User's Manual

32

RX260 Group, RX261 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX200 Series

Renesas Electronics
www.renesas.com

Rev.1.00 Jul 2024

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.
SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

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. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.
The manual comprises an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes can be found within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RX260 Group, RX261 Group. Make sure to refer to the latest versions of these documents. The latest versions of the listed documents are available from the Renesas Electronics website.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	RX260 Group, RX261 Group Datasheet	R01DS0430EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	RX260 Group, RX261 Group User's Manual: Hardware	This User's manual
User's Manual: Software	Description of CPU 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 register initial setting	RX260 Group, RX261 Group Initial Setting Examples	—
	Information on using peripheral functions and application examples Sample programs	Available from Renesas Electronics website.	
Renesas Technical Update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

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 (3)	R/W (1)
b3 to b1	—	Reserved (2)	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. (3)	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
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver / Transmitter
VCO	Voltage Controlled Oscillator

Contents

RX260 Group, RX261 Group

Renesas MCUs

R01UH1045EJ0100
Rev.1.00
Jul 31, 2024

64-MHz, 32-bit RX MCUs, on-chip FPU, 355 Coremark, up to 512-KB flash memory, up to 36 pins capacitive touch sensing unit, up to 11 comms channels, 12-bit A/D, D/A, RTC, IEC60730 compliance, 1.6-V to 5.5-V single supply, Encryption functions (optional)

Features

■ 32-bit RXv3 CPU core

- Maximum operating frequency: 64 MHz
  Capable of 355 Coremark in operation at 64 MHz
- Enhanced DSP instructions: 32-bit multiply-accumulate instructions, and 16-bit multiply-subtract instructions are supported.
- On-chip FPU: 32-bit single-precision floating point compliant with IEEE-754
- On-chip divider that operated at the fastest of two clock cycles
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions, ultra-compact code
- On-chip debugging circuit
- Memory protection unit (MPU) supported

■ Low power design and architecture

- Operation from a single 1.6-V to 5.5-V supply
- Four low power consumption modes
- Low power timer (LPT) that operates during the software standby state
- Supply current
  High-speed operating mode: 84 µA/MHz
  Supply current in software standby mode: 1.01 µA (typ.) (T_a = 25°C)
- Recovery time from software standby mode: 6.3 µs (typ.) (Clock Source: HOCO 64 MHz, T_a = 25°C)

■ On-chip flash memory for code

- 256 K/384 K/512 Kbytes size capacities
- User code is programmable by on-board programming.
- Programmable at 1.6 V
- For instructions and operands

■ On-chip data flash memory

- 8 Kbytes (1,000,000 program/erase cycles (typ.))
- BGO (Background Operation)

■ On-chip SRAM, no wait states

- 128 Kbytes size capacities

■ Data transfer functions

- DMAC: Incorporates four channels
- DTC: Five transfer modes

■ ELC

- Module operation can be initiated by event signals without using interrupts.
- Linked operation between modules is possible while the CPU is sleeping.

■ Reset and supply management

- Eight types of reset, including the power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

■ Clock functions

- External main clock input frequency: Up to 20 MHz
- External sub clock input frequency: 32.768 kHz
- Main clock oscillator frequency: 1 to 20 MHz
- Sub clock oscillator frequency: 32.768 kHz
- PLL/PLL2 circuit input: 4 MHz to 12.5 MHz
- Low-speed on-chip oscillator: 4 MHz
- High-speed on-chip oscillator: 24/32/48/64 MHz ± 1%
- IWDT-dedicated on-chip oscillator: 15 kHz
- Generate a 32.768 kHz clock for the real-time clock
- On-chip clock frequency accuracy measurement circuit (CAC)

