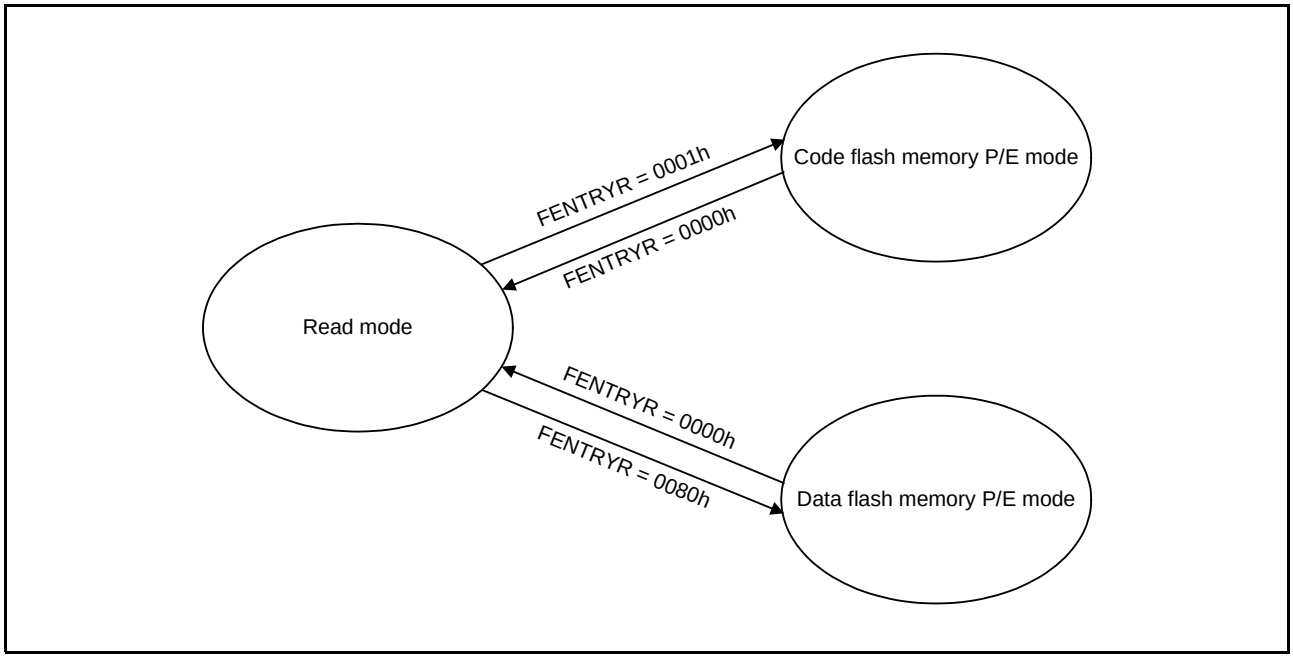


Figure 44.8 Modes of the Flash Sequencer

44.6.2 Read Mode

Read mode is for high-speed reading of the code flash memory or data flash memory.

(1) Code Flash Memory

Special settings are not required to read code flash memory in single-chip mode or user boot mode. Data can simply be read out through access to addresses in the code flash memory.

When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

(2) Data Flash Memory

Special settings are not required to read data flash memory in single-chip mode or user boot mode. Data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

44.6.3 P/E Modes

The P/E mode is for programming and erasure of the code flash memory or data flash memory.

(1) Code Flash Memory P/E Mode

In this mode, the code flash memory is in P/E mode, and the data flash memory is in read mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD bit to 0 and the FENTRYR.FENTRYC bit to 1.

(2) Data Flash Memory P/E Mode

In this mode, the code flash memory is in read mode, and the data flash memory is in P/E mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD bit to 1 and the FENTRYR.FENTRYC bit to 0.

44.6.4 Transitions of Operating Modes

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, ERSERR, PRGERR, and FLWEERR flags in the FSTATR register, and the CFAE and DFAE flags in the FASTAT register. Therefore, the occurrence of errors can be checked by reading the value of the FASTAT.CMDLK flag.
Table 44.13 lists the available commands in each operating mode.

Table 44.13 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 Lock-bit programming Lock-bit read
Data flash memory P/E mode	0080h	Programming Block erase P/E suspend P/E resume Status clear Forced stop Blank check Configuration setting

Table 44.14 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 44.14 Acceptable FACI Commands and the State of the Flash Sequencer

	Processing of Programming or Erasure*5	Processing of Configuration Setting	Processing to Suspend Programming or Erasure	Blank checking or lock bit reading	Programming Suspended	Erasure Suspended	Programming while Erasure is Suspended	Command-Locked State (FRDY = 1)	Command-Locked State (FRDY = 0)	Lock-Bit Programming	Processing of Forced Stop Command	Other State
FRDY flag	0	0	0	0	1	1	0	1	0	0	0	1
SUSRDY flag	1	0	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	0
PRGSPD flag	0	0	0/1	0/1	1	0	0	0/1	0/1	0	0	0
CMDLK flag	0	0	0	0	0	0	0	1	1	0	0	0
Programming	×	×	×	×	×	✓*3	×	×	×	×	×	✓
Block erase	×	×	×	×	×	×	×	×	×	×	×	✓
P/E suspend	✓	×	×	×	×	×	×	—	×	×	×	—
P/E resume	×	×	×	×	✓	✓	×	×	×	×	×	×
Status clear	×	×	×	×	✓	✓	×	✓	×	×	×	✓
Forced stop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Blank check	×	×	×	×	✓*1	✓*1	×	×	×	×	×	✓*1
Configuration setting	×	×	×	×	×	×	×	×	×	×	×	✓*1
Lock-bit programming	×	×	×	×	×	×	×	×	×	×	×	✓*2
Lock-bit read	×	×	×	×	✓*2	✓*2, *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. The value read out is undefined when a lock-bit read command is issued for a block where erasure was suspended.
Note 5. Cases that the programming or erasure processing is completed while the P/E suspend command is accepted are also included.

44.6.4.1 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 shift to code flash memory P/E mode, set the FENTRYR.FENTRYC bit to 1.

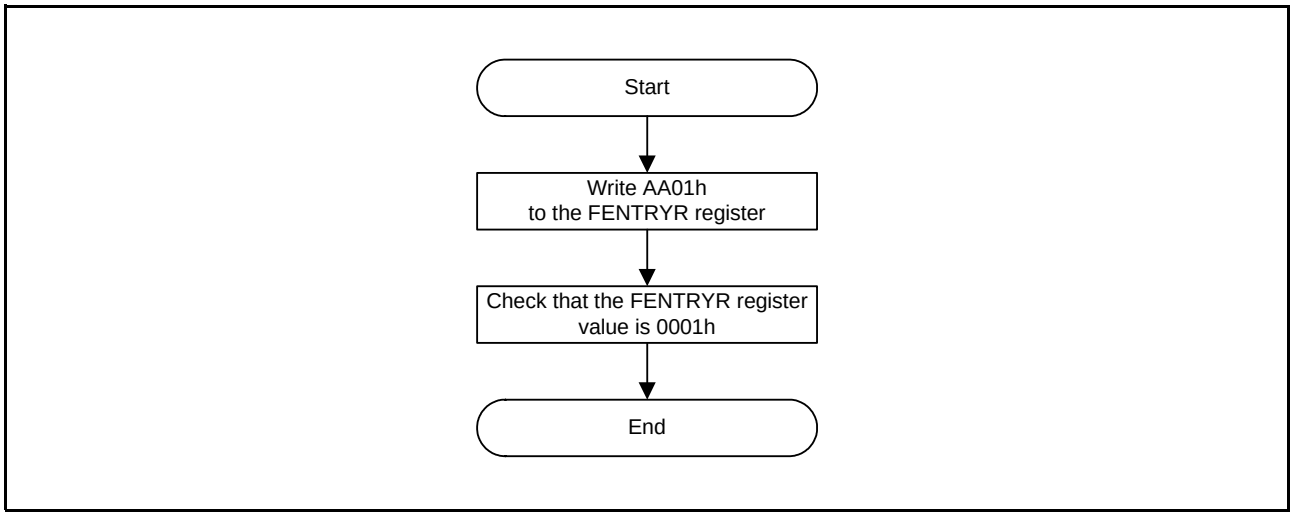


Figure 44.9 Procedure for Transition to Code Flash Memory P/E Mode

44.6.4.2 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.FENTRYD bit to 1.

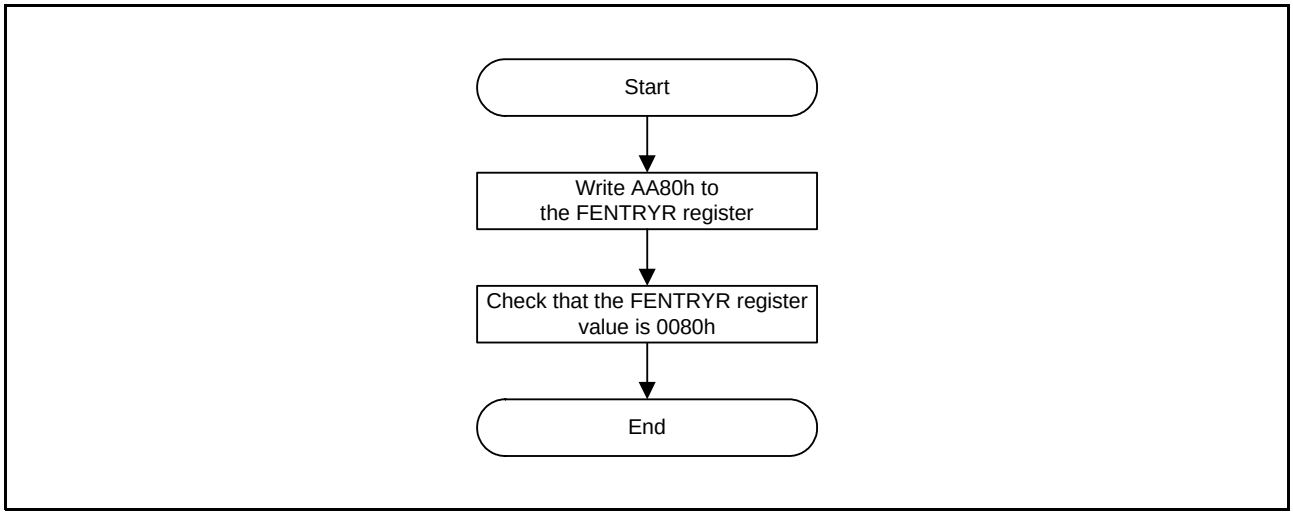
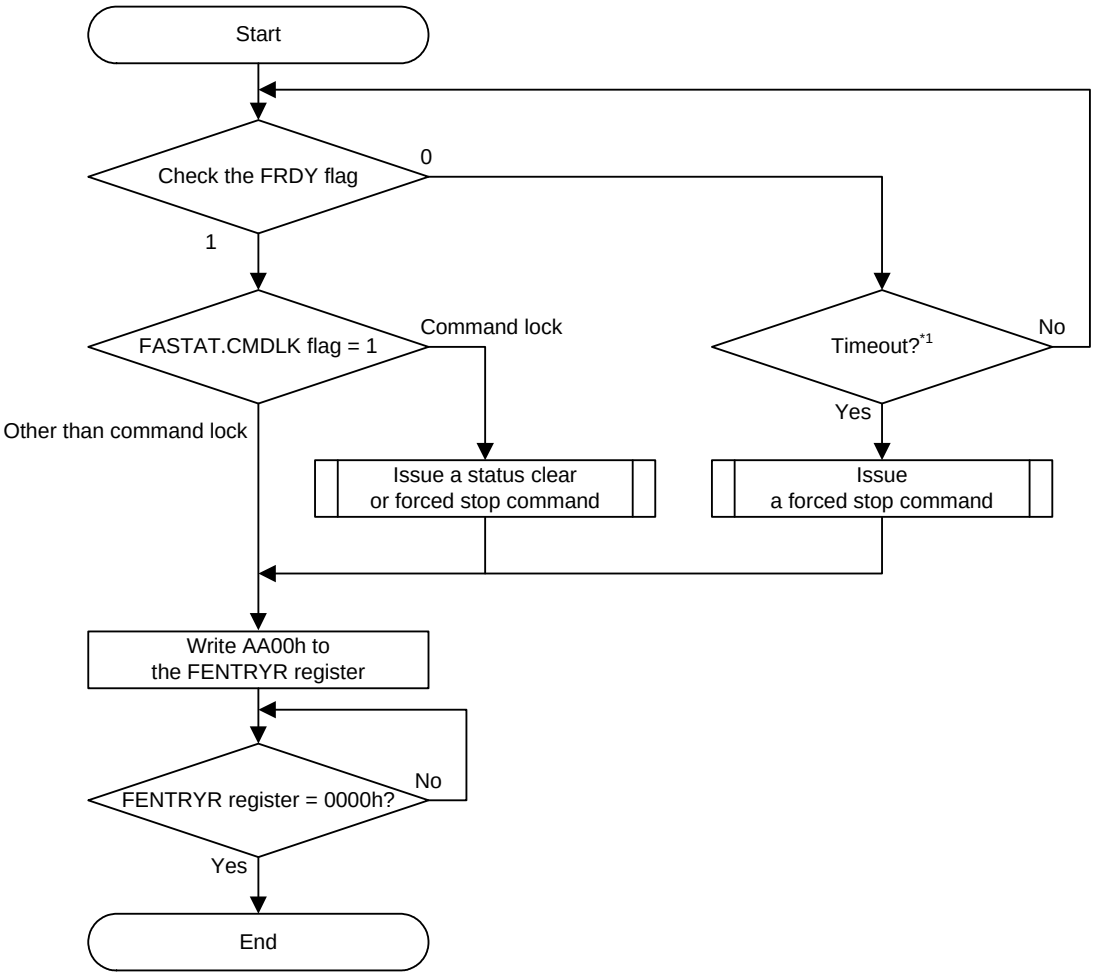


Figure 44.10 Procedure for Transition to Data Flash Memory P/E Mode

44.6.4.3 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 section 45, Electrical Characteristics).

Figure 44.11 Procedure for Transition to Read Mode

44.6.5 List of FACI Commands

Table 44.15 List of FACI Commands

FACI Command	Description
Programming	This is used to program the user area and data area. Units of programming are 256 bytes for the user area and 4 bytes for the data area.
Block erase	This is used to erase the user area (including the lock bit) and data area. The unit of erasure is one block. (user area: 8 K or 32 Kbytes, data area: 64 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, and PRGERR flags in the FSTATR register, and the CMDLK flag in the FASTAT register and releases the flash sequencer from the command-locked state.
Forced stop	This forcibly stops processing of FACI commands and initializes the FSTATR register.
Blank check	This is used to check if data areas are blank. Units of blank checking: 4 bytes to 32 Kbytes (specified in 4-byte units).
Configuration setting	This is used to set the configuration setting area. Units of setting: 16 bytes.
Lock-bit programming	This is used to program the lock bit for a user area. Units of programming: 1 bit (the lock bit for one block)
Lock-bit read	The lock bit for a user area is read and the result is stored in the FLKSTAT register. Units of reading: 1 bit (the lock bit for one block)

The FACI commands are issued by writing to the FACI command-issuing area (refer to Table 44.2). When write access as shown in Table 44.16 proceeds in the specified state, the flash sequencer executes the processing corresponding to the given command (refer to section 44.6.4, Transitions of Operating Modes).

Table 44.16 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 (user area) 256-byte programming, N = 128	131	E8h	80h (= N)	WD_1 to WD_{128}	D0h
Programming (data area) 4-byte programming: N = 2	5	E8h	02h (= N)	WD_1 and WD_2	D0h
Block erase	2	20h	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
Lock-bit programming	2	77h	D0h	—	—
Lock-bit read	2	71h	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.

44.6.6 Usage of FACI Commands

This section gives an overview of the usage of FACI commands.

44.6.6.1 Overview Flow when FACI Command is Used

Figure 44.12 shows an overview flow when the FACI command is used.
When using the BGO function, 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 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 44.4.18, Flash Sequencer Processing Clock Frequency Notification Register (FPCKAR).

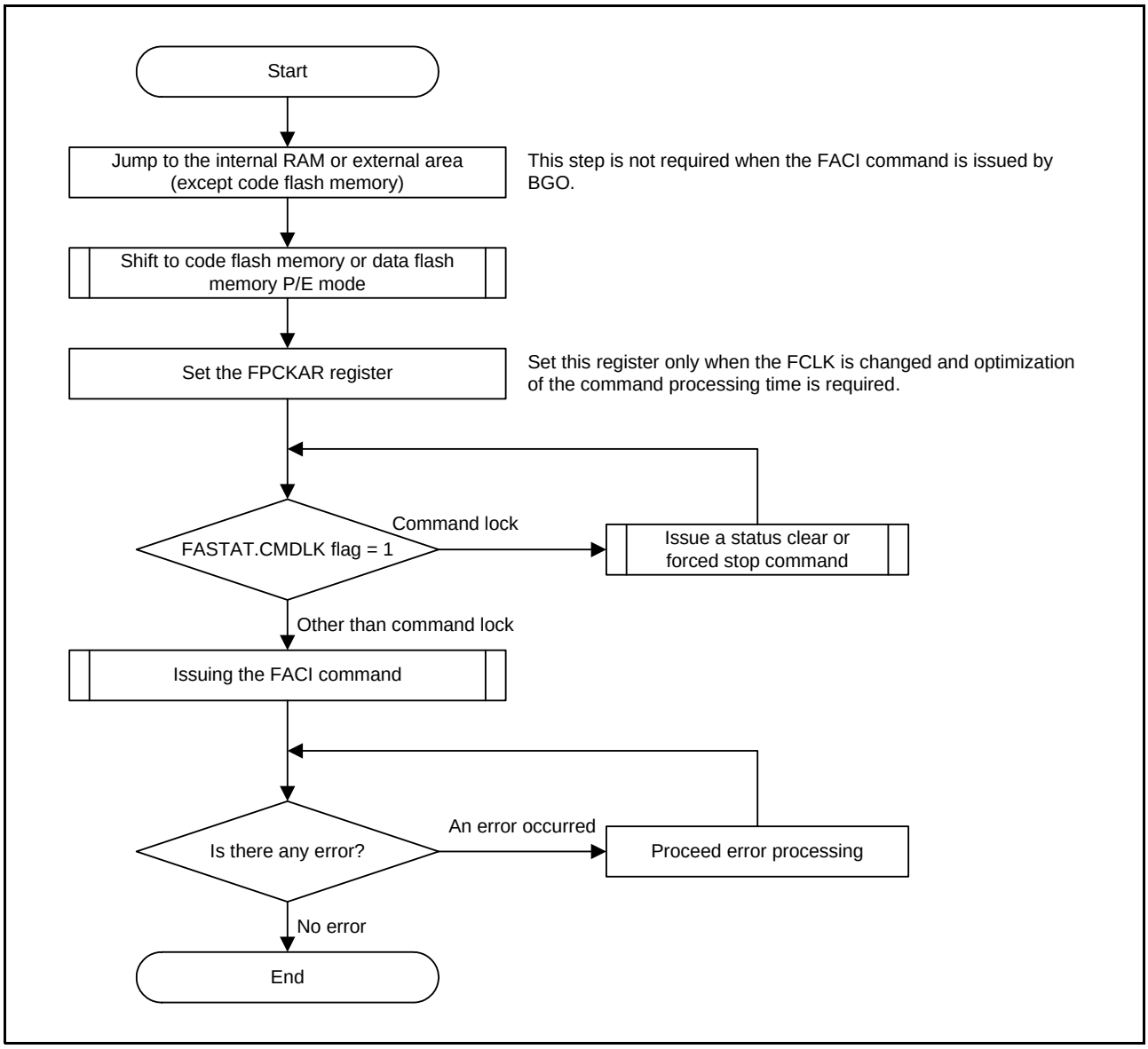
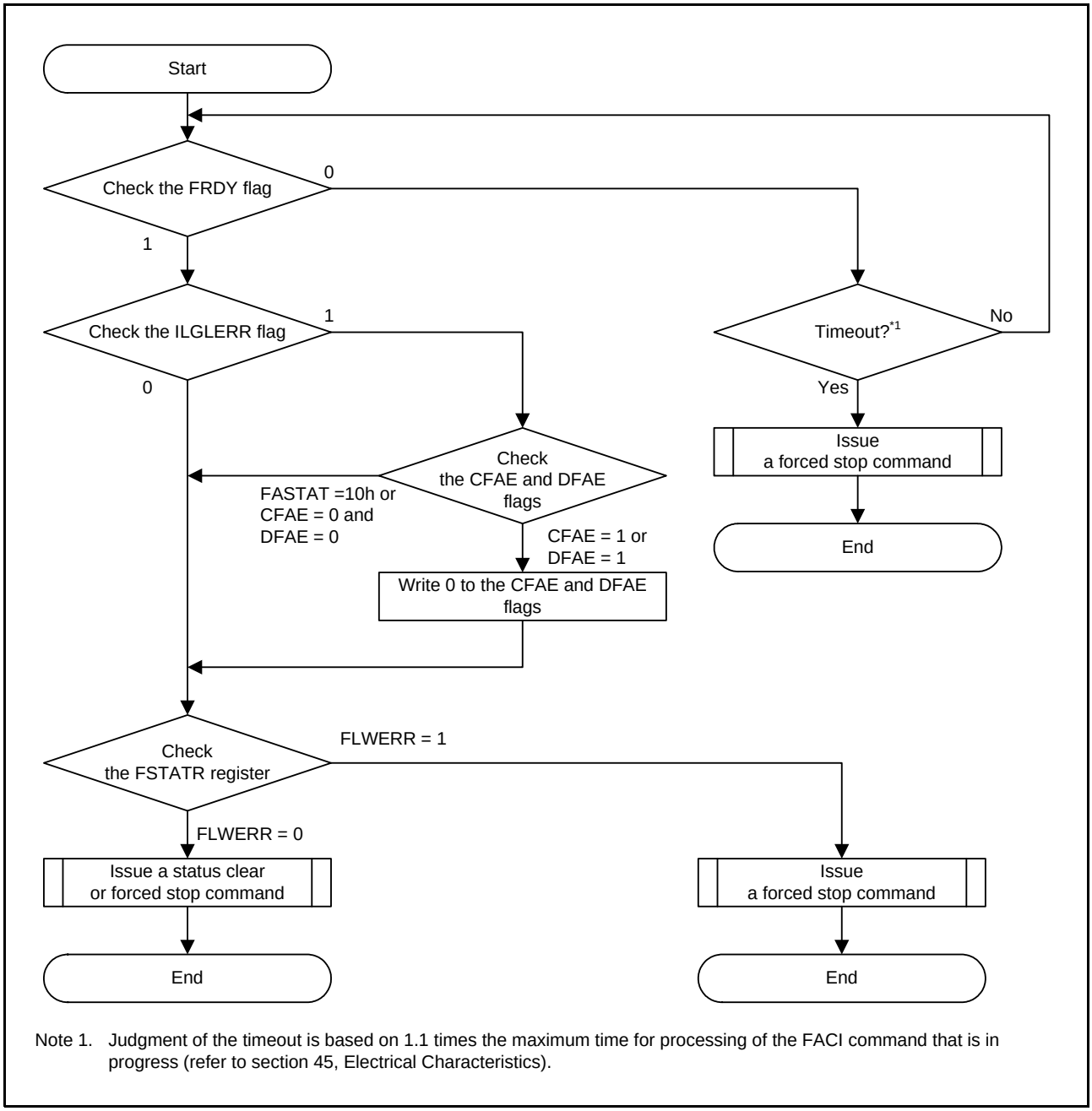


Figure 44.12 Overview Flow when FACI Command is Used

44.6.6.2 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, or FASTAT register. 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. If processing is not completed within the maximum programming/erasure time specified in section 45, Electrical Characteristics, this can be considered a timeout, and the flash sequencer should be stopped by the forced stop command.
When the FSTATR.ILGLERR flag is 1, check the FASTAT value. If the CFAE or DFAE flag in the FASTAT register is 1, set the CFAE or DFAE flag in the FASTAT register to 0 and then issue the status clearing and forced stop commands. The FSTATR.FLWERR flag is not changed from 1 to 0 by the status clear command. When these bits are set to 1, use the forced stop command for release from the command-locked state. The other bits that indicate the command-locked state can be changed from 1 to 0 by the status clear or forced stop command.

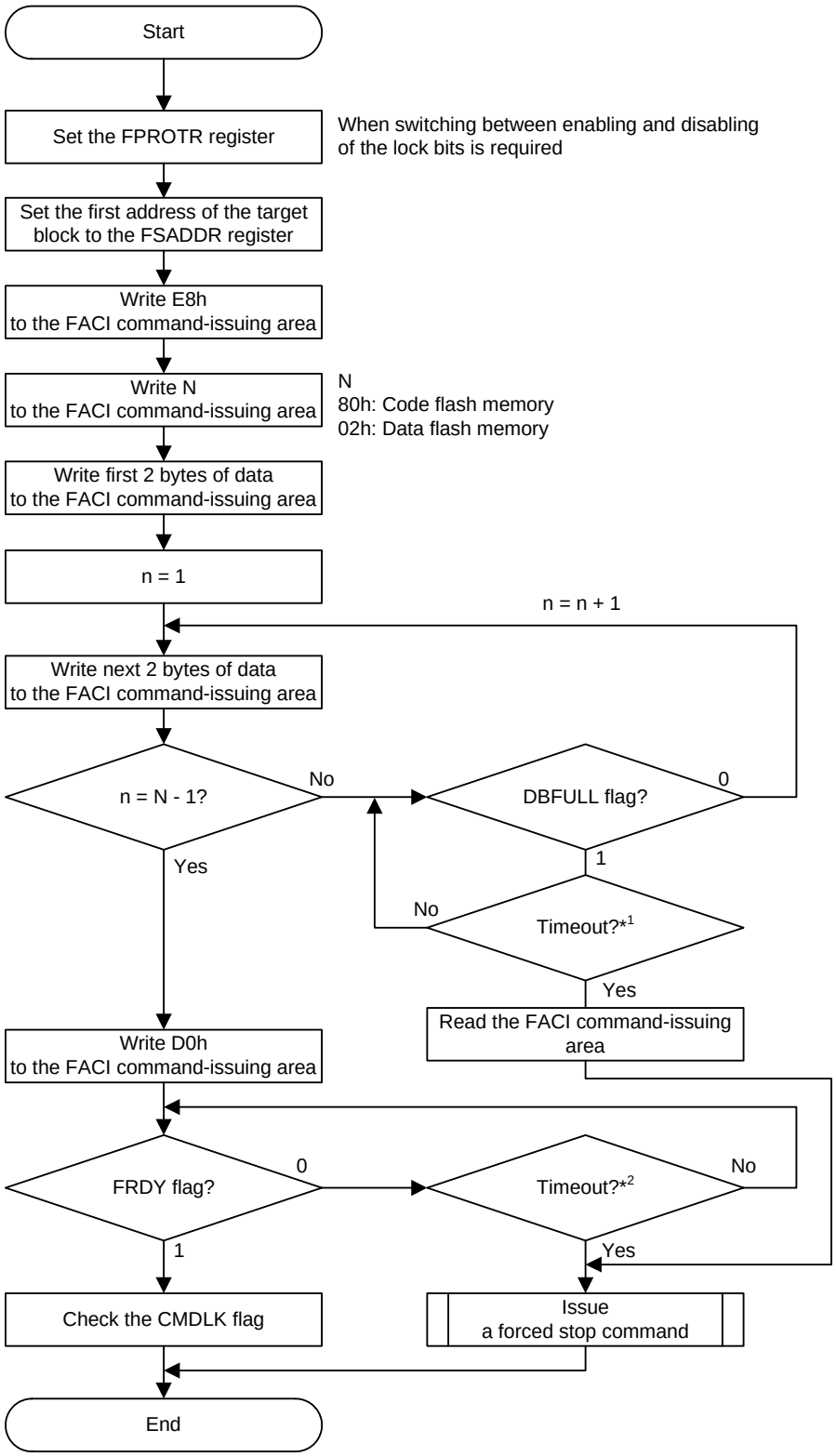


Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (refer to section 45, Electrical Characteristics).

Figure 44.13 Recovery from the Command-Locked State

44.6.6.3 Programming Command

A programming command is used for writing to the user area and data area.
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.
The FPROTR register must be set before issuing a programming command. To switch between enabling and disabling of the lock bits, the setting of the FPROTR register must be changed.
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.
Writing to the data area will not lead to the data buffer becoming full.

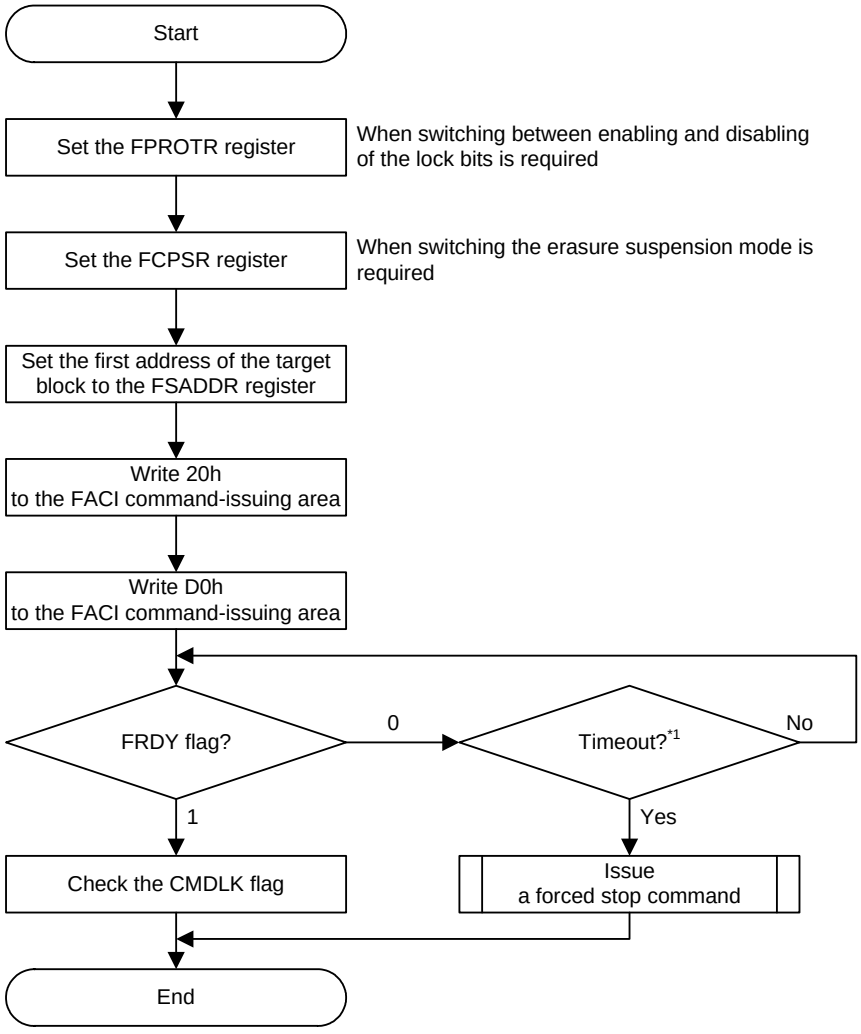


Note 1. Judgment of the timeout is based on 1.4 µs elapsing (when FCLK ≥ 20 MHz)
Note 2. Judgment of the timeout is based on 1.1 times 256-byte programming time (max) for code flash memory. For data flash memory, judgment is based on 1.1 times 4-byte programming time (max). For the 256-byte programming time and 4-byte programming time, refer to section 45, Electrical Characteristics.

Figure 44.14 Usage of the Programming Command

44.6.6.4 Block Erase Command

A block erase command is used to erase the user area, lock bit, and data area 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 checked by reading the FSTATR.FRDY flag.
The FPROTR and FCPSR registers must be set before issuing the block erase command. To switch between enabling and disabling of the lock bits, the setting of the FPROTR register must be changed. To erase the lock bit, issue a block erase command while the FPROTR.FPROTCN bit is 1. 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 256-byte 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 256-byte erasure time and 64-byte erasure time, refer to section 45, Electrical Characteristics.

Figure 44.15 Usage of the Block Erase Command

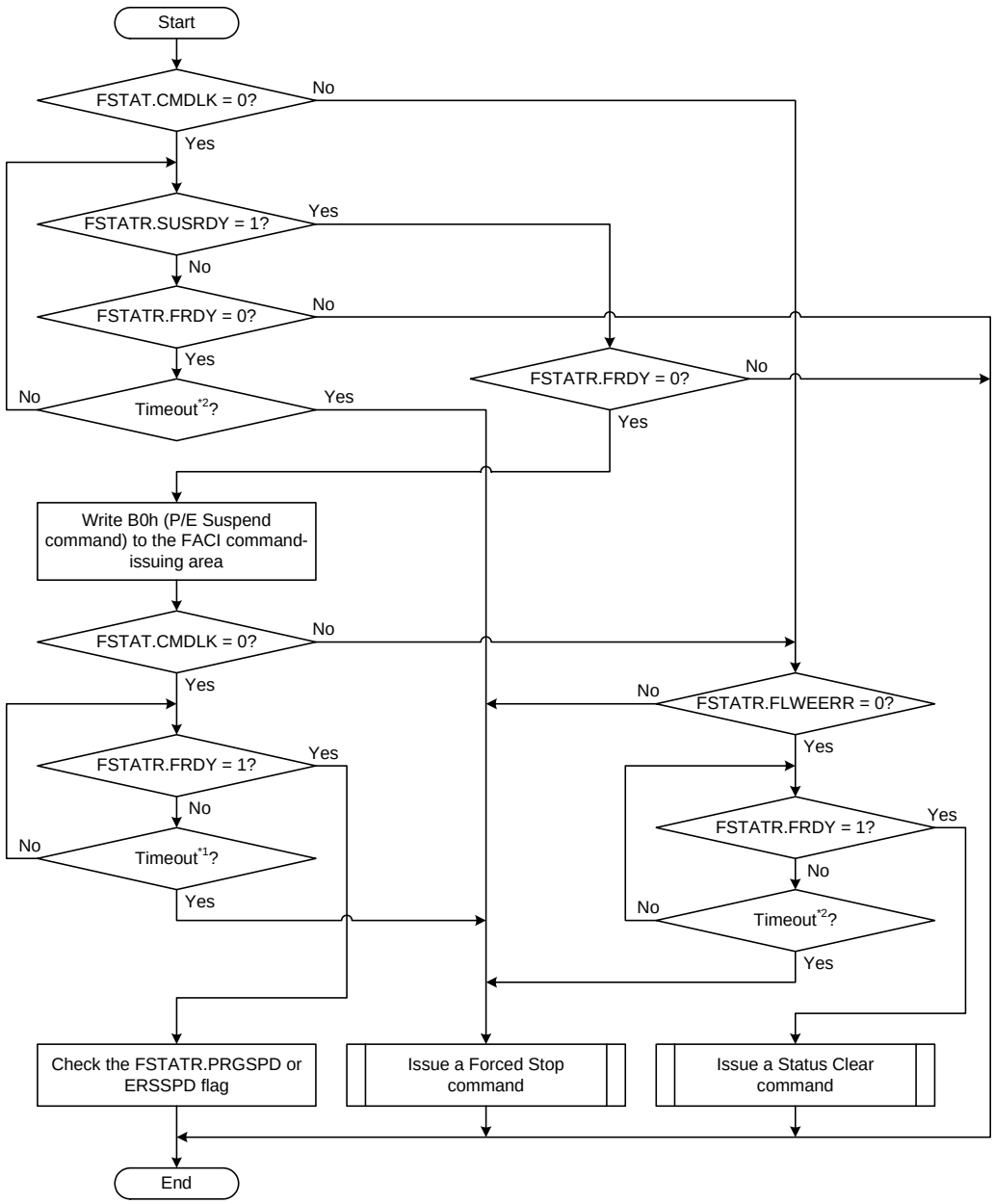
44.6.6.5 P/E Suspend Command

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 (refer to section 44.5.3.2, 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. For both the code flash and data flash memory, judgment of the timeout is based on 1.1 times the maximum value of the erase suspend latency.

Note 2. For both the code flash and data flash memory, judgment of the timeout is based on 1.1 times the maximum value of the erasure time.

For the erase suspend latency and the erasure time, refer to section 45, Electrical Characteristics.

Figure 44.16 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 44.17 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.

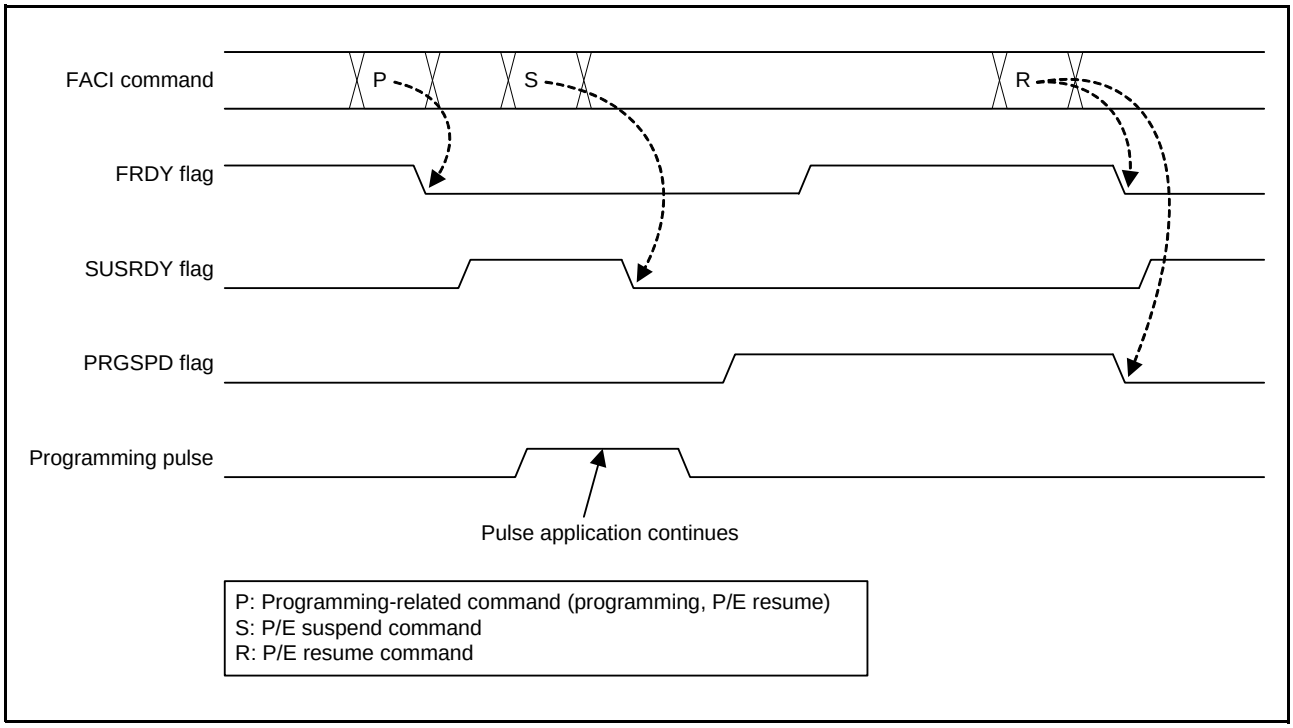


Figure 44.17 Suspension during Programming

(2) Suspension during Erasure (Suspension Priority Mode)

This MCU has a suspension priority mode for the suspension of erasure. Figure 44.18 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.

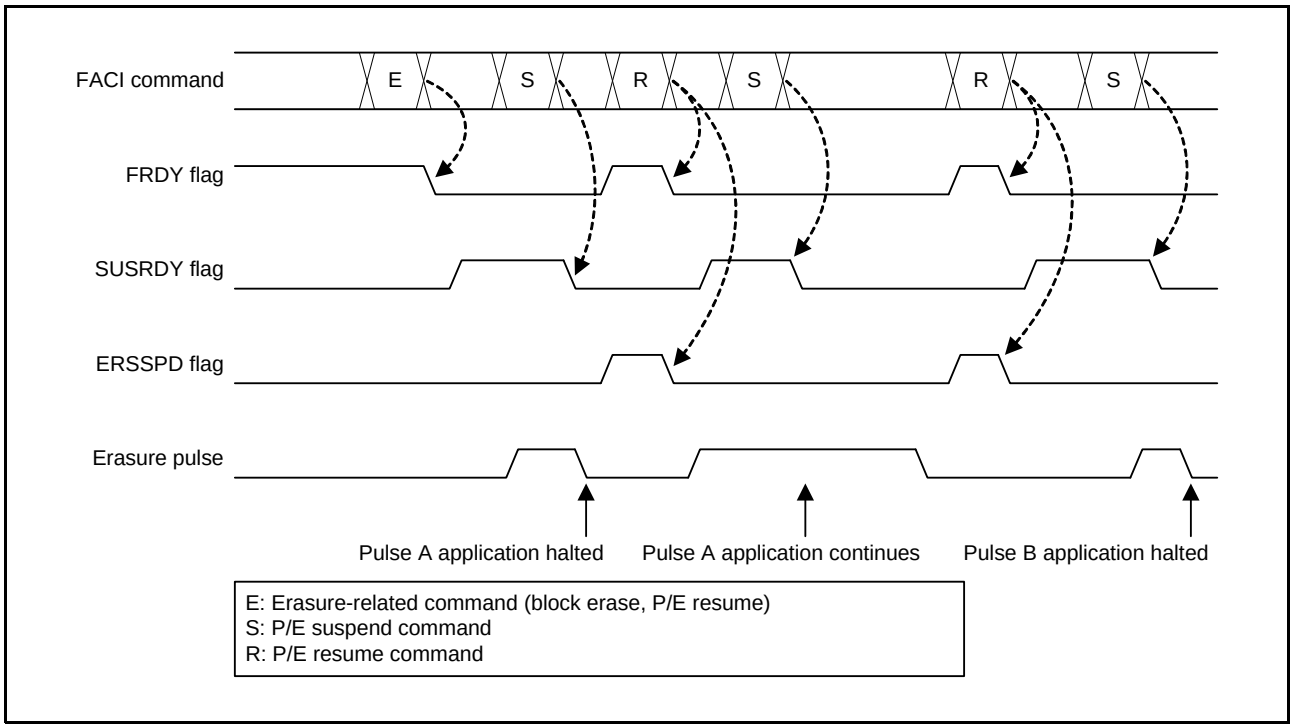


Figure 44.18 Suspension during Erasure (Suspension Priority Mode)

(3) Suspension during Erasure (Erasure Priority Mode)

This MCU has an erasure priority mode for the suspension of erasure.

Figure 44.19 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.