■ Realtime clock

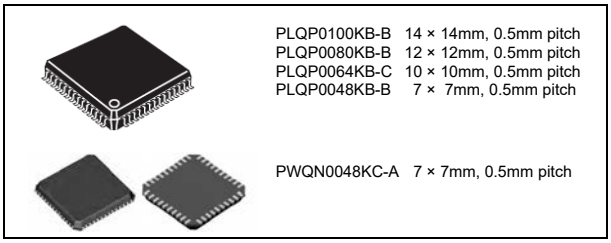
- Adjustment functions (30 seconds, leap year, and error)
- Calendar count mode or binary count mode selectable
- Time capture function
- Time capture on event-signal input through external pins

■ Independent watchdog timer

- 15-kHz on-chip oscillator produces a dedicated clock signal to drive IWDT operation.

■ Useful functions for IEC60730 compliance

- Self-diagnostic and disconnection-detection assistance functions for the A/D converter, clock frequency accuracy measurement circuit, independent watchdog timer, RAM test assistance functions using the DOC, etc.



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

PWQN0048KC-A 7 × 7mm, 0.5mm pitch

■ MPC

- Input/output functions selectable from multiple pins

■ Up to 11 communication functions

- USB 2.0 full-speed host/function/On-The-Go (OTG) (1 channel), full-speed = 12 Mbps, low-speed = 1.5 Mbps (host only), and isochronous transfer
- CAN FD: Compliant with ISO11898-1:2015, standard frame and extended frame (1 channel)
- SCI with multiple functionalities (up to 4 channels)
  Choose from among asynchronous mode, clock-synchronous mode, smart-card interface mode, simplified SPI, simplified I^2C, and extended serial mode.
- Up to three RSCIs with Manchester encoding and HBS functionality
- I^2C bus interface: Transfer at up to 400 kbps, capable of SMBus operation (one channel)
- RSPI (one channel): Transfer at up to 16 Mbps

■ Remote control signal reception

■ Up to 16 extended-function timers

- 32-bit (2 channels) or 16-bit (6 channels) GPTW: operation at 64 MHz, input capture, output compare, PWM waveforms: 10 output channels in single-phase complementary PWM mode/3 output channels in 3-phase complementary PWM mode/2 output channels in 5-phase complementary PWM mode, phase-counting mode, linkage with comparator (counting operation, PWM negate control)
- 8-bit TMR (four channels)
- 16-bit compare-match timers (four channels)

■ 12-bit A/D converter

- Capable of conversion within 0.5 µs
- 24 (external pin input) +1 (internal input) channels
- Sampling time can be set for each channel
- Conversion results compare features
- Self-diagnostic function and analog input disconnection detection assistance function
- Double trigger (data duplication) function for motor control

■ D/A converter

- Two channels

■ Capacitive touch sensing unit

- Self-capacitance method: A single pin configures a single key, supporting up to 36 keys
- Mutual capacitance method: Matrix configuration with 8 × 8, supporting up to 64 keys

■ Comparator B

- Two channels

■ General I/O ports

- 5-V tolerant, open drain, input pull-up

■ Renesas Secure IP (RSIP-E11A) (optional)

- AES128/256, ECC, True-random number generator (TRNG), SHA224, and SHA256

■ Temperature sensor

■ Unique ID

- 32-byte ID code for the MCU

■ Operating temperature range

- –40 to +85°C
- –40 to +105°C

■ Applications

- General industrial and consumer equipment

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 gives a comparison of the functions of the products in different packages.
Table 1.1 shows the outline of maximum specifications. The peripheral functions and the number of their channels vary depending on the number of pins of the package. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/6)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 64 MHz • 32-bit RX CPU (RX v3) • Minimum instruction execution time: One instruction per clock cycle • Address space: 4-Gbyte linear • Register set General purpose: Sixteen 32-bit registers Control: Ten 32-bit registers Accumulator: Two 72-bit registers • 111 instructions Standard provided instructions: 111 Basic instructions: 77 Single precision floating point instructions: 11 DSP instructions: 23 • Addressing modes: 11 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32-bit × 32-bit → 64-bit • On-chip divider: 32-bit ÷ 32-bit → 32 bits • Barrel shifter: 32 bits • Memory protection unit (MPU)
	FPU	• Single precision (32-bit) floating-point number • Data types and exceptions in conformance with the IEEE754 standard
Memory	ROM	• Capacity: 512 Kbytes, 384 Kbytes, 256 Kbytes • 32 MHz ≤: No-wait cycle access • 32 MHz to 64 MHz: One-wait cycle access • Programming/erasing method: Serial programming (asynchronous serial communication/USB communication), self-programming
	RAM	• Capacity: 128 Kbytes • 64 MHz, no-wait memory access • Parity error detection
	E2 DataFlash	• Capacity: 8 Kbytes • Number of erase/write cycles: 1,000,000 (typ.)
MCU operating mode		Single-chip mode
Clock	Clock generation circuit	• Main clock oscillator, sub-clock oscillator, low-speed on-chip oscillator, high-speed on-chip oscillator, PLL frequency synthesizer, PLL2 frequency synthesizer, and IWDT-dedicated on-chip oscillator • Oscillation stop detection: Available • Clock frequency accuracy measurement circuit (CAC) • Independent settings for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 64 MHz (at max.) Peripheral modules of GPTW, and the ECC function control registers in the CANFD module run in synchronization with PCLKA: 64 MHz (at max.) ADCLK in the S12AD runs in synchronization with PCLKD: 64 MHz (at max.) Other peripheral modules run in synchronization with PCLKB: 32 MHz (at max.) The flash peripheral circuit runs in synchronization with the FCLK: 64 MHz (at max.)

Table 1.1 Outline of Specifications (2/6)

Classification	Module/Function	Description
Resets		RES# pin reset, power-on reset, voltage monitoring reset, watchdog timer reset, independent watchdog timer reset, and software reset
Voltage detection circuit (LVDAb)		When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated. • Voltage detection circuit 0 is capable of selecting the detection voltage from 5 levels • Voltage detection circuit 1 is capable of selecting the detection voltage from 16 levels • Voltage detection circuit 2 is capable of selecting the detection voltage from 4 levels
Low power consumption	Low power consumption functions	• Module stop function • Four low power consumption modes Sleep mode, deep sleep mode, software standby mode, and snooze mode
	Function for lower operating power consumption	Operating power control modes • High-speed operating mode • middle-speed operating mode (default) • middle-speed operating mode 2 • low-speed operating mode
Interrupt	Interrupt controller (ICUb)	• Interrupt vectors: 256 • External interrupts: 9 (NMI, IRQ0 to IRQ7 pins) • Non-maskable interrupts: 7 (The NMI pin, oscillation stop detection interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, WDT interrupt, IWDT interrupt, and RAM error interrupt) • 16 levels specifiable for the order of priority
DMA	DMA controller (DMACA)	• 4 channels • Transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	Data transfer controller (DTCb)	• Transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: External interrupts and interrupt requests from peripheral functions • Sequence transfer
I/O ports	General I/O ports	• I/O ports for the 100-pin LFQFP I/O pins: 89 (RX260 group), 87 (RX261 group) Input pin: 3 Pull-up resistors: 89 (RX260 group), 87 (RX261 group) Open-drain outputs: 63 5-V tolerance: 4 • I/O ports for the 80-pin LFQFP I/O pins: 69 (RX260 group), 67 (RX261 group) Input pin: 3 Pull-up resistors: 69 (RX260 group), 67 (RX261 group) Open-drain outputs: 47 5-V tolerance: 4 • I/O ports for the 64-pin LFQFP I/O pins: 53 (RX260 group), 51 (RX261 group) Input pin: 3 Pull-up resistors: 53 (RX260 group), 51 (RX261 group) Open-drain outputs: 35 5-V tolerance: 2 • I/O ports for the 48-pin LFQFP, 48-pin HWQFN I/O pins: 39 (RX260 group), 37 (RX261 group) Input pin: 1 Pull-up resistors: 39 (RX260 group), 37 (RX261 group) Open-drain outputs: 27 5-V tolerance: 2
Event link controller (ELC)		• Event signals of 116 types can be directly connected to the module • Operations of timer modules are selectable at event input • Capable of event link operation for ports B and E
Multi-function pin controller (MPC)		Capable of selecting the input/output function from multiple pins