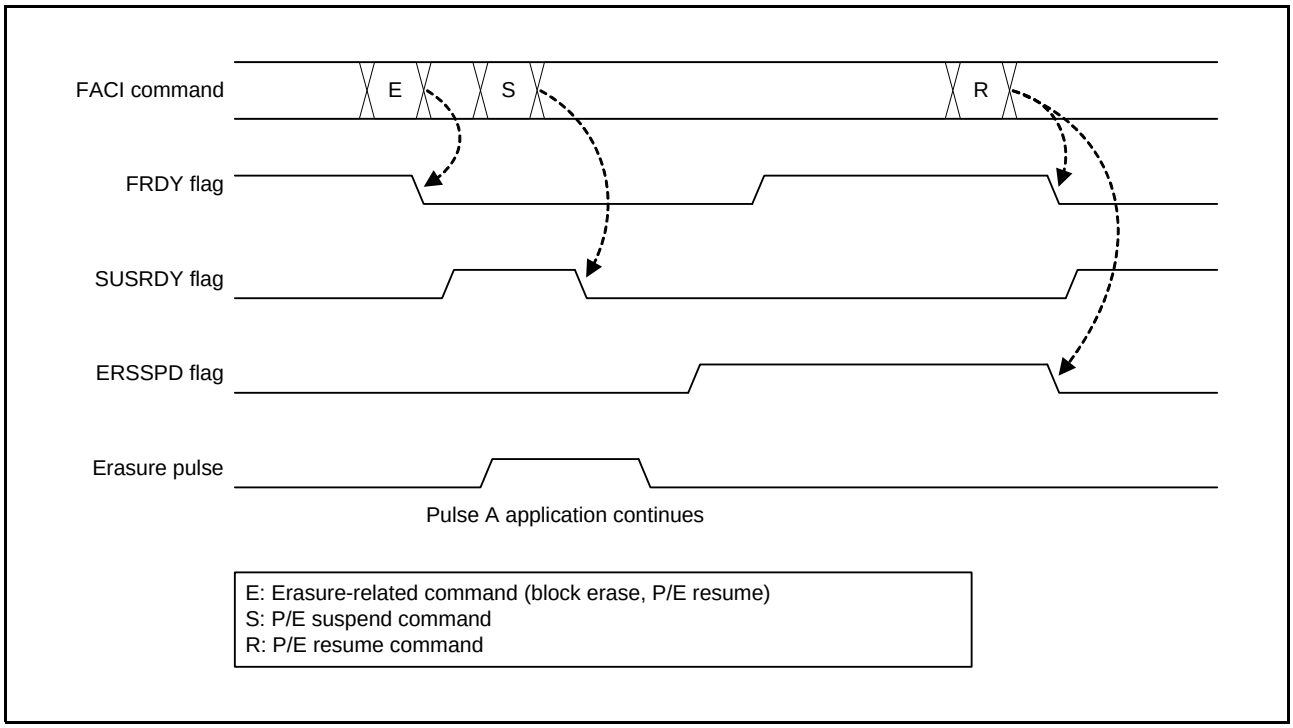
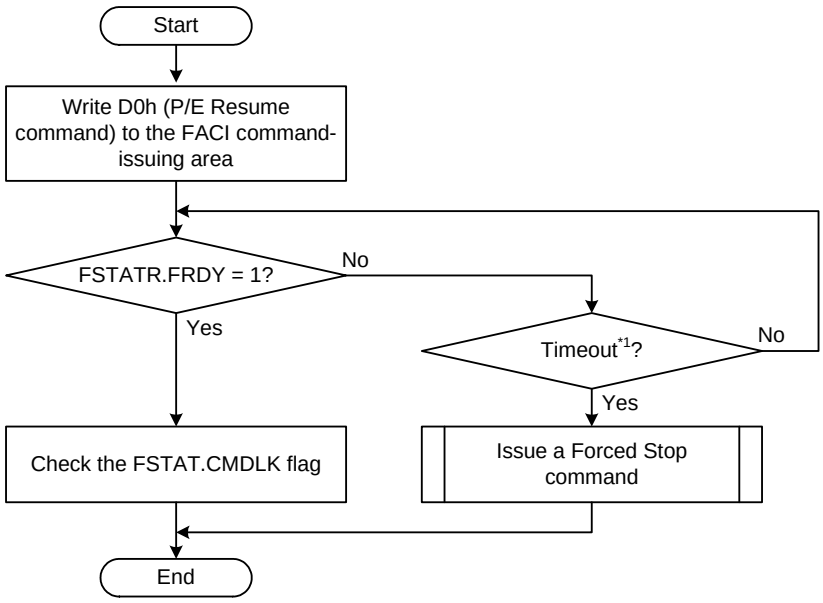


Figure 44.19 Suspension during Erasure (Erasure Priority Mode)

44.6.6.6 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 section 45, Electrical Characteristics).

Figure 44.20 Usage of the P/E Resume Command

44.6.6.7 Status Clear Command

When one of the ILGLERR, 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 (refer to section 44.6.6.2, Recovery from the Command-Locked State). To clear the CMDLK flags in the FASTAT register, and the ILGLERR, ERSERR, and PRGERR flags in the FSTATR register in the command-locked state, the status clear command is available.

The FSTATR.FLWEERR flag cannot only be cleared by the status clearing command, but can only be cleared by the forced end command.

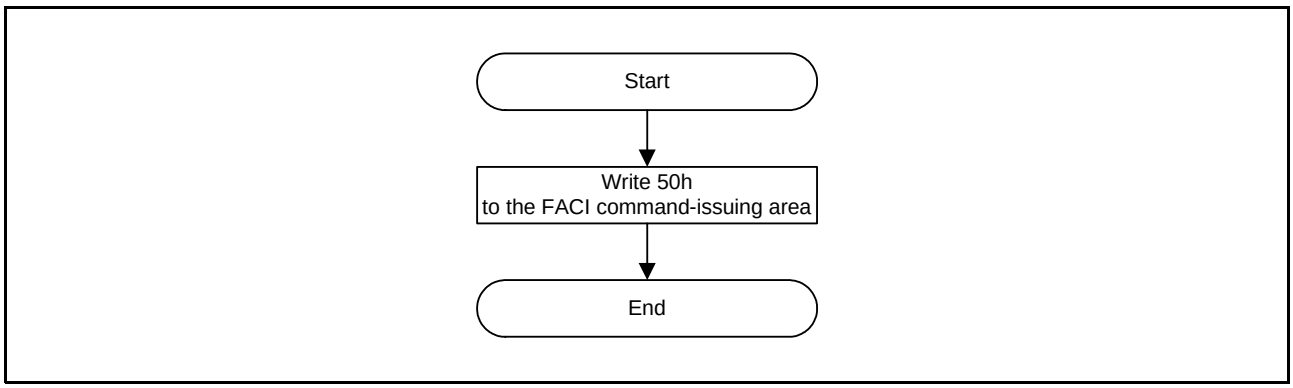
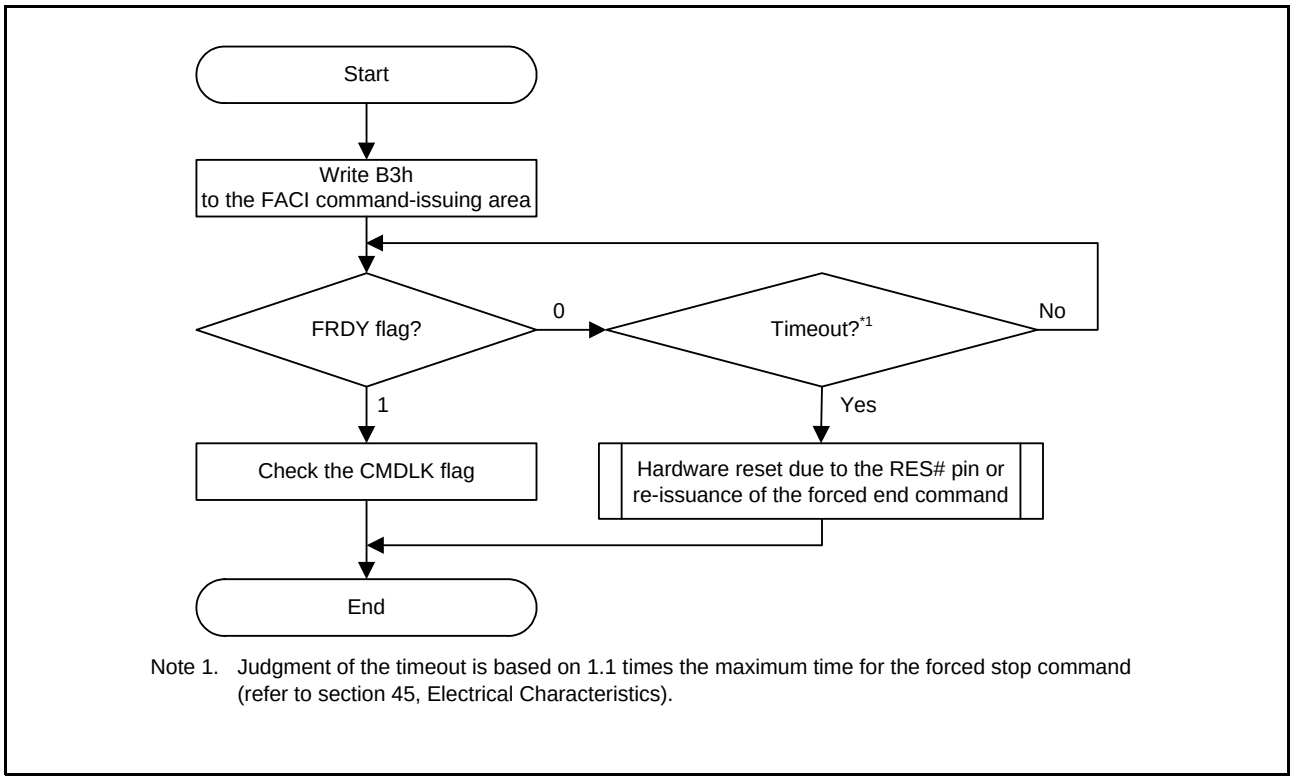


Figure 44.21 Usage of the Status Clear Command

44.6.6.8 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 initializes the whole FCU and a part of the FACI. Also, it initializes the FASTAT.CMDLK flag, and the FSTATR register. 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 (refer to section 44.6.6.2, 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 section 45, Electrical Characteristics).

Figure 44.22 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.

44.6.6.9 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 nonprogrammed state) are undefined. Use the blank check command when you need to confirm that an area is in the nonprogrammed state.

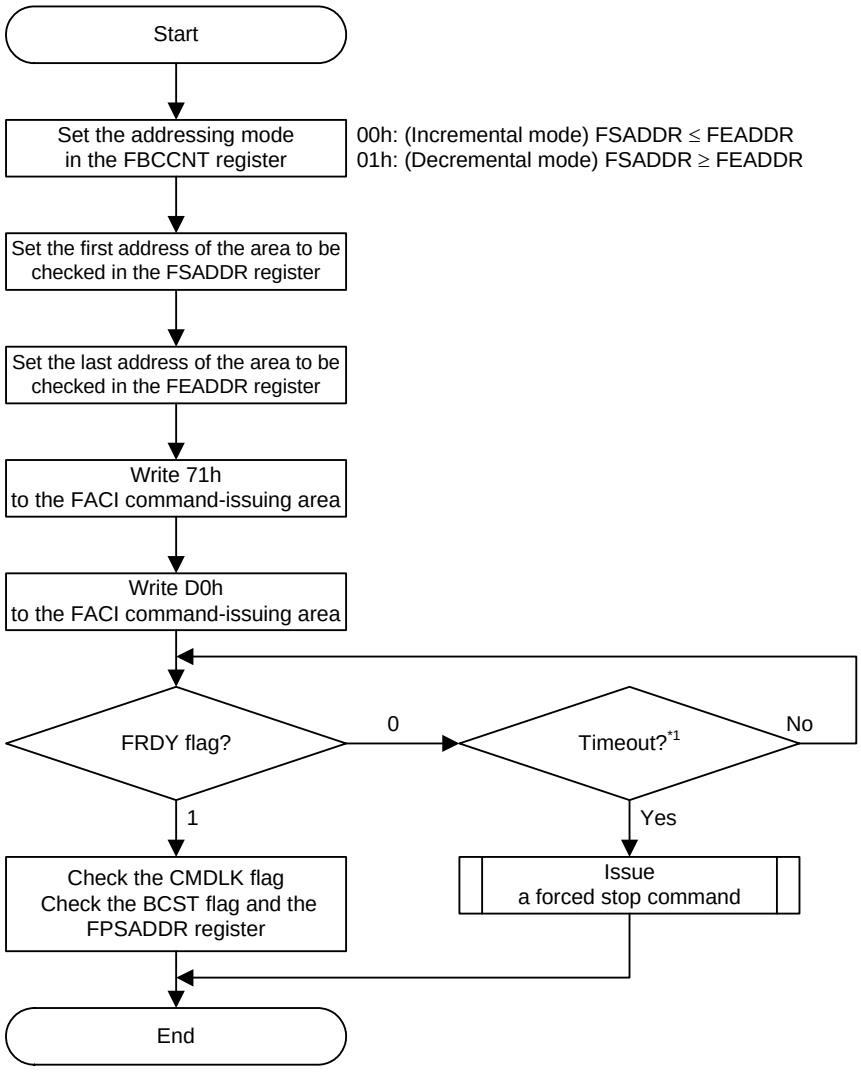
Before issuing a blank check command, set addressing mode, start and end addresses of the target area for blank checking to the FBCCNT, FSADDR, and FEADDR registers.

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 32 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.

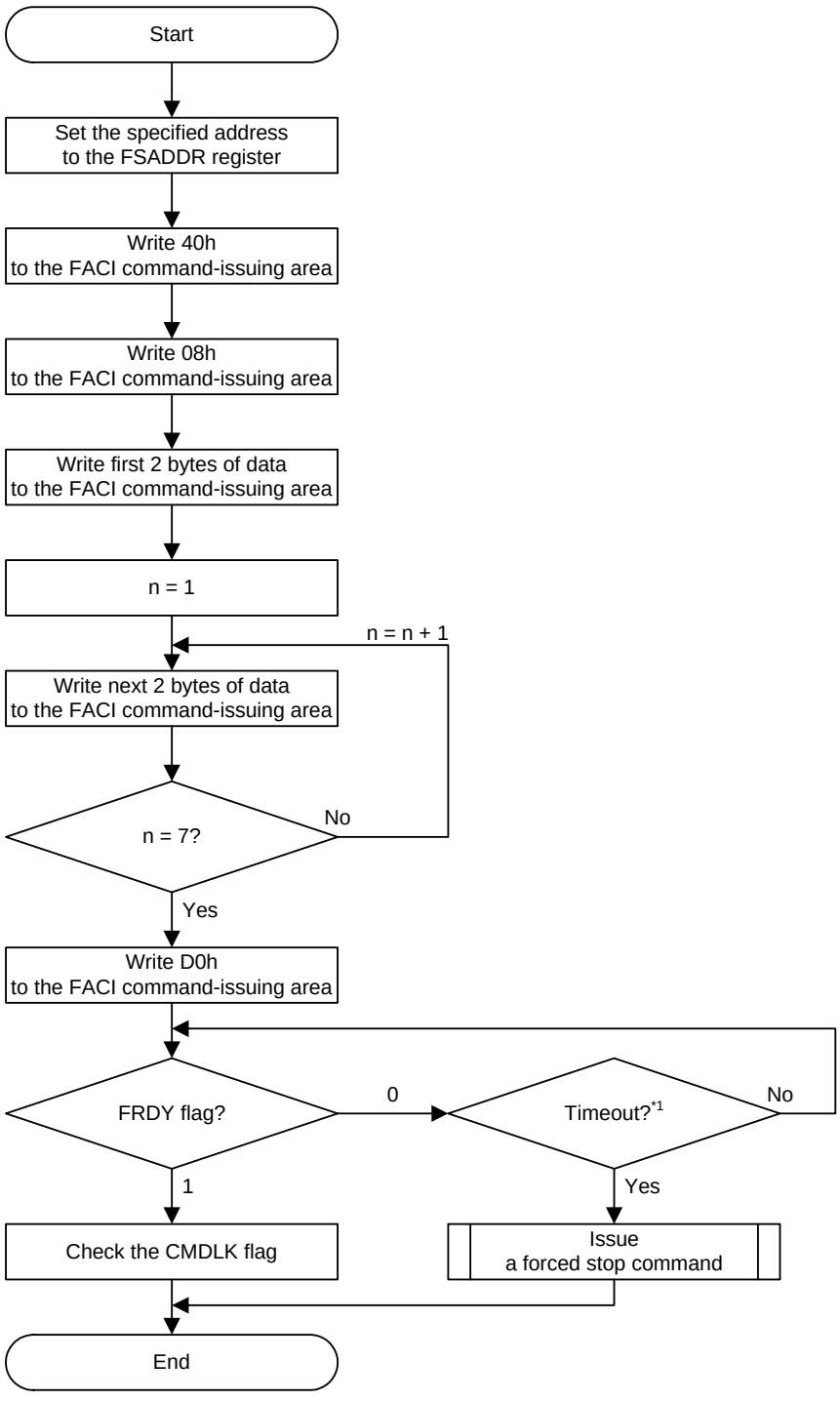


Note 1. Judgment of the timeout is based on 1.1 times the blank checking time (max.) for the entire target area. For the blank checking time, refer to section 45, Electrical Characteristics.

Figure 44.23 Usage of the Blank Check Command

44.6.6.10 Configuration Set Command

The configuration set command is used to set the configuration setting area.
Before issuing a configuration set command, set the specified address (shown in Table 44.17) 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 70ms (when 20 MHz ≤ FCLK ≤ 60 MHz), or 140 ms (when FCLK = 4 MHz).

Figure 44.24 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 44.17. For details on the FSADDR register, refer to section 44.4.5, FACI Command Processing Start Address Register (FSADDR).

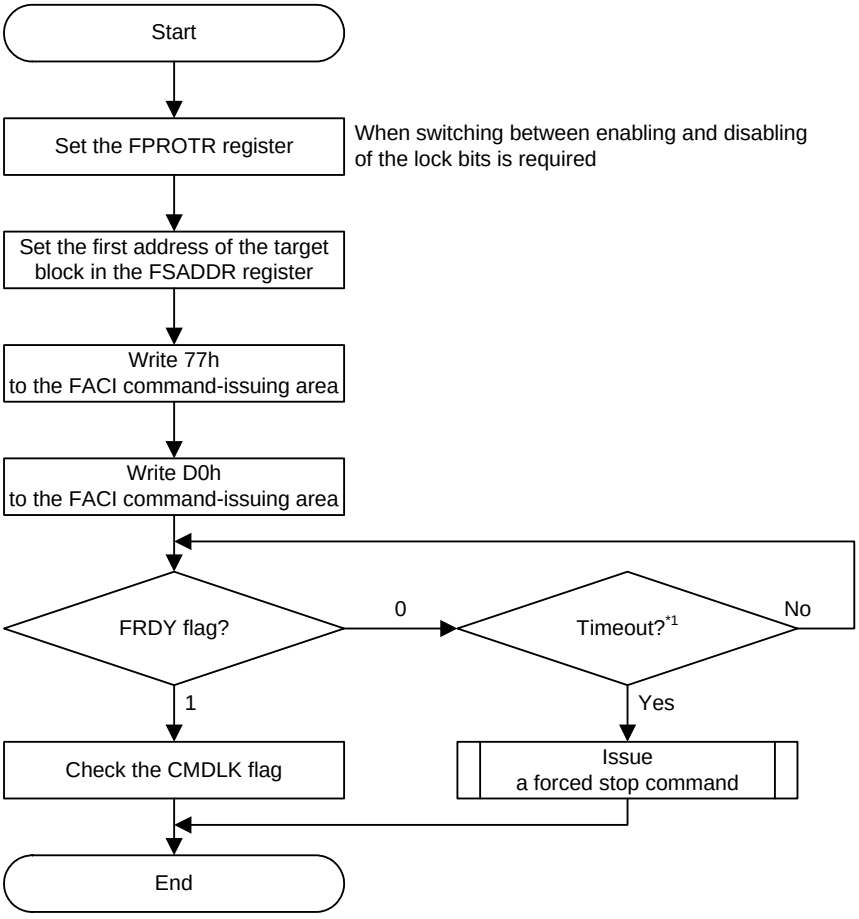
Table 44.17 Address Used by Configuration Set Command

Address	FSADDR Register Value	Setting Data	Timing when the Setting is Enabled
0012 0040h	0000 0040h	Serial programmer command control register (SPCC), TM enable flag register (TMEF)	When a reset or command is executed*1
0012 0050h	0000 0050h	OCD/serial programmer ID setting register (OSIS)	At a reset
0012 0060h	0000 0060h	TM identification data register (TMINF), option function select register 0 (OFS0), option function select register 1 (OFS1), endian select register (MDE)	At a reset
0012 007Ch	0000 0070h	ROM code protection register (ROMCODE)	At a reset

Note 1. 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.

44.6.6.11 Lock-Bit Programming Command

The lock-bit programming command is used to write to a lock bit. The block erase command is used to erase a lock bit (refer to section 44.6.6.4, Block Erase Command).
Before issuing a lock-bit programming command, set the first address of the target block in the FSADDR register.
Writing 77h and D0h to the FACI command-issuing area starts processing of a lock-bit programming command.
The FPROTR register must be set before issuing a lock-bit programming command. To switch between enabling and disabling of the lock bits, the setting of the FPROTR register must be changed.

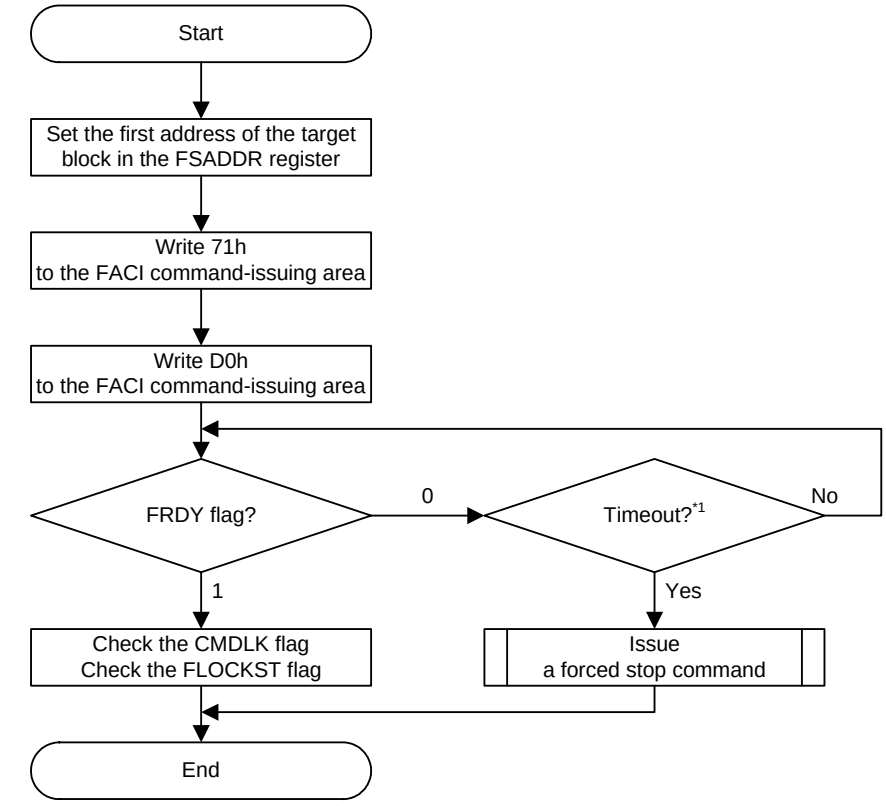


Note 1. Judgment of the timeout is based on 25 ms elapsing (when 20 MHz ≤ FCLK ≤ 60 MHz), 50 ms elapsing (when FCLK = 4 MHz)

Figure 44.25 Usage of the Lock-Bit Programming Command

44.6.6.12 Lock-Bit Read Command

The lock-bit read command is used to read a lock bit.
Before issuing a lock-bit read command, set the first address of the target block in the FSADDR register. Writing 71h and D0h to the FACI command-issuing area starts processing of a lock-bit read command.
Completion of command processing can be confirmed with the FSTATR.FRDY flag. After command processing is completed, the result of reading the lock bit is stored in the FLKSTAT.FLOCKST flag.



Note 1. Judgment of the timeout is based on 10 µs elapsing (when 20 MHz ≤ FCLK ≤ 60 MHz), 20 µs elapsing (when FCLK = 4 MHz)

Figure 44.26 Usage of the Lock-Bit Read Command

44.7 Boot Mode

There are two serial programming modes; the boot mode (SCI interface) with SCI and the boot mode (FINE interface) with FINE. Table 44.18 lists the I/O pins used in boot mode.
I/O pins that are not used in boot mode are in the same state as that after a reset.

Table 44.18 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
PN6/MD	Input	Boot mode (SCI interface) User boot mode	Selection of operating mode
PC7/UB	Input		Selection of boot mode (SCI interface) or user boot mode*1
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)
PN6/MD/FINED	I/O	Boot mode (FINE interface)	Selection of operating mode, FINE data I/O

Note 1. To enable user boot mode, the settings for UB code A and UB code B are required.

44.7.1 Boot Mode (SCI Interface)

In boot mode (SCI interface), the host sends control commands and data for programming, and the user area or data flash memory, and the user boot area can be programmed or erased. 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 (boot program) in a dedicated area within the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.
Figure 44.27 shows the system configuration for operations in boot mode (SCI interface).

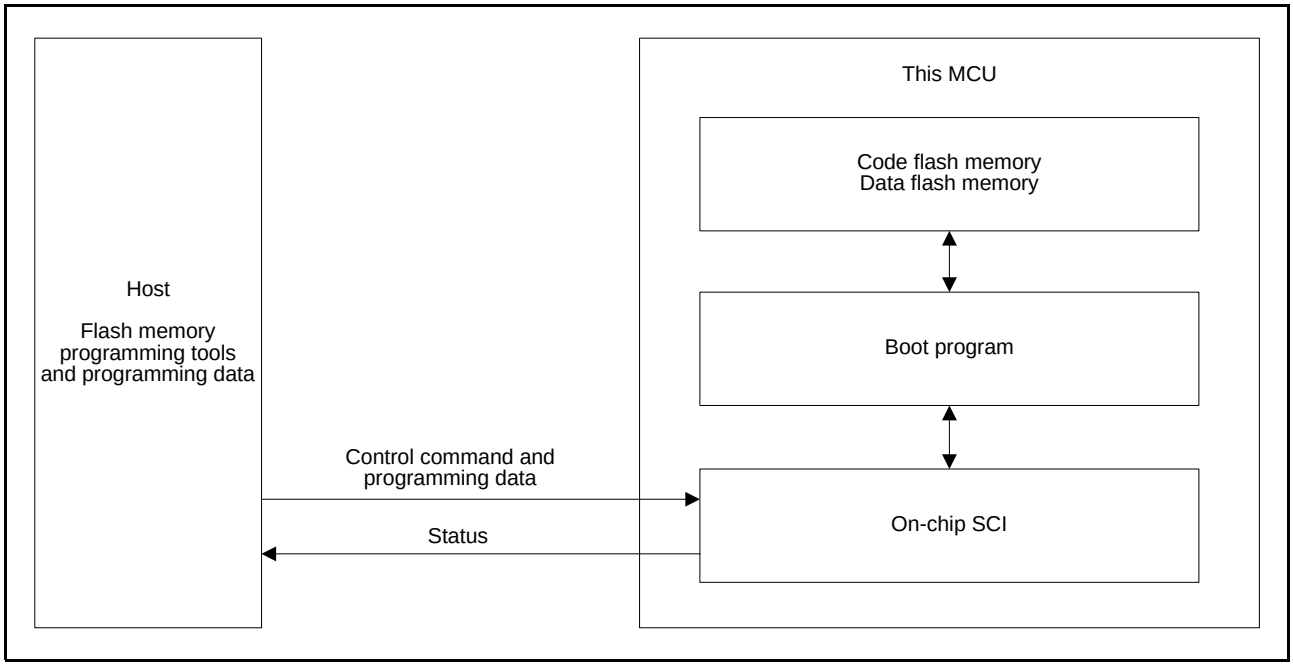


Figure 44.27 System Configuration for Operations in Boot Mode (SCI Interface)

44.7.2 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The user area or data flash memory, and the user boot area can be rewritten.

44.7.2.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE Interface).
Figure 44.28 shows an Example of Pin Connections in Boot Mode (FINE Interface). Table 44.19 lists Pin Handling in Boot Mode (FINE Interface).
The example of pin connections shown in Figure 44.28 is a simplified circuit. Operations are not guaranteed in all systems.

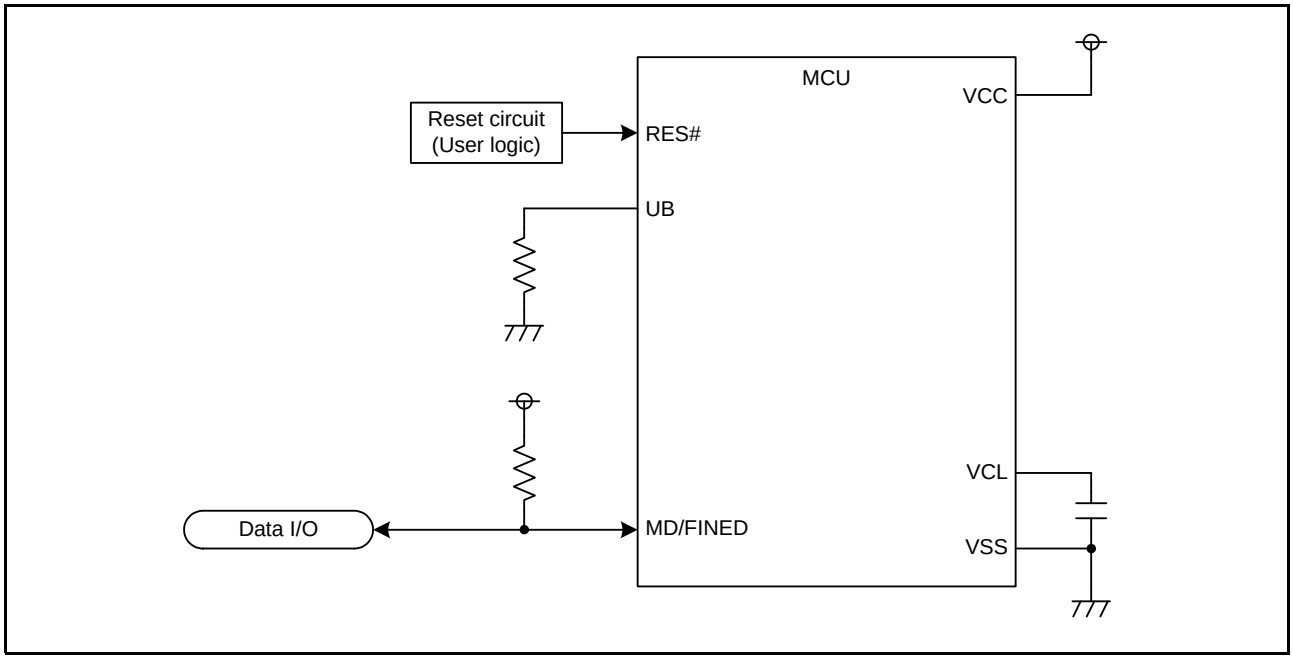


Figure 44.28 Example of Pin Connections in Boot Mode (FINE Interface)

Table 44.19 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 0.47-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).
UB	Operating mode control	Input	Connect the VSS pin via a resistor (pull down).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

44.8 Flash Memory Protection

Flash memory protection prevents the flash memory from being read or rewritten by the third party.
The serial programmer ID code protection is for connecting the serial programmer, and the on-chip debugger ID code protection is for connecting the on-chip debugger. ROM code protection is for connecting the parallel programmer.

44.8.1 ID Code Protection

There are two types of ID code protection: The serial programmer ID code protection for connecting the serial programmer and on-chip debugger ID code protection is for connecting the on-chip debugger. The same ID codes are used for both functions, but operations differ.
For details, refer to section 7.2.2, OCD/Serial Programmer ID Setting Register (OSIS).

44.8.1.1 On-Chip Debugger ID Code Protection

This function is used to prohibit connection with the on-chip debugger (OCD).
The code sent from the OCD is compared with the ID code in the OCD/serial programmer ID setting register (OSIS) to determine whether they match. If they match, connection with the OCD is allowed. If they do not match, the OCD cannot be connected.

44.8.1.2 Serial Programmer ID Code Protection

This function is used to prohibit connection with the serial programmer. When the MCU is connected to a serial programmer, the ID code set in the OCD/serial programmer ID setting register (OSIS) and written in the option-setting memory is used to judge ID code protection on connection of the serial programmer.
The code sent from the serial programmer is compared with the ID code set in the OCD/serial programmer ID setting register (OSIS) to determine whether they match. If they match, connection with the serial programmer is allowed. If they do not match, the serial programmer cannot be connected. 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.
In user boot mode, ID code protection function is not available.

44.8.2 ROM Code Protection

ROM code protection is a function to prohibit reading from or programming/erasure to the flash memory when the parallel programmer is used during off-board programming.
For details of the ROM code protection register, refer to section 7.2.8, ROM Code Protection Register (ROMCODE).

44.9 Boot Mode Communications Protocol

This section describes the communications protocol for use in boot mode. When developing a programmer, use this communications protocol to control it.

44.9.1 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 44.29 shows the states of pins up to communications (through the SCI interface) in boot mode becoming possible.

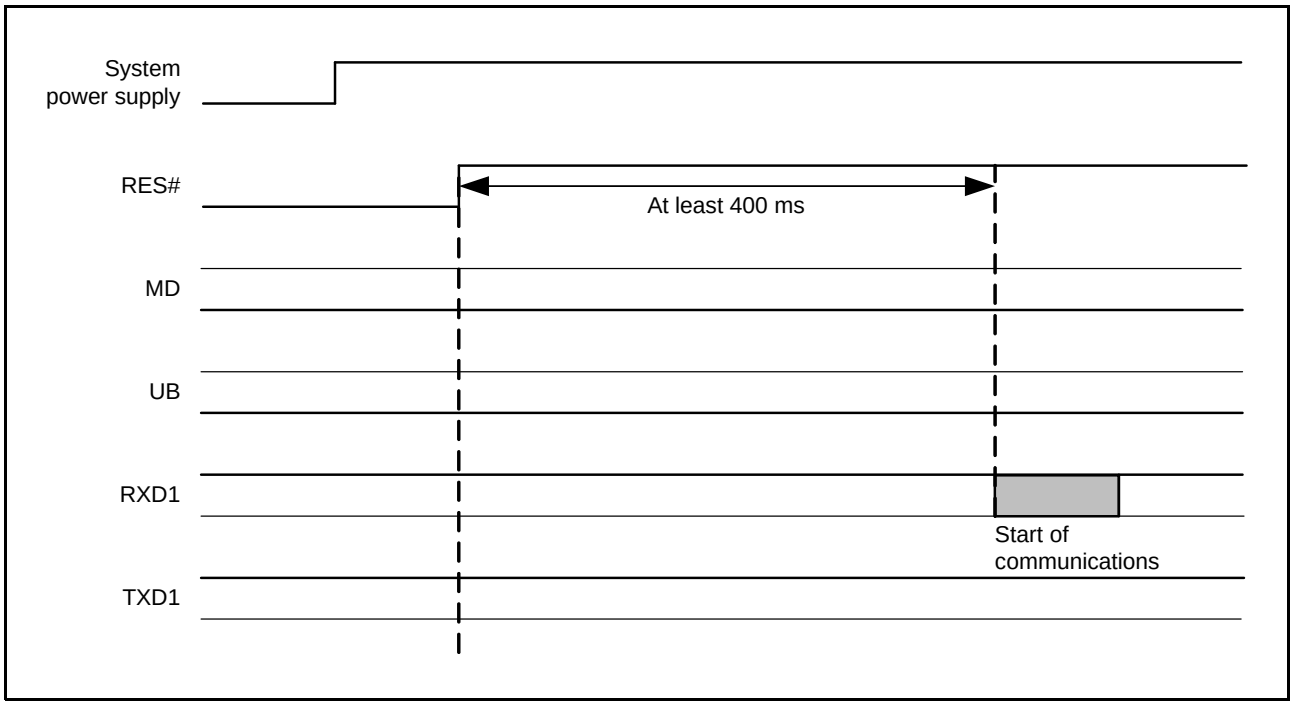


Figure 44.29 States of Pins Up to Communications (through the SCI Interface) in Boot Mode Becoming Possible

44.9.2 State Transitions in Boot Mode

44.9.2.1 State Transitions in Boot Mode (SCI Interface)

Figure 44.30 shows a flowchart for transition to boot mode (SCI interface).

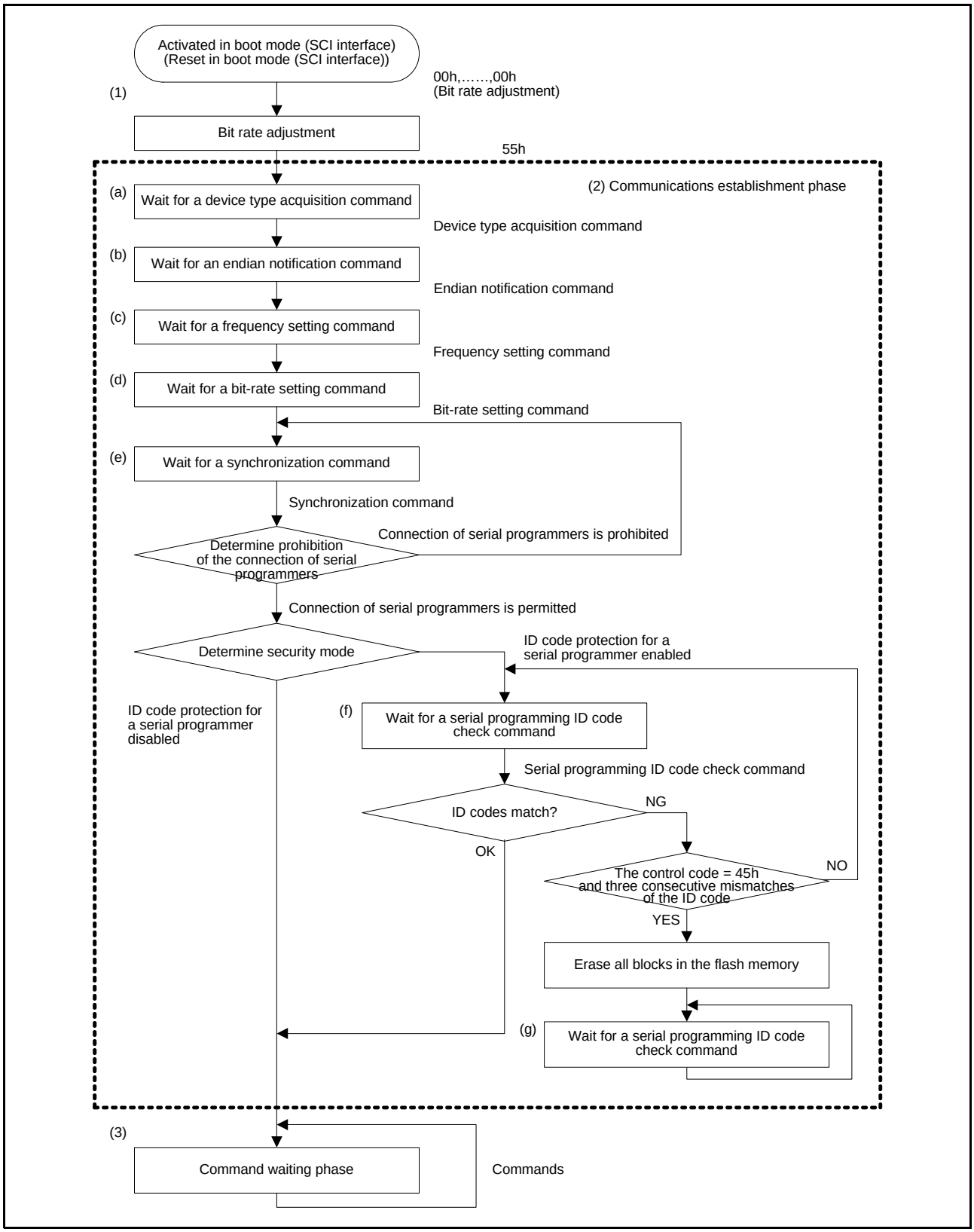


Figure 44.30 Flowchart for Transition to Boot Mode (SCI Interface)

(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, refer to section 44.9.3, Automatic Adjustment of the Bit Rate.

(2) Communications establishment phase

The device, endian, frequency, and bit rate are selected in this phase. The ID code from the serial programmer is judged in this phase to see if protection should be applied. For the commands to use in the communications establishment phase, refer to section 44.9.5, 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, refer to section 44.9.9, 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, refer to section 44.9.10, 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, refer to section 44.9.11, 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, refer to section 44.9.12, 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 determines whether ID code protection is enabled or disabled. When ID code protection is disabled, the MCU enters the command waiting phase. When ID code protection is enabled, it enters the state of waiting for a serial programming ID code check command. If the MCU has been set to prohibit the connection of a serial programmer, this MCU transmits an error code to indicate that connecting a serial programmer is prohibited and remains in the state of waiting for a synchronization command. For details of the synchronization command, refer to section 44.9.13, 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 ID code sent is compared with the ID code written in the option-setting memory area, and the command waiting phase is entered if the two match. If they do not match, the next transition is back to the state of waiting for a serial programming ID code check command. 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.
For details of the ID code check command, refer to section 44.9.15, Serial Programming ID Code Check Command.

(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, refer to section 44.9.6, Command Waiting Phase.

44.9.3 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 C1h to the host, and if successful reception of 55h by this MCU is not possible, it responds by sending FFh to the host.

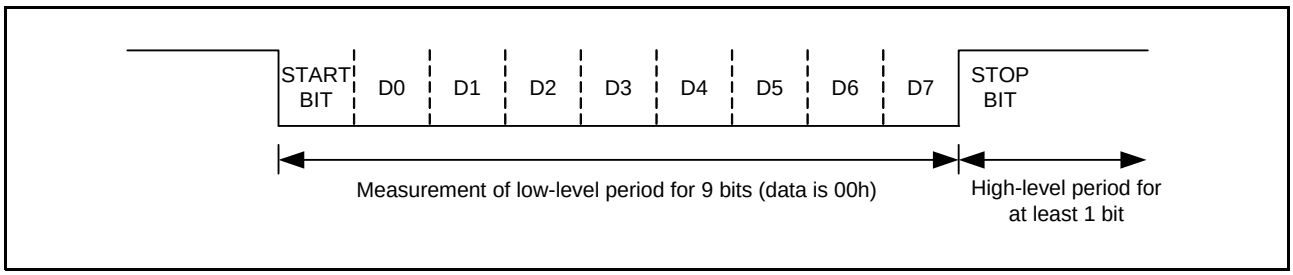


Figure 44.31 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

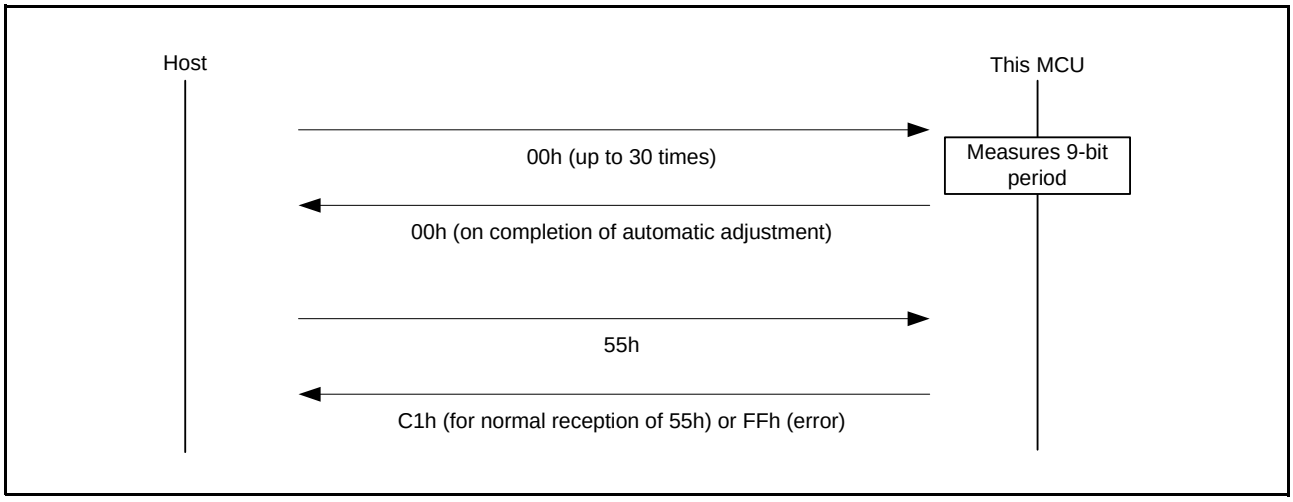


Figure 44.32 Sequence of Transfer between Host and This MCU

For the host's SCI bit rate, ensure that SCI communications proceed under the conditions given in Table 44.20.