Table 1.1 Outline of Specifications (3/6)

Classification	Module/Function	Description
Timers	General PWM timer (GPTWa)	• (32 bits × 2 channels, 16 bits × 6 channels) × 1 unit • Counting up or down (sawtooth-wave), counting up and down (triangle-wave) selectable for all channels • Clock sources independently selectable for each channel • 2 input/output pins per channel • 2 output compare/input capture registers per channel • For the 2 output compare/input capture registers of each channel, 4 registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at peaks or troughs, enabling the generation of laterally asymmetrically PWM waveforms. • Registers for setting up frame intervals on each channel (with capability for generating interrupts on overflow or underflow) • Generation of dead times in PWM operation • Capable of synchronous start, stop, or clearing of counter for any channel • Capable of a start, stop, clearing, or up-/down-counting of the counter supporting maximum of 8 ELC events • Capable of a start, stop, clearing, or up-/down-counting of the counter supporting input level comparison • Capable of a start, stop, clearing, or up-/down-counting of the counter supporting maximum of 4 external triggers • Output pin disabling function by a short circuit detection among output pins • Capable of generating conversion start triggers for the A/D converters • Capable of outputting events, such as compare-match from A to F and overflow/ underflow, to ELC • Capable of using noise filter of input capture
	Port output enable for GPTW (POEGc)	• Controlling the output disable for GPTW waveform output • Initiation by input level detection of GTETRG pins • Initiation by output disable request from GPTW • Initiation by detection of comparator interrupt request • Initiation by detection of oscillation stop or by software
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	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 • Count clock: Dedicated low-speed on-chip oscillator for the IWDT Frequency divided by 1, 16, 32, 64, 128, or 256
	Realtime clock (RTCBa)	• Clock source: Sub-clock • Clock and calendar functions • Interrupts: Alarm interrupt, periodic interrupt, and carry interrupt • Time capture function (up to 3 pins)
	Low power timer (LPTa)	• 16 bits × 1 channel • Clock source: Sub-clock, LOCO clock divided by 4, or dedicated low-speed clock for the IWDT selectable • Clock division ratio: Frequency divided by 1, 2, 4, 8, 16, or 32 selectable • PWM output mode
	8-bit timer (TMRa)	• (8 bits × 2 channels) × 2 units • Seven internal clocks (PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, and PCLK/8192) and an external clock can be selected • Pulse output and PWM output with any duty cycle are available • Two channels can be cascaded and used as a 16-bit timer

Table 1.1 Outline of Specifications (4/6)

Classification	Module/Function	Description
Communication functions	Serial communications interfaces (SCIk, SCIh)	• 4 channels SCIk: SCI1, SCI5, SCI6 SCIh: SCI12 • SCIk, SCIh Serial communications modes: Asynchronous, clock synchronous, and smart-card interface Multi-processor function On-chip baud rate generator allows selection of the desired bit rate Choice of LSB-first or MSB-first transfer Average transfer rate clock can be input from TMR timers for SCI5, SCI6, and SCI12 Start-bit detection: Level or edge detection is selectable. Simple I^2C Simple SPI 7, 8, 9-bit transfer mode Bit rate modulation Double-speed mode Data match detection (SCI12 is not supported) Event linking by the ELC (supported by SCI5 only) • SCIk Only Data match detection Adjustment of the timing of sampling of the RXD signals • SCIh Only Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format

Table 1.1 Outline of Specifications (5/6)

Classification	Module/Function	Description
Communication functions	Serial communications interfaces (RSCI)	• 3 channels (RSCI0, RSCI8, RSCI9) • 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 (RSCI9) • Data can be transmitted or received in sequence by the 32-byte FIFO buffers of the transmission and reception unit • Manchester encoding is supported (RSCI9). • 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)	• 1 channel • Communications formats: I^2C bus format/SMBus format • Master mode or slave mode selectable • Supports fast mode
	Serial peripheral interface (RSPIc)	• 1 channel • 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 Choice of LSB-first or MSB-first transfer The number of bits in each transfer can be changed to 8 to 16, 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) • Transit/receive data can be swapped in byte units • Double buffers for both transmission and reception • RSPCK can be stopped with the receive buffer full for master reception.
	CANFD module (CANFD)	• 1 channel • Compliance with the ISO11898-1:2015 specification (standard frame and extended frame)
	USB 2.0 FS host/ function module (USBe)	• Includes a UDC (USB Device Controller) and transceiver for USB 2.0 FS • One port • Compliance with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps), low speed (1.5 Mbps) (host only) • Both self-powered mode and bus-powered mode are supported • OTG (On the Go) operation is possible (low-speed is not supported) • Incorporates 2 Kbytes of RAM as a transfer buffer • External pull-up and pull-down resistors are not required
	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 PCLKB, sub-clock, IWDTCLK, and TMR.

Table 1.1 Outline of Specifications (6/6)

Classification	Module/Function	Description
12-bit A/D converter (S12ADE)		• 12 bits (25 channels × 1 unit*1) • 12-bit resolution • Minimum conversion time: 0.50 µs per channel when the ADCLK is operating at 64 MHz • Operating modes Scan mode (single scan mode, continuous scan mode, and group scan mode) Group A priority control (only for group scan mode) • Sampling variable Sampling time can be set up for each channel. • Self-diagnostic function • Double trigger mode (A/D conversion data duplicated) • Detection of analog input disconnection • Conversion results compare features • A/D conversion start conditions A software trigger, a trigger from a timer (GPTW), an external trigger signal, or ELC • Event linking by the ELC
Temperature sensor (TEMPSA)		• 1 channel • The voltage output from the temperature sensor is converted into a digital value by the 12-bit A/D converter.
D/A converter (DAa)		• 2 channels • 8-bit resolution • Output voltage: 0V to AVCC0
CRC calculator (CRC)		• CRC code generation for arbitrary amounts of data in 8-bit units • Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable.
Comparator B (CMPBa)		• 2 channels • Function to compare the reference voltage and the analog input voltage • Window comparator operation or standard comparator operation is selectable
Capacitive touch sensing unit (CTSU2SLa)		• Self-capacitance method: A single pin configures a single key, supporting up to 36 keys • Mutual capacitance method: Matrix configuration with 8 × 8, supporting up to 64 keys • Automatic correction • Automatic judgment
Encryption function	Unique ID	32-byte ID code for the MCU
	Renesas Secure IP (RSIP-E11A)	• Symmetric-key cryptography: AES • Public-key cryptography: ECC • Hash functions: SHA224, SHA256 • True-random number generator
Data operation circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Power supply voltages/Operating frequencies		VCC = 1.6 to 1.8 V: 4 MHz, VCC = 1.8 to 2.4 V: 48 MHz, VCC = 2.4 to 5.5 V: 64 MHz
Operating temperature range		D version: –40 to +85°C, G version: –40 to +105°C
Packages		100-pin LFQFP (PLQP0100KB-B) 14 × 14 mm, 0.5 mm pitch 80-pin LFQFP (PLQP0080KB-B) 12 × 12 mm, 0.5 mm pitch 64-pin LFQFP (PLQP0064KB-C) 10 × 10 mm, 0.5 mm pitch 48-pin LFQFP (PLQP0048KB-B) 7 × 7 mm, 0.5 mm pitch 48-pin HWQFN (PWQN0048KC-A) 7 × 7 mm, 0.5 mm pitch
Debugging interfaces		FINE interface

Note 1. 25 channel pins consist of 24 external input pins, and an internal input pin for the CTSU.