Table 44.20 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host
9,600 bps

44.9.4 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.

44.9.5 Communications Establishment Phase

Table 44.21 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 44.21 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/disabled state of ID code protection for serial programmers to the host.
Serial programming ID code check	Determines whether the control code or ID code written in the option-setting memory matches the control code or ID code sent by the host.

In the communications establishment phase, send commands from the host in the order of the device type acquisition, endian notification, frequency setting, bit-rate setting, and synchronization commands according to the responses to commands. When ID code protection for serial programmers is enabled, send the ID authentication mode acquisition or serial programming ID code check command following the synchronization command.
If commands are issued in an incorrect order or other commands are issued, this MCU returns a response indicating a flow error.

44.9.6 Command Waiting Phase

Table 44.22 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 44.22 Commands Available in the Command Waiting Phase

Command Name	Function
Synchronization	Refer to Table 44.21.
Blank check	Check that a selected area is blank.
Block erase	Erases a selected single block.
Area erasure	Erases the specified area.
Programming	Programs the selected area.
Read	Reads data from a selected area.
Lock-bit setting	Sets the lock bits.
Lock-bit acquisition	Obtains the settings of lock bits.
Lock bit enable	Enables the setting of lock bits.
Lock bit disable	Disables the setting of lock bits.
ID authentication mode acquisition	Refer to Table 44.21.
Command protection setting	Enables prohibition of block erase commands, programming commands, or read commands.
Command protection acquisition	Obtains the settings for protection against block erase commands, programming commands, and read commands.
Serial programming ID code setting	Sets the OSIS register. Also, sets the SPCC register and enables ID code protection.
ID code setting	Sets the OSIS register.
ID code acquisition	Obtains the OSIS setting.
Serial programmer connection prohibition	Sets the SPCC.SPE bit and prohibits the connection of serial programmers.
On-chip debugger connection prohibition	Sets the SPCC.OCDE bit and prohibits the connection of on-chip debuggers.
OCDE acquision	Obtains the OCDE bit setting.
OFS setting	Sets the OFS0 and OFS1 registers.
OFS acquisition	Obtains the setting of the OFS0 and OFS1 registers.
Endian setting	Sets the MDE register.
Endian acquisition	Obtain the setting of the MDE register.
Configuration clearing	Erases the configuration setting areas and the TM target areas.
TM setting	Sets the TMEF and TMINF registers and enables the TM function.
TM acquisition	Acquires an indicator of whether the TM function is currently enabled or disabled, the contents of the TMINF register, and the addresses where the TM target areas start and end.
Simple addition checksum	Calculates the sum of values in a selected area.
Signature acquisition	Obtains information of the flash memory configuration.
ROM code setting	Sets the ROM code protection register.
ROM code acquisition	Obtains the settings of the ROM code protection register.

If the host has sent an undefined command, this MCU returns a response indicating an error in the form of a non-supported command.

44.9.7 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, refer to the sections on the individual commands for details.

(1) Common transfer sequence for the commands that only make settings

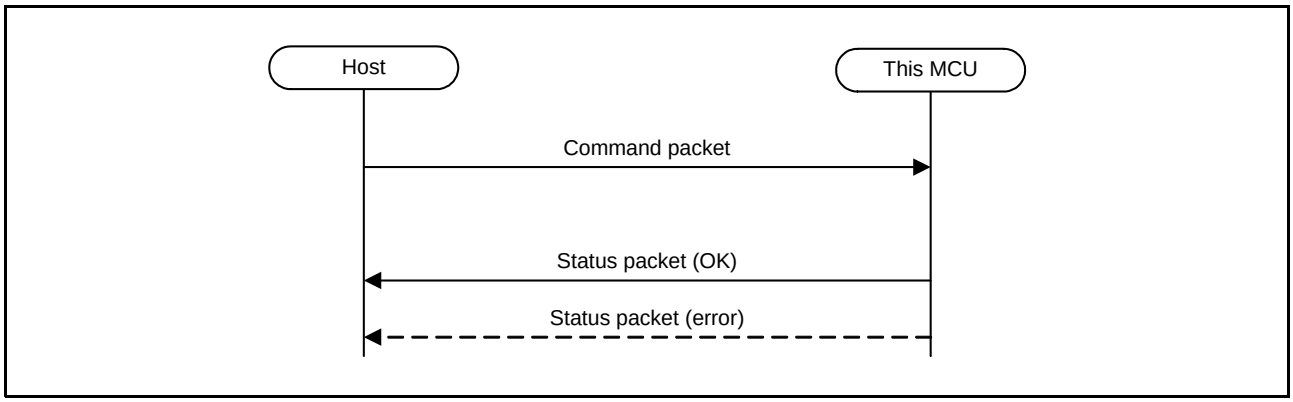


Figure 44.33 Common Transfer Sequence for the Commands that Only Make Settings

(2) Common transfer sequence for the commands that obtain information on settings

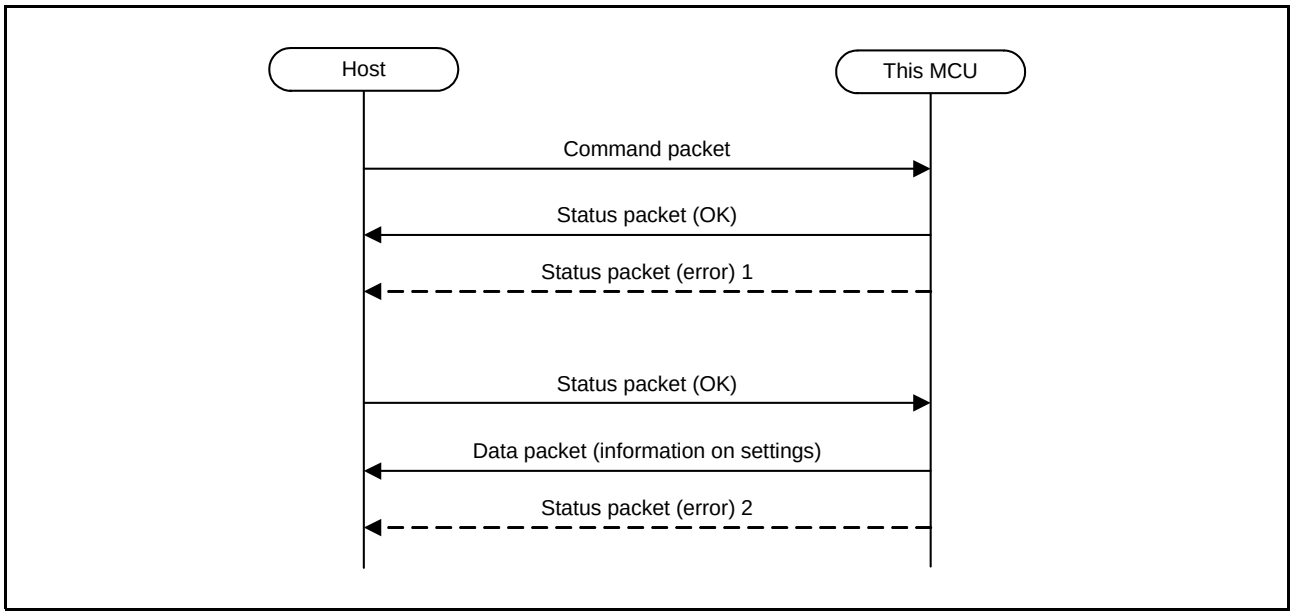


Figure 44.34 Common Transfer Sequence for the Commands that Obtain Information on Settings

Table 44.23 Common Transfer Sequence

Command Name	Common Transfer Sequence Type
Device type acquisition	Command that obtains information on settings
Endian notification	Command that only makes a setting
Frequency setting	Command that obtains information on settings
Bit rate setting	Not in a common transfer sequence
Synchronization	Command that only makes a setting
ID authentication mode acquisition	Command that obtains information on settings
Serial programming ID code check	Command that only makes a setting
Blank check	Command that only makes a setting
Block erase	Command that only makes a setting
Area erasure	Command that only makes a setting
Programming	Not in a common transfer sequence
Read	Not in a common transfer sequence
Lock-bit setting	Command that only makes a setting
Lock-bit acquisition	Command that obtains information on settings
Lock bit enable	Command that only makes a setting
Lock bit disable	Command that only makes a setting
Command protection setting	Command that only makes a setting
Command protection acquisition	Command that obtains information on settings
Serial programming ID code setting	Command that only makes a setting
ID code setting	Command that only makes a setting
ID code acquisition	Command that obtains information on settings
Serial programmer connection prohibition	Command that only makes a setting
On-chip debugger connection prohibition	Command that only makes a setting
OCDE acquision	Command that obtains information on settings
OFS setting	Command that only makes a setting
OFS acquisition	Command that obtains information on settings
Endian setting	Command that only makes a setting
Endian acquisition	Command that obtains information on settings
Configuration clearing	Command that only makes a setting
TM setting	Command that only makes a setting
TM acquisition	Command that obtains information on settings
Simple addition checksum	Command that obtains information on settings
Signature acquisition	Command that obtains information on settings
ROM code setting	Command that only makes a setting
ROM code acquisition	Command that obtains information on settings

For the command transfer sequences that are not in a common transfer sequence, refer to the sections on the individual commands.

44.9.8 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 44.22.

(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

44.9.9 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

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.

44.9.10 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.

44.9.11 Frequency Setting Command

This command is used to set the values of oscillation frequency and CPU operating frequency (in Hz).
In boot mode (SCI interface) or boot mode (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. Furthermore, in boot mode (SCI interface) or boot mode (FINE interface), the FCLK and PCLKB run at 60 MHz, so the MCU sends that the frequency of the peripheral module clock is 60 MHz.

This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OC1	OC2	OC3	OC4	CC1	CC2	CC3	CC4	SUM	ETX

When the input frequency is 16,000,000 Hz and the system clock frequency is 120,000,000 Hz, send the values as below.

OC1: 00h	CC1: 07h
OC2: F4h	CC2: 27h
OC3: 24h	CC3: 0Eh
OC4: 00h	CC4: 00h

SOH: 01h
LNH: 00h
LNL: 09h
COM: 32h
OC1: Input frequency
OC2: Input frequency
OC3: Input frequency
OC4: Input frequency
CC1: System clock frequency
CC2: System clock frequency
CC3: System clock frequency
CC4: System clock frequency
SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	FQ1	FQ2	FQ3	FQ4	PF1	PF2	PF3	PF4	SUM	ETX

An example of the values sent is given below.
System clock frequency = 120,000,000 Hz
Peripheral clock frequency = 60,000,000 Hz

FQ1: 07h	PF1: 03h
FQ2: 27h	PF2: 93h
FQ3: 0Eh	PF3: 87h
FQ4: 00h	PF4: 00h

SOD: 81h
LNH: 00h
LNL: 09h
RES: 32h
FQ1: System clock frequency
FQ2: System clock frequency
FQ3: System clock frequency
FQ4: System clock frequency
PF1: Peripheral clock frequency
PF2: Peripheral clock frequency
PF3: Peripheral clock frequency
PF4: Peripheral clock frequency
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 32h (OK)
SUM: CDh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D1h (input frequency error)
D2h (system clock (ICLK) frequency error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.12 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 (FINE interface), switching of the bit rate does not proceed. Set the desired value for the bit rate.

(1) Procedure

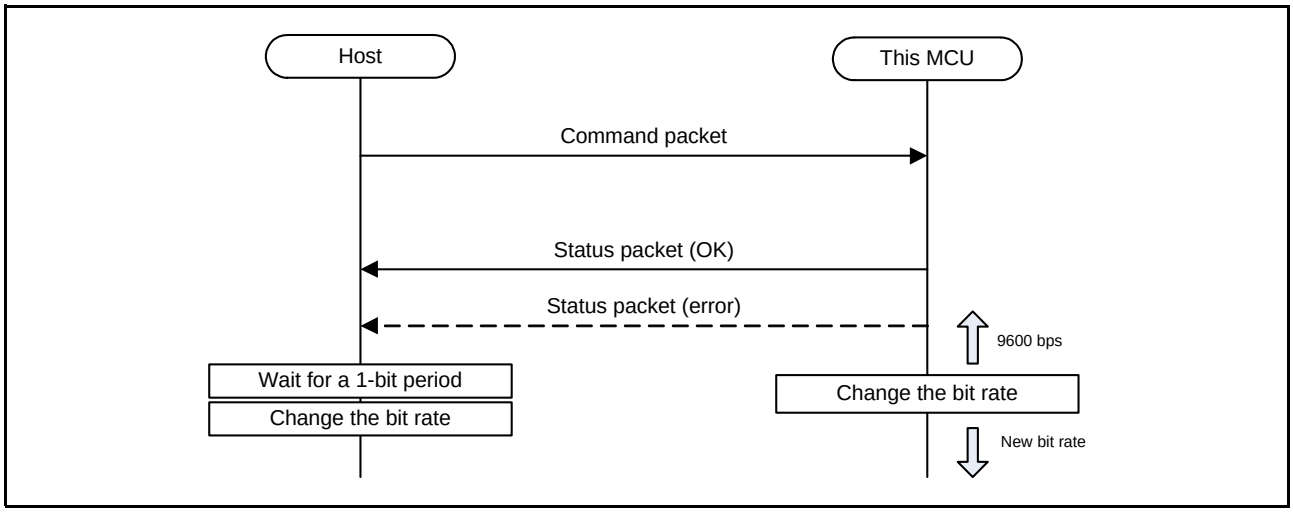


Figure 44.35 Bit-Rate Setting Command Transfer Sequence

(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.

44.9.13 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

44.9.14 ID Authentication Mode Acquisition Command

This command sends the enabled/disabled state of ID code protection for serial programmers 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 (ID code protection for serial programmers enabled)
FFh (ID code protection for serial programmers disabled)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Ch (OK)
SUM: D3h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.15 Serial Programming ID Code Check Command

This command determines whether the ID code set in the OSIS register matches the ID code sent by the host, and sends the result to the host.
This command can be accepted in the communications establishment phase. When ID code protection for serial programmers 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 and the codes do not match three times in a row, all data in the flash memory are erased.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 30h
IDC: Control code or 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 (erase error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.16 Blank Check Command

This command checks whether the specified area is blank.
Specify a range in the user area or user boot area with addresses on 256-byte boundaries or in the data area with addresses on 16-byte boundaries. When the TM function is enabled, an attempt at blank checking of the code flash memory in blocks 8 and 9 leads to an error since they are being handled as TM target area.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 10h
SHH: Blank check start address (24 to 31 bits)
SHL: Blank check start address (16 to 23 bits)
SLH: Blank check start address (8 to 15 bits)
SLL: Blank check start address (0 to 7 bits)
EHH: Blank check end address (24 to 31 bits)
EHL: Blank check end address (16 to 23 bits)
ELH: Blank check end address (8 to 15 bits)
ELL: Blank check end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 10h (OK)
SUM: EFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 90h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
E0h (non-blank error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.17 Block Erase Command

This command is used to erase a specified single block.
Specify the block for erasure as the first address in the block.
If the lock-bit function is disabled and the lock bit of a block for erasure is set, the lock bit is also erased so the block is released from locking. While the TM function is enabled, blocks 8 and 9 in the user area are handled as TM target areas and so cannot be erased.
To erase areas being handled as TM target, use the configuration clearing command described in section 44.9.37, Configuration Clearing Command to restore them to normal handling.
If programming commands are prohibited, follow the procedure for erasure prescribed in section 44.9.45, Flow for Erasure when Programming Commands are Prohibited.
This command cannot be used when block erase commands are prohibited.
If processing of erasure is forcibly stopped in response to a command due to a reset, the lock bit may be set. In such a case, refer to (6) Abnormal Termination During Programming/Erasure or Blank Checking in section 44.12, Usage Notes.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 05h
COM: 12h
SHH: First address in the block for erasure (24 to 31 bits)
SHL: First address in the block for erasure (16 to 23 bits)
SLH: First address in the block for erasure (8 to 15 bits)
SLL: First address in the block for erasure (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 12h (OK)
SUM: EDh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 92h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
E1h (erase error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.18 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 user area, user boot area, or data area.
When an area for which the lock bit function is disabled and the lock bit has been set is erased, the lock-bit setting is cleared.
When the TM function is enabled, only the blocks other than those in the TM target areas are erased.
If programming commands are prohibited, follow the procedure for erasure prescribed in section 44.9.45, Flow for Erasure when Programming Commands are Prohibited.
This command cannot be used when block erase commands are prohibited.
If processing in response to the command is forcibly stopped due to a reset, etc., while erasure is in progress, the lock bit may be set. In such a case, refer to (6) Abnormal Termination During Programming/Erasure or Blank Checking in section 44.12, Usage Notes.
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 (user area)
10h (user boot area)
20h (data area)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 50h (OK)
SUM: AFh
ETX: 03h

(3) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	SUM	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D0h (Error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D5h (area error)
DAh (protection error)
E1h (erase error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.19 Programming Command

This command sets up the MCU for data to be programmed to its flash memory and specifies the area where the data are to be programmed.
Specify addresses for a data length that is a multiple of 256 bytes for the user area or user boot area, or a multiple of 16 bytes for the data area. Furthermore, as the address where programming is to start, specify an address on a 256-byte boundary for the user area or user boot area or on a 16-byte boundary for the data area. When the TM function is enabled, blocks 8 and 9 in the user area are handled as TM target areas and so cannot be programmed.
This command cannot be used when programming commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Procedure

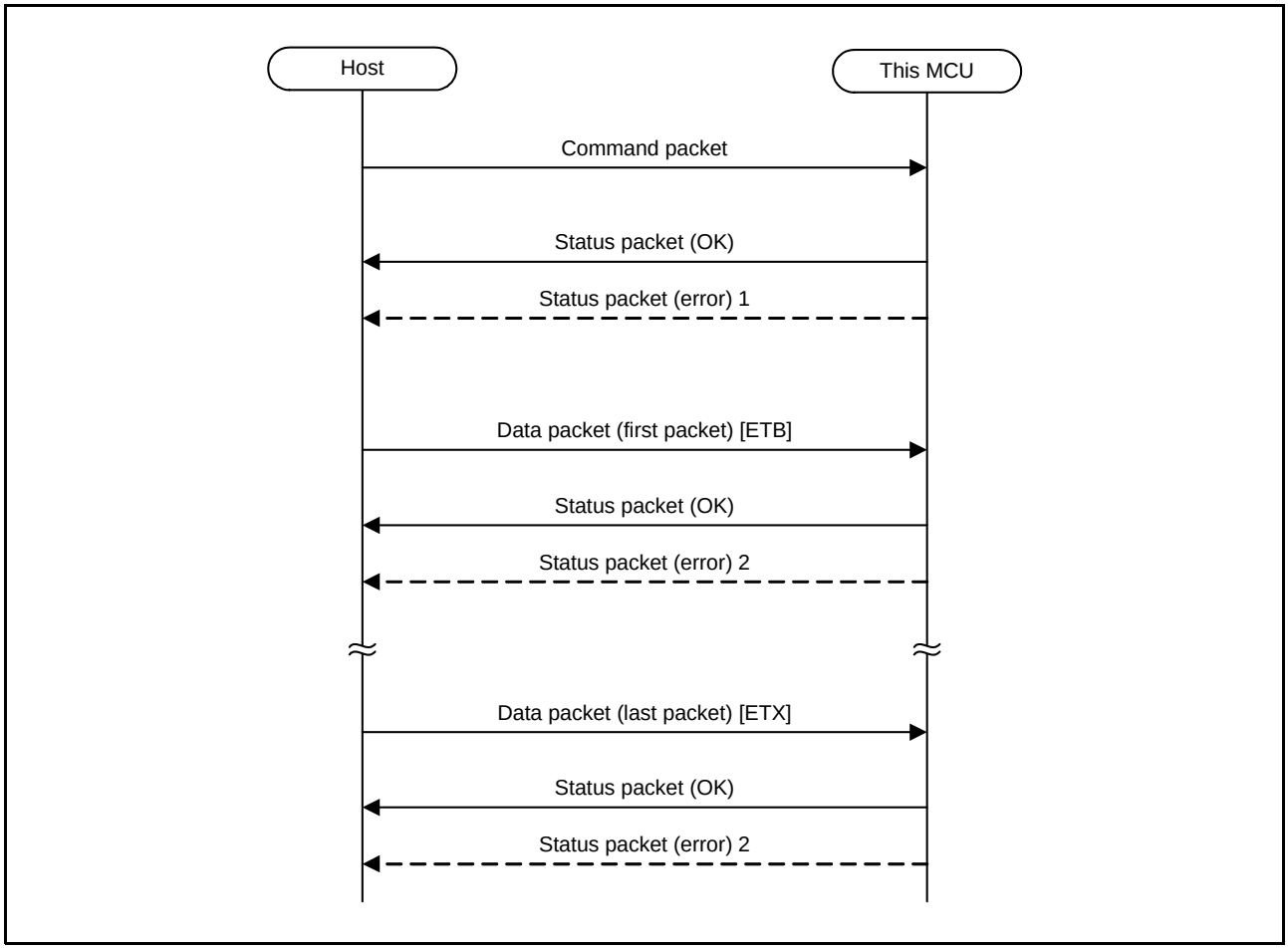


Figure 44.36 Programming Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 13h
SHH: Program start address (24 to 31 bits)
SHL: Program start address (16 to 23 bits)
SLH: Program start address (8 to 15 bits)
SLL: Program start address (0 to 7 bits)
EHH: Program end address (24 to 31 bits)
EHL: Program end address (16 to 23 bits)
ELH: Program end address (8 to 15 bits)
ELL: Program end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 13h (OK)
Data: Program data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 13h (OK)
SUM: ECh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.20 Read Command

This command is used to read data from the specified area in the flash memory and send it to the host.
Specify addresses for a data length that is a multiple of 256 bytes for the user area or user boot area, or a multiple of 16 bytes for the data area. Furthermore, as the address where reading is to start, specify an address on a 256-byte boundary for the user area or user boot area or on a 16-byte boundary for the data area. When the TM function is enabled, reading blocks 8 and 9 in the user area being handled as TM target areas returns 0.
This command cannot be used when read commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Procedure

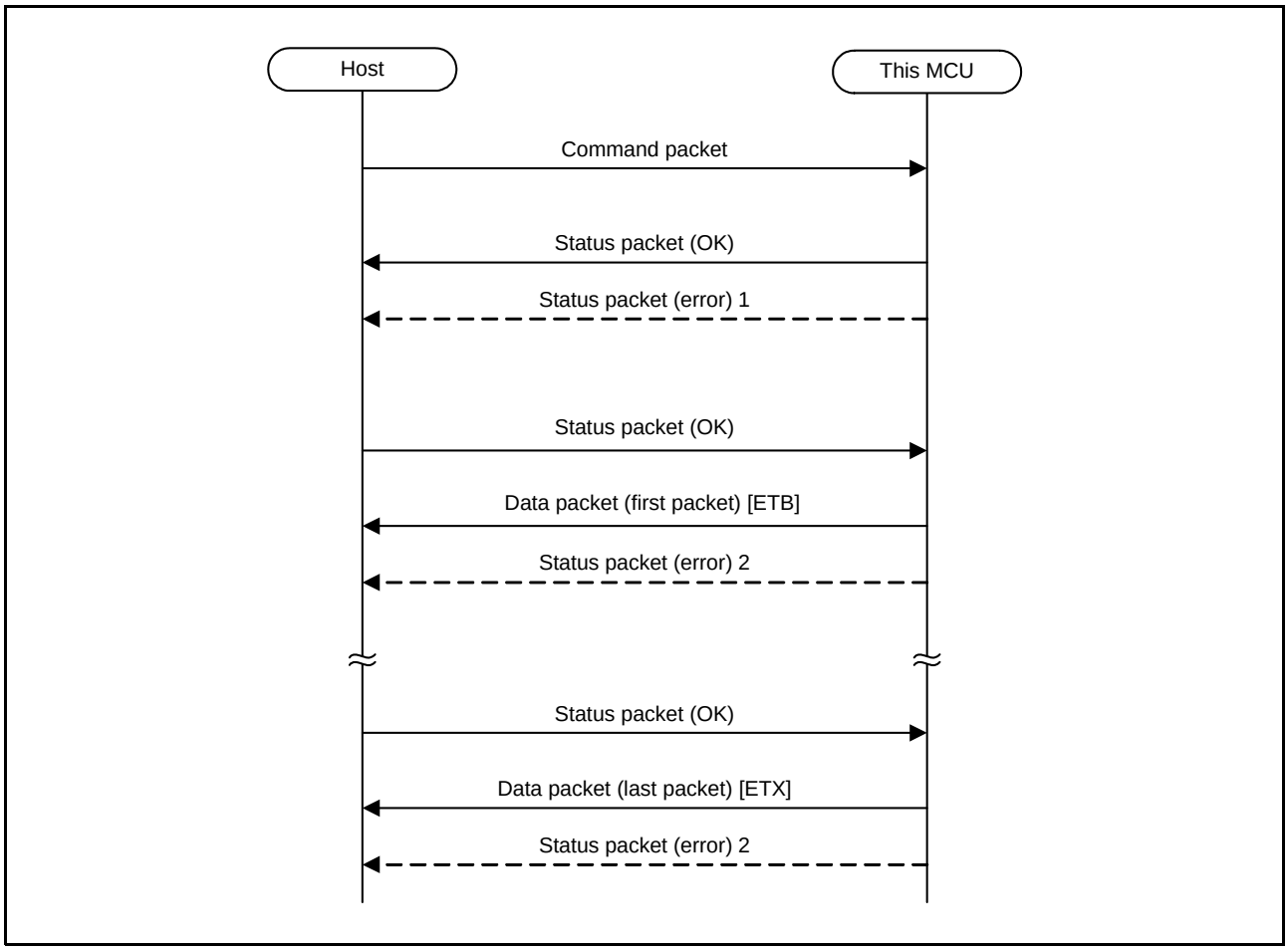


Figure 44.37 Read Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 15h
SHH: Read start address (24 to 31 bits)
SHL: Read start address (16 to 23 bits)
SLH: Read start address (8 to 15 bits)
SLL: Read start address (0 to 7 bits)
EHH: Read end address (24 to 31 bits)
EHL: Read end address (16 to 23 bits)
ELH: Read end address (8 to 15 bits)
ELL: Read end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 15h (OK)
Data: Read data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 15h (OK)
SUM: EAh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.21 Lock-Bit Setting Command

This command is used to set lock bits for the user area.
When the chip starts up in boot mode, programming or erasure of blocks for which the lock bit has been set is possible since the lock bit function is disabled. To enable the lock bit function during operation in boot mode, use the lock-bit enable command.
This command cannot be used to clear a lock bit that has already been set. Canceling the setting of a lock bit requires erasure of the target block.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	LB1	LB2	LBU	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 63h
COM: 22h
LB1: User area (65 bytes)
LB2: Reserved data (32 bytes)
LBU: Reserved data (1 byte)
SUM: Sum of values
ETX: 03h

[Lock-bit setting]

(0: Set the lock bit; 1: Do not set the lock bit.)

	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	Block 7	Block 6	Block 5	Block 4	Block 3	Block 2	Block 1	Block 0
⋮								
65th byte	Fixed to 1	Fixed to 1	Block 517	Block 516	Block 515	Block 514	Block 513	Block 512
66th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1
⋮								
97th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1
98th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1

Set the value to 1 for blocks that do not exist.

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 22h (OK)
SUM: DDh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E2h (program error)
DDh (Lock-bit setting error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.22 Lock-Bit Acquisition Command

This command is used to make the MCU send the lock bit information for the user area to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 23h
SUM: DCh
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	LB1	LB2	LBU	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 63h
RES: 23h (OK)
LB1: User area (65 bytes)
LB2: Reserved data (32 bytes)
LBU: Reserved data (1 byte)
SUM: Sum of values
ETX: 03h

[Lock-bit setting]

(0: The lock bit has been set; 1: The lock bit has not been set.)

	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	Block 7	Block 6	Block 5	Block 4	Block 3	Block 2	Block 1	Block 0
65th byte	Fixed to 1	Fixed to 1	Block 517	Block 516	Block 515	Block 514	Block 513	Block 512
66th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1
97th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1
98th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1

The value is set to 1 for blocks that do not exist.

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 23h (OK)
SUM: DCh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A3h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A3h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.23 Lock-Bit Enable Command

This command can be used to enable the lock-bit function.
When the function is enabled, blocks for which the lock bit is set cannot be programmed or erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 24h
SUM: DBh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 24h (OK)
SUM: DBh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.24 Lock-Bit Disable Command

This command can be used to disable the lock-bit function.
When the function is disabled, blocks for which the lock bit is set can be programmed or erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 25h
SUM: DAh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 25h (OK)
SUM: DAh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A5h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.25 Command Protection Setting Command

This command is used to prohibit block erase commands, programming commands, and read commands.
Command protection cannot be disabled if it is already enabled. This command cannot be used when ID code protection for serial programmers is enabled.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SEC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 20h
SEC: Security data (1 byte)
- Bit 7: Prohibition of read commands (1: Disabled; 0: Enabled)
- Bit 6: Prohibition of programming commands (1: Disabled; 0: Enabled)
- Bit 5: Prohibition of block erase commands (1: Disabled; 0: Enabled)
- Bit 4: Reserved bit (fixed to 1)
- Bit 3: Reserved bit (fixed to 1)
- Bit 2: Reserved bit (fixed to 1)
- Bit 1: Reserved bit (fixed to 1)
- Bit 0: Reserved bit (fixed to 1)

SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 20h (OK)
SUM: DFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A0h (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)
- DAh (protection error)
- E1h (erase error)
- E2h (program error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.26 Command Protection Acquisition Command

This command is used to make the MCU send information on the settings for prohibition of block erase commands, programming commands, and read commands to the host. This command cannot be used when ID code protection for serial programmers is enabled.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 21h
SUM: DEh
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	SEC	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 21h (OK)
SEC: Security data (1 byte)
Bit 7: Prohibition of read commands
(1: Disabled; 0: Enabled)
Bit 6: Prohibition of programming commands
(1: Disabled; 0: Enabled)
Bit 5: Prohibition of block erase commands
(1: Disabled; 0: Enabled)
Bit 4: Reserved bit (fixed to 1)
Bit 3: Reserved bit (fixed to 1)
Bit 2: Reserved bit (fixed to 1)
Bit 1: Reserved bit (fixed to 1)
Bit 0: Reserved bit (fixed to 1)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 21h (OK)
SUM: DEh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A1h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A1h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.27 Serial Programming ID Code Setting Command

This command is used to set the ID code to the OSIS register. Also, it enables the ID code protection for the serial programmer with the setting of the SPCC.IDE, SPCC.SEPR, SPCC.ERPR, and SPCC.RDPR bits to 0.
When this command is used, ID code protection for serial programmers is enabled at reconnection.
This command cannot be used if block erase commands, programming commands, or read commands are prohibited.
Apply a reset after executing this command.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 28h
IDC: Control code or 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: 28h (OK)
SUM: D7h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (program error)
E3h (verify error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.28 ID Code Setting Command

This command is used to set the ID code for the OSIS register.
Setting the ID code with this command does not enable ID code protection for serial programmers.
This command cannot be used when ID code protection for serial programmers is enabled.
This command cannot be used if block erase commands, programming commands, or read commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 2Ah
IDC: ID code (16 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(ID code 1: 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: 2Ah (OK)
SUM: D5h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: AAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.29 ID Code Acquisition Command

This command is used to send the value set in the OSIS register to the host.
This command cannot be used when ID code protection for serial programmers is enabled, or when read commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 2Bh
SUM: D4h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	IDC	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 11h
RES: 2Bh (OK)
IDC: ID code (16 bytes) *1
SUM: Sum of values
ETX: 03h

Note 1. The values are sent as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(ID code 1: 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	…

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Bh (OK)
SUM: D4h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ABh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ABh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.30 Serial Programmer Connection Prohibition Command

This command is used to prohibit the serial programmer connection by setting the SPCC.SPE bit to 0.
Applying a reset on execution of this command disables the connection of serial programmers.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 29h
SUM: D6h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 29h (OK)
SUM: D6h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A9h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.31 On-Chip Debugger Connection Prohibition Command

This command is used to prohibit the on-chip debugger connection by setting the SPCC.OCDE bit to 0.
Applying a reset on execution of this command disables the connection of on-chip debuggers.
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: 56h
SUM: A9h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 56h (OK)
SUM: A9h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D6h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.32 OCDE Acquisition Command

This command is used to make the MCU send information on the setting of the SPCC.OCDE bit 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: 57h
SUM: A8h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	OCD	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 57h (OK)
OCD: OCDE information
00h (connection of on-chip debugger is prohibited)
FFh (connection of on-chip debugger is permitted)
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: 57h (OK)
SUM: A8h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D7h (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: D7h (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.

44.9.33 OFS Setting Command

This command is used to set the values in the OFS0 and OFS1 registers.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OFS	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 48h
OFS: OFS (8 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
< OFS >
OFS0 register = 01234567h
OFS1 register = 89ABCDEFh
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
67h	45h	23h	01h	EFh	CDh	ABh	89h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 48h (OK)
SUM: B7h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: C8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.34 OFS Acquisition Command

This command is used to make the MCU send the values of the OFS0 and OFS1 registers to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 49h
SUM: B6h
ETX: 03h

(2) Data packet structure (data packet)

SOD	LNH	LNL	RES	OFS	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 09h
RES: 49h (OK)
OFS: OFS (8 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. The values are sent as below.
< OFS >
OFS0 register = 01234567h
OFS1 register = 89ABCDEFh
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
67h	45h	23h	01h	EFh	CDh	ABh	89h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 49h (OK)
SUM: B6h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: C9h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: C9h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.35 Endian Setting Command

This command is used to set the values in the MDE register.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OFS	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 4Ah
END: Endian information
00h (big-endian)
Other than the above (little-endian)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Ah (OK)
SUM: B5h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.36 Endian Acquisition Command

This command is used to make the MCU send information on the setting of the MDE register to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 4Bh
SUM: B4h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	END	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 4Bh (OK)
END: Endian information
00h (big-endian)
FFh (little-endian)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Bh (OK)
SUM: B4h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CBh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CBh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.37 Configuration Clearing Command

This command is used to erases the configuration setting areas. When the TM function is enabled, the command can also be used to erase blocks 8 and 9 in the user area being handled as TM target areas.
For the states after execution of this command, refer to (4) Configuration data after clearing.
This command cannot be issued in the following cases: the user area, user boot area, and data area are not blank; the lock bit is set for any block; or prohibition of block erase commands is enabled.
When ID code protection for serial programmers is enabled, a reset is required to disable it.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 1Ch
SUM: E3h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 1Ch (OK)
SUM: E3h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 9Ch (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E0h (non-blank error)
E1h (erase error)
E2h (program error)
SUM: Sum error
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(4) Configuration data after clearing

Data	Setting
Prohibition of block erase commands	Disabled
Prohibition of programming commands	Disabled
Prohibition of read commands	Disabled
ID code protection for serial programmers	Disabled
Prohibition of serial programmer connection	Disabled
Prohibition of on-chip debugger connection	Disabled
ID code	All FFh
MDE	Little-endian
OFS0 and OFS1	All FFh
TM function	Disabled
ROM code protection	Disabled

44.9.38 TM Setting Command

The TM setting command writes 000b to the TMEF.TMEF[2:0] bits and the desired 4-byte value to the TMINF register to enable the TM function. The TMINF register is used to store codes for identification of the programs, etc., in the areas being handled as TM. The TM function is enabled after a reset.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	TMI	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 05h
COM: 4Eh
TMI: TMINF (4 bytes)*1
SUM: Sum of value
ETX: 03h

Note 1. Send the values as below.
< TMINF >
TMINF = 01234567h

<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte
67h	45h	23h	01h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Eh (OK)
SUM: B1h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CEh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (write error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.39 TM Acquisition Command

The TM acquisition command acquires an indicator of whether the TM function is enabled or disabled, the setting of the TMINF register, and the addresses where the areas being handled as TM target start and end, and conveys the results to the 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: 4Fh
SUM: B0h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	TME	TMI	SHH	SHL	SLH	SLL
EHH	EHL	ELH	ELL	SUM	EXT				

SOD: 81h
LNH: 00h
LNL: 0Eh
RES: 4Fh (OK)
TME: TM function is enabled or disabled
00h (TM function is enabled)
FFh (TM function is disabled)
TMI: TMINF (4 bytes*1)
SHH: TM target area start addresses (high and high)
SHL: TM target area start addresses (high and low)
SLH: TM target area start addresses (low and high)
SLL: TM target area start addresses (low and low)
EHH: TM target area end addresses (high and high)
EHL: TM target area end addresses (high and low)
ELH: TM target area end addresses (low and high)
ELL: TM target area end addresses (low and low)
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
< TMINF >
TMINF = 01234567h

<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte
67h	45h	23h	01h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Fh (OK)
SUM: B0h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CEh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CEh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.40 Simple Addition Checksum Command

This command is used to calculate the sum of values in the specified area and send the result to the host. While the TM function is enabled, however, the values in the TM target areas are not included in the calculation.
The target area of this command can be selected from the user area, user boot area, and data area. Calculation is by simple addition. The initial value is 0 and the sum of values in the specified area is obtained by adding the values of all bytes.
If this command is issued for a data area that includes erased blocks, the result is undefined. When a simple addition checksum is to be executed for the data area, make sure that data have been written throughout the specified area.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 4Dh
ARE: Area information
- 00h (user area)
- 10h (user boot area)
- 20h (data area)

SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	SD1	SD2	SD3	SD4	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 05h
RES: 4Dh (OK)
SD1: Sum of values
SD2: Sum of values
SD3: Sum of values
SD4: Sum of values
- If the sum of values is 01234567h, the settings are as follows.
- SD1 = 01h
- SD2 = 23h
- SD3 = 45h
- SD4 = 67h

SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Dh (OK)
SUM: B2h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)
- D5h (area error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.41 Signature Acquisition Command

This command is used to make the MCU send information on the flash memory configuration to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 3Ah
SUM: C5h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	DEV	CF1	CS1	CN1	CF2	CS2	CN2		
			SP1	SS1	SN1	CF3	CSS	CN3	UF1	US1	UN1	
								DF1	DS1	DN1	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 3Bh
RES: 3Ah (OK)
DEV: Reserved data (16 bytes)
CF1: 00h (8-Kbyte-block portion of the user area)
CS1: Size of the 8-Kbyte-block portion of the user area
(in bytes; 4-byte value)
CN1: Number of 8-Kbyte blocks in the user area (2 bytes)
CF2: 00h (32-Kbyte-block portion of the user area)
CS2: Size of the 32-Kbyte-block portion of the user area
(in bytes; 4-byte value)
CN2: Number of 32-Kbyte blocks in the user area (2 bytes)
SP1: Reserved data (1 byte)
SS1: Reserved data (4 bytes)
SN1: Reserved data (2 bytes)
CF3: Reserved data (1 byte)
CS3: Reserved data (4 bytes)
CN3: Reserved data (2 bytes)
UF1: 02h (user boot area)
US1: Block size of the user boot area (in bytes; 4-byte value)
UN1: Number of blocks in the user boot area (2 bytes)
DF1: 03h (data area)
DS1: Block size of the data area (in bytes; 4-byte value)
DN1: Number of blocks in the data area (2 bytes)
SUM: Sum of values
ETX: 03h

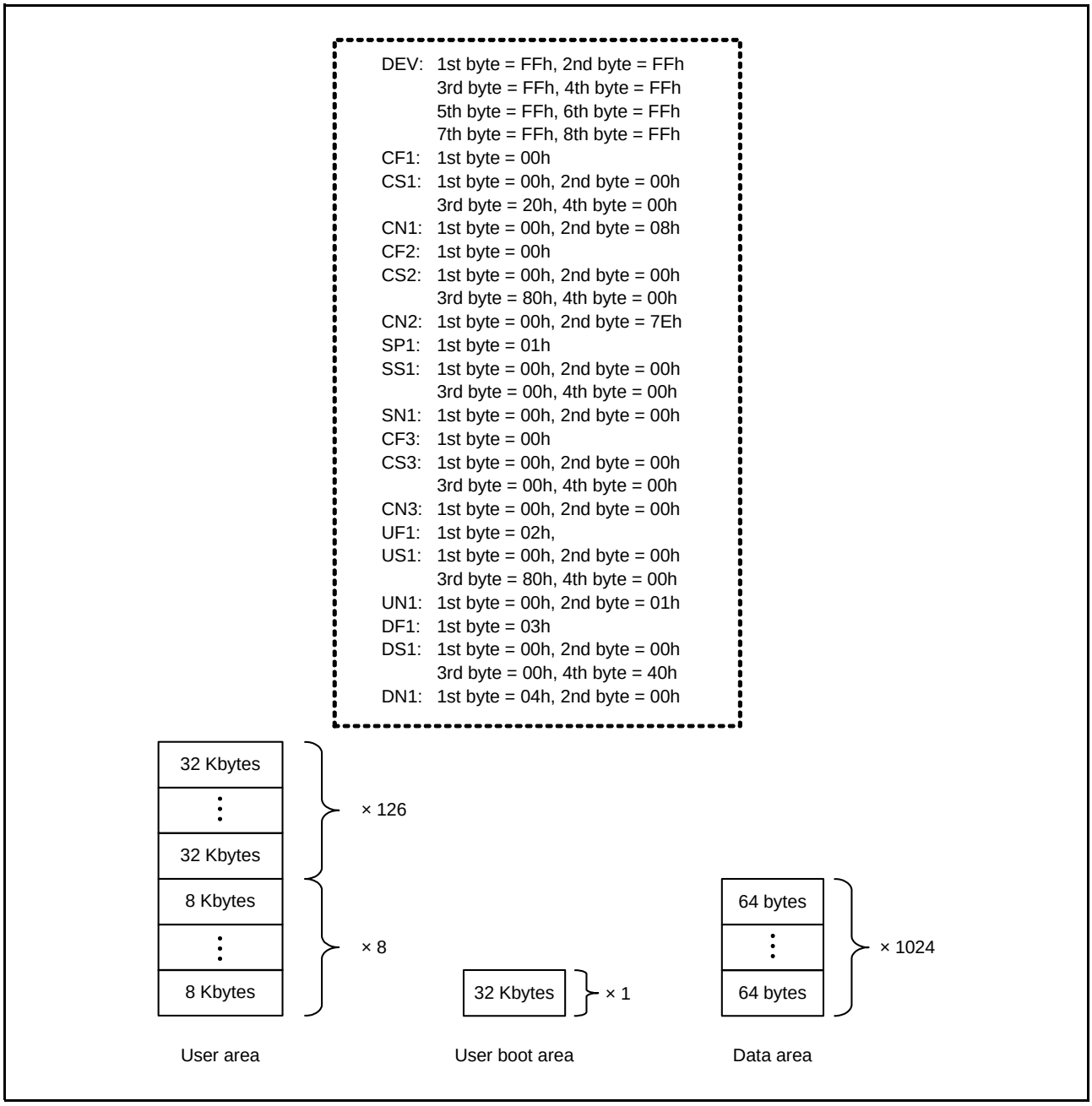


Figure 44.38 Example of Flash Memory Configuration Information

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 3Ah (OK)
SUM: C5h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: BAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: BAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.42 ROM Code Setting Command

This command is used to set the values in the ROM code protection register.
This command cannot be used when programming commands or block erase commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	RS1	ROM	RS2	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 26h
RS1: Settings for the configuration setting area, which is in the address range from 0012 0070h to 0012 007Bh (12 bytes)*1
ROM: Settings for the ROM code protection register (4 bytes)*2
RS2: Reserved data (16 bytes)*3
SUM: Sum of values
ETX: 03h

Note 1. The data are used for writing to the configuration setting area, which is in the address range from 0012 0070h to 0012 007Bh. Send FFh, since the configuration setting area is reserved.
Note 2. Send the values as below.
< ROM code >
ROM code protection register = 00000001h

<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte
01h	00h	00h	00h

Note 3. Send FFh.