Table 1.2 Comparison of Functions for Different Packages

Module/Functions		RX260 Group				RX261 Group			
		100 Pins	80 Pins	64 Pins	48 Pins	100 Pins	80 Pins	64 Pins	48 Pins
Interrupts	External interrupts	NMI, IRQ0 to IRQ7		NMI, IRQ0 to IRQ2, IRQ4 to IRQ7		NMI, IRQ0 to IRQ7		NMI, IRQ0 to IRQ2, IRQ4 to IRQ7	
DMA	DMA controller	4 channels (DMAC0 to DMAC3)				4 channels (DMAC0 to DMAC3)			
	Data transfer controller	Available				Available			
Timers	General PWM timer	8 channels				8 channels			
	Port output enable for GPTW	Available				Available			
	8-bit timer	2 channels × 2 units				2 channels × 2 units			
	Compare match timer	2 channels × 2 units				2 channels × 2 units			
	Low power timer	1 channel				1 channel			
	Realtime clock	Available			Not available	Available			Not available
	Watchdog timer	Available				Available			
	Independent watchdog timer	Available				Available			
Communication functions	Serial communications interfaces (SCIk)	Ch. 1, 5, and 6				Ch. 1, 5, and 6			
	Serial communications interfaces (SCIh)	Ch. 12				Ch. 12			
	Serial communications interfaces (RSCI)	Ch. 0, 8, and 9			Ch. 0 and 8	Ch. 0, 8, and 9			Ch. 0 and 8
	I^2C bus interface	1 channel				1 channel			
	Serial peripheral interface	1 channel				1 channel			
	CANFD module (CANFD)	Not available				1 channel			
	USB 2.0 FS host/ function module	Not available				1 channel			
	Remote control signal receiver (REMC)	1 channel				1 channel			
Capacitive touch sensing unit		36 channels		32 channels	24 channels	34 channels		30 channels	22 channels
12-bit A/D converter		25 channels	18 channels	15 channels	11 channels	25 channels	18 channels	15 channels	11 channels
Temperature sensor		Available				Available			
D/A converter		2 channels			Not available	2 channels			Not available
CRC calculator		Available				Available			
Event link controller		Available				Available			
Comparator B		2 channels				2 channels			
Renesas Secure IP (RSIP-E11A)		Not available				Available/Not available			
Packages		100-pin LFQFP	80-pin LFQFP	64-pin LFQFP	48-pin LFQFP 48-pin HWQFN	100-pin LFQFP	80-pin LFQFP	64-pin LFQFP	48-pin LFQFP 48-pin HWQFN

1.2 List of Products

Table 1.3 is a lists of products, and Figure 1.1 shows how to read the product part no., memory capacity, and package type.