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 26h (OK)
SUM: D9h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A6h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

44.9.43 ROM Code Acquisition Command

The ROM code acquisition command causes an indicator of the setting of the ROM code protection register to be sent 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: 27h
SUM: D8h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	RS1	ROM	RS2	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 21h
RES: 27h (OK)
RS1: Settings for the configuration setting area, which is in the address range from 0012 0070h to 0012 007Bh (12 bytes)
ROM: Settings for the ROM code protection register (4 bytes)*1
RS2: Reserved data (16 bytes)*2
SUM: Sum of values
ETX: 03h

Note 1. The values are sent as below.
< ROM code >
ROM code protection register = 00000001h
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte
01h	00h	00h	00h

Note 2. FFh is sent.

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 27h (OK)
SUM: D8h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A7h (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: A7h (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.

44.9.44 Usage Example

(1) Example of the Procedure for Reprogramming

Figure 44.39 shows an example of the procedure for reprogramming.

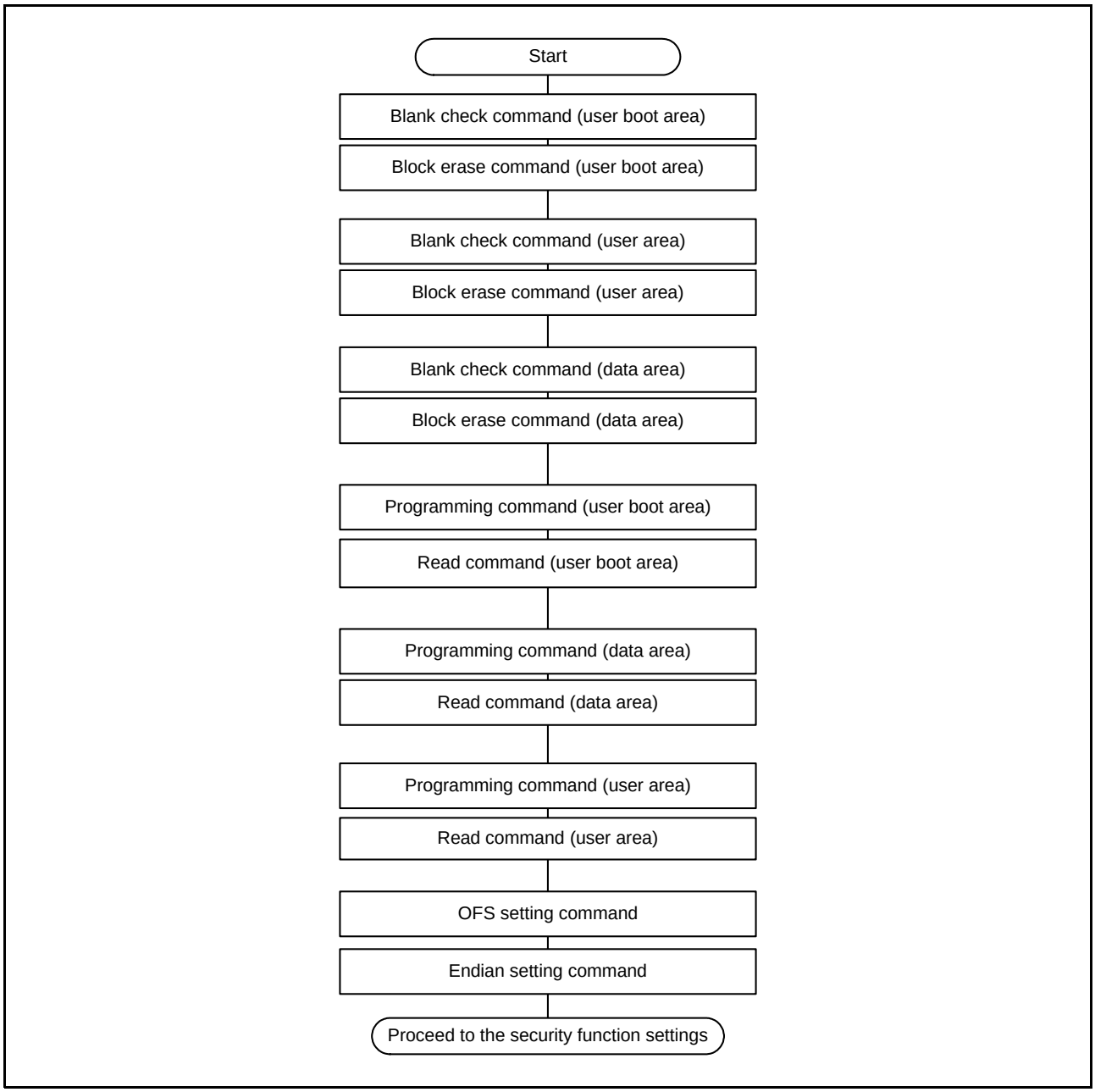
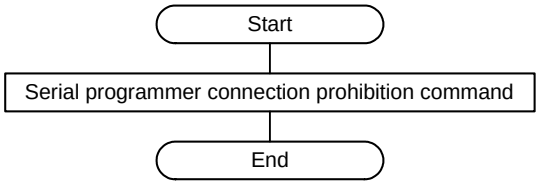


Figure 44.39 Example of Reprogramming Procedure

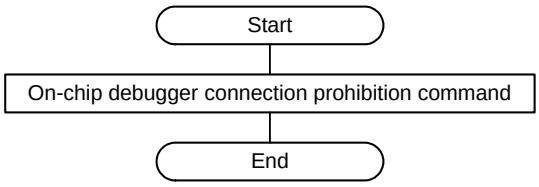
(2) Security Function Settings

When the security functions are to be used, follow the procedure below to set the required security functions.

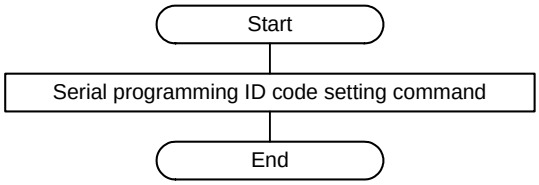
1. Prohibiting the connection of serial programmers



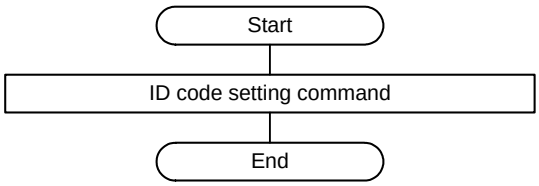
2. Prohibiting the connection of on-chip debuggers



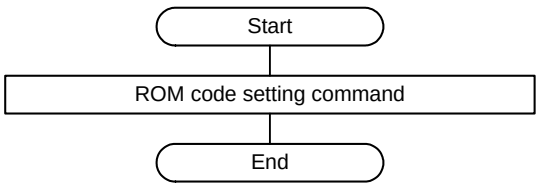
3. Setting the OCD ID code and enabling ID code protection for serial programmers



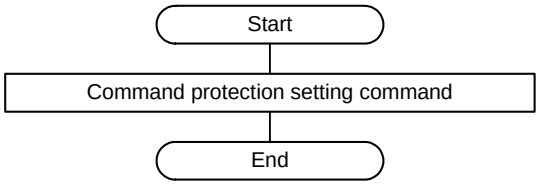
4. Setting the OCD ID code



5. Enabling ROM code protection



6. Prohibiting block erase commands, programming commands, and read commands



Note: When issuing both the on-chip debugger connection prohibition command and the command protection setting command, issue the on-chip debugger connection prohibition command first, and then issue the command protection setting command.

Note: When issuing both the ID code setting command and the command protection setting command, issue the ID code setting command first, and then issue the command protection setting command.

Note: When issuing both the ROM code setting command and the command protection setting command, issue the ROM code setting command first, and then issue the command protection setting command.

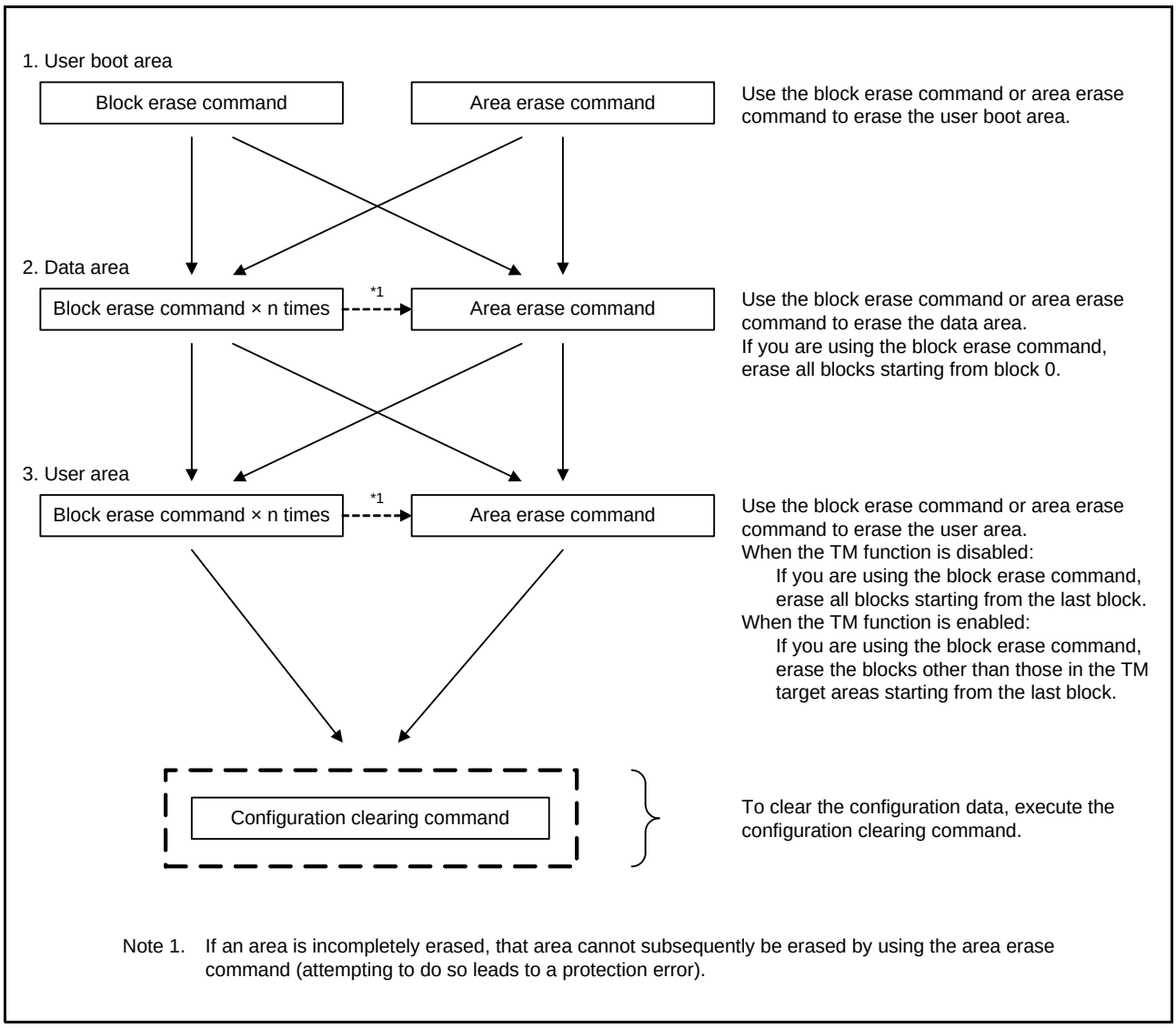
Figure 44.40 Settings for Use of the Security Functions

44.9.45 Flow for Erasure when Programming Commands are Prohibited

When programming commands are prohibited, follow the procedure below for erasure.

1. Erase the user boot area.
2. Erase the data area in order from the first block.
3. Erase the user area in order from the last block.

Note that neither a reset nor interrupts, including NMI, should be applied during the steps from erasure of the user boot area to erasure of the user area.



Note 1. If an area is incompletely erased, that area cannot subsequently be erased by using the area erase command (attempting to do so leads to a protection error).

Figure 44.41 Flow for Erasure when Programming Commands are Prohibited

44.10 Using the Serial Programmer for Rewriting

A dedicated serial programmer can be used to rewrite flash memory in boot mode.

(1) Serial Programming

A serial programmer can program this MCU by providing a connector to the board while the MCU is implemented to the board that the user will use.

44.10.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU with data are described below.

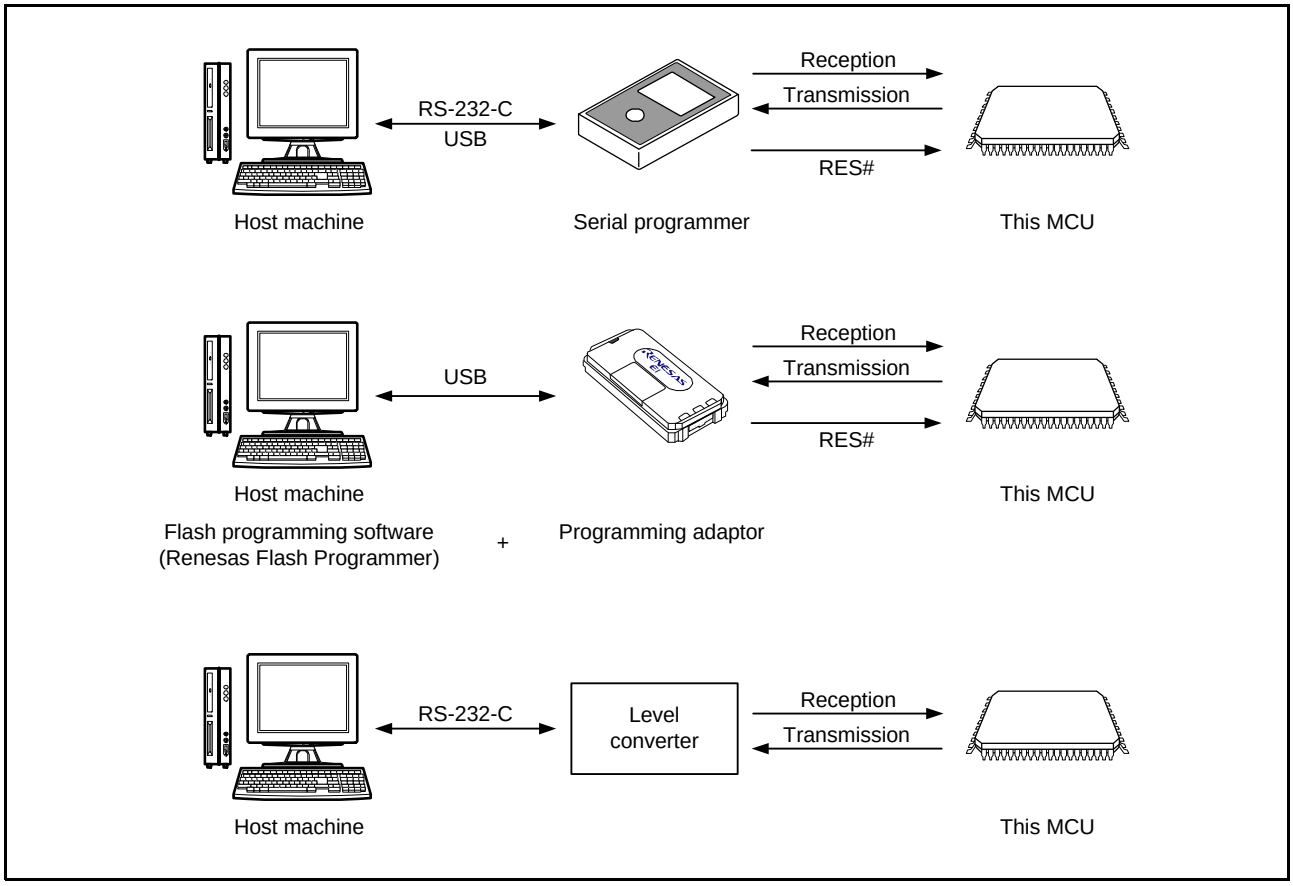


Figure 44.42 Environments for Rewriting the Flash Memory

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.

44.11 Programming through Self-Programming

44.11.1 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 user area and data area. This allows upgrading of user programs and rewriting of constant data fields.
The BGO can be used for rewriting to the data area. The program for rewriting must be transferred to the internal RAM or external memory in advance when the BGO is disabled.

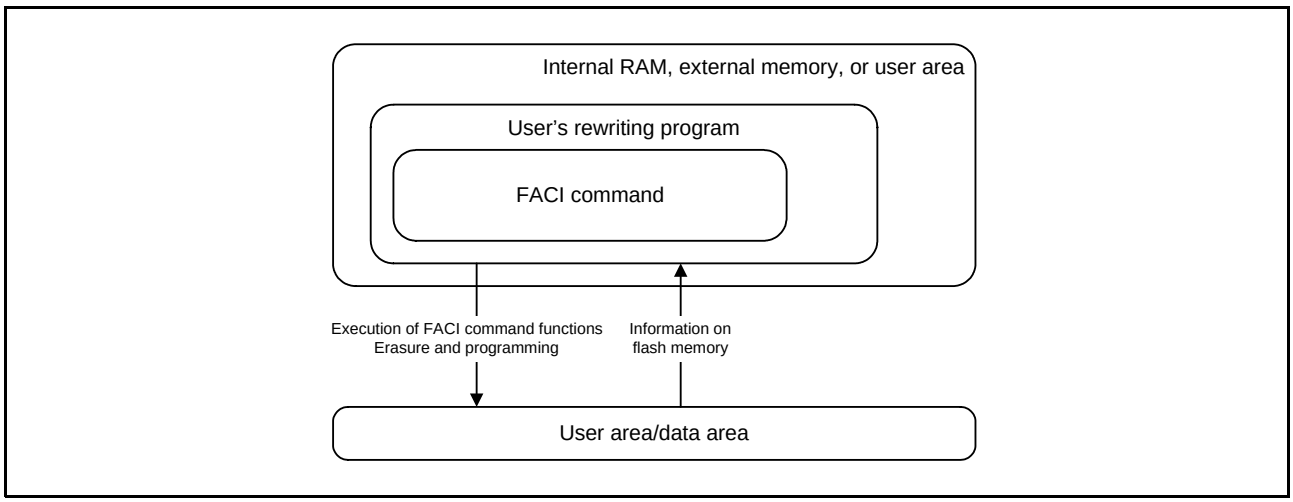


Figure 44.43 Schematic View of Self-Programming

44.12 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 to the user area, data area, or user boot area twice is not possible. To program user area, data area or user boot area which has already been programmed, erase the relevant area.

(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, wait for at least t_{RESWF} (refer to section 45, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal, and then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming/Erasure

Generation of an interrupt or other exception during programming/erasure may lead to fetching of the vector from the code flash memory. If the conditions for using the background operation (BGO) are not satisfied, set the address for vector to an address that is not in the code flash memory.

(6) Abnormal Termination During Programming/Erasure or Blank Checking

Even if programming/erasure ends abnormally due to the generation of a reset by the RES# pin, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming/erasure ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again and confirm that the corresponding area is completely erased before using.
When programming/erasure or blank checking is not completed successfully due to a voltage change that exceeds the operational voltage range, a reset on the RES# pin, the command locked state in response to error detection, or prohibited actions described in (7), the lock bit may be enabled.
In this case, erase the target block while the lock bit is disabled to erase the lock bit.

(7) Items Prohibited During Programming/Erasure or Blank Checking

High voltage is applied to the flash memory during programming/erasure or blank checking. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0] bits.
- Change the SYSCR0.ROME bit.
- Change the 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.

45. Electrical Characteristics

45.1 Absolute Maximum Ratings

Table 45.1 Absolute Maximum Rating

Conditions: VSS = AVSS0 = 0 V

Item		Symbol	Value	Unit
Power supply voltage		VCC	–0.3 to +6.5	V
Analog power supply voltage		AVCC0*1	–0.3 to +6.5	V
Reference power supply voltage		VREFH0	–0.3 to AVCC0 + 0.3 (up to 6.5)	V
Input voltage	P12, P13, P16, P17	V_{in}	–0.3 to +6.5	V
	P03, P05 to P7, P40 to P47, PJ6, PJ7		–0.3 to AVCC0 + 0.3 (up to 6.5)	
	Other than above		–0.3 to VCC + 0.3 (up to 6.5)	
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. Insert capacitors of high frequency characteristics between the AVCC0 and AVSS0 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.

45.2 Recommended Operating Conditions

Table 45.2 Recommended Operating Conditions (1)

Item		Symbol	Min.	Typ.	Max.	Unit
Power supply voltage*1		VCC	2.7	—	5.5	V
		VSS	—	0	—	
Analog power supply voltage*1, *2		AVCC0	3.0	—	5.5	V
		AVSS0	—	0	—	
		VREFH0	AVCC0 – 1.0	—	AVCC0	
		VREFL0	—	0	—	
Input voltage	P12, P13, P16, P17	V_{in}	–0.3	—	5.8	V
	P03, P05 to P7, P40 to P47, PJ6, PJ7		–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

Note 2. For details, see section 38.6.10, Voltage Range of Analog Power Supply Pins.

Table 45.3 Recommended Operating Conditions (2)

Item	Symbol	Value
Decoupling capacitance to stabilize the internal voltage	C_{VCL}	0.47 µF ± 30%*1

Note 1. Use a multilayer ceramic capacitor whose nominal capacitance is 0.47 µF and a capacitance tolerance is ±30% or better.

45.3 DC Characteristics

Table 45.4 DC Characteristics (1)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ input pin MTU input pin POE input pin TMR input pin CMTW input pin RTC input pin SCI input pin CANFD input pin REMC input pin ADTRG# input pin RES#, NMI	V_{IH}	0.8 × VCC	—	—	V	
		V_{IL}	—	—	0.2 × VCC		
		ΔV_T	0.06 × VCC	—	—		
	RIIC input pin (except for SMBus)	V_{IH}	0.7 × VCC	—	—		
		V_{IL}	—	—	0.3 × VCC		
		ΔV_T	0.05 × VCC	—	—		
	Ports for 5 V tolerant (P12, P13, P16, P17)	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, WAIT#		0.8 × VCC	—	—		
	D0 to D15		0.7 × VCC	—	—		
	RIIC (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, WAIT#		—	—	0.2 × VCC		
	D0 to D15		—	—	0.3 × VCC		
	RIIC (SMBus)		—	—	0.8		

Table 45.5 DC Characteristics (2)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	P03, P05 to P7, P40 to P47, PJ6, PJ7	V_{OH}	AVCC0 – 0.5	—	—	V	I_{OH} = –1 mA
	Other than above		VCC – 0.5	—	—		
Output low voltage	RIIC pins	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA
			—	—	0.6		I_{OL} = 6.0 mA
	Other than above		—	—	0.5		I_{OL} = 1.0 mA
Input leakage current	RES#, EMLE	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	RIIC pins	$\lvert I_{TSI} \rvert$	—	—	5.0	µA	V_{in} = 0 V V_{in} = VCC
	Other than above		—	—	1.0		
Input pull-up resistors	P03, P05 to P7, P40 to P47, PJ6, PJ7	R_{PU}	10	—	100	kΩ	AVCC0 = 3.0 to 5.5 V V_{in} = 0 V
	Other than above		10	—	100		VCC = 2.7 to 5.5 V V_{in} = 0 V
Input pull-down resistors	EMLE	R_{PD}	5	—	50	kΩ	V_{in} = VCC = AVCC0
Input capacitance	RIIC pins	C_{in}	—	—	16	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz T_a = 25°C
	PJ6, PJ7		—	—	12		
	Other than above		—	—	8		
Output voltage of the VCL pin		V_{CL}	—	1.25	—	V	

Table 45.6 DC Characteristics (3)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item				Symbol	D version		G version		Unit	Test Conditions
					Typ.	Max.	Typ.	Max.		
Supply current*1	Normal operating mode	Full operation		I_{CC}*2	—	68	—	76	mA	ICLK = 120 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKD = 60 MHz, FCLK = 60 MHz, BCLK = 60 MHz, BCLK pin = 30 MHz
		Normal operation	Peripheral module clocks are supplied		21	—	21	—		
			Peripheral module clocks are stopped		12	—	12	—		
		CoreMark	Peripheral module clocks are stopped		19	—	19	—		
		Sleep mode	Peripheral module clocks are supplied		16	36	16	44		
		All module clock stop mode			7.8	24	7.8	32		
		Increased by BGO operation*3	Reading from the code flash memory while the data flash memory is being programmed		12	—	12	—		
	Software standby mode				0.9	9	0.9	14	mA	
	Deep software standby mode				15	23	15	32	µA	
		Increase current by operating RTC			2.6	—	2.6	—		When a standard C_L crystal is in use

Note 1. Supply current values are measured when all output pins are unloaded and all input pull-up resistors are disabled.
Note 2. I_{CC} depends on the f (ICLK) as follows.
- D version
  I_{CC} max = 0.500 × f + 8 (full operation in normal operating mode)
  I_{CC} typ = 0.144 × f + 4 (normal operation in normal operating mode)
  I_{CC} max = 0.234 × f + 8 (sleep mode)
- G version
  I_{CC} max = 0.534 × f + 12 (full operation in normal operating mode)
  I_{CC} typ = 0.144 × f + 4 (normal operation in normal operating mode)
  I_{CC} max = 0.267 × f + 12 (sleep mode)

Note 3. This is an increase caused by programming/erasing of the data flash memory during execution of the user program from the code flash memory.

Table 45.7 DC Characteristics (4)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

	Item				Symbol	Typ.	Max.	Unit
	A/D	D/A (2 channels)	CMPC (4 channels)	TEMPS				
Analog power supply current	During conversion	Waiting for conversion	Waiting	Waiting	A_{ICC}	0.9	1.4	mA
	Waiting for conversion	During conversion	Waiting	Waiting		0.6	0.8	
	Waiting for conversion	Waiting for conversion	During operation	Waiting		0.4	0.5	
	During conversion	Waiting for conversion	Waiting	During operation		1.0	1.5	
	When waiting					0.4	7.7	µA
	In the module-stop state					0.4	6.5	

Table 45.8 DC Characteristics (5)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM retension voltage		V_{RAM}	2.7	—	—	V	
VCC ramp rate at power-on	At normal startup	SrVCC	0.02	—	8	ms/V	Figure 45.1
	Voltage monitoring 0 reset enabled at startup*1, *2		0.02	—	20		
VCC ramp rate at power fluctuation		dt/dVCC	1.0	—	—		When VCC change exceeds VCC ±10%

Note 1. When OFS1.LVDAS = 0.
Note 2. Settings of the OFS1 register are not read in boot mode or user boot mode, so turn on the power supply voltage with a ramp rate at normal startup.

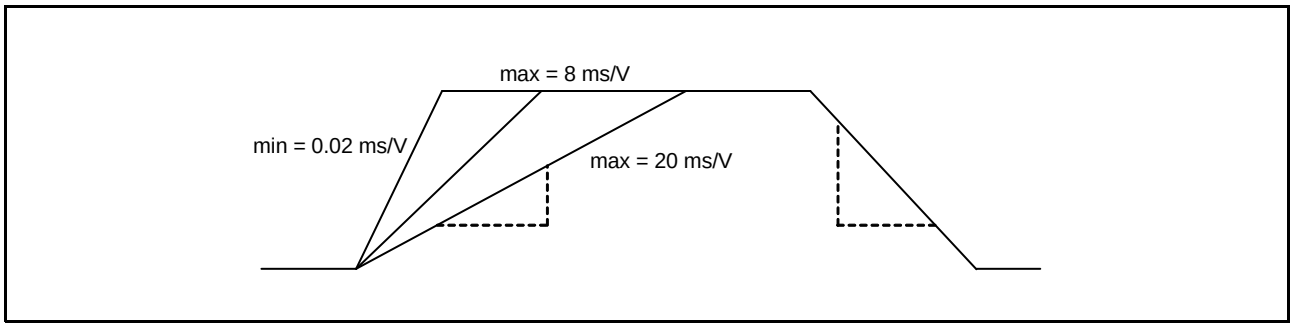


Figure 45.1 VCC Ramp Rate at Power-On

Table 45.9 Permissible Output Currents

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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	
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	
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	
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	
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.

Table 45.10 Standard Output Characteristics (1)

Conditions: VCC = AVCC0 = 5.0 V,
VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Normal drive output	V_{OH}	—	4.97	—	V	I_{OH} = –0.5 mA
			—	4.94	—		I_{OH} = –1.0 mA
			—	4.87	—		I_{OH} = –2.0 mA
			—	4.74	—		I_{OH} = –4.0 mA
	High-drive output		—	4.98	—		I_{OH} = –0.5 mA
			—	4.97	—		I_{OH} = –1.0 mA
			—	4.94	—		I_{OH} = –2.0 mA
			—	4.87	—		I_{OH} = –4.0 mA
Output low voltage	Normal drive output	V_{OL}	—	0.02	—	V	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
	High-drive output		—	0.01	—		I_{OL} = 0.5 mA
			—	0.03	—		I_{OL} = 1.0 mA
			—	0.05	—		I_{OL} = 2.0 mA
			—	0.10	—		I_{OL} = 4.0 mA

Table 45.11 Standard Output Characteristics (2)

Conditions: VCC = AVCC0 = 3.3V,
VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Normal drive output	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 output		—	3.28	—		I_{OH} = –0.5 mA
			—	3.26	—		I_{OH} = –1.0 mA
			—	3.22	—		I_{OH} = –2.0 mA
			—	3.13	—		I_{OH} = –4.0 mA
Output low voltage	Normal drive output	V_{OL}	—	0.03	—	V	I_{OL} = 0.5 mA
			—	0.06	—		I_{OL} = 1.0 mA
			—	0.12	—		I_{OL} = 2.0 mA
			—	0.25	—		I_{OL} = 4.0 mA
	High-drive output		—	0.02	—		I_{OL} = 0.5 mA
			—	0.03	—		I_{OL} = 1.0 mA
			—	0.07	—		I_{OL} = 2.0 mA
			—	0.13	—		I_{OL} = 4.0 mA

Table 45.12 Thermal Resistance Value (Reference)

Item	Package	Symbol	Max.	Unit	Test Conditions
Thermal resistance	144-pin LFQFP (PLQP0144KA-B)	θ_{ja}	52.1	°C/W	JESD51-2 and JESD51-7 compliant T_a = 105°C
	100-pin LFQFP (PLQP0100KB-B)		51.3		
	80-pin LFQFP (PLQP0080KB-B)		52.0		
	64-pin LFQFP (PLQP0064KB-C)		50.8		
	48-pin LFQFP (PLQP0048KB-B)		58.4		
	144-pin LFQFP (PLQP0144KA-B)	Ψ_{jt}	1.3	°C/W	JESD51-2 and JESD51-7 compliant
	100-pin LFQFP (PLQP0100KB-B)		1.3		
	80-pin LFQFP (PLQP0080KB-B)		1.3		
	64-pin LFQFP (PLQP0064KB-C)		1.3		
	48-pin LFQFP (PLQP0048KB-B)		1.8		

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.

45.4 AC Characteristics

Table 45.13 Operating Frequency

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Operating frequency	System clock (ICLK)	f	—	—	120	MHz	
	Peripheral module clock (PCLKA)		—	—	120		
	Peripheral module clock (PCLKB)		—	—	60		
	Peripheral module clock (PCLKD)		—*1	—	60		
	Flash-IF clock (FCLK)		—*2	—	60		
	External bus clock (BCLK)		—	—	60		
	BCLK pin output		—	—	40		VCC ≥ 4.5 V, High-drive output is selected in the driving ability control register.
			—	—	32		VCC < 4.5 V, High-drive output is selected in the driving ability control register.

Note 1. When the 12-bit A/D converter is to be used, the frequency of PCLKD must be set to at least 8 MHz.
Note 2. The FCLK must run at a frequency of at least 4 MHz when the contents of flash memory are to be changed.

45.4.1 Reset Timing

Table 45.14 Reset Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	t_{RESWP}	2.0	—	—	ms	Figure 45.2
	Deep software standby mode	t_{RESWD}	0.6	—	—	ms	Figure 45.3
	Software standby mode	t_{RESWS}	0.3	—	—	ms	
	Programming or erasure of the code flash memory, or programming, erasure or blank checking of the data flash memory	t_{RESWF}	200	—	—	µs	
	Other than above	t_{RESW}	200	—	—	µs	
Waiting time after release from the RES# pin reset		t_{RESWT}	62	—	63	t_{Lcyc}	Figure 45.2
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)		t_{RESW2}	108	—	116	t_{Lcyc}	

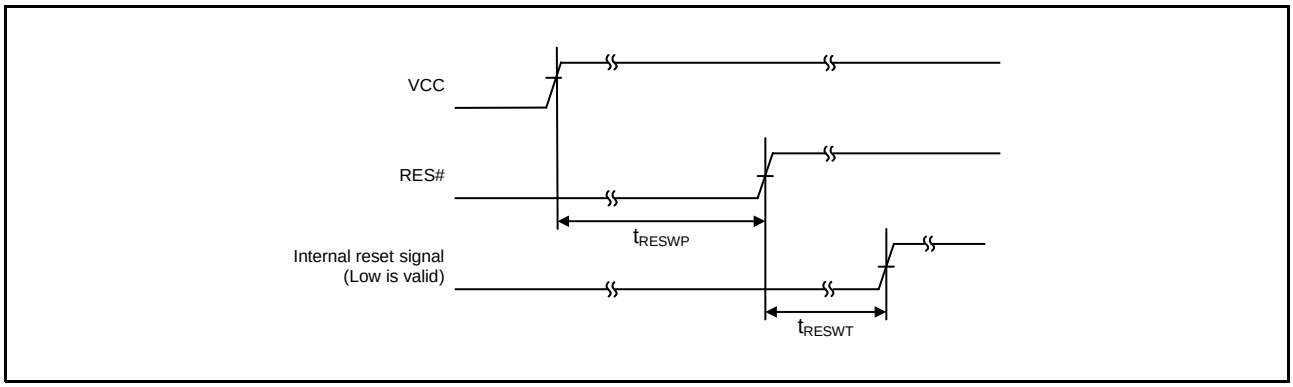


Figure 45.2 Reset Input Timing at Power-On

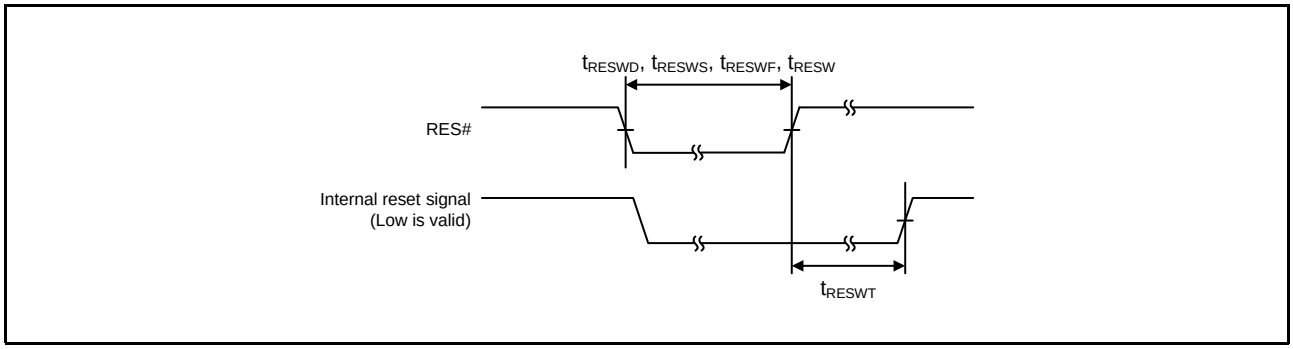


Figure 45.3 Reset Input Timing

45.4.2 Clock Timing

Table 45.15 BCLK Pin Output Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
BCLK pin output cycle time	t_{Bcyc}	25	—	—	ns	VCC ≥ 4.5 V	Figure 45.4
		31.25	—	—		VCC < 4.5 V	
BCLK pin output high pulse width	t_{CH}	7.5	—	—	ns	VCC ≥ 4.5 V	
		10.625	—	—		VCC < 4.5 V	
BCLK pin output low pulse width	t_{CL}	7.5	—	—	ns	VCC ≥ 4.5 V	
		10.625	—	—		VCC < 4.5 V	
BCLK pin output rising time	t_{Cr}	—	—	5	ns		
BCLK pin output falling time	t_{Cf}	—	—	5	ns		

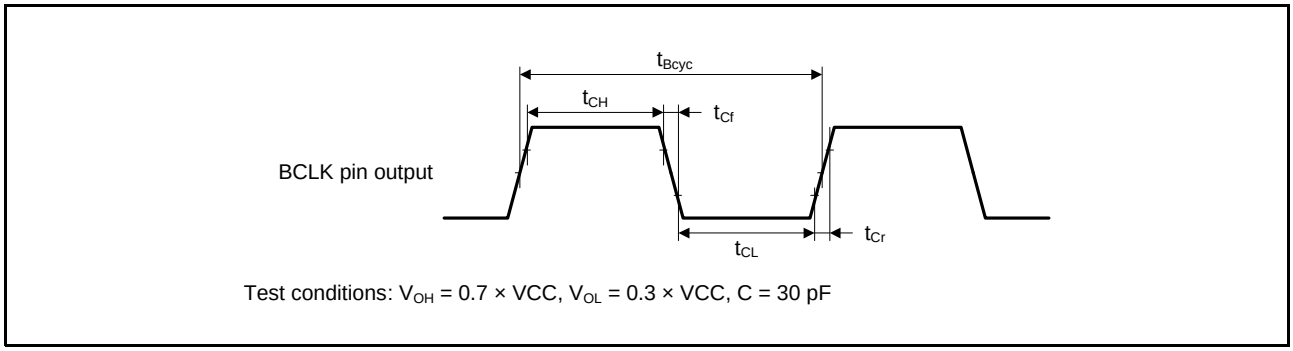


Figure 45.4 BCLK Pin Output Timing

Table 45.16 EXTAL Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	t_{EXcyc}	41.66	—	—	ns	Figure 45.5
EXTAL external clock input frequency	f_{EXMAIN}	—	—	24	MHz	
EXTAL external clock input high pulse width	t_{EXH}	15.83	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	15.83	—	—	ns	
EXTAL external clock rising time	t_{EXr}	—	—	5	ns	
EXTAL external clock falling time	t_{EXf}	—	—	5	ns	

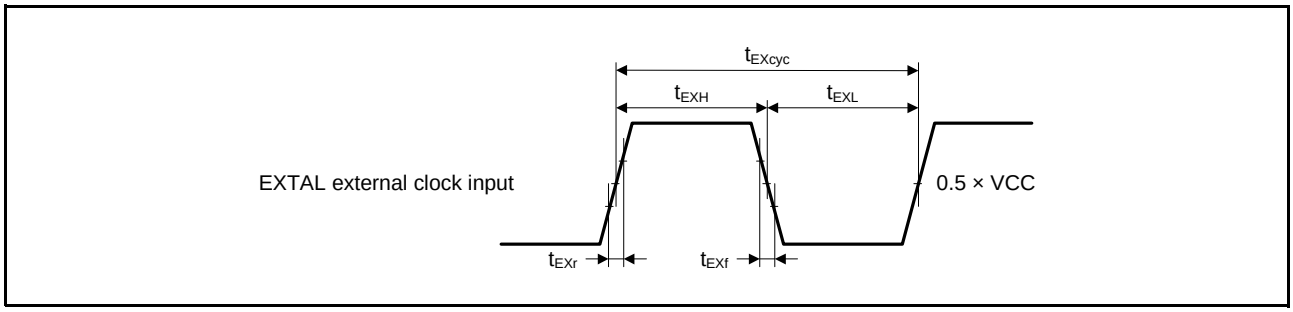


Figure 45.5 EXTAL External Clock Input Timing

Table 45.17 Main Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.6
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}

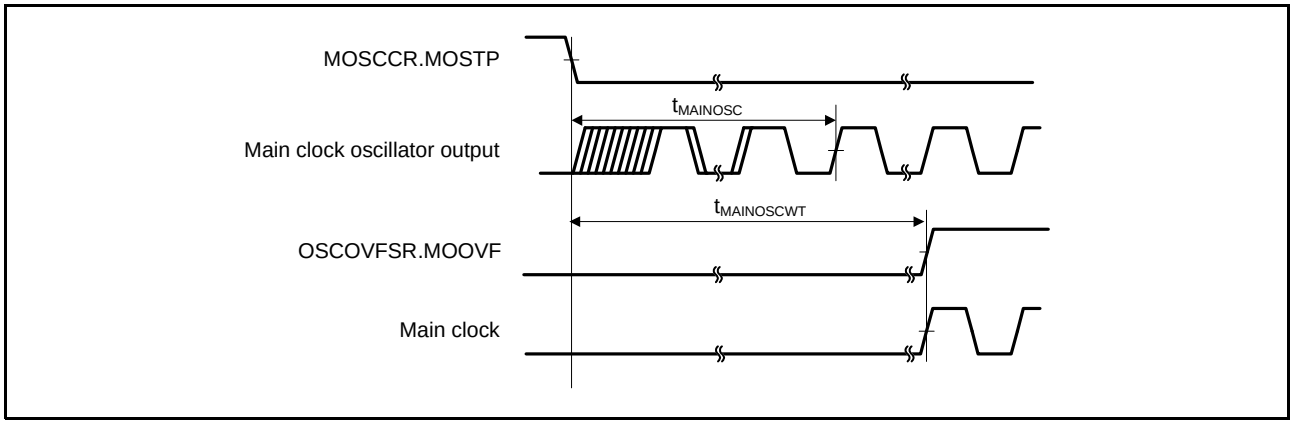


Figure 45.6 Main Clock Oscillation Start Timing

Table 45.18 LOCO and IWDT-Dedicated Low-Speed Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.7
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 45.8

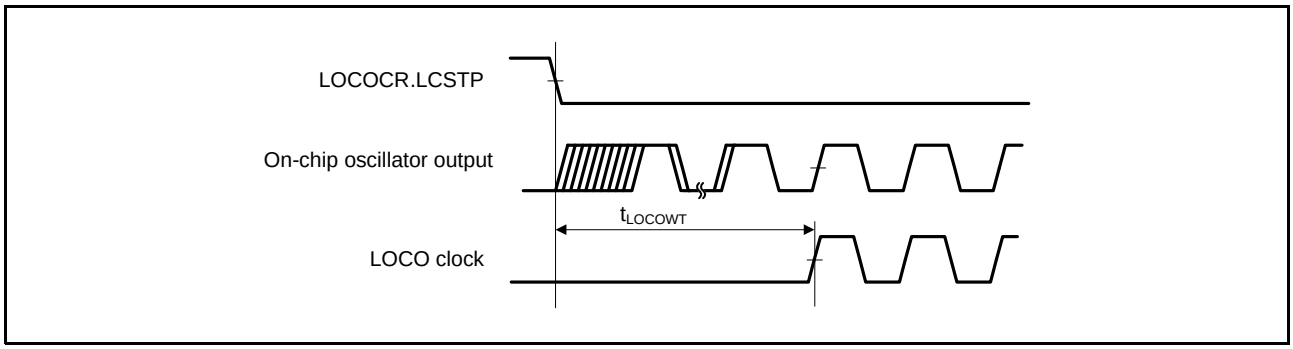


Figure 45.7 LOCO Clock Oscillation Start Timing

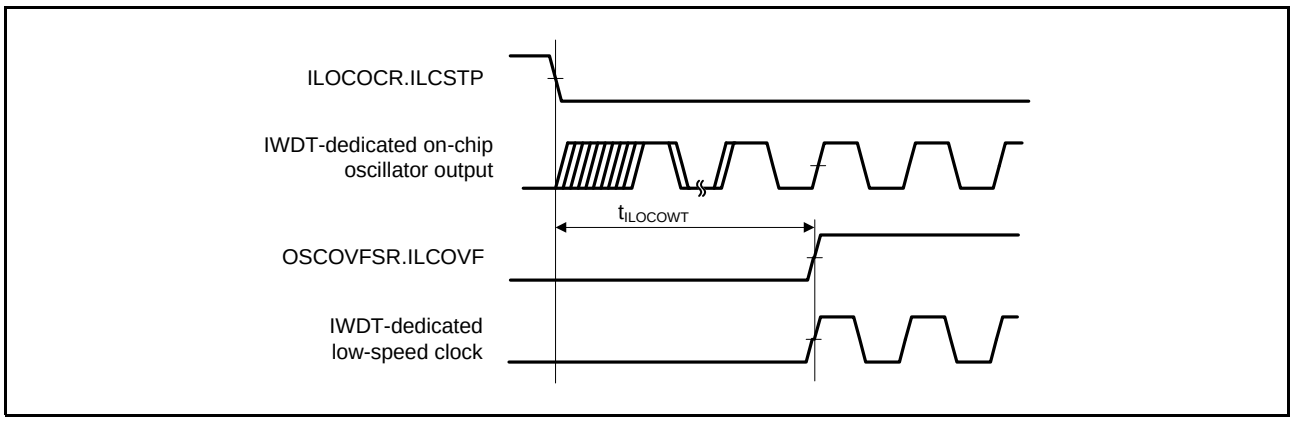


Figure 45.8 IWDT-dedicated Low-Speed Clock Oscillation Start Timing

Table 45.19 HOCO Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HOCO clock oscillation frequency	FLL not in use	f_{HOCO}	15.84 (–1.0%)	16	16.16 (+1.0%)	MHz	–20°C ≤ T_a
			17.82 (–1.0%)	18	18.18 (+1.0%)		
			19.80 (–1.0%)	20	20.20 (+1.0%)		
			15.68 (–2.0%)	16	16.16 (+1.0%)		T_a < –20°C
			17.64 (–2.0%)	18	18.18 (+1.0%)		
			19.60 (–2.0%)	20	20.20 (+1.0%)		
	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 45.9
HOCO clock power supply stabilization time		t_{HOCOP}	—	—	150	µs	Figure 45.10
FLL stabilization waiting time		t_{FLLWT}	—	—	1.8	ms	

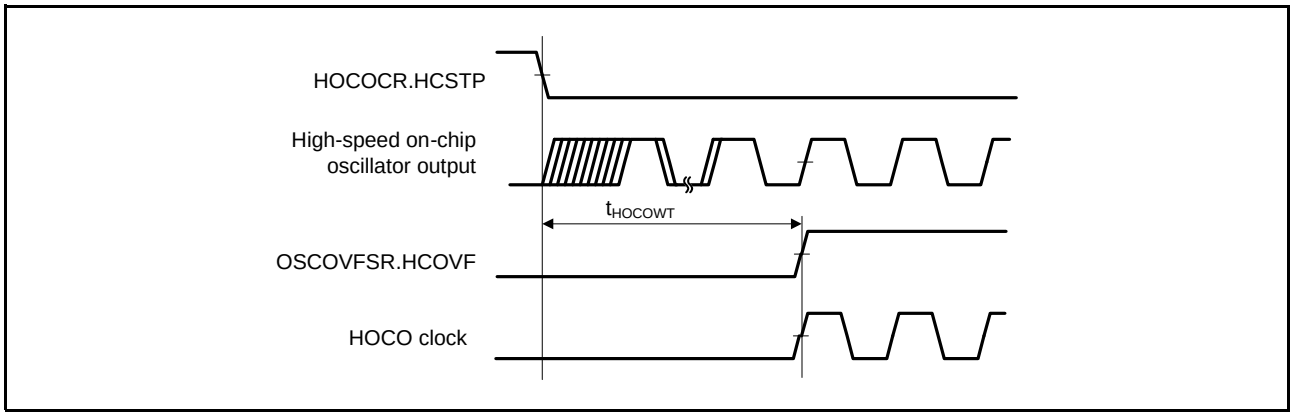


Figure 45.9 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

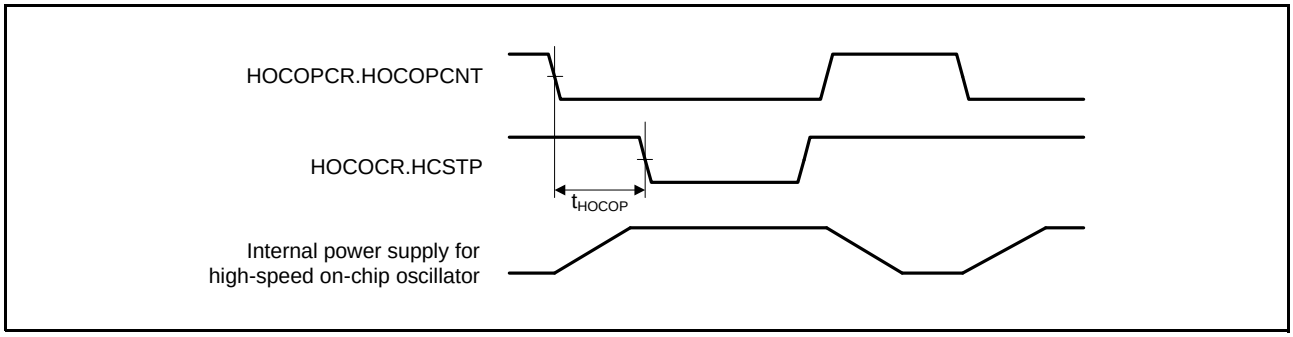


Figure 45.10 High-Speed On-Chip Oscillator Power Supply Control Timing

Table 45.20 PLL Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.11

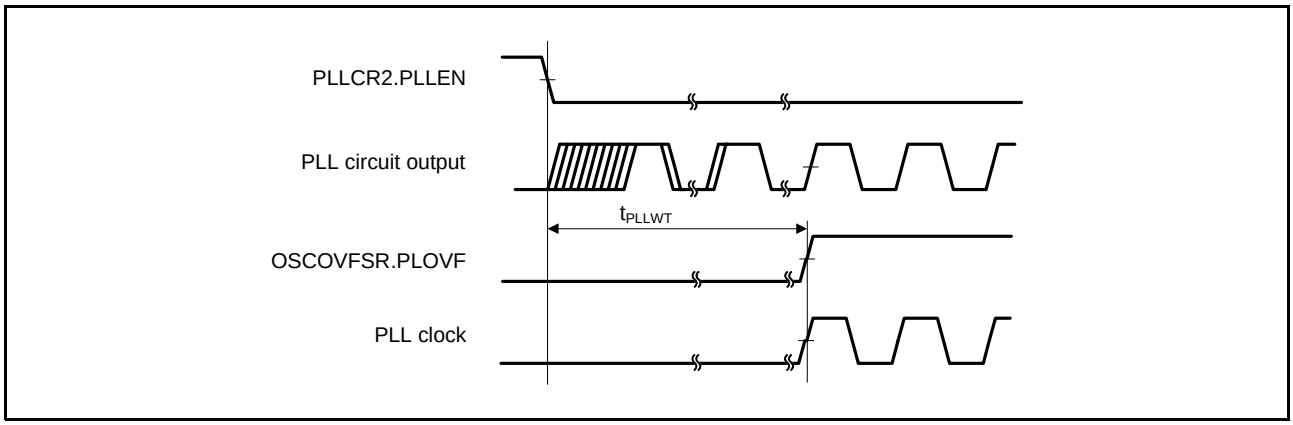


Figure 45.11 PLL Clock Oscillation Start Timing

Table 45.21 Sub-Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 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 45.12
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}

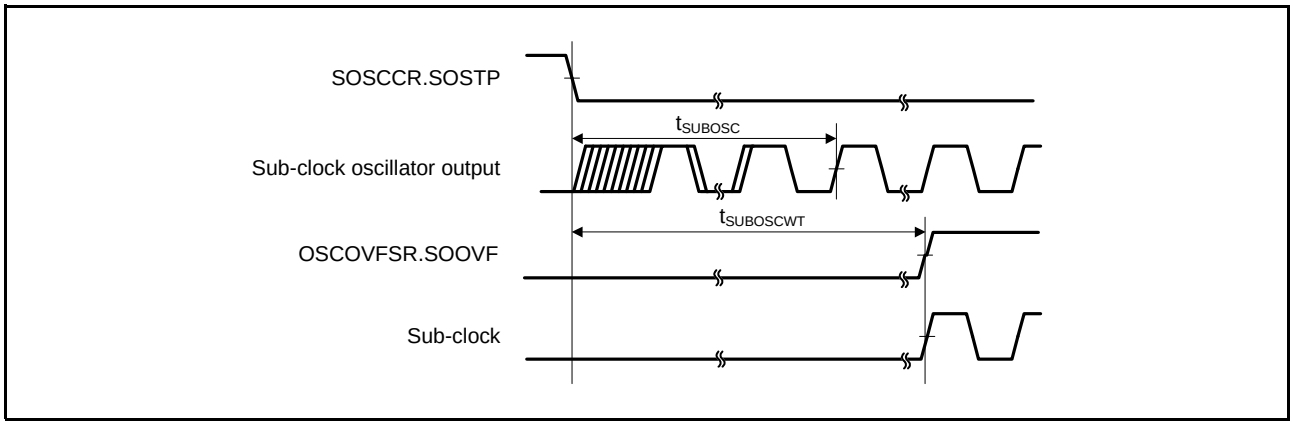


Figure 45.12 Sub-Clock Oscillation Start Timing

45.4.3 Timing of Recovery from Low Power Consumption Modes

Table 45.22 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.13
		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.

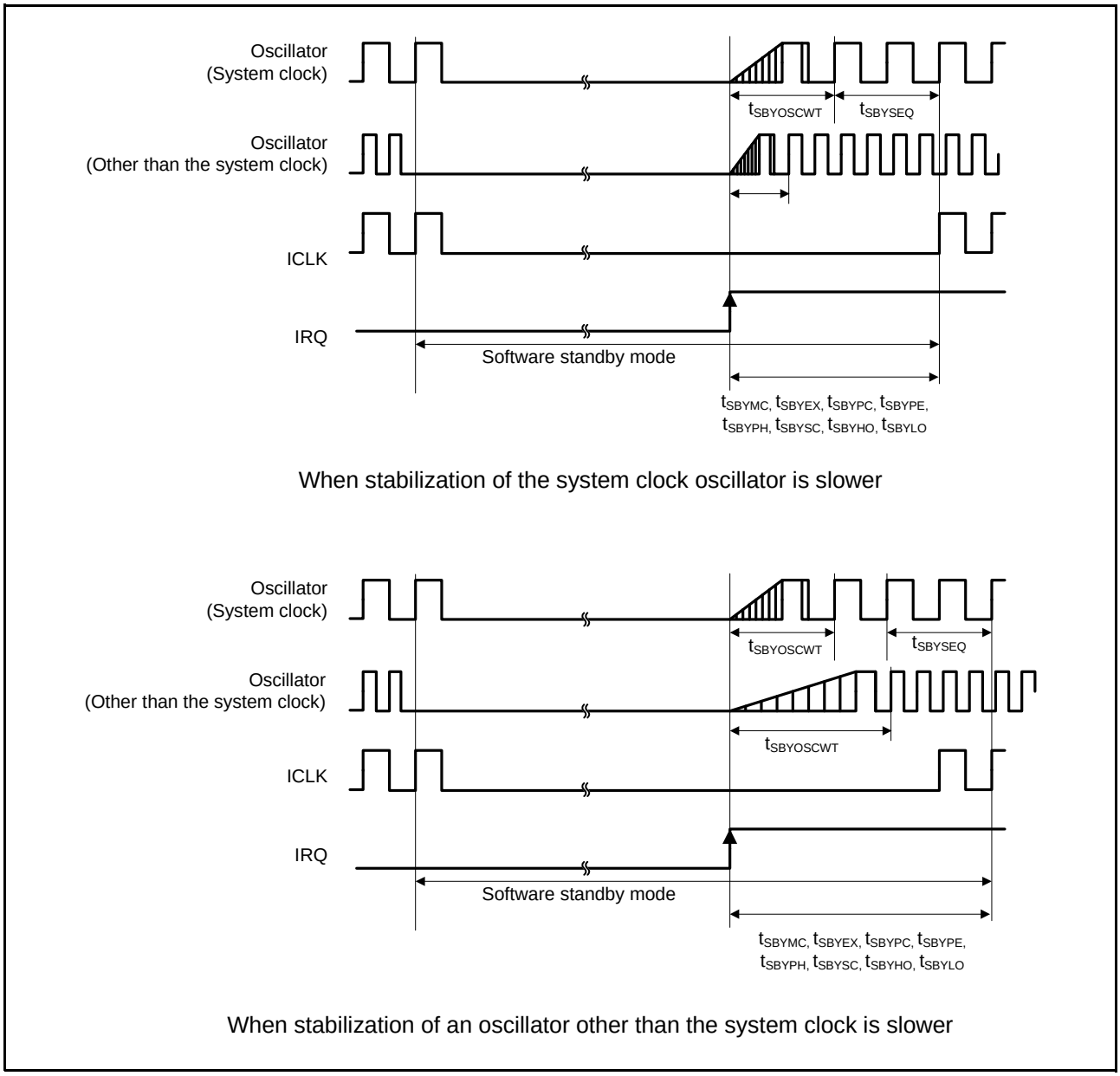


Figure 45.13 Software Standby Mode Recovery Timing

Table 45.23 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.14
Waiting time after recovery from deep software standby mode	t_{DSBYWT}	31	—	32	t_{Lcyc}	

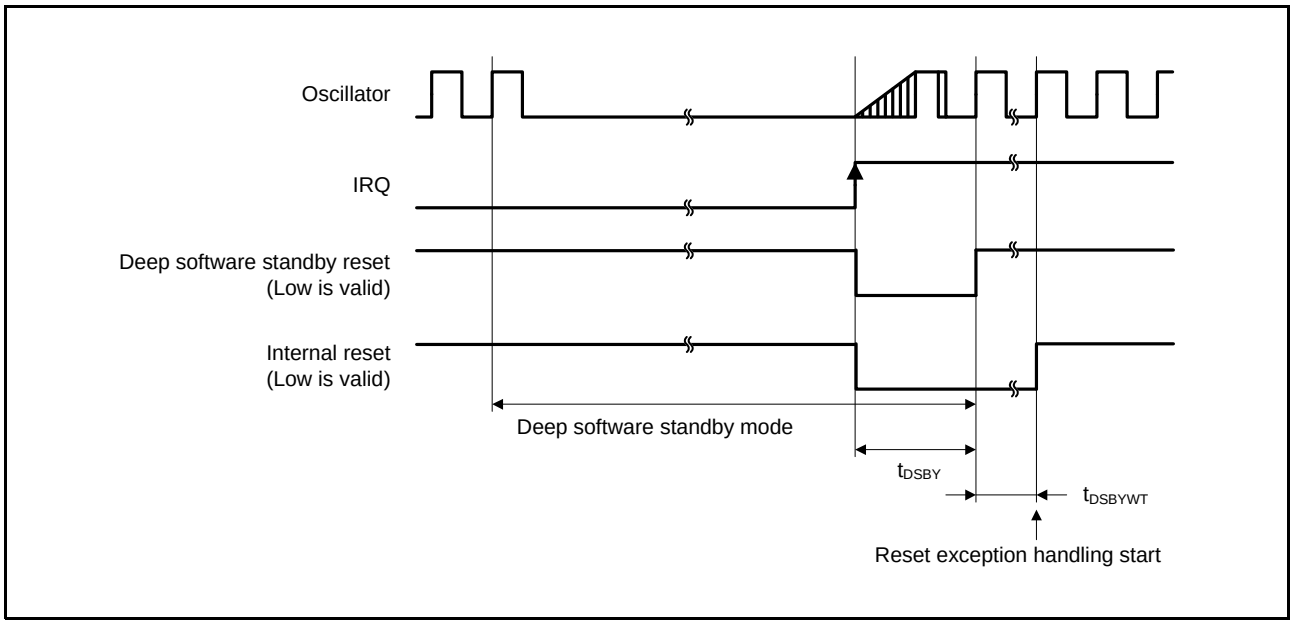


Figure 45.14 Deep Software Standby Mode Recovery Timing

45.4.4 Control Signal Timing

Table 45.24 Control Signal Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.*1	Typ.	Max.	Unit	Test Conditions*1
NMI pulse width	t_{NMIW}	200	—	—	ns	$2 \times t_{PBcyc} \leq 200$ ns, Figure 45.15
		$2 \times t_{PBcyc}$	—	—		$2 \times t_{PBcyc} > 200$ ns, Figure 45.15
IRQ pulse width	t_{IRQW}	200	—	—	ns	$2 \times t_{PBcyc} \leq 200$ ns, Figure 45.16
		$2 \times t_{PBcyc}$	—	—		$2 \times t_{PBcyc} > 200$ ns, Figure 45.16

Note 1. t_{PBcyc}: PCLKB cycle

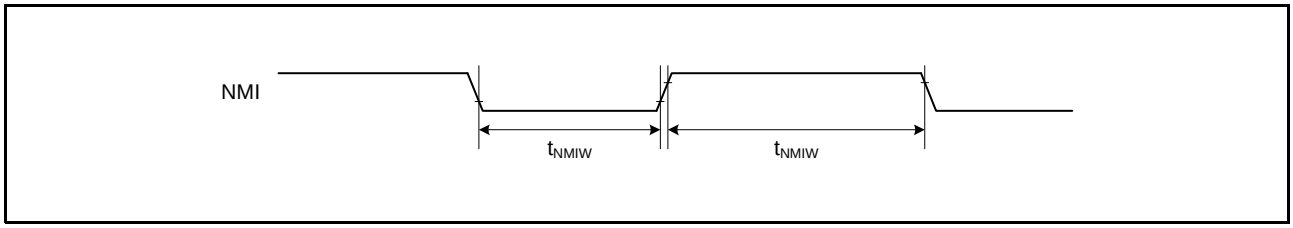


Figure 45.15 NMI Interrupt Input Timing

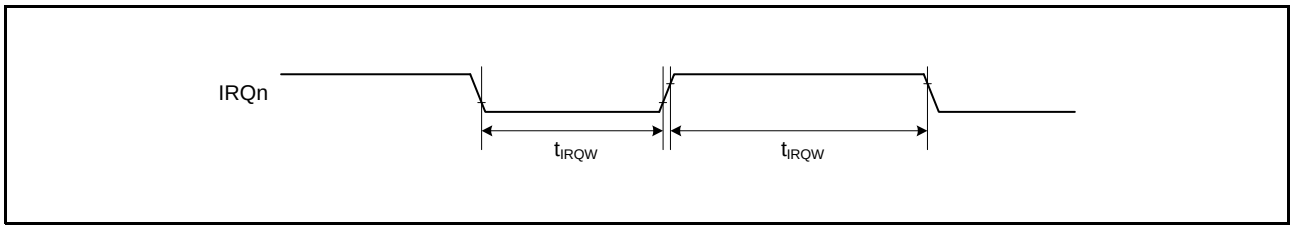


Figure 45.16 IRQ Interrupt Input Timing

45.4.5 Bus Timing

Table 45.25 Bus Timing (1)

Conditions: 4.5 V ≤ VCC ≤ 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	12.5	ns	Figure 45.17 to Figure 45.22
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 45.23
WAIT# hold time	t_{WTH}	0	—	ns	

Table 45.26 Bus Timing (2)

Conditions: 2.7 V ≤ VCC < 4.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	25	ns	Figure 45.17 to Figure 45.22
Byte control delay time	t_{BCD}	—	25	ns	
CS# delay time	t_{CSD}	—	25	ns	
ALE delay time	t_{ALED}	—	25	ns	
RD# delay time	t_{RSD}	—	25	ns	
Read data setup time	t_{RDS}	25	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	—	25	ns	
Write data delay time	t_{WDD}	—	25	ns	
Write data hold time	t_{WDH}	0	—	ns	
WAIT# setup time	t_{WTS}	25	—	ns	Figure 45.23
WAIT# hold time	t_{WTH}	0	—	ns	

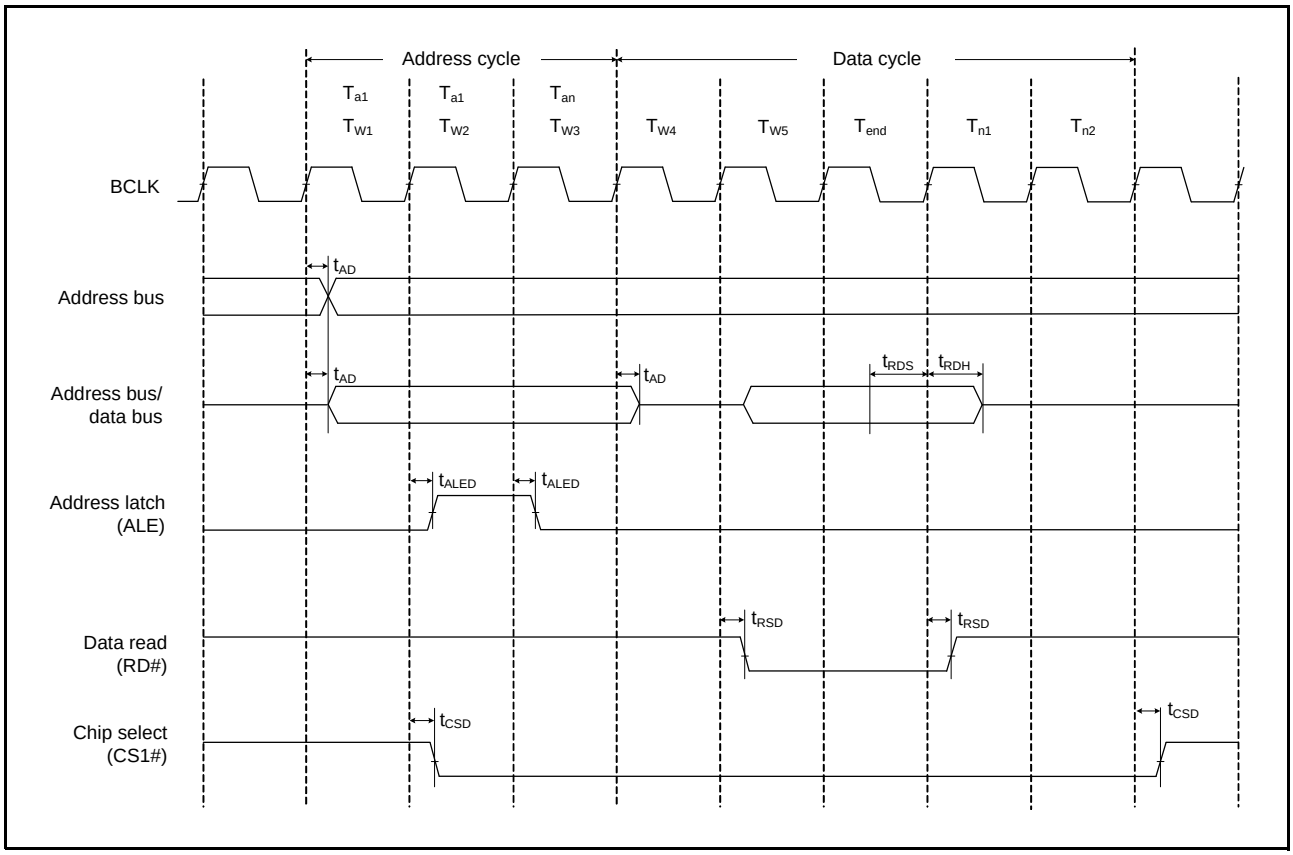


Figure 45.17 Address/Data Multiplexed Bus Read Access Timing

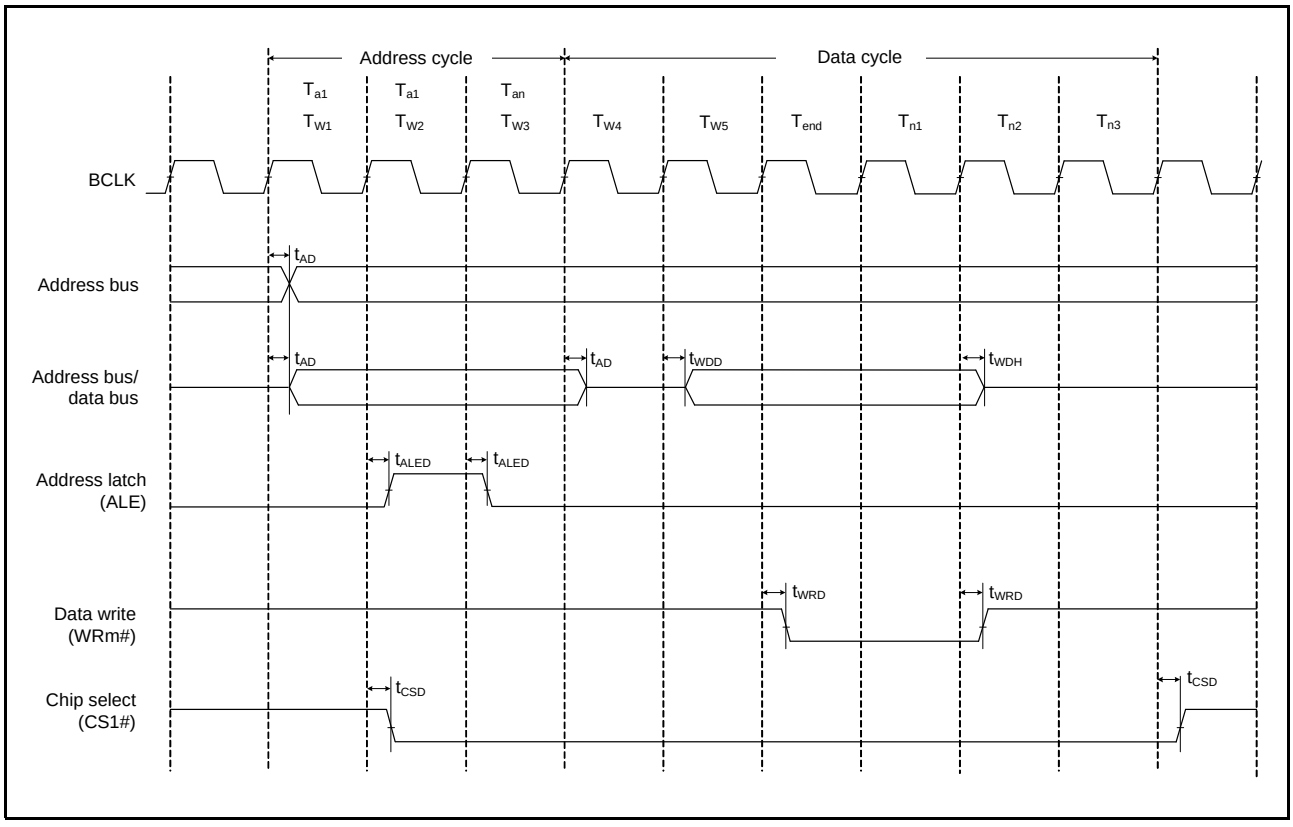


Figure 45.18 Address/Data Multiplexed Bus Write Access Timing

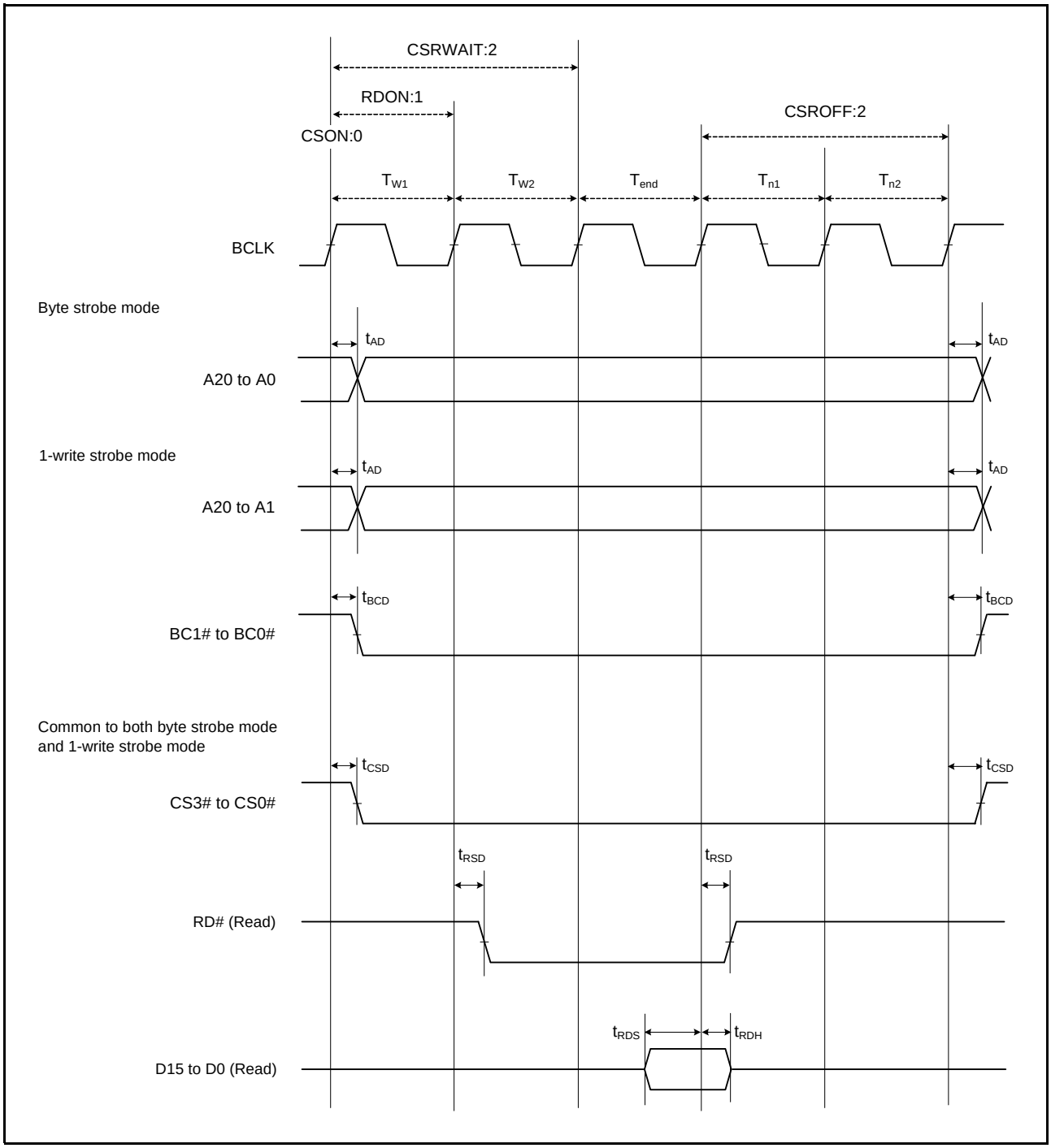
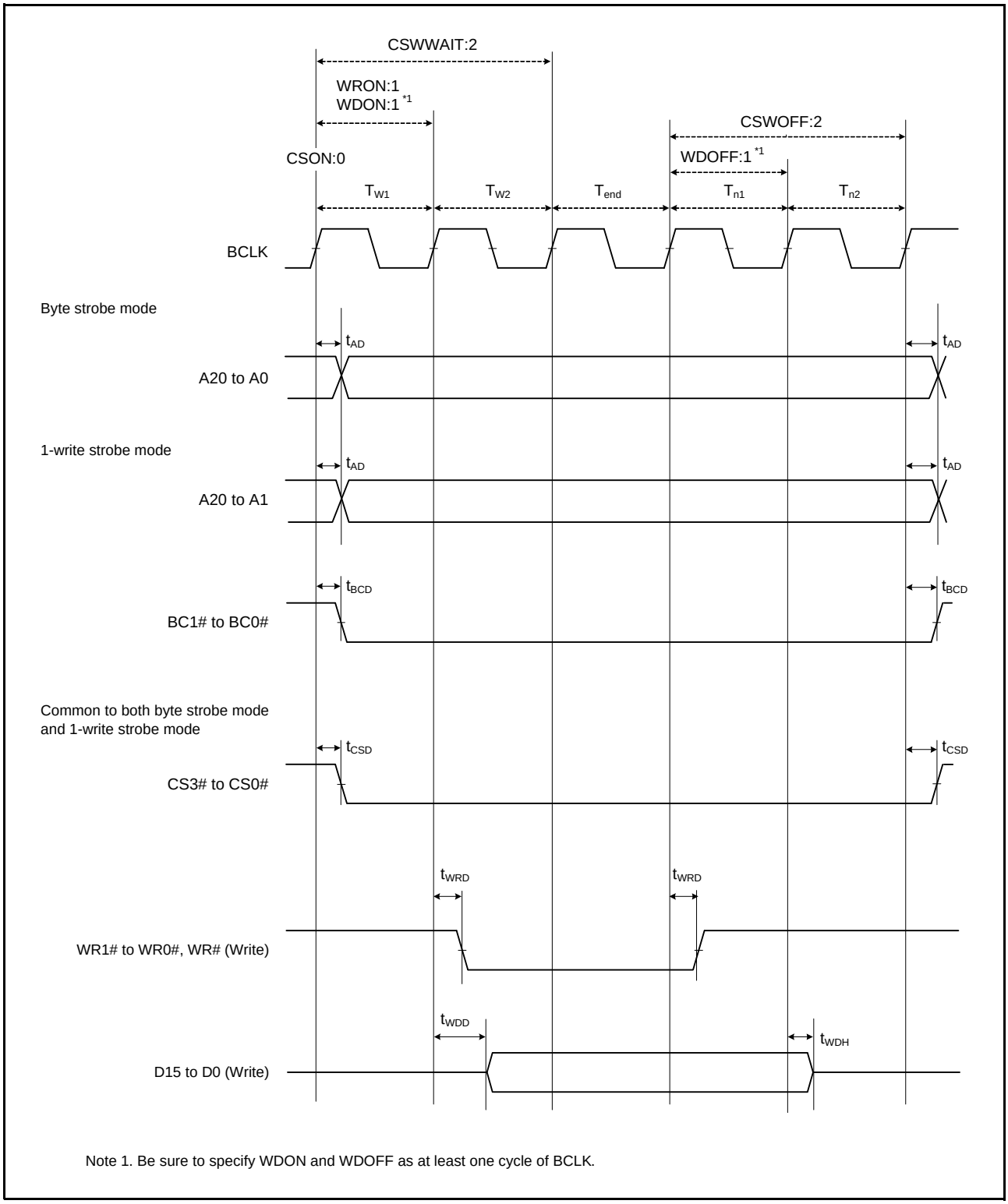


Figure 45.19 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of BCLK.

Figure 45.20 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

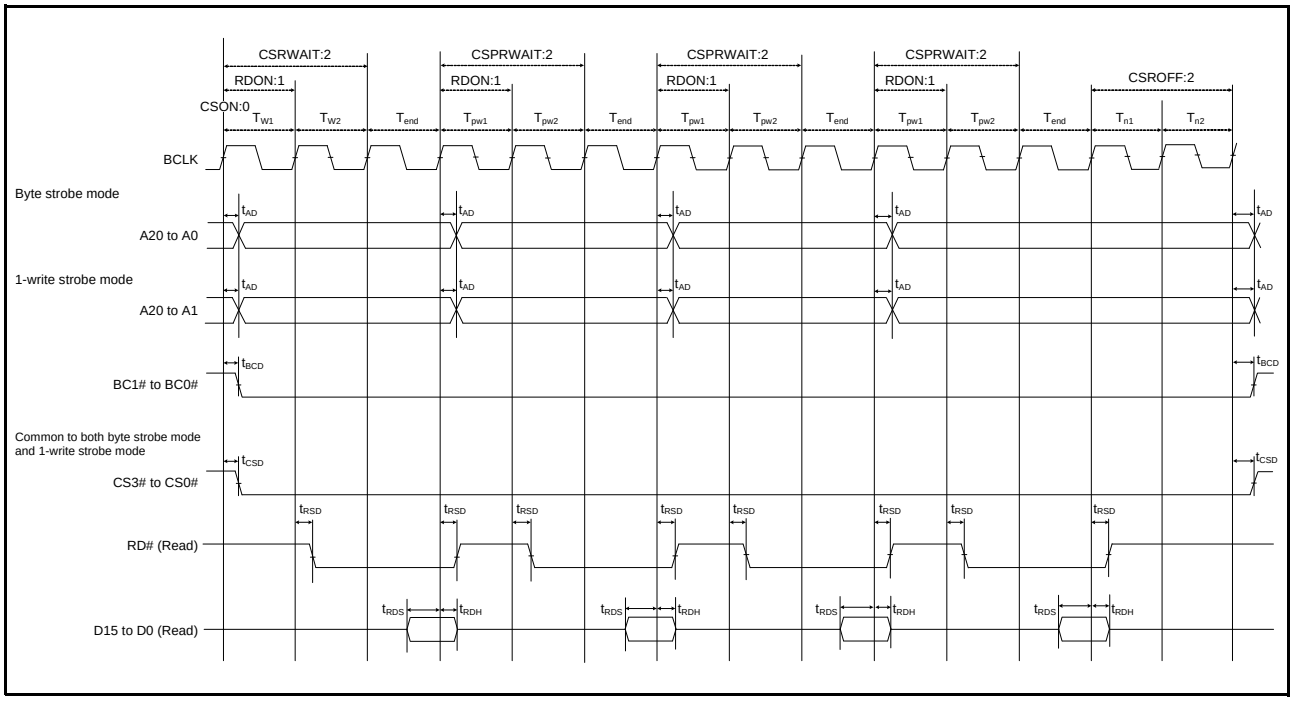


Figure 45.21 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)

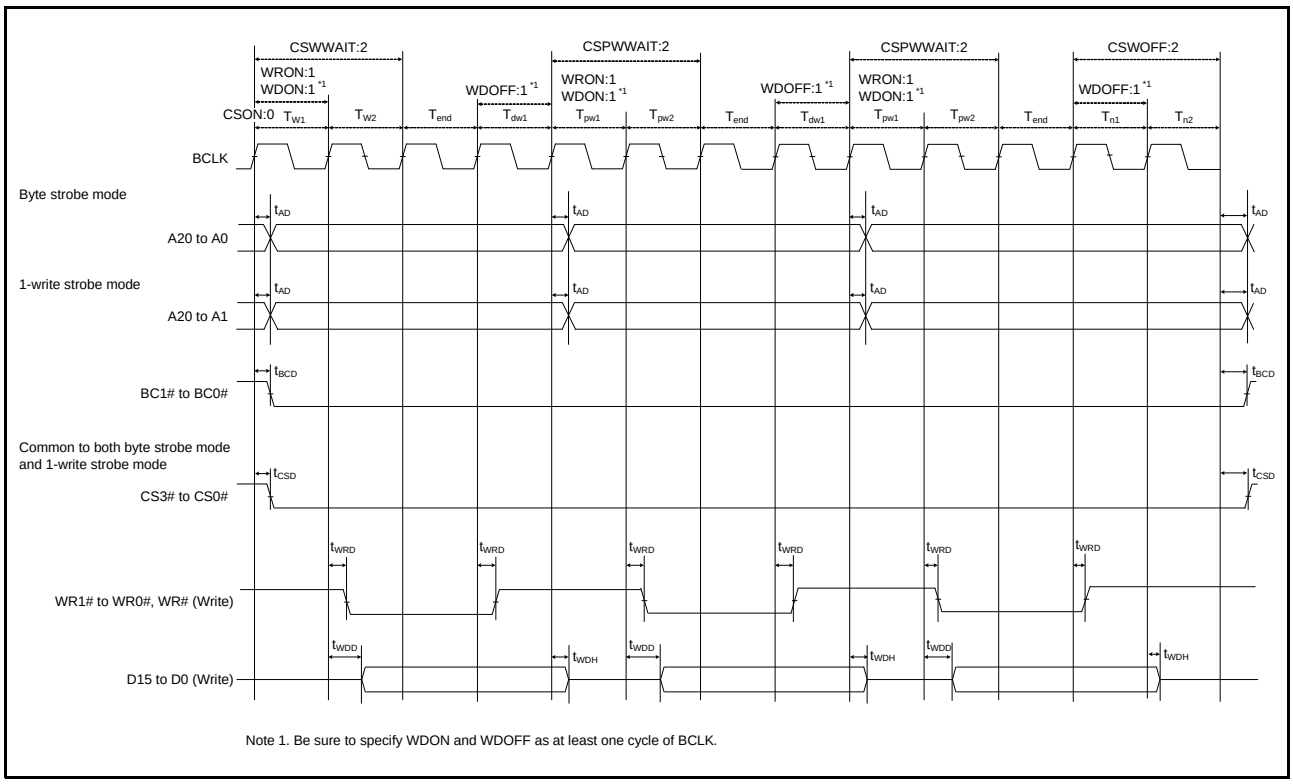


Figure 45.22 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

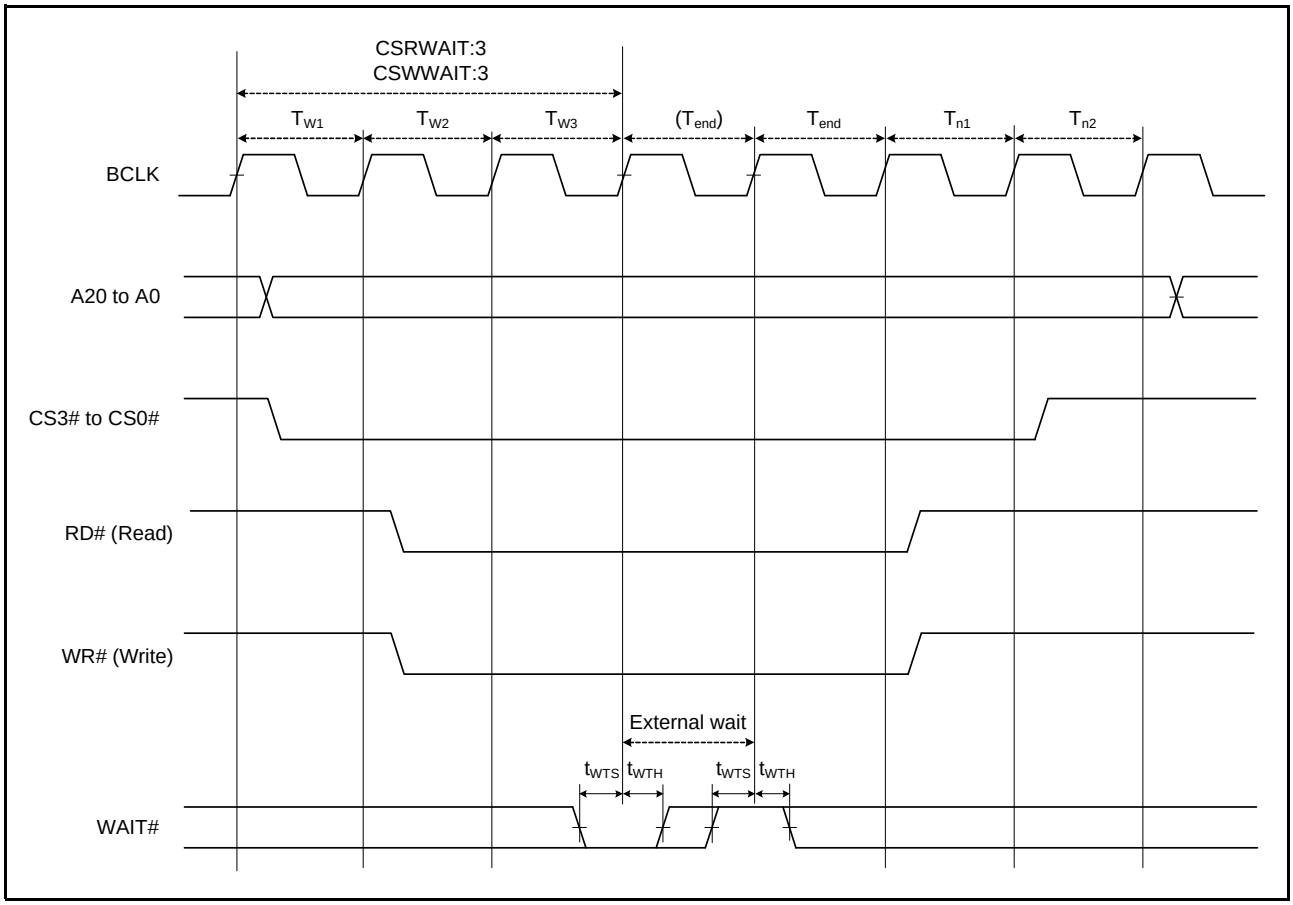


Figure 45.23 External Bus Timing/External Wait Control

45.4.6 Timing of On-Chip Peripheral Modules

45.4.6.1 I/O Port

Table 45.27 I/O Port Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.24

Note 1. t_{PBcyc}: PCLKB cycle

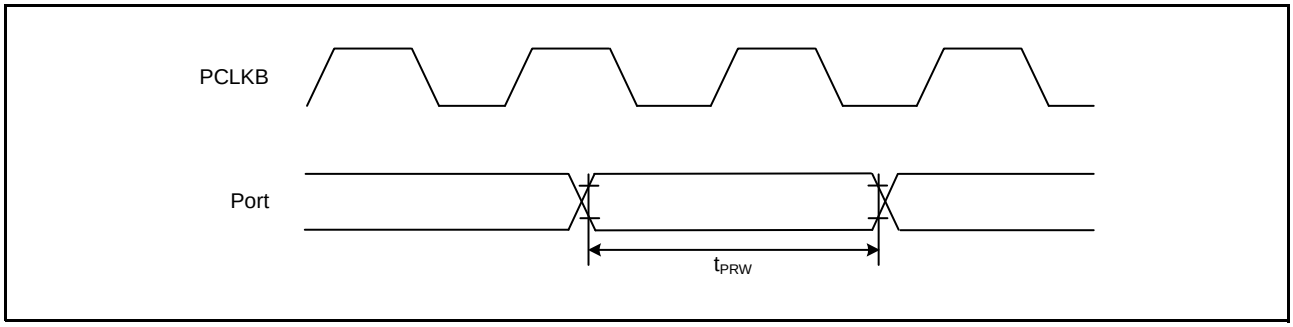


Figure 45.24 I/O Port Input Timing

45.4.6.2 TMR

Table 45.28 TMR Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.25
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