Table 1.3 List of Products (1/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	E2 DataFlash	Operating Frequency	Encryption Module	CANFD	USB	Operating Temperature
RX261 (D-version)	R5F52618ADFP	PLQP0100KB-B	512 Kbytes	128 Kbytes	8 Kbytes	64 MHz	Not available	Available*1	Available	–40 to +85°C
	R5F52618BDFP	PLQP0100KB-B					Available	Available	Available	
	R5F52618ADFN	PLQP0080KB-B					Not available	Available*1	Available	
	R5F52618BDFN	PLQP0080KB-B					Available	Available	Available	
	R5F52618ADFM	PLQP0064KB-C					Not available	Available*1	Available	
	R5F52618BDFM	PLQP0064KB-C					Available	Available	Available	
	R5F52618ADFL	PLQP0048KB-B					Not available	Available*1	Available	
	R5F52618BDFL	PLQP0048KB-B					Available	Available	Available	
	R5F52618ADNE	PWQN0048KC-A					Not available	Available*1	Available	
	R5F52618BDNE	PWQN0048KC-A					Available	Available	Available	
	R5F52617ADFP	PLQP0100KB-B	384 Kbytes				Not available	Available*1	Available	
	R5F52617BDFP	PLQP0100KB-B					Available	Available	Available	
	R5F52617ADFN	PLQP0080KB-B					Not available	Available*1	Available	
	R5F52617BDFN	PLQP0080KB-B					Available	Available	Available	
	R5F52617ADFM	PLQP0064KB-C					Not available	Available*1	Available	
	R5F52617BDFM	PLQP0064KB-C					Available	Available	Available	
	R5F52617ADFL	PLQP0048KB-B					Not available	Available*1	Available	
	R5F52617BDFL	PLQP0048KB-B					Available	Available	Available	
	R5F52617ADNE	PWQN0048KC-A					Not available	Available*1	Available	
	R5F52617BDNE	PWQN0048KC-A					Available	Available	Available	
	R5F52616ADFP	PLQP0100KB-B	256 Kbytes				Not available	Available*1	Available	
	R5F52616BDFP	PLQP0100KB-B					Available	Available	Available	
	R5F52616ADFN	PLQP0080KB-B					Not available	Available*1	Available	
	R5F52616BDFN	PLQP0080KB-B					Available	Available	Available	
	R5F52616ADFM	PLQP0064KB-C					Not available	Available*1	Available	
	R5F52616BDFM	PLQP0064KB-C					Available	Available	Available	
	R5F52616ADFL	PLQP0048KB-B					Not available	Available*1	Available	
	R5F52616BDFL	PLQP0048KB-B					Available	Available	Available	
	R5F52616ADNE	PWQN0048KC-A					Not available	Available*1	Available	
	R5F52616BDNE	PWQN0048KC-A					Available	Available	Available	

Table 1.3 List of Products (2/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	E2 DataFlash	Operating Frequency	Encryption Module	CANFD	USB	Operating Temperature
RX261 (G-version)	R5F52618AGFP	PLQP0100KB-B	512 Kbytes	128 Kbytes	8 Kbytes	64 MHz	Not available	Available*1	Available	–40 to +105°C
	R5F52618BGFP	PLQP0100KB-B					Available	Available	Available	
	R5F52618AGFN	PLQP0080KB-B					Not available	Available*1	Available	
	R5F52618BGFN	PLQP0080KB-B					Available	Available	Available	
	R5F52618AGFM	PLQP0064KB-C					Not available	Available*1	Available	
	R5F52618BGFM	PLQP0064KB-C					Available	Available	Available	
	R5F52618AGFL	PLQP0048KB-B					Not available	Available*1	Available	
	R5F52618BGFL	PLQP0048KB-B					Available	Available	Available	
	R5F52618AGNE	PWQN0048KC-A					Not available	Available*1	Available	
	R5F52618BGNE	PWQN0048KC-A					Available	Available	Available	
	R5F52617AGFP	PLQP0100KB-B	384 Kbytes				Not available	Available*1	Available	
	R5F52617BGFP	PLQP0100KB-B					Available	Available	Available	
	R5F52617AGFN	PLQP0080KB-B					Not available	Available*1	Available	
	R5F52617BGFN	PLQP0080KB-B					Available	Available	Available	
	R5F52617AGFM	PLQP0064KB-C					Not available	Available*1	Available	
	R5F52617BGFM	PLQP0064KB-C					Available	Available	Available	
	R5F52617AGFL	PLQP0048KB-B					Not available	Available*1	Available	
	R5F52617BGFL	PLQP0048KB-B					Available	Available	Available	
	R5F52617AGNE	PWQN0048KC-A					Not available	Available*1	Available	
	R5F52617BGNE	PWQN0048KC-A					Available	Available	Available	
	R5F52616AGFP	PLQP0100KB-B	256 Kbytes				Not available	Available*1	Available	
	R5F52616BGFP	PLQP0100KB-B					Available	Available	Available	
	R5F52616AGFN	PLQP0080KB-B					Not available	Available*1	Available	
	R5F52616BGFN	PLQP0080KB-B					Available	Available	Available	
	R5F52616AGFM	PLQP0064KB-C					Not available	Available*1	Available	
	R5F52616BGFM	PLQP0064KB-C					Available	Available	Available	
	R5F52616AGFL	PLQP0048KB-B					Not available	Available*1	Available	
	R5F52616BGFL	PLQP0048KB-B					Available	Available	Available	
	R5F52616AGNE	PWQN0048KC-A					Not available	Available*1	Available	
	R5F52616BGNE	PWQN0048KC-A					Available	Available	Available	