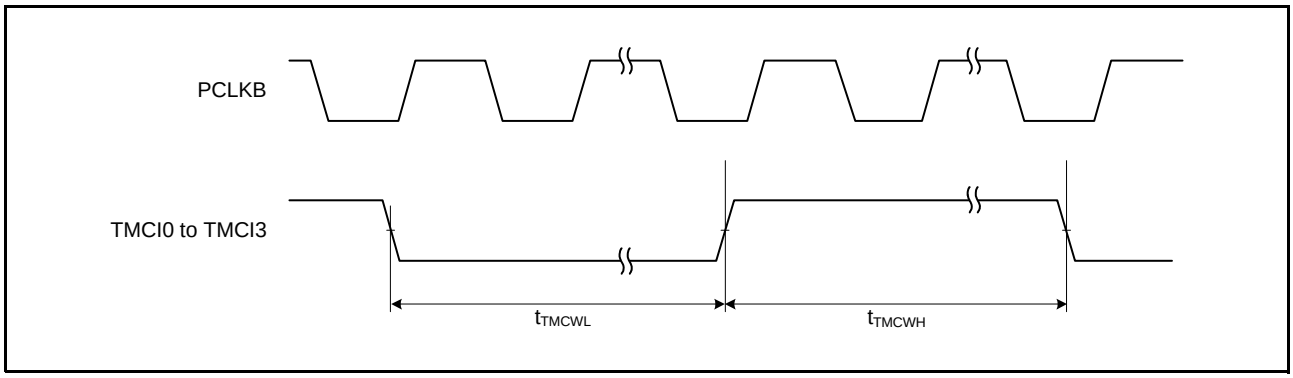


Figure 45.25 TMR Clock Input Timing

45.4.6.3 CMTW

Table 45.29 CMTW Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.26
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

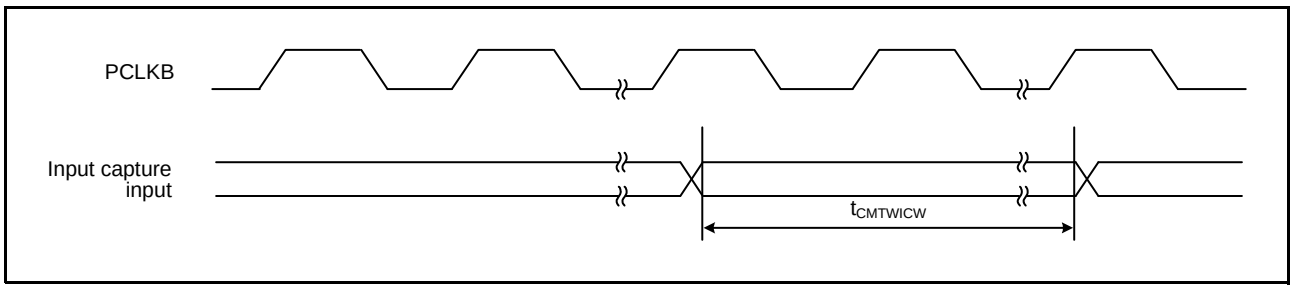


Figure 45.26 CMTW Input Capture Input Timing

45.4.6.4 MTU

Table 45.30 MTU Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.27
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{MTCKWH}, t_{MTCKWL}	1.5	—	t_{PAcyc}	Figure 45.28
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{PAcyc}: PCLKA cycle

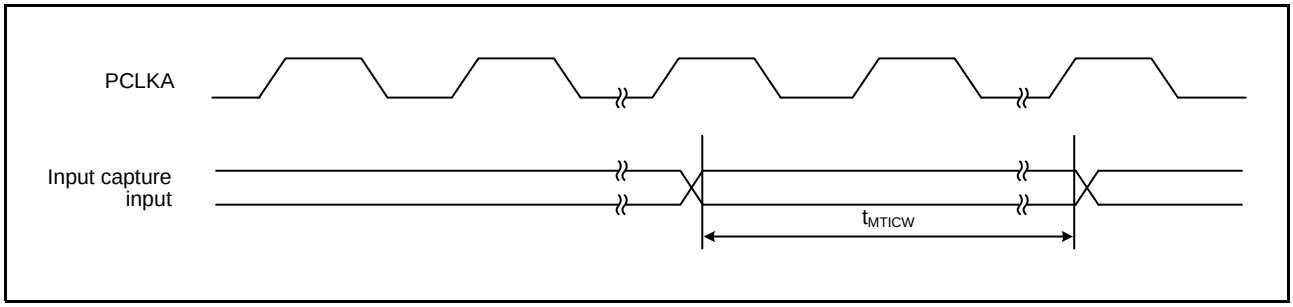


Figure 45.27 MTU Input Capture Input Timing

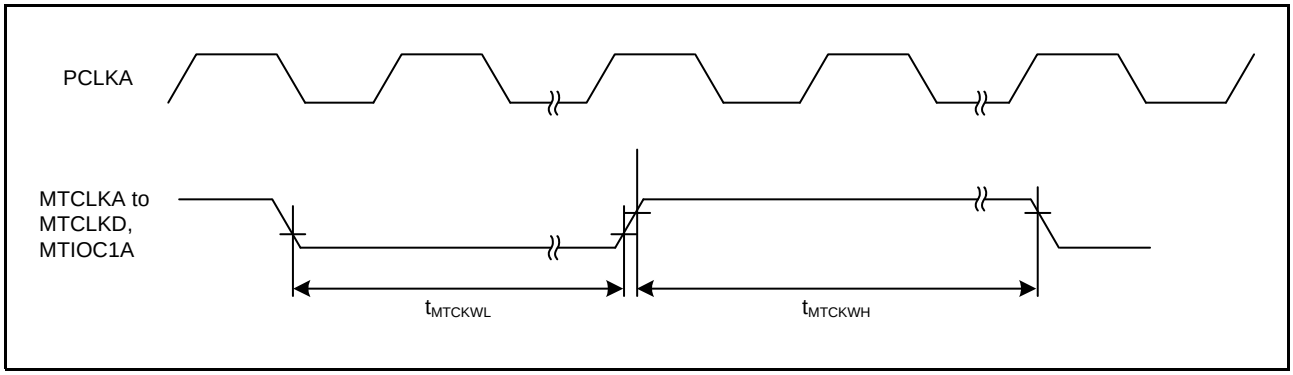


Figure 45.28 MTU Clock Input Timing

45.4.6.5 POE3

Table 45.31 POE3 Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.29
	Output disable time	Transition of the POEn# signal level	t_{POEDI}	—	—	5 PCLKB + 0.24	µs	Figure 45.30 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 45.31
		Register setting	t_{POEDS}	—	—	1 PCLKB + 0.2	µs	Figure 45.32 Time for access to the register is not included.
		Oscillation stop detection	t_{POEDOS}	—	—	21	µs	Figure 45.33

Note 1. t_{PBcyc}: PCLKB cycle

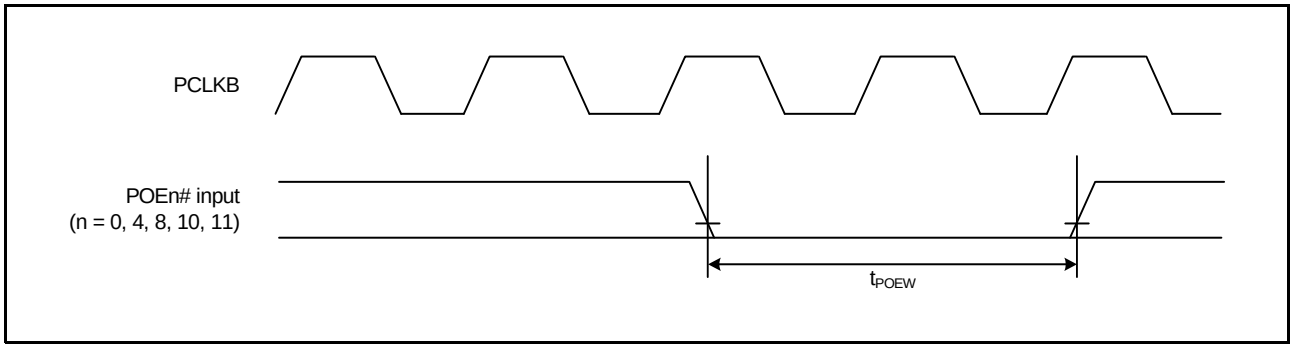


Figure 45.29 POE# Pin Input Timing

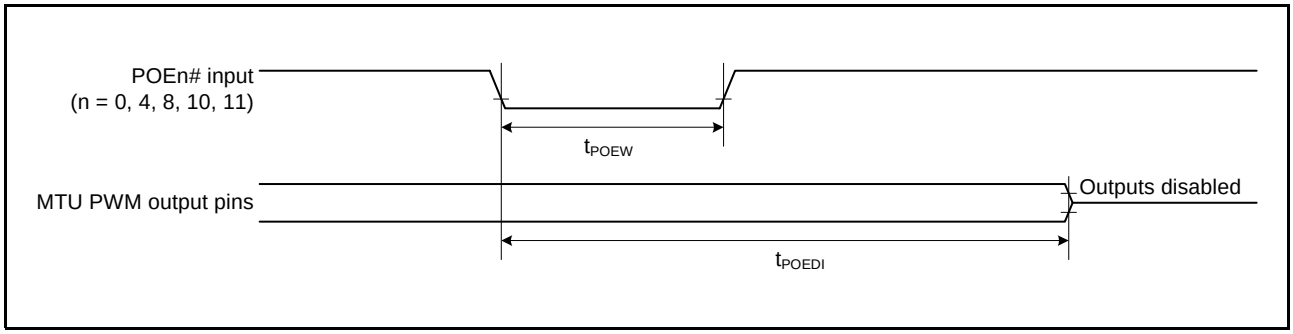
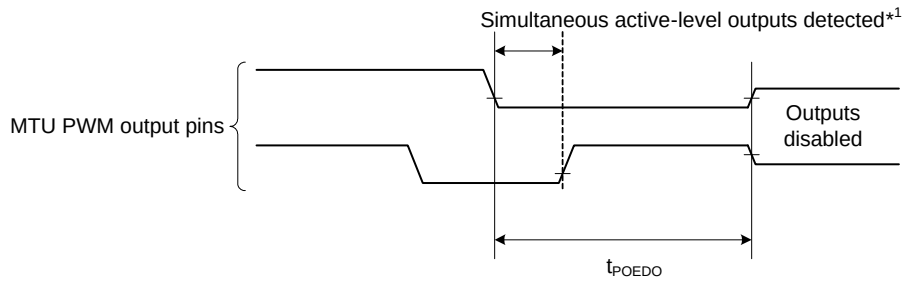


Figure 45.30 Output Disable Time for POE in Response to Transition of the POEn# Signal Level



Note 1. When the active level is set to low.

Figure 45.31 Output Disable Time for POE in Response to the Simultaneous Conduction of Output Pins

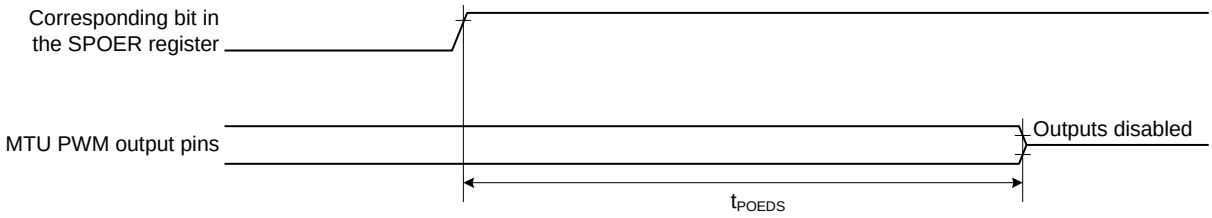


Figure 45.32 Output Disable Time for POE in Response to the Register Setting

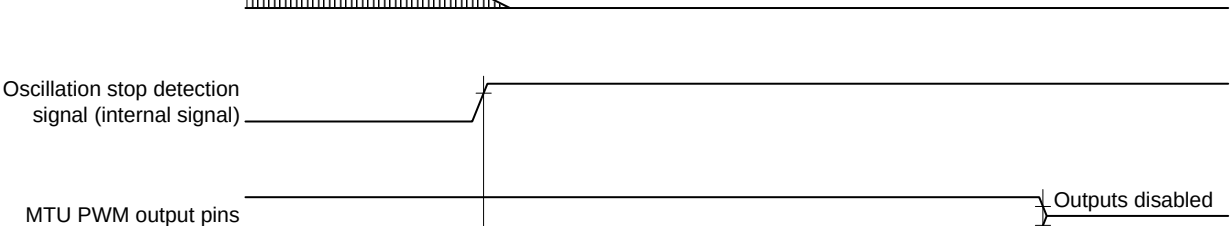


Figure 45.33 Output Disable Time for POE in Response to the Oscillation Stop Detection

45.4.6.6 A/D Converter Trigger

Table 45.32 A/D Converter Trigger Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.34

Note 1. t_{PBcyc}: PCLKB cycle

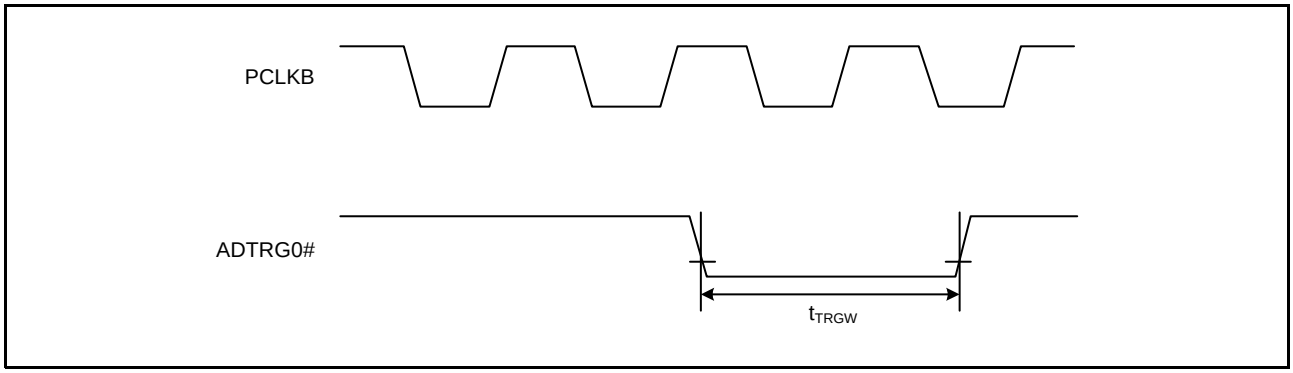


Figure 45.34 A/D Converter Trigger Input Timing

45.4.6.7 CAC

Table 45.33 CAC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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

45.4.6.8 SCI

Table 45.34 SCIk, SCIh, and SCIm Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.35	
		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	VCC ≥ 4.5 V	Figure 45.36
				—	33		VCC < 4.5 V	
	Receive data setup time	Clock synchronous	t_{RXS}	15	—	ns	Figure 45.36	
	Receive data hold time	Clock synchronous	t_{RXH}	5	—	ns		
SCIm	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PAcyc}	Figure 45.35	
		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	VCC ≥ 4.5 V	Figure 45.36
				—	20		VCC < 4.5 V	
		Slave		—	28		VCC ≥ 4.5 V	
				—	33		VCC < 4.5 V	
	Receive data setup time	Clock synchronous	t_{RXS}	20	—	ns	Figure 45.36	
	Receive data hold time	Clock synchronous	t_{RXH}	5	—	ns		

Note 1. t_{PBcyc}: PCLKB cycle; t_{PAcyc}: PCLKA cycle

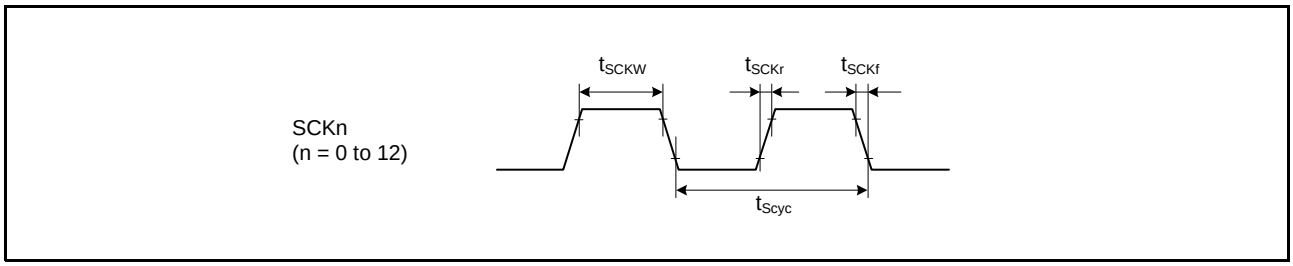


Figure 45.35 SCK Clock Input Timing

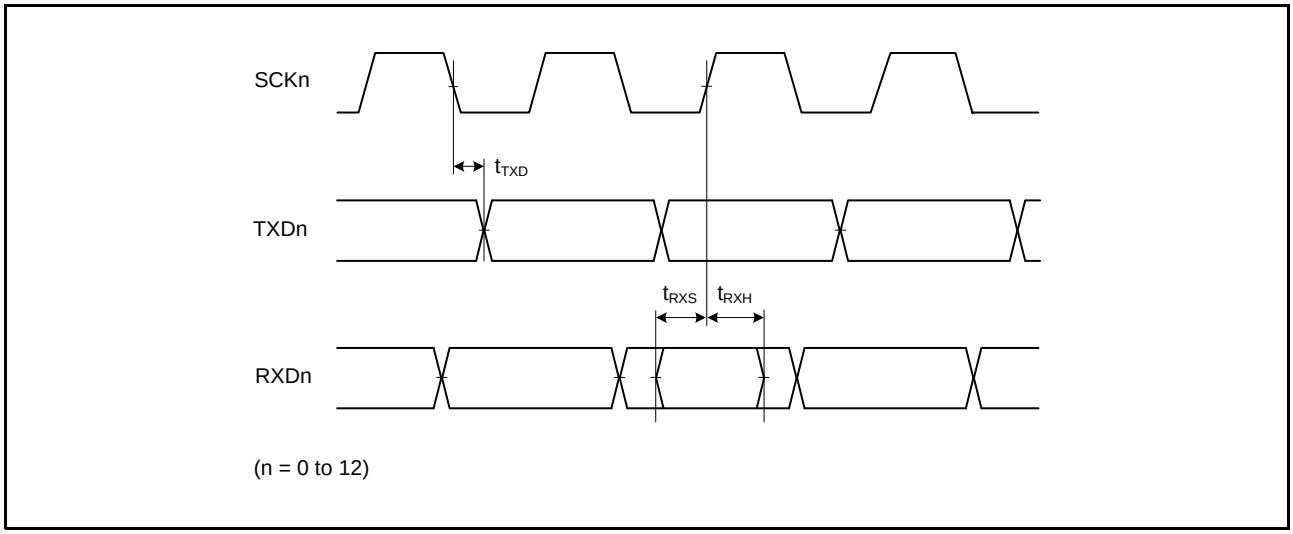


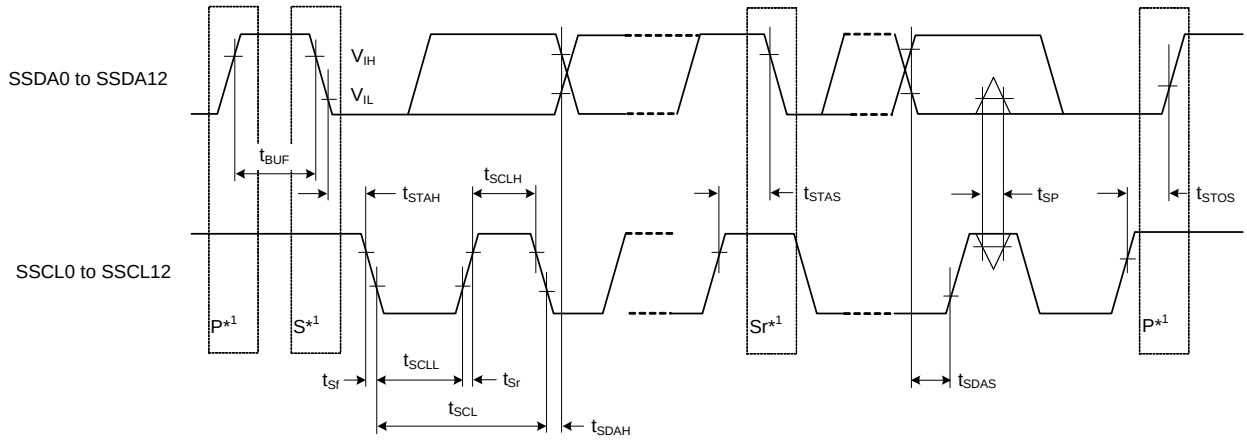
Figure 45.36 SCI Input/Output Timing: Clock Synchronous Mode

Table 45.35 Simple IIC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.37
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	$4 \times t_{Pcyc}$	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	$4 \times t_{Pcyc}$	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
$V_{IH} = 0.7 \times VCC$, $V_{IL} = 0.3 \times VCC$
$V_{OL} = 0.6$ V, $I_{OL} = 6$ mA

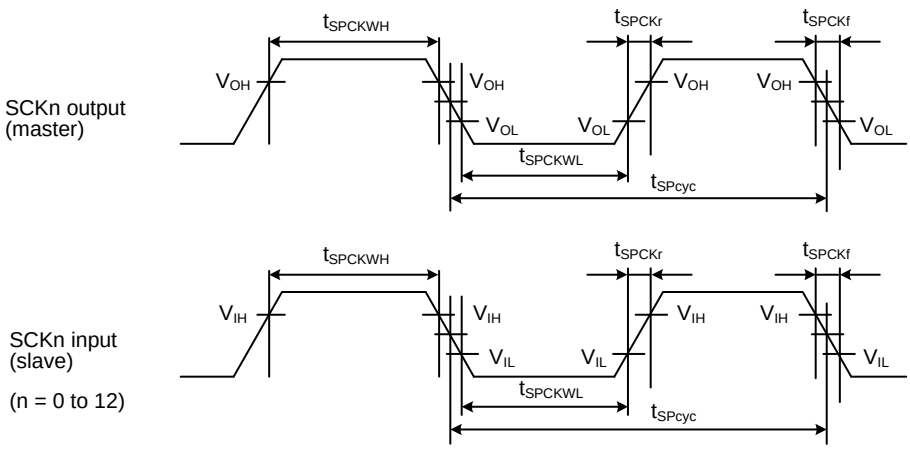
Figure 45.37 Simple IIC Bus Interface Input/Output Timing

Table 45.36 Simple SPI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.38
	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 45.39 to Figure 45.42
	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 45.41, Figure 45.42
	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.



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Figure 45.38 Simple SPI Clock Timing

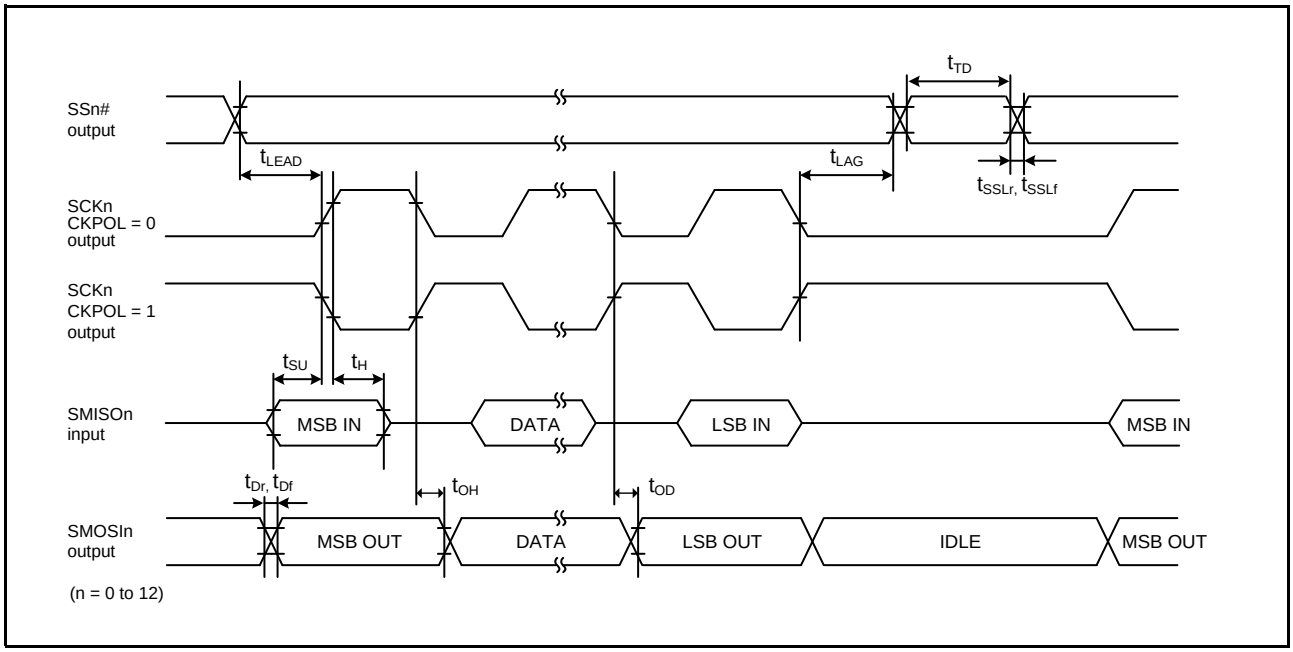


Figure 45.39 Simple SPI Timing (Master, CKPH = 1)

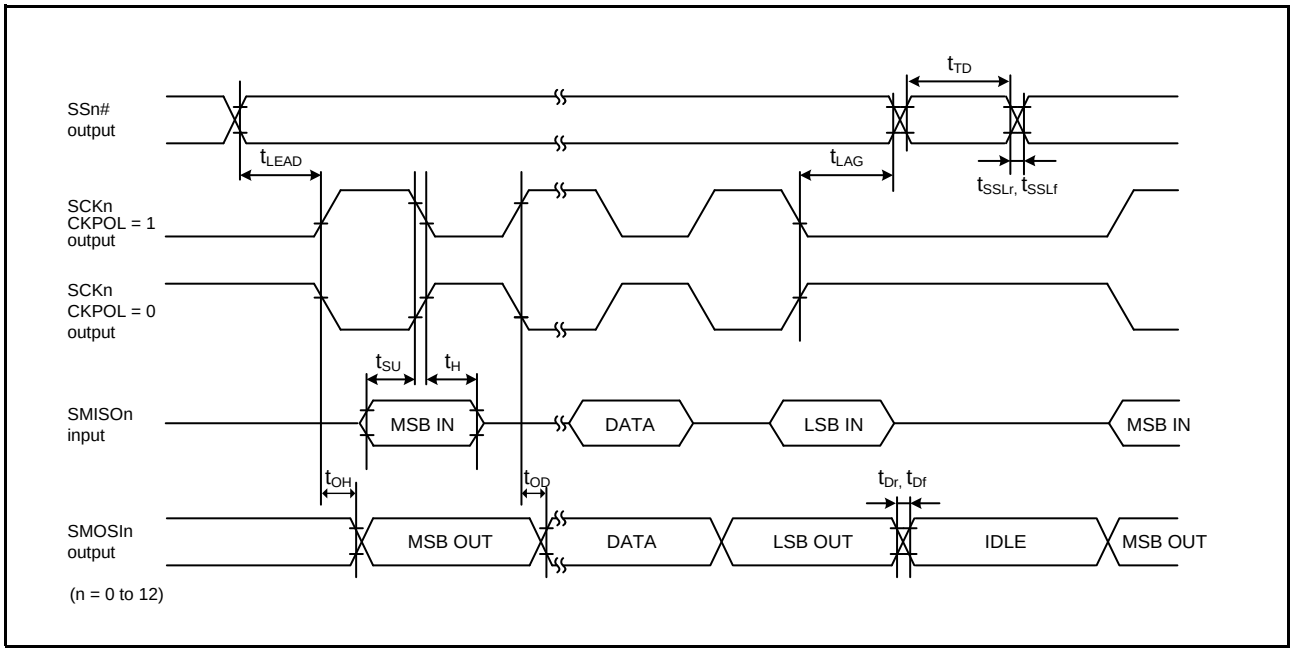


Figure 45.40 Simple SPI Timing (Master, CKPH = 0)

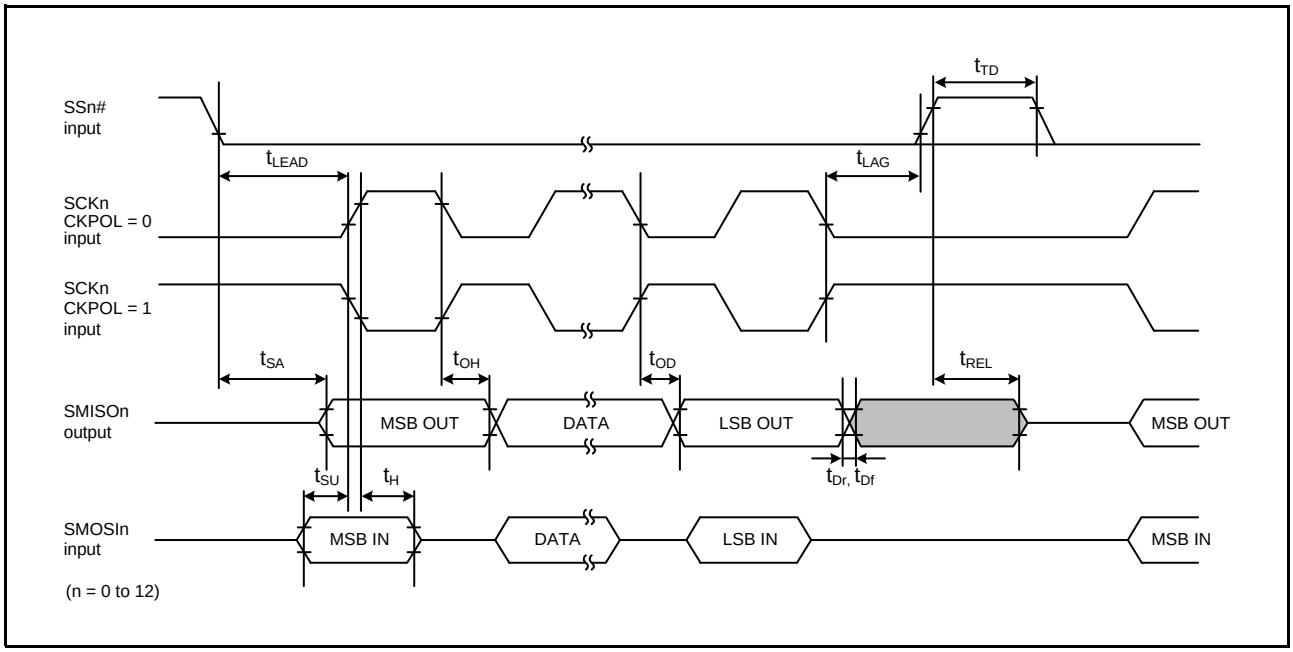


Figure 45.41 Simple SPI Timing (Slave, CKPH = 1)

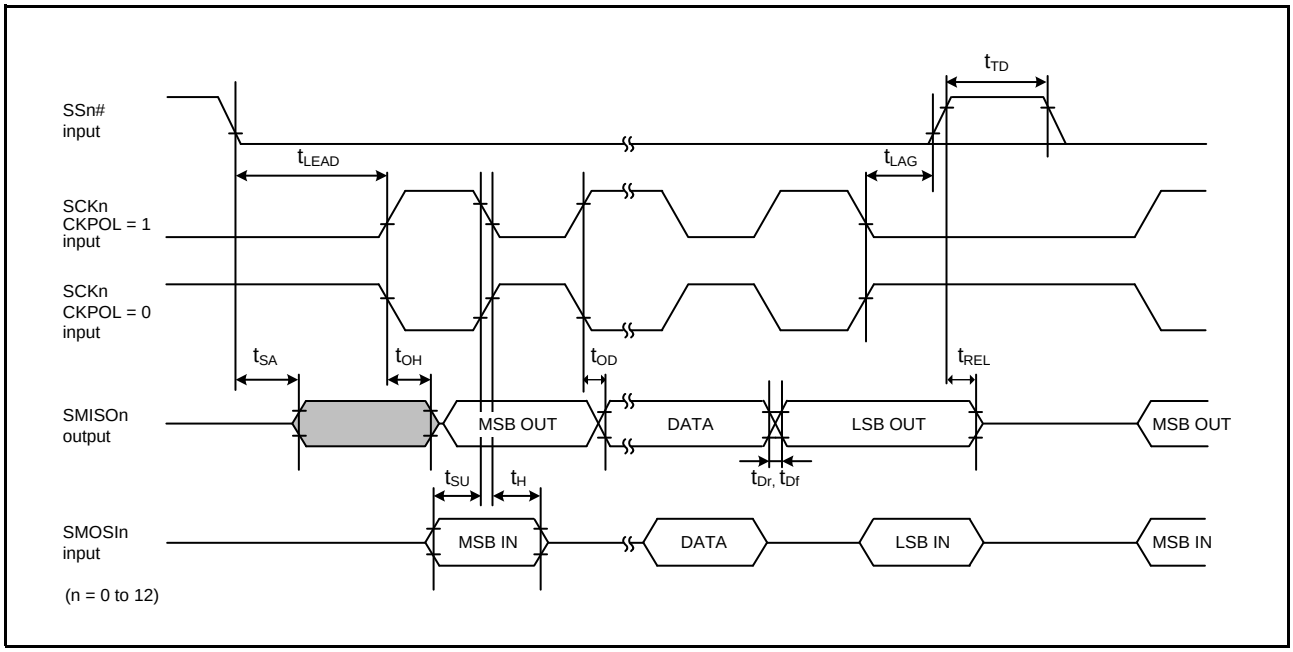


Figure 45.42 Simple SPI Timing (Slave, CKPH = 0)

45.4.6.9 RSCI

Table 45.37 RSCI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, C = 30 pF,
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 45.43	
		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}	–1.5	—	ns	VCC ≥ 4.5 V	Figure 45.44
				3.5	—		VCC < 4.5 V	
		Slave		2.5	—		Figure 45.44	
	Receive data hold time	Master	t_{RXH}	11	—	ns		
		Slave		2.5	—			
	Transmit data delay time	Master	t_{TXD}	—	4	ns		
		Slave		—	17		VCC ≥ 4.5 V	Figure 45.44
				—	22		VCC < 4.5 V	

Note 1. t_{PAcyc}: PCLKA cycle; t_{Scyc}: SCK cycle

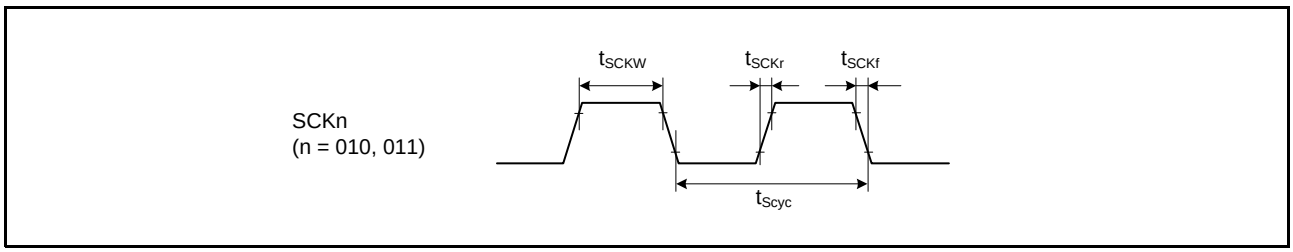


Figure 45.43 SCK Clock Input Timing

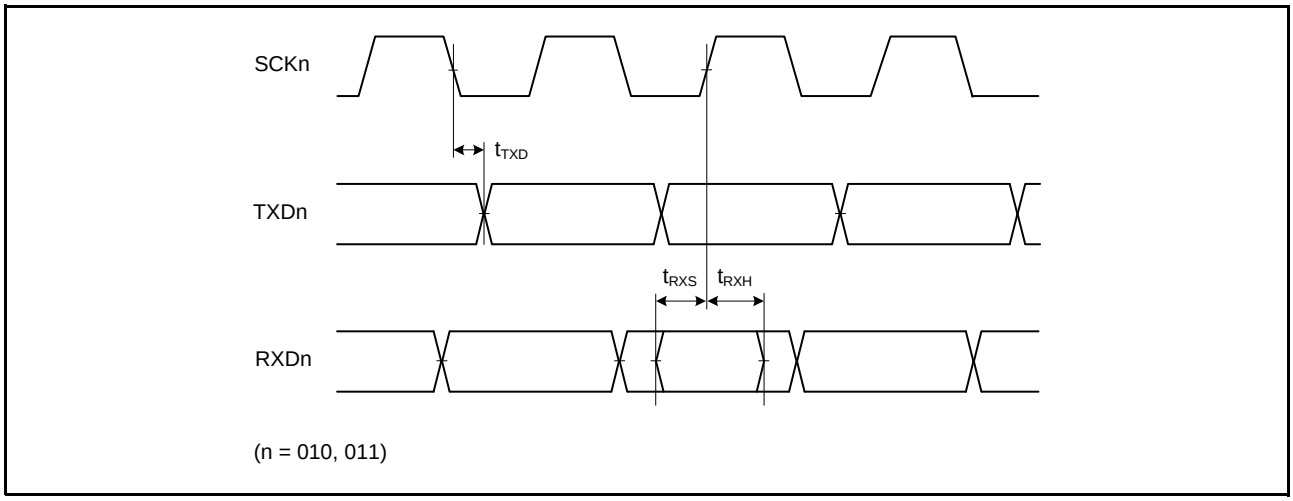


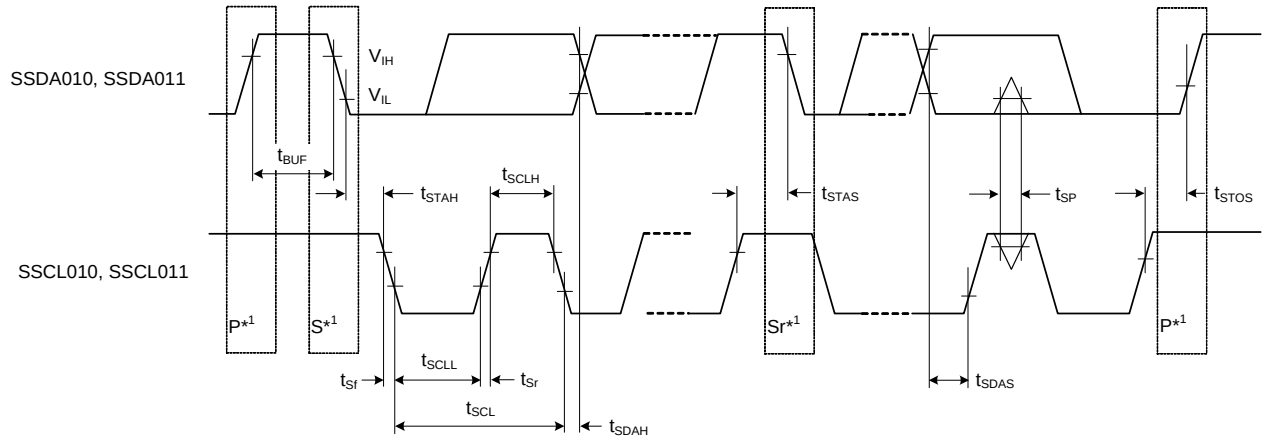
Figure 45.44 RSCI Input/Output Timing: Clock Synchronous Mode

Table 45.38 Simple IIC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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 45.45
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	$4 \times t_{PAcyc}$	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	$4 \times t_{PAcyc}$	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
$V_{IH} = 0.7 \times VCC$, $V_{IL} = 0.3 \times VCC$
$V_{OL} = 0.6$ V, $I_{OL} = 6$ mA

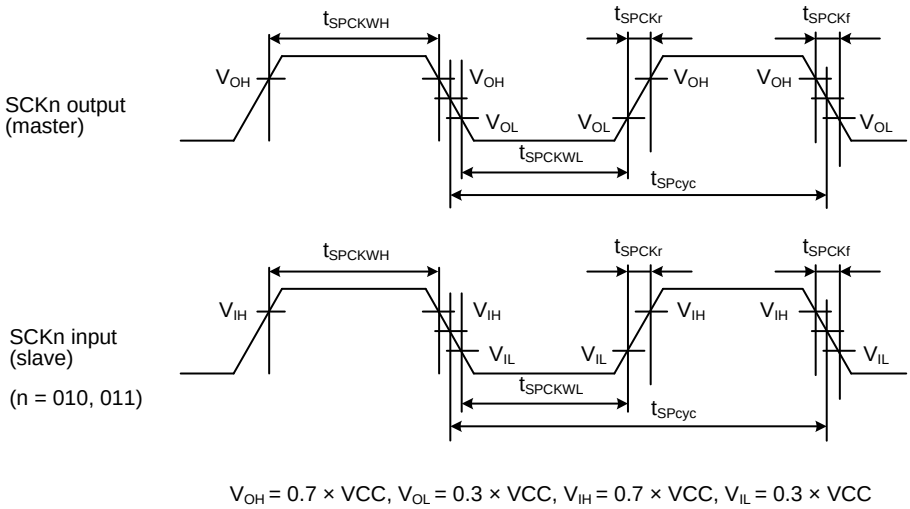
Figure 45.45 Simple IIC Bus Interface Input/Output Timing

Table 45.39 Simple SPI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 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
Simple SPI	SCK clock cycle output (master)		t_{SPcyc}	2	—	t_{PAcyc}	Figure 45.46
	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 45.47 to Figure 45.50
		Slave		2.5	—		
	Data input hold time	Master	t_H	11	—	ns	
		Slave		2.5	—		
	Data output delay time	Master	t_{OD}	—	4	ns	Figure 45.47 to Figure 45.50
		Slave		—	17		VCC ≥ 4.5V: Figure 45.47 to Figure 45.50
				—	22		VCC < 4.5V: Figure 45.47 to Figure 45.50
	Data output hold time	Master	t_{OH}	–1	—	ns	Figure 45.47 to Figure 45.50
		Slave		0	—		
	Data rise/fall time	Output	t_{Dr}, t_{Df}	—	5	ns	
		Input		—	1	—	
	Slave access time		t_{SA}	—	5	t_{PAcyc}	Figure 45.49, Figure 45.50
	Slave output release time		t_{REL}	—	5	t_{PAcyc}	
	SS input setup time		t_{LEAD}	1	—	t_{SPcyc}	Figure 45.47 to Figure 45.50
	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



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Figure 45.46 Simple SPI Clock Timing

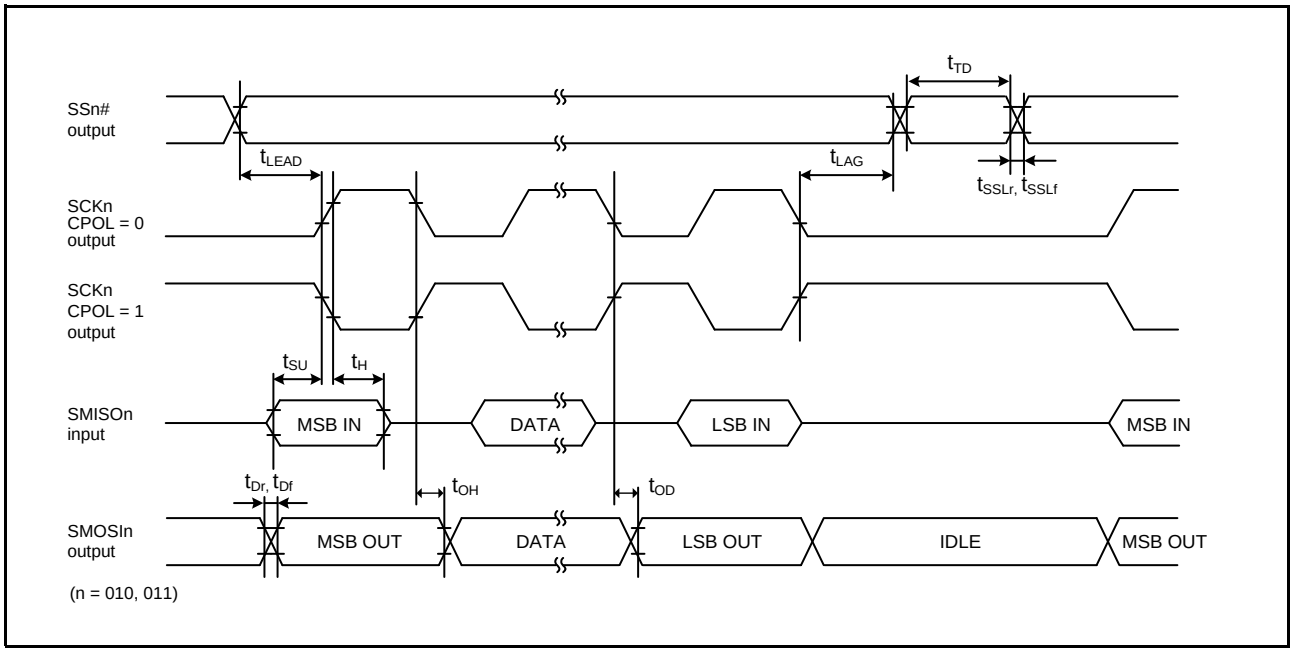


Figure 45.47 Simple SPI Timing (Master, CPHA = 0)

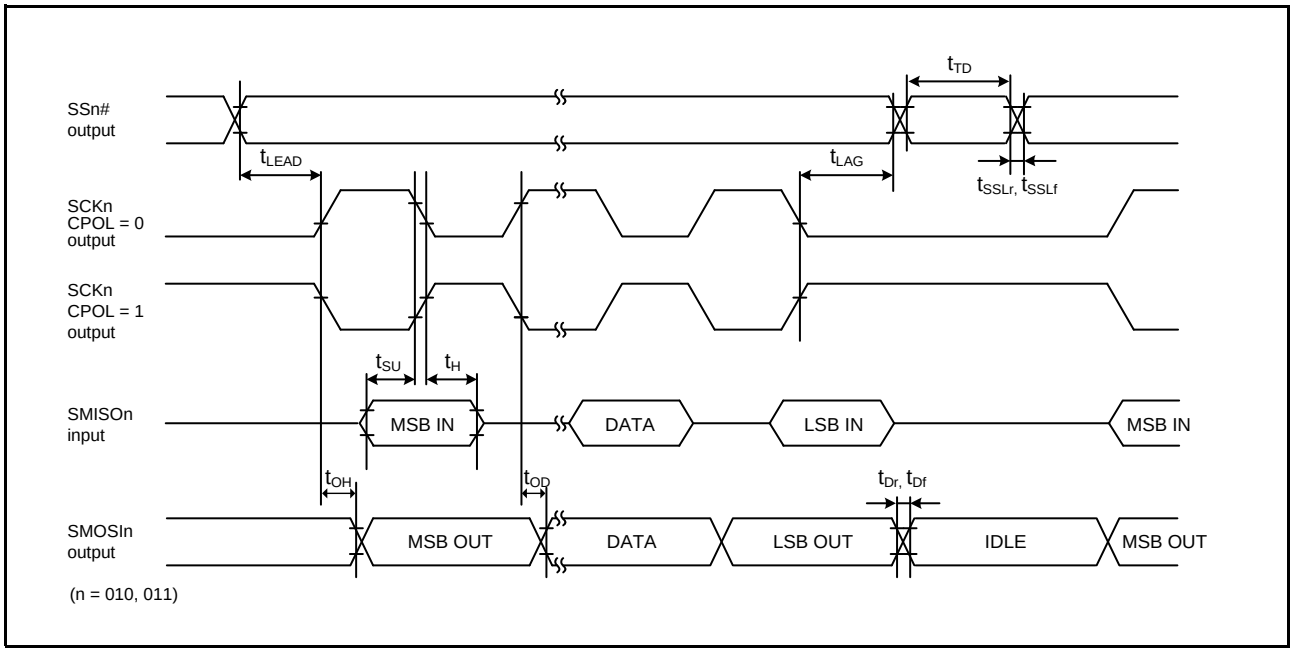


Figure 45.48 Simple SPI Timing (Master, CPHA = 1)

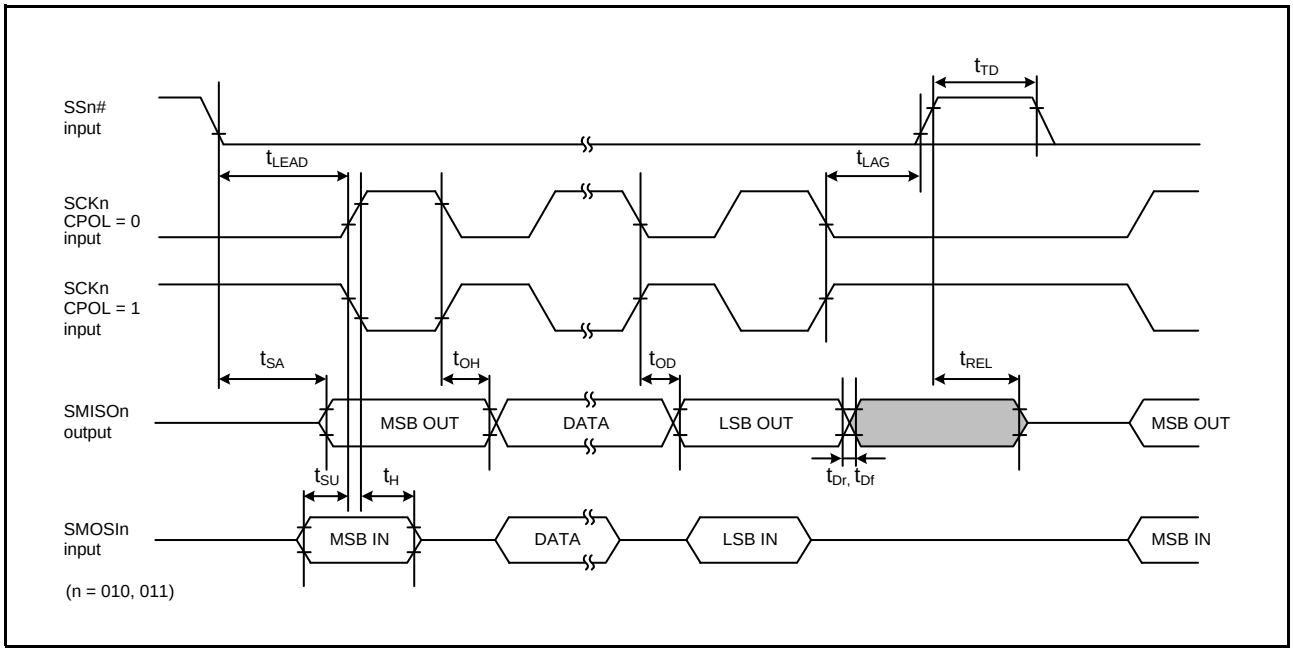


Figure 45.49 Simple SPI Timing (Slave, CPHA = 0)

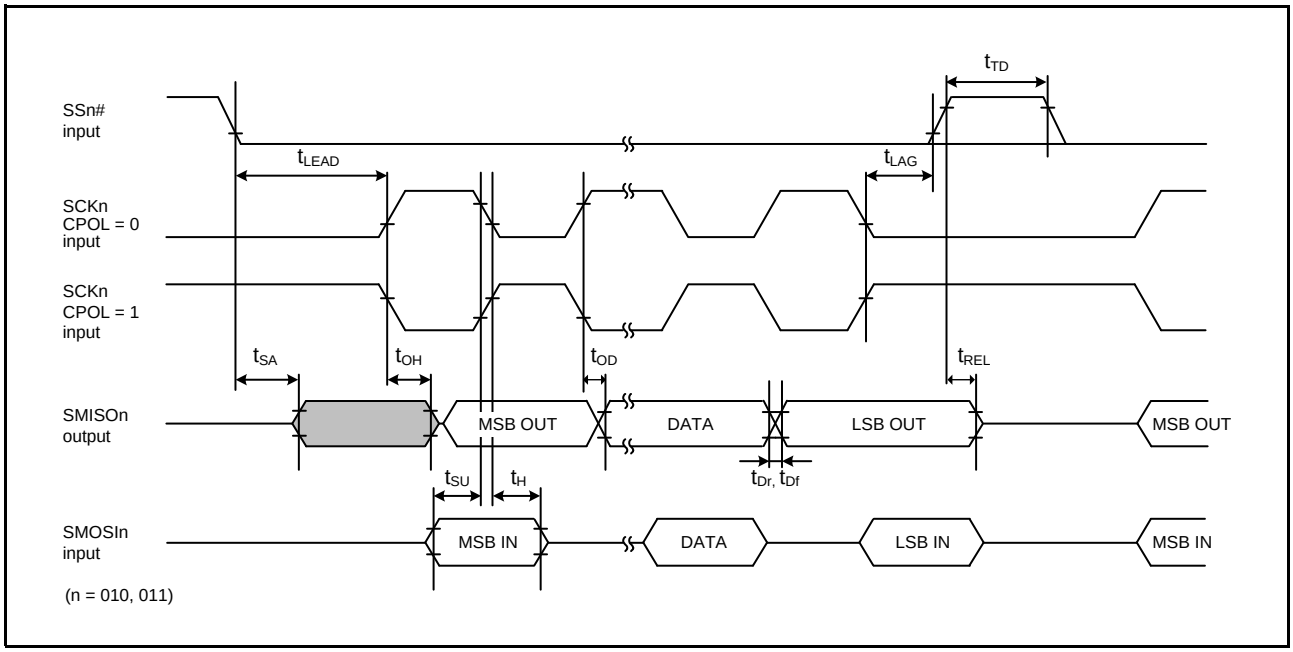


Figure 45.50 Simple SPI Timing (Slave, CPHA = 1)

45.4.6.10 RSPI

Table 45.40 RSPI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, 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 45.51	
		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	VCC ≥ 4.5 V	Figure 45.52 to Figure 45.57
					11	—		VCC < 4.5 V	
		Slave			8.3	—		Figure 45.52 to Figure 45.57	
	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	VCC ≥ 4.5 V	Figure 45.52 to Figure 45.57
					—	11.3		VCC < 4.5 V	
		Slave			—	28		VCC ≥ 4.5 V	
					—	33		VCC < 4.5 V	
	Data output hold time	Master		t_{OH}	0	—	ns	Figure 45.52 to Figure 45.57	
		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	VCC ≥ 4.5 V	Figure 45.56, Figure 45.57
					—	33		VCC < 4.5 V	
	Slave output release time			t_{REL}	—	28	ns	VCC ≥ 4.5 V	
					—	33		VCC < 4.5 V	

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.

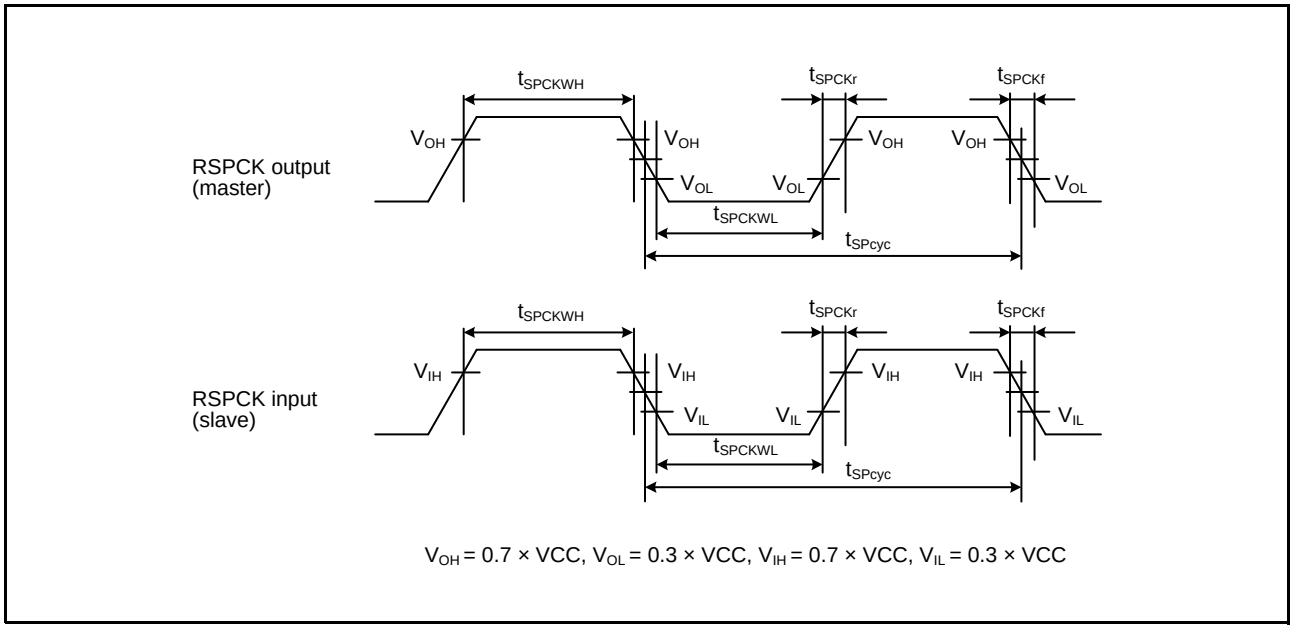


Figure 45.51 RSPI Clock Timing

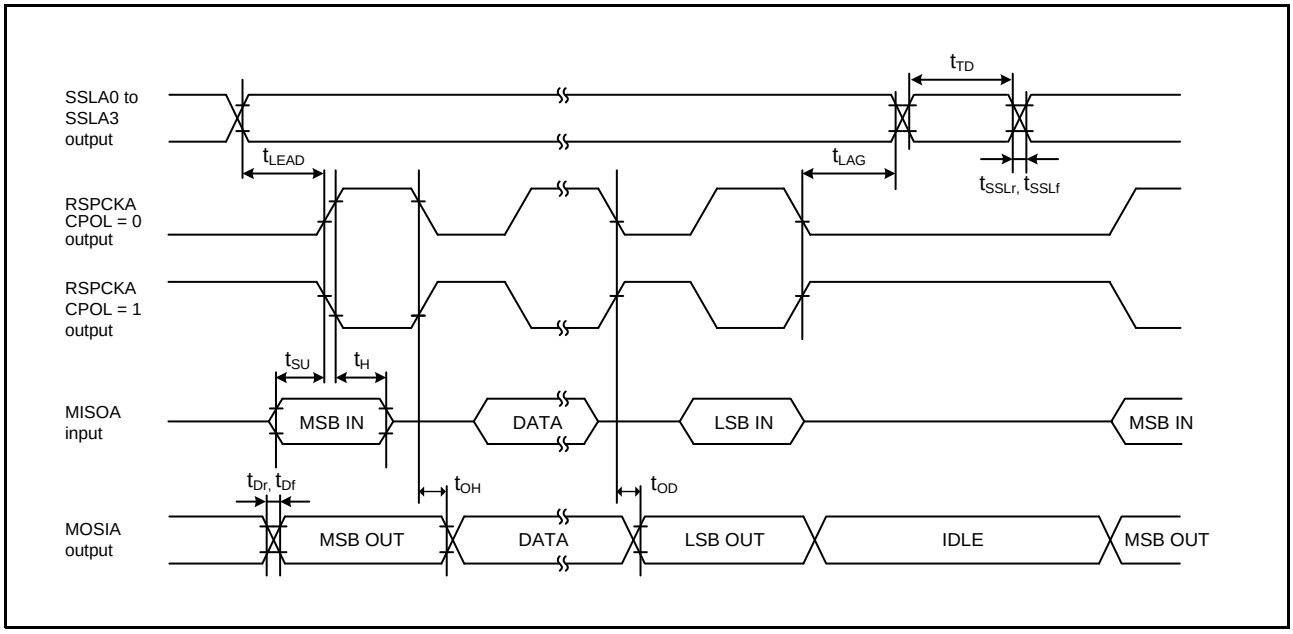


Figure 45.52 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

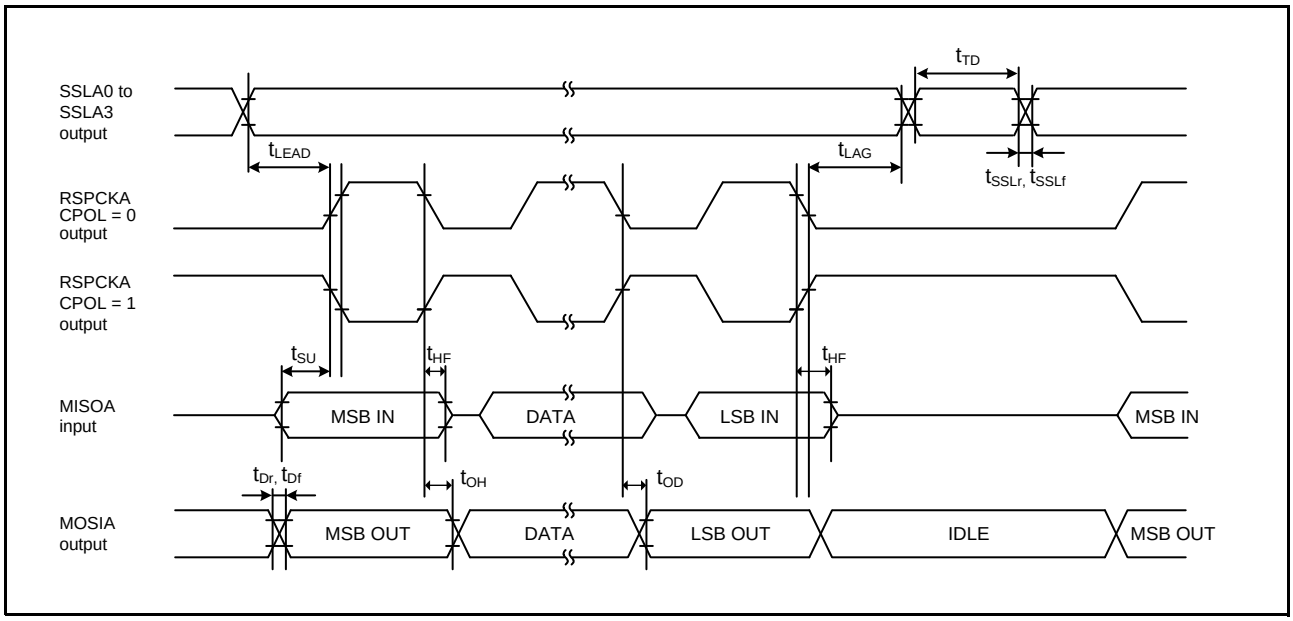


Figure 45.53 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to 1/2)

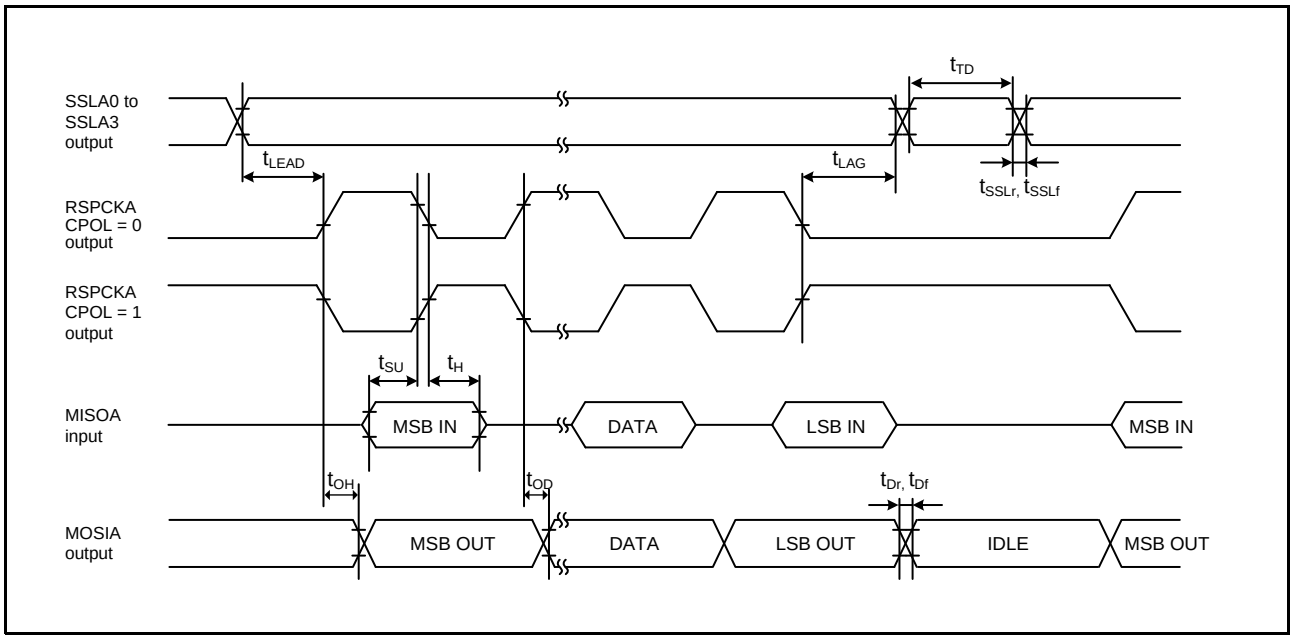


Figure 45.54 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

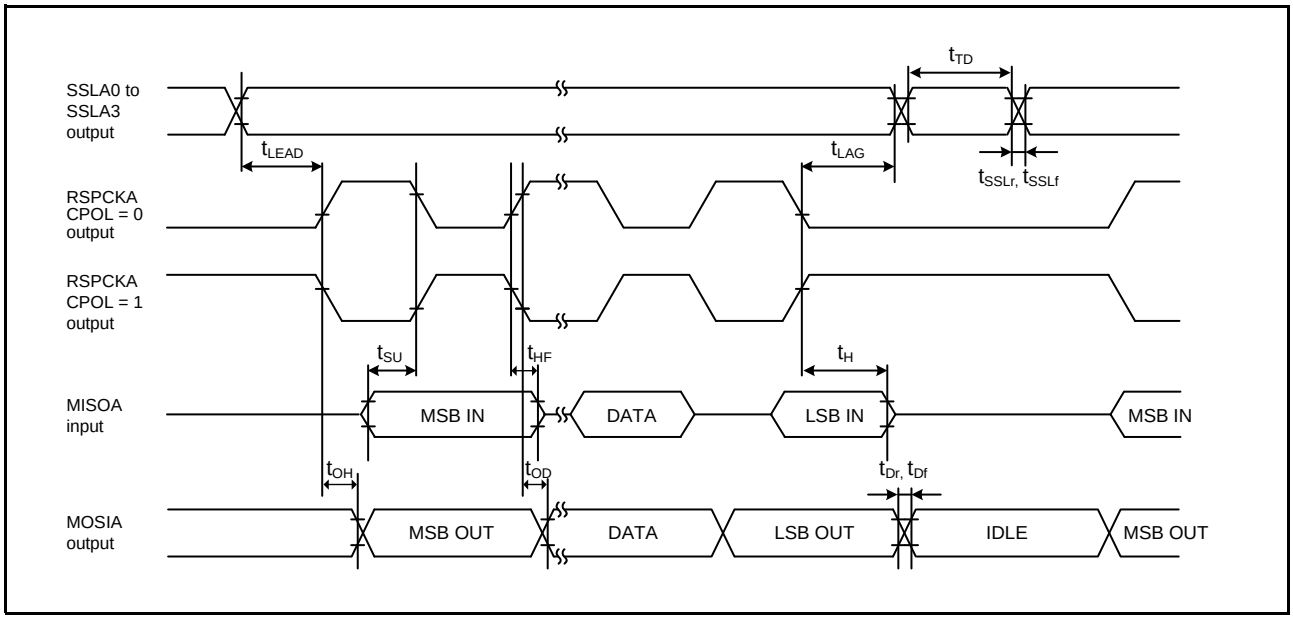


Figure 45.55 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to 1/2)

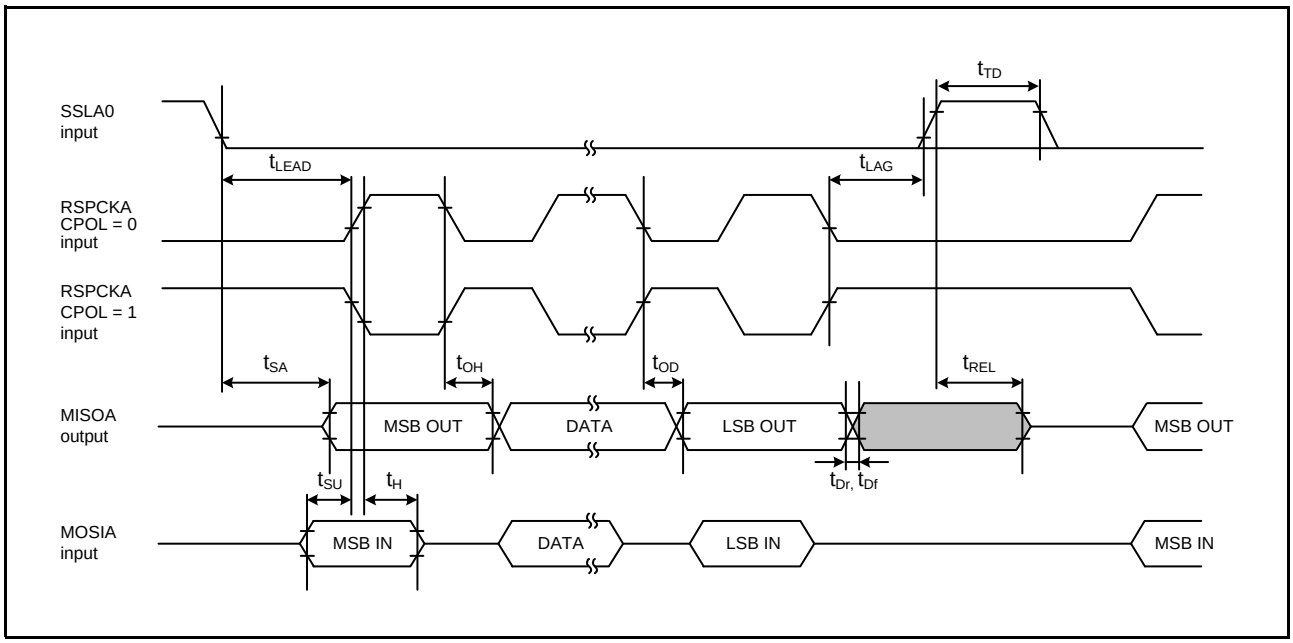


Figure 45.56 RSPI Timing (Slave, CPHA = 0)

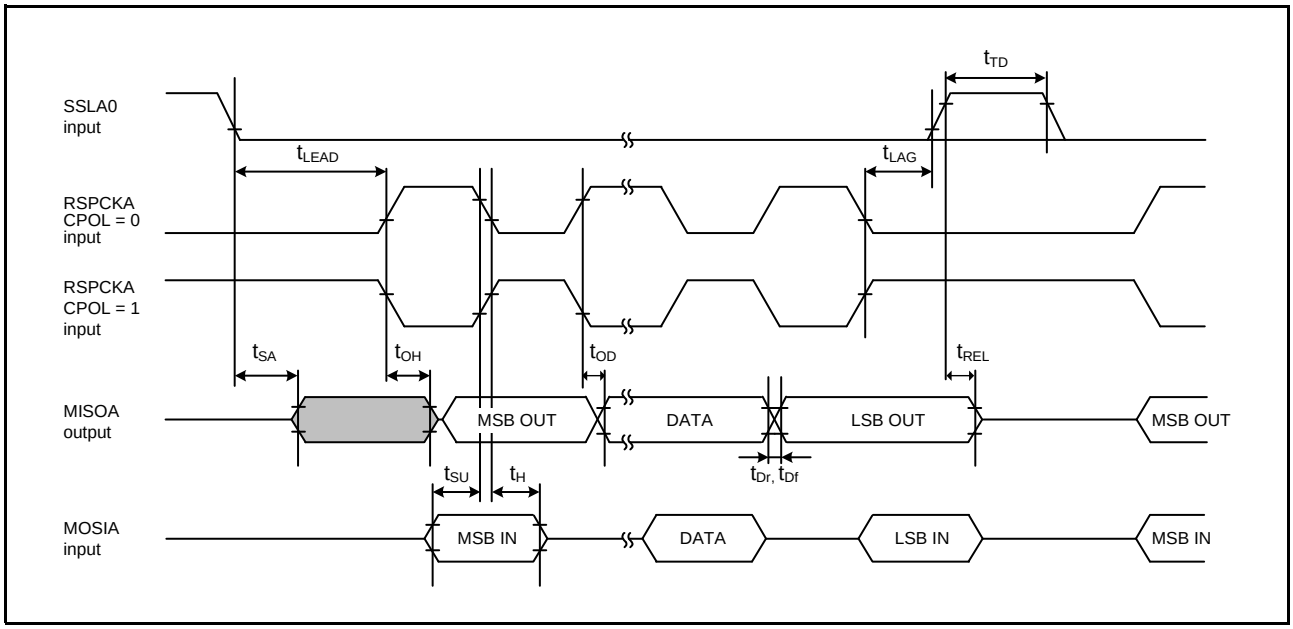


Figure 45.57 RSPI Timing (Slave, CPHA = 1)

45.4.6.11 RIIC

Table 45.41 RIIC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.*1	Max.	Unit	Test Conditions*3
RIIC (Standard-mode, SMBus)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 45.58
	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)	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.
Note 3. When VCC ≥ 4.5V, VOLSR.RICVLS = 0
When VCC < 4.5V, VOLSR.RICVLS = 1

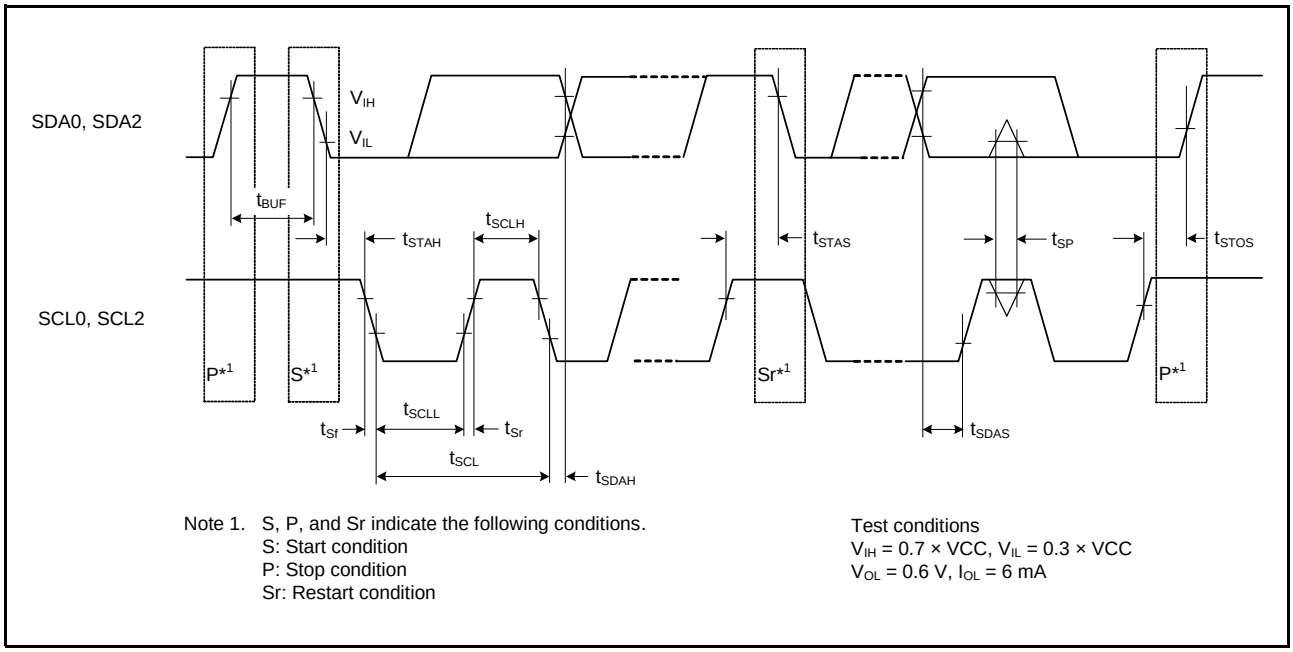


Figure 45.58 RIIC Bus Interface Input/Output Timing

45.4.6.12 CANFD

Table 45.42 CANFD Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Max.	Unit
Classic CAN mode	Bit rate for communications		—	1	Mbps
CAN FD mode	Bit rate for communications		—	1	
	Bit rate for communications (only for data)		—	5	

45.5 A/D Conversion Characteristics

Table 45.43 12-Bit A/D Conversion Characteristics (1)

Conditions: VCC = 2.7 to 5.5 V, 3.0 V ≤ AVCC0 ≤ 5.5 V, AVCC0 – 1.0 ≤ VREFH0 ≤ AVCC0, 3.0 V ≤ VREFH0,
VSS = AVSS0 = VREFL0 = 0 V, $T_a = T_{opr}$, Source impedance = 1.0 kΩ,
The VREFH0 and VREFL0 pins are selected as the reference voltage (ADVREFCR.VREFSEL bit = 1).

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		12	12	12	Bit	
Analog input capacitance		—	—	30	pF	
Conversion time*1 (Operation at PCLKD = 60 MHz)	AN000 to AN007	0.90 (0.50)*2	—	—	µs	Sampling in 30 states
	AN008 to AN015	0.95 (0.55)*2	—	—		Sampling in 33 states
	AN017, AN019, AN021, AN023	0.95 (0.55)*2	—	—		Sampling in 33 states
	AN016, AN018, AN020, AN022	1.05 (0.65)*2	—	—		Sampling in 39 states
Offset error		—	±1.5	±5.0	LSB	
Full-scale error		—	±1.5	±4.5		
Quantization error		—	±0.5	—		
Absolute accuracy		—	±2.5	±5.5		
DNL differential nonlinearity error		—	±1.0	±1.5		
INL integral nonlinearity error		—	±1.5	±2.5		

Note: The characteristics apply when no pin functions other than A/D converter input are used. The absolute accuracy includes the quantization error. The offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error does not include the quantization error.

Note 1. The conversion time includes the sampling time and the comparison time. The numbers of sampling states is indicated as test conditions for the respective items.

Note 2. The value in parentheses indicates the sampling time.

Table 45.44 12-Bit A/D Conversion Characteristics (2) (144-Pin Products)

Conditions: VCC = 2.7 to 5.5 V, 3.0 V ≤ AVCC0 ≤ 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$, Source impedance = 1.0 kΩ,
The AVCC0 and AVSS0 pins are selected as the reference voltage (ADVREFCR.VREFSEL bit = 0).

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		12	12	12	Bit	
Analog input capacitance		—	—	30	pF	
Conversion time*1 (Operation at PCLKD = 60 MHz)	AN000 to AN007	0.90 (0.50)*2	—	—	µs	Sampling in 30 states
	AN008 to AN015	0.95 (0.55)*2	—	—		Sampling in 33 states
	AN017, AN019, AN021, AN023	0.95 (0.55)*2	—	—		Sampling in 33 states
	AN016, AN018, AN020, AN022	1.05 (0.65)*2	—	—		Sampling in 39 states
Conversion time*1 (Operation at PCLKD = 30 MHz)	AN000 to AN007	1.30 (0.50)*2	—	—	µs	Sampling in 15 states
	AN008 to AN015	1.39 (0.60)*2	—	—		Sampling in 18 states
	AN017, AN019, AN021, AN023	1.39 (0.60)*2	—	—		Sampling in 18 states
	AN016, AN018, AN020, AN022	1.49 (0.70)*2	—	—		Sampling in 21 states
Offset error	Operation at PCLKD = 60 MHz	—	±4.0	—	LSB	
	Operation at PCLKD = 30 MHz	—	±1.5	—		
Full-scale error	Operation at PCLKD = 60 MHz	—	±2.5	—		
	Operation at PCLKD = 30 MHz	—	±1.5	—		
Quantization error		—	±0.5	—		
Absolute accuracy	Operation at PCLKD = 60 MHz	—	±7.0	—		
	Operation at PCLKD = 30 MHz	—	±4.0	—		
DNL differential nonlinearity error	Operation at PCLKD = 60 MHz	—	±4.0	—		
	Operation at PCLKD = 30 MHz	—	±1.0	—		
INL integral nonlinearity error	Operation at PCLKD = 60 MHz	—	±4.0	—		
	Operation at PCLKD = 30 MHz	—	±1.5	—		

Note: The characteristics apply when no pin functions other than A/D converter input are used. The absolute accuracy includes the quantization error. The offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error does not include the quantization error.

Note 1. The conversion time includes the sampling time and the comparison time. The numbers of sampling states is indicated as test conditions for the respective items.

Note 2. The value in parentheses indicates the sampling time.

Table 45.45 12-Bit A/D Conversion Characteristics (2) (100-, 80-, and 64-Pin Products)

Conditions: VCC = 2.7 to 5.5 V, 3.0 V ≤ AVCC0 ≤ 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$, Source impedance = 1.0 kΩ,
The AVCC0 and AVSS0 pins are selected as the reference voltage (ADVREFCR.VREFSEL bit = 0).

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		12	12	12	Bit	
Analog input capacitance		—	—	30	pF	
Conversion time*1 (Operation at PCLKD = 60 MHz)	AN000 to AN007	0.90 (0.50)*2	—	—	µs	Sampling in 30 states
	AN008 to AN015	0.95 (0.55)*2	—	—		Sampling in 33 states
	AN017, AN019, AN021, AN023	0.95 (0.55)*2	—	—		Sampling in 33 states
	AN016, AN018, AN020, AN022	1.05 (0.65)*2	—	—		Sampling in 39 states
Conversion time*1 (Operation at PCLKD = 30 MHz)	AN000 to AN007	1.30 (0.50)*2	—	—	µs	Sampling in 15 states
	AN008 to AN015	1.39 (0.60)*2	—	—		Sampling in 18 states
	AN017, AN019, AN021, AN023	1.39 (0.60)*2	—	—		Sampling in 18 states
	AN016, AN018, AN020, AN022	1.49 (0.70)*2	—	—		Sampling in 21 states
Offset error	Operation at PCLKD = 60 MHz	—	±2.5	—	LSB	
	Operation at PCLKD = 30 MHz	—	±1.5	—		
Full-scale error	Operation at PCLKD = 60 MHz	—	±2.5	—		
	Operation at PCLKD = 30 MHz	—	±1.5	—		
Quantization error		—	±0.5	—		
Absolute accuracy	Operation at PCLKD = 60 MHz	—	±4.5	—		
	Operation at PCLKD = 30 MHz	—	±2.5	—		
DNL differential nonlinearity error	Operation at PCLKD = 60 MHz	—	±1.5	—		
	Operation at PCLKD = 30 MHz	—	±1.0	—		
INL integral nonlinearity error	Operation at PCLKD = 60 MHz	—	±2.5	—		
	Operation at PCLKD = 30 MHz	—	±1.5	—		

Note: The characteristics apply when no pin functions other than A/D converter input are used. The absolute accuracy includes the quantization error. The offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error does not include the quantization error.

Note 1. The conversion time includes the sampling time and the comparison time. The numbers of sampling states is indicated as test conditions for the respective items.

Note 2. The value in parentheses indicates the sampling time.

Table 45.46 12-Bit A/D Conversion Characteristics (2) (48-Pin Products)

Conditions: VCC = 2.7 to 5.5 V, 3.0 V ≤ AVCC0 ≤ 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$, Source impedance = 1.0 kΩ,
The AVCC0 and AVSS0 pins are selected as the reference voltage (ADVREFCR.VREFSEL bit = 0).

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		12	12	12	Bit	
Analog input capacitance		—	—	30	pF	
Conversion time*1 (Operation at PCLKD = 60 MHz)	AN000 to AN002, AN005 to AN007	0.90 (0.50)*2	—	—	µs	Sampling in 30 states
	AN009 to AN012	0.95 (0.55)*2	—	—		Sampling in 33 states
Offset error		—	±1.5	—	LSB	
Full-scale error		—	±1.5	—		
Quantization error		—	±0.5	—		
Absolute accuracy		—	±2.5	—		
DNL differential nonlinearity error		—	±1.0	—		
INL integral nonlinearity error		—	±1.5	—		

Note: The characteristics apply when no pin functions other than A/D converter input are used. The absolute accuracy includes the quantization error. The offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error does not include the quantization error.

Note 1. The conversion time includes the sampling time and the comparison time. The numbers of sampling states is indicated as test conditions for the respective items.

Note 2. The value in parentheses indicates the sampling time.

Table 45.47 A/D Internal Reference Voltage Characteristics

Conditions: VCC = 2.7 to 5.5 V, 3.0 V ≤ AVCC0 ≤ 5.5 V, AVCC0 – 1.0 ≤ VREFH0 ≤ AVCC0, 3.0 V ≤ VREFH0,
VSS = AVSS0 = VREFL0 = 0 V, $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D internal reference voltage	1.20	1.25	1.30	V	

Note: The above specification values apply during normal operations.

45.6 D/A Conversion Characteristics

Table 45.48 D/A Conversion Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	12	12	12	Bit	
Absolute accuracy	—	—	±6.0	LSB	2-MΩ resistive load 10-bit conversion
Differential nonlinearity error (DNL)	—	±1.0	±2.0	LSB	2-MΩ resistive load
Output resistance (R_O)	—	5.7	—	kΩ	
Conversion time	—	—	3	µs	20-pF capacitive load

45.7 Temperature Sensor Characteristics

Table 45.49 Temperature Sensor Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	±1	—	°C	
Temperature slope	—	–2.0	—	mV/°C	
Output voltage	—	0.63	—	V	T_a = 25°C
Temperature sensor start time	—	—	200	µs	
Sampling time*1	3	—	—	µs	

Note 1. Set the S12AD.ADSSTRT register such that the sampling time of the 12-bit A/D converter satisfies this specification.

45.8 Comparator Characteristics

Table 45.50 Comparator Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input offset voltage	V_{IO}	—	8	40	mV	
Reference input voltage range	V_{ref}	0	—	VCC	V	
Analog input voltage rang	V_{ain}	0	—	VCC	V	
Response time	$t_{tot(r)}$	—	—	200	ns	VOD = 100 mV CMPCTL.CDFS[1:0] = 00b Figure 45.59
	$t_{tot(f)}$	—	—	200		
Waiting time for stabilization following switching of the input	t_{cwait}	300	—	—	ns	
Operation stabilization time	t_{cmp}	—	—	1	µs	

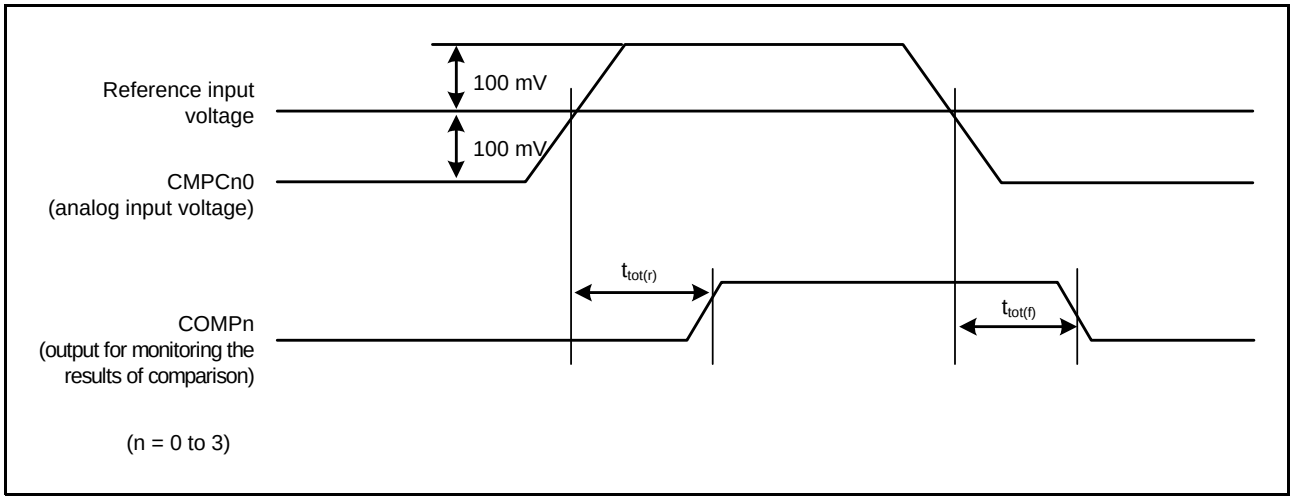


Figure 45.59 Comparator Response Time

45.9 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 45.51 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	2.46	2.58	2.70	V	Figure 45.60
	Voltage detection circuit (LVD0)	V_{det0_1}	4.04	4.22	4.40		Figure 45.61
		V_{det0_2}	2.71	2.83	2.95		
	Voltage detection circuit (LVD1)	V_{det1_0}	4.39	4.57	4.75		Figure 45.62
		V_{det1_1}	4.29	4.47	4.65		
		V_{det1_2}	4.14	4.32	4.50		
		V_{det1_3}	2.81	2.93	3.05		
		V_{det1_4}	2.76	2.88	3.00		
	Voltage detection circuit (LVD2)	V_{det2_0}	4.39	4.57	4.75		Figure 45.63
		V_{det2_1}	4.29	4.47	4.65		
		V_{det2_2}	4.14	4.32	4.50		
		V_{det2_3}	2.81	2.93	3.05		
		V_{det2_4}	2.76	2.88	3.00		
Internal reset time	Power-on reset time	t_{POR}	—	15.5	—	ms	Figure 45.60
	LVD0 reset time	t_{LVD0}	—	0.70	—		Figure 45.61
	LVD1 reset time	t_{LVD1}	—	0.57	—		Figure 45.62
	LVD2 reset time	t_{LVD2}	—	0.57	—		Figure 45.63
Minimum VCC down time		t_{VOFF}	200	—	—	µs	Figure 45.60, Figure 45.61
Response delay time		t_{det}	—	—	200	µs	Figure 45.60 to Figure 45.63
LVD operation stabilization time (after LVD is enabled)		$t_{d(E-A)}$	—	—	20	µs	Figure 45.62, Figure 45.63
Hysteresis width (LVD1 and LVD2)		V_{LVH}	—	80	—	mV	

Note: The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for the POR/ LVD.

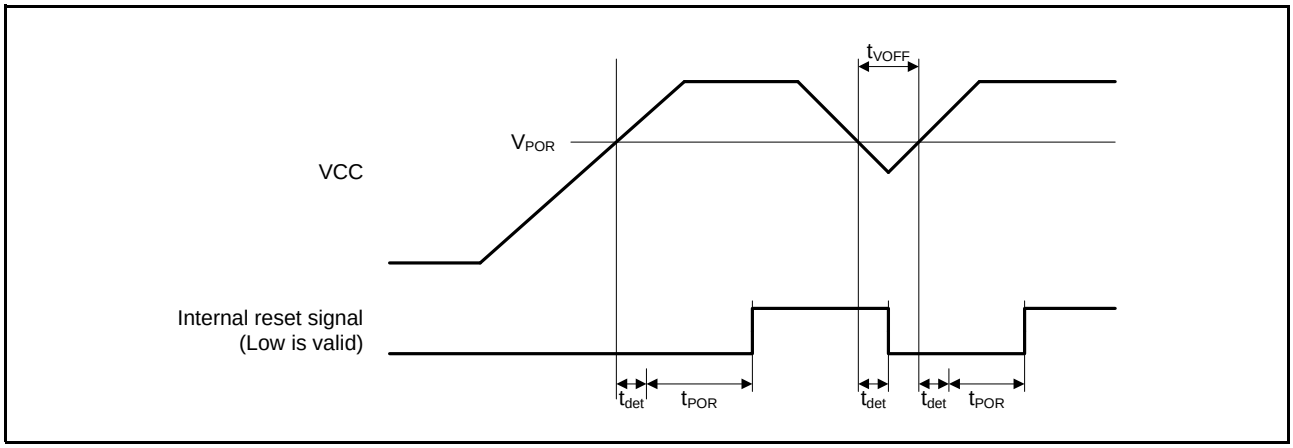


Figure 45.60 Power-on Reset Timing

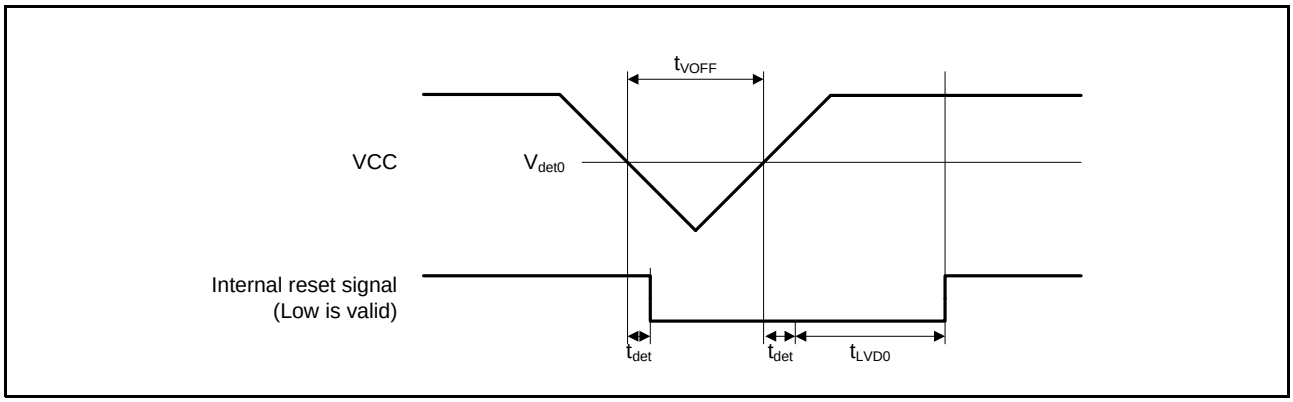


Figure 45.61 Voltage Detection Circuit Timing (V_{det0})

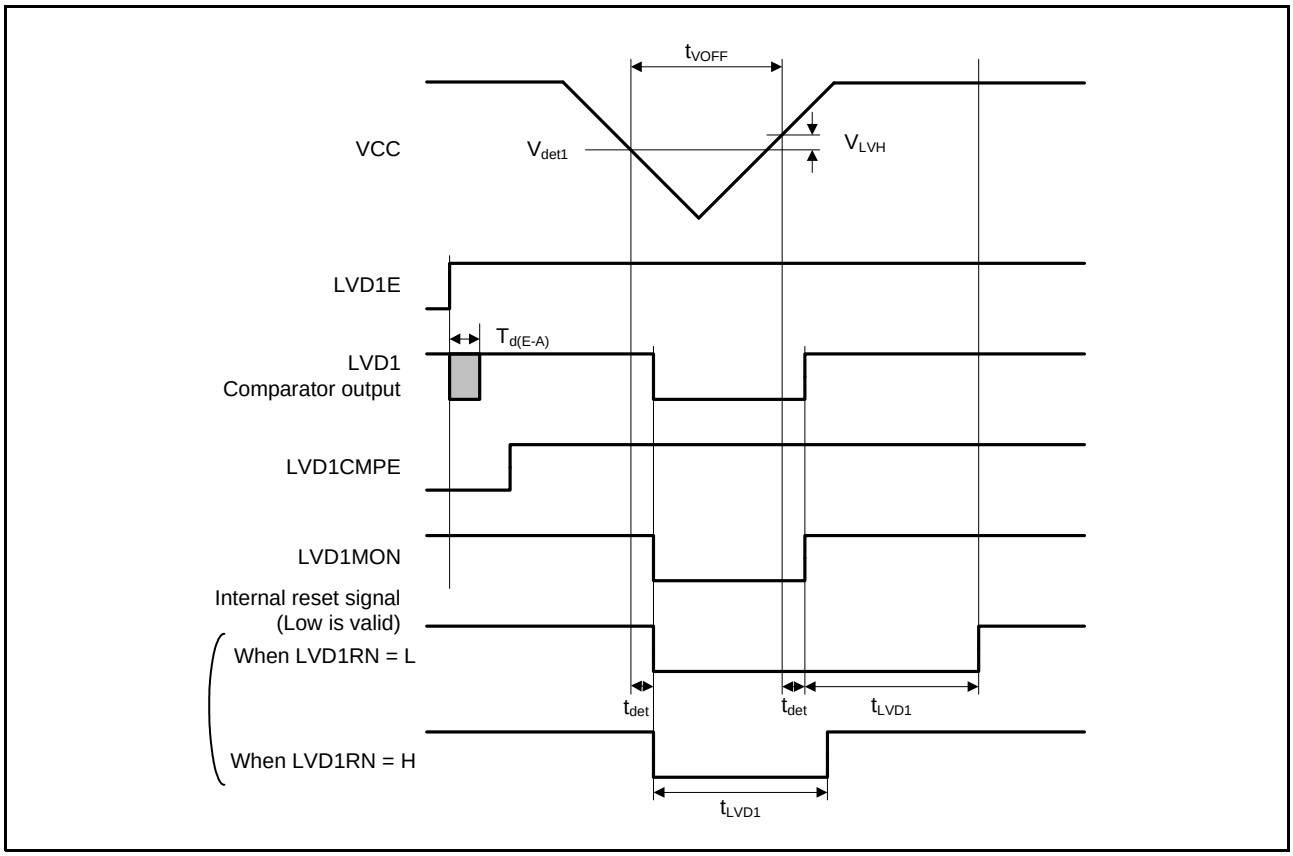


Figure 45.62 Voltage Detection Circuit Timing (V_{det1})

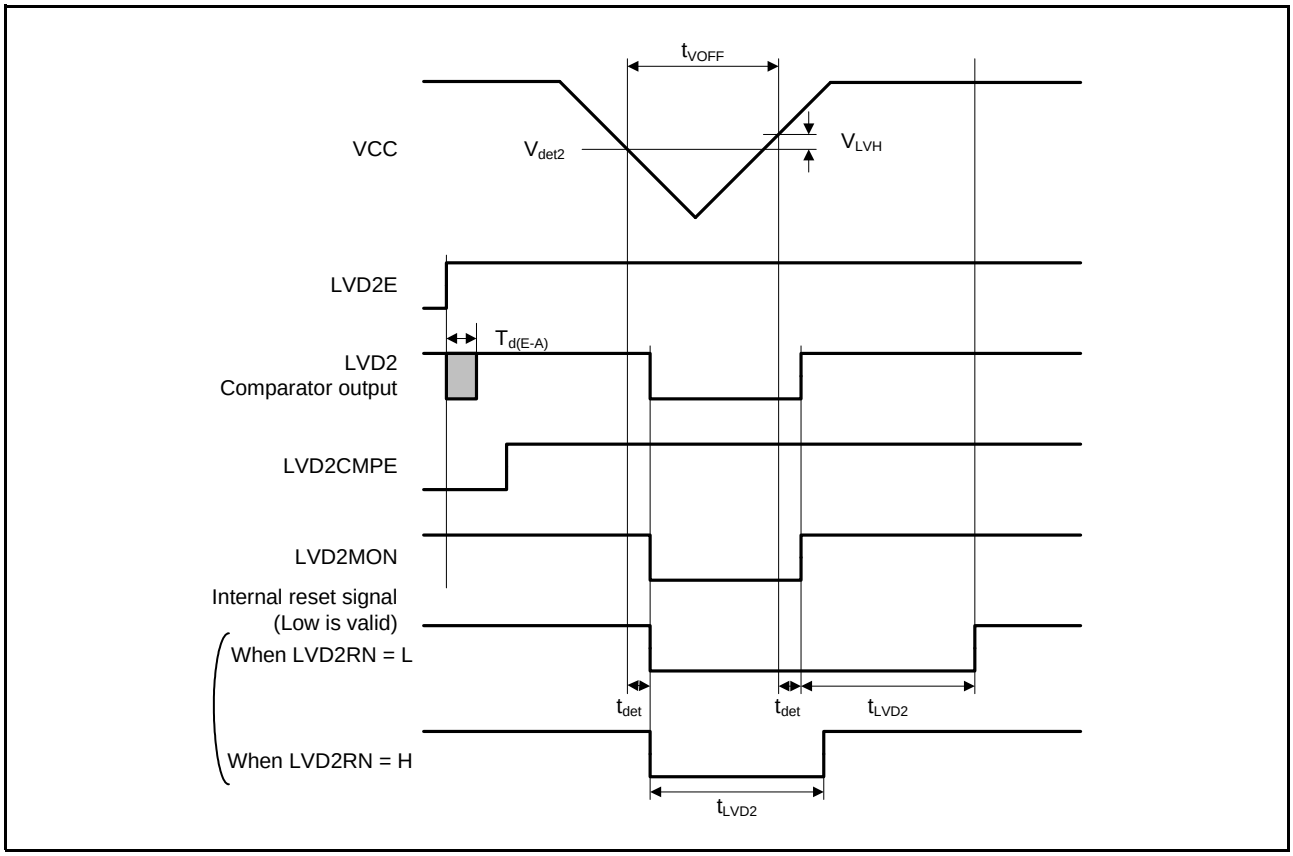


Figure 45.63 Voltage Detection Circuit Timing (V_{det2})

45.10 Oscillation Stop Detection Timing

Table 45.52 Oscillation Stop Detection Circuit Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 45.64

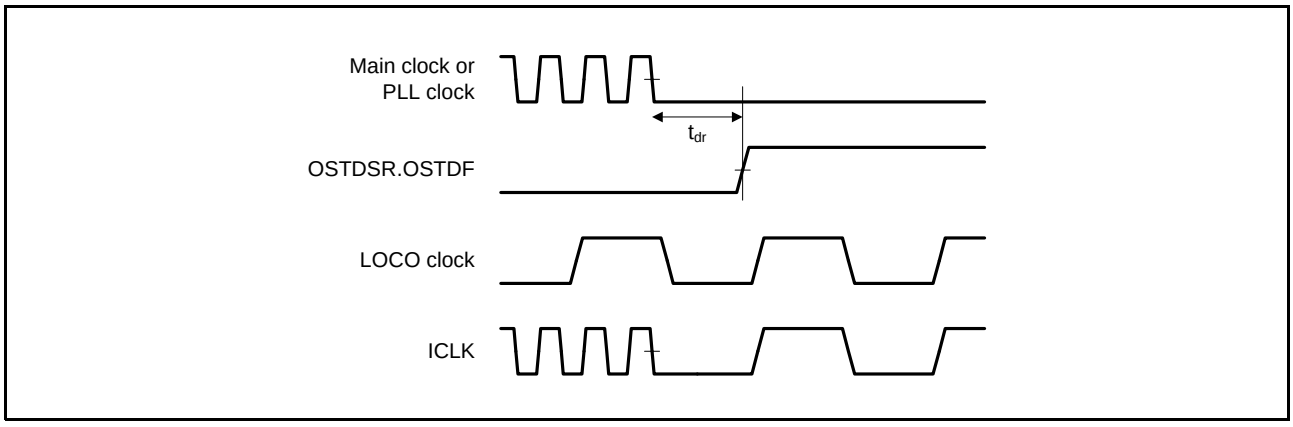


Figure 45.64 Oscillation Stop Detection Timing

45.11 Flash Memory Characteristics

Table 45.53 Code Flash Memory Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
Temperature range for programming/erasure: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Programming time N_{PEC} ≤ 100 times	256 bytes	t_{P256}	—	0.9	13.2	—	0.4	6	ms	
	8 Kbytes	t_{P8K}	—	29	176	—	13	80	ms	
	32 Kbytes	t_{P32K}	—	116	704	—	52	320	ms	
Programming time N_{PEC} > 100 times	256 bytes	t_{P256}	—	1.1	15.8	—	0.5	7.2	ms	
	8 Kbytes	t_{P8K}	—	35	212	—	16	96	ms	
	32 Kbytes	t_{P32K}	—	140	848	—	64	384	ms	
Erasure time N_{PEC} ≤ 100 times	8 Kbytes	t_{E8K}	—	71	216	—	39	120	ms	
	32 Kbytes	t_{E32K}	—	254	864	—	141	480	ms	
Erasure time N_{PEC} > 100 times	8 Kbytes	t_{E8K}	—	85	260	—	47	144	ms	
	32 Kbytes	t_{E32K}	—	304	1040	—	169	576	ms	
Reprogramming/erasure cycle*1		N_{PEC}	1000*2	—	—	1000*2	—	—	Times	
Suspend delay time during programming		t_{SPD}	—	—	264	—	—	120	µs	
First suspend delay time during erasing (in suspend priority mode)		t_{SESD1}	—	—	216	—	—	120	µs	
Second suspend delay time during erasure (in suspend priority mode)		t_{SESD2}	—	—	1.7	—	—	1.7	ms	
Suspend delay time during erasure (in erasure priority mode)		t_{SEED}	—	—	1.7	—	—	1.7	ms	
Forced stop command		t_{FD}	—	—	32	—	—	20	µs	
Data hold time*3, *4		t_{DRP}	20	—	—	20	—	—	Year	T_a ≤ 85°C
			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 256-byte program is performed 32 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 45.54 Data Flash Memory Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 3.0 to 5.5 V,
VSS = AVSS0 = 0 V,
Temperature range for programming/erasure: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Programming time	4 bytes	t_{DP4}	—	0.36	3.8	—	0.16	1.7	ms	
Erasure time	64 bytes	t_{DP64}	—	3.1	18	—	1.7	10	ms	
Blank check time	4 bytes	t_{DBC4}	—	—	84	—	—	30	µs	
	64 bytes	t_{DBC64}	—	—	280	—	—	100	µs	
	2 Kbytes	t_{DBC2K}	—	—	6160	—	—	2200	µs	
Reprogramming/erasure cycle*1		N_{DPEC}	100000 *2	—	—	100000 *2	—	—	Times	
Suspend delay time during programming		t_{DSPD}	—	—	264	—	—	120	µs	
First suspend delay time during erasure (in suspend priority mode)		t_{DSESD1}	—	—	216	—	—	120	µs	
Second suspend delay time during erasure (in suspend priority mode)		t_{DSESD2}	—	—	300	—	—	300	µs	
Suspend delay time during erasure (in erasure priority mode)		t_{DSEED}	—	—	300	—	—	300	µs	
Forced stop command		t_{FD}	—	—	32	—	—	20	µs	
Data hold time*3, *4		t_{DDRP}	20	—	—	20	—	—	Year	$T_a \leq 85°C$
			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.

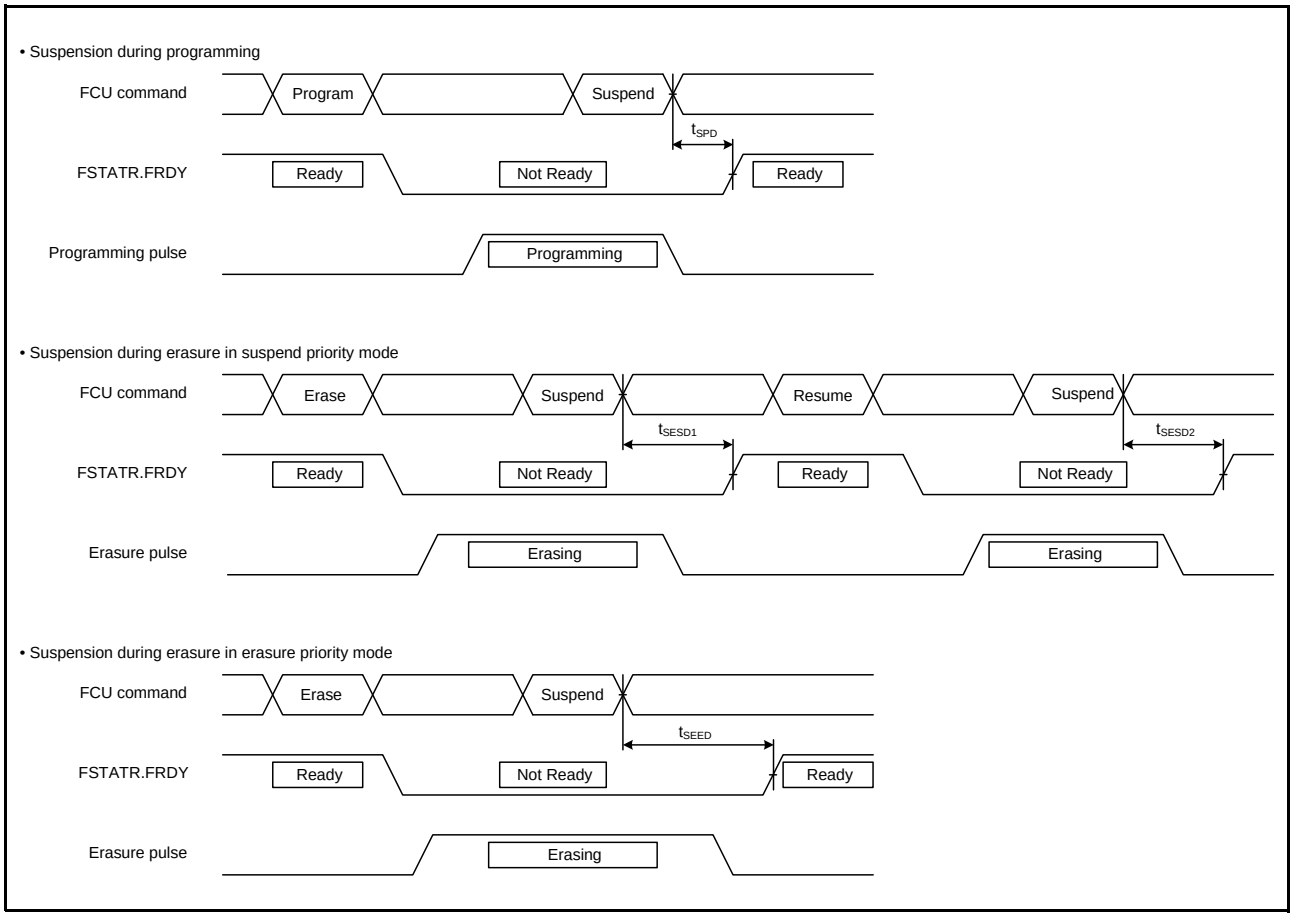


Figure 45.65 Flash Memory Programming/Erasure Suspension Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1 / 4)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
P00/IRQ8, P01/IRQ9, P02/IRQ10, P03/IRQ11, P04, P05/IRQ13, P06, P07/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P12/IRQ2, P13/IRQ3, P14/IRQ4, P15/IRQ5, P16/IRQ6, P17/IRQ7	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P20/IRQ8, P21/IRQ9, P22/IRQ15, P23/IRQ3	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P24/IRQ12/CS0#, P25/IRQ5/CS1#, P26/IRQ6/CS2#, P27/IRQ7/CS3#	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, P31/IRQ1-DS/ RTCIC1, P32/IRQ2-DS/ RTCIC2, P33/IRQ3-DS, P34/IRQ4, P35/NMI, P36/IRQ5, P37/IRQ4	All	Hi-Z	Keep-O*2		Keep-O*3	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	All	Hi-Z	[Clock output] H [Other than the above] Keep-O*2		Keep	Keep	Hi-Z
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			

Table 1.1 Port States in Each Processing State (2 / 4)

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
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#	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 [Other than the above] Keep-O	[CS1# output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
P62/IRQ2/CS2#	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 [Other than the above] Keep-O	[CS2# output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
P63/IRQ3/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 [Data output] Hi-Z [Other than the above] Keep-O	[CS3# output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
P64/IRQ4	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				
P65/IRQ13, P66/IRQ14, P67/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P70/IRQ0	All	Hi-Z	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			
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	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			
P75/IRQ13, P76/IRQ14, P77/IRQ7	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P80/IRQ8, P81/IRQ9, P82/IRQ2, P83/IRQ3, P86/IRQ14, P87/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (3 / 4)

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
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 [Other than the above] Keep-O	[Address output] Hi-Z [Other than the above] Keep-O			
PA0/IRQ0/BC0#	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 [BC0# output] H [Other than the above] Keep-O	[Address output] Hi-Z [BC0# output] Hi-Z [Other than the above] Keep-O			
PA1/IRQ11, 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, 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	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)			[Address output] Address output retained [Other than the above] Keep-O*2	[Address output] Hi-Z [Other than the above] Keep-O*2			
PC4/IRQ12/CS3#	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 [CS3# output] H [Other than the above] Keep-O*2	[Address output] Hi-Z [CS3# output] Hi-Z [Other than the above] Keep-O*2			
PC5/IRQ5/CS2#, PC6/IRQ13/CS1#	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 [Data output] Hi-Z [Other than the above] Keep-O*2	[CSn# output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O*2			
PC7/IRQ14/CS0#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)			[CS0# output] H [Other than the above] Keep-O*2	[CS0# 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				
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, PF6, PF7	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (4 / 4)

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
PH0, PH1/IRQ0, PH2/IRQ1, PH3, PH6, PH7	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PJ1, PJ3/IRQ11, PJ4, PJ5/IRQ13, PJ6, PJ7	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PK2, PK3, PK4, PK5	All	Hi-Z	Keep		Keep	Keep	Hi-Z
PL0, PL1	All	Hi-Z	Keep		Keep	Keep	Hi-Z
PN6, PN7	All	Hi-Z	Keep		Keep	Keep	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

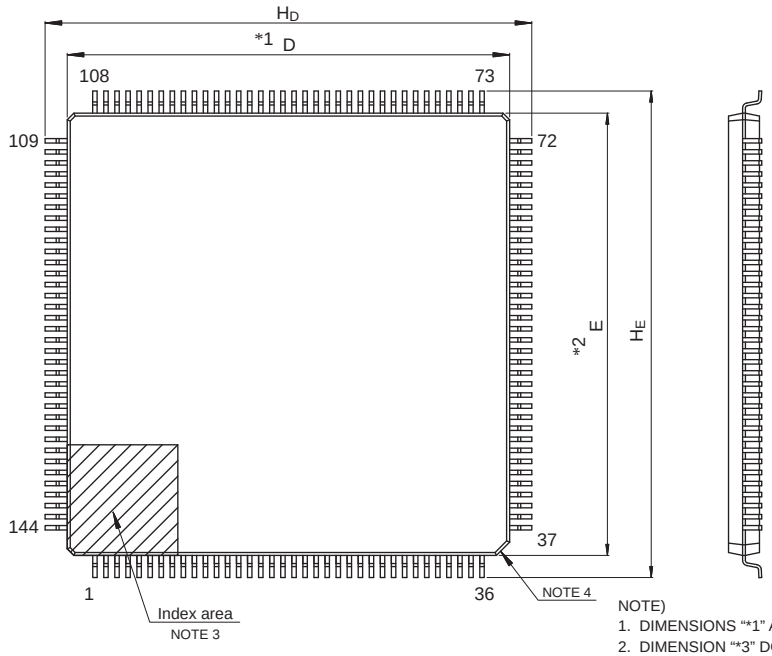
Note 1. The I/O port state is retained until the DPSBYCR.IOKEEP bit is cleared to 0.
Note 2. Input is enabled if the pin is specified as the software standby cancelling source while it is used as an external interrupt pin. (IRQ* pin)
Note 3. Input is enabled if the pin is specified as the deep software standby cancelling source. (IRQ*-DS pin)

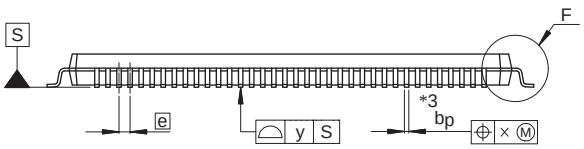
Appendix 2. Package Dimensions

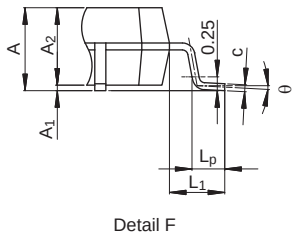
Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP144-20x20-0.50	PLQP0144KA-B	—	1.2

Unit: mm







Detail F

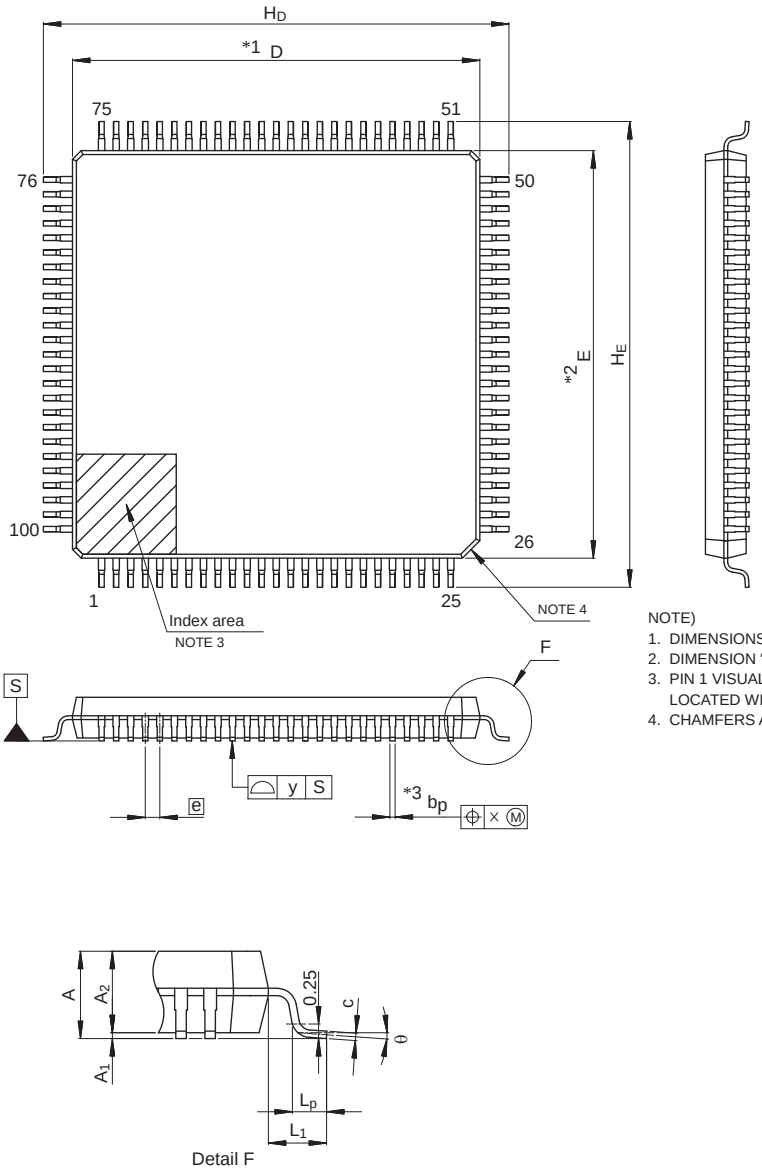
NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

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 A 144-Pin LFQFP (PLQP0144KA-B)

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm



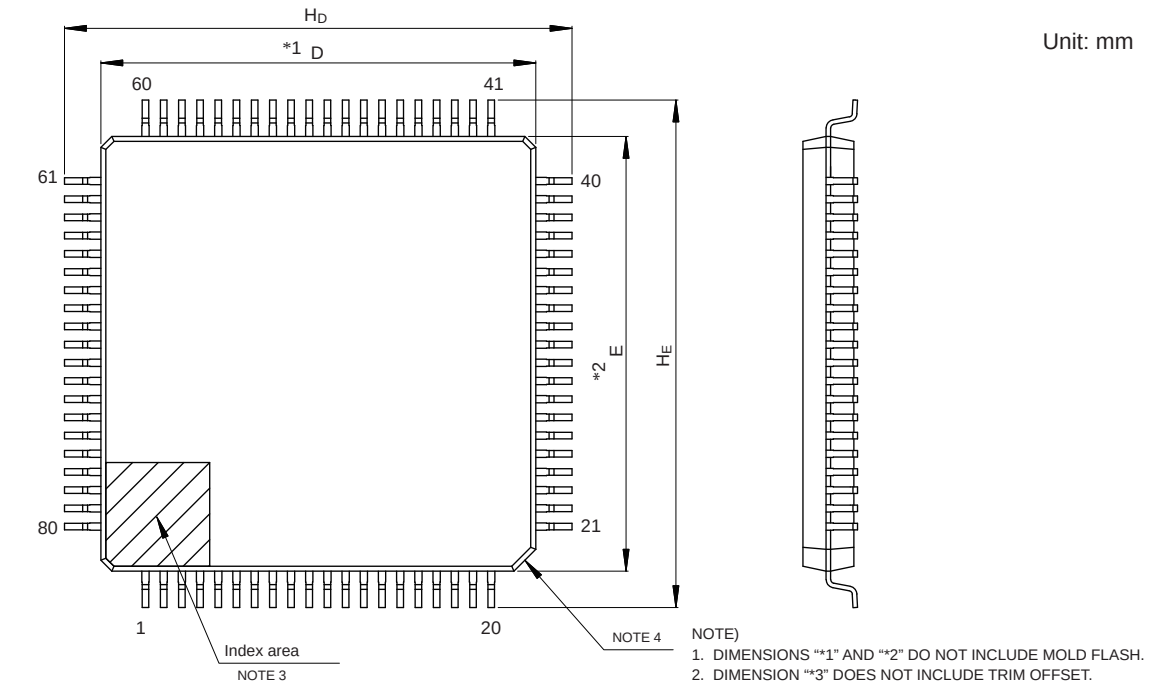
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

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	—

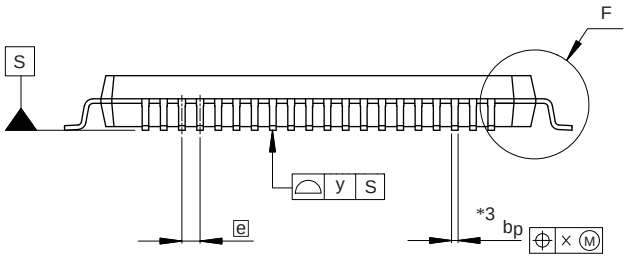
Figure B 100-Pin LFQFP (PLQP0100KB-B)

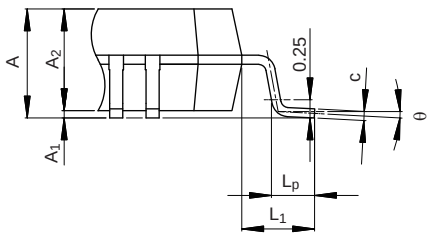
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5

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.





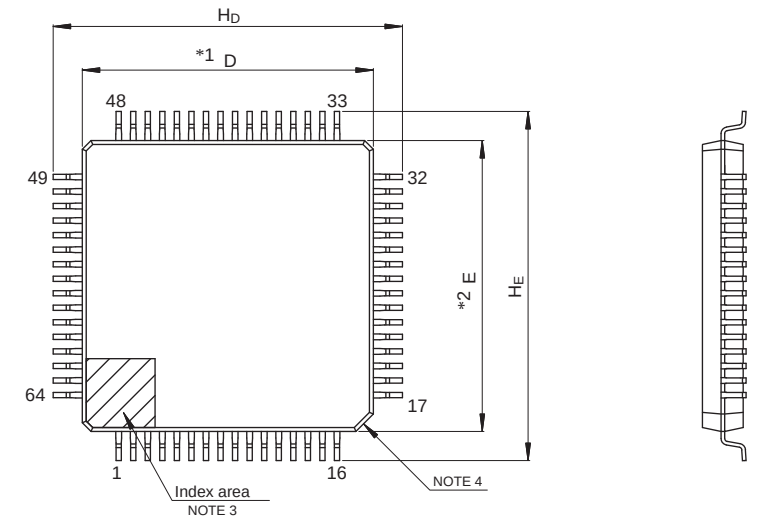
Detail F

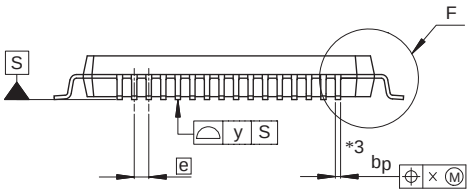
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

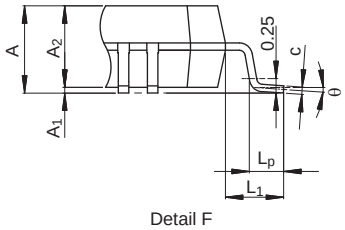
Figure C 80-Pin LFQFP (PLQP0080KB-B)

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm







Detail F

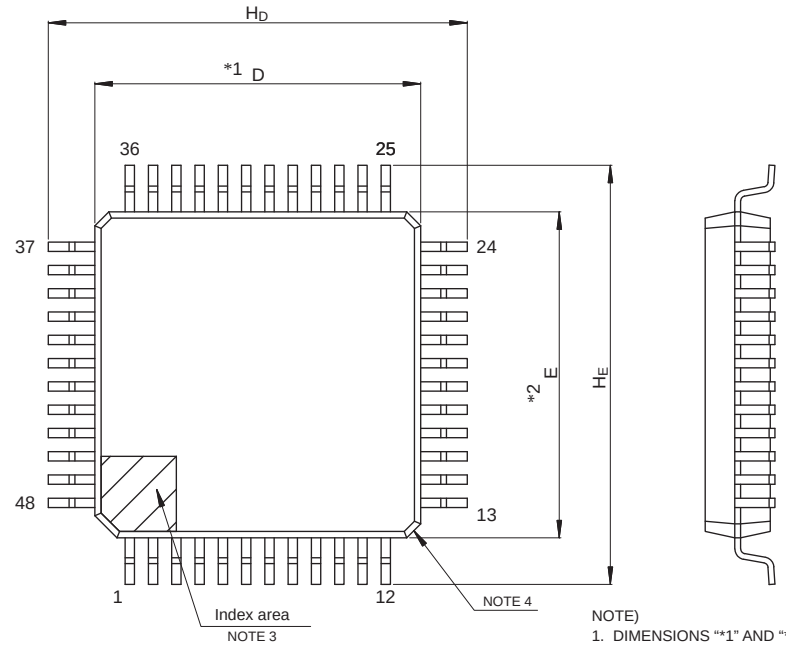
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
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	—

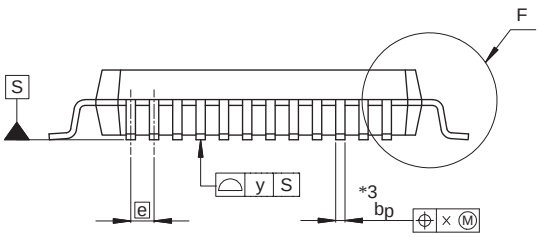
Figure D 64-Pin LFQFP (PLQP0064KB-C)

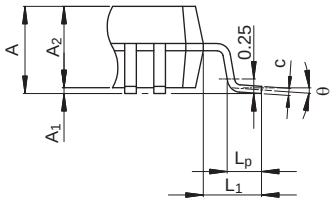
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2

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.





Detail F

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
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	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure E 48-Pin LFQFP (PLQP0048KB-B)

REVISION HISTORY	RX660 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 18, 2022	—	First edition, issued	

RX660 Group User’s Manual: Hardware

Publication Date: Rev.1.00 Mar 18, 2022

Published by: Renesas Electronics Corporation

RX660 Group

Renesas Electronics Corporation

R01UH0937EJ0100