Table 1.3 List of Products (3/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	E2 DataFlash	Operating Frequency	Encryption Module	CANFD	USB	Operating Temperature
RX260 (D-version)	R5F52608ADFP	PLQP0100KB-B	512 Kbytes	128 Kbytes	8 Kbytes	64 MHz	Not available	Not available	Not available	–40 to +85°C
	R5F52608ADFN	PLQP0080KB-B					Not available	Not available	Not available	
	R5F52608ADFM	PLQP0064KB-C					Not available	Not available	Not available	
	R5F52608ADFL	PLQP0048KB-B					Not available	Not available	Not available	
	R5F52608ADNE	PWQN0048KC-A					Not available	Not available	Not available	
	R5F52607ADFP	PLQP0100KB-B	384 Kbytes				Not available	Not available	Not available	
	R5F52607ADFN	PLQP0080KB-B					Not available	Not available	Not available	
	R5F52607ADFM	PLQP0064KB-C					Not available	Not available	Not available	
	R5F52607ADFL	PLQP0048KB-B					Not available	Not available	Not available	
	R5F52607ADNE	PWQN0048KC-A					Not available	Not available	Not available	
	R5F52606ADFP	PLQP0100KB-B	256 Kbytes				Not available	Not available	Not available	
	R5F52606ADFN	PLQP0080KB-B					Not available	Not available	Not available	
	R5F52606ADFM	PLQP0064KB-C					Not available	Not available	Not available	
	R5F52606ADFL	PLQP0048KB-B					Not available	Not available	Not available	
	R5F52606ADNE	PWQN0048KC-A					Not available	Not available	Not available	
RX260 (G-version)	R5F52608AGFP	PLQP0100KB-B	512 Kbytes	128 Kbytes	8 Kbytes	64 MHz	Not available	Not available	Not available	–40 to +105°C
	R5F52608AGFN	PLQP0080KB-B					Not available	Not available	Not available	
	R5F52608AGFM	PLQP0064KB-C					Not available	Not available	Not available	
	R5F52608AGFL	PLQP0048KB-B					Not available	Not available	Not available	
	R5F52608AGNE	PWQN0048KC-A					Not available	Not available	Not available	
	R5F52607AGFP	PLQP0100KB-B	384 Kbytes				Not available	Not available	Not available	
	R5F52607AGFN	PLQP0080KB-B					Not available	Not available	Not available	
	R5F52607AGFM	PLQP0064KB-C					Not available	Not available	Not available	
	R5F52607AGFL	PLQP0048KB-B					Not available	Not available	Not available	
	R5F52607AGNE	PWQN0048KC-A					Not available	Not available	Not available	
	R5F52606AGFP	PLQP0100KB-B	256 Kbytes				Not available	Not available	Not available	
	R5F52606AGFN	PLQP0080KB-B					Not available	Not available	Not available	
	R5F52606AGFM	PLQP0064KB-C					Not available	Not available	Not available	
	R5F52606AGFL	PLQP0048KB-B					Not available	Not available	Not available	
	R5F52606AGNE	PWQN0048KC-A					Not available	Not available	Not available	

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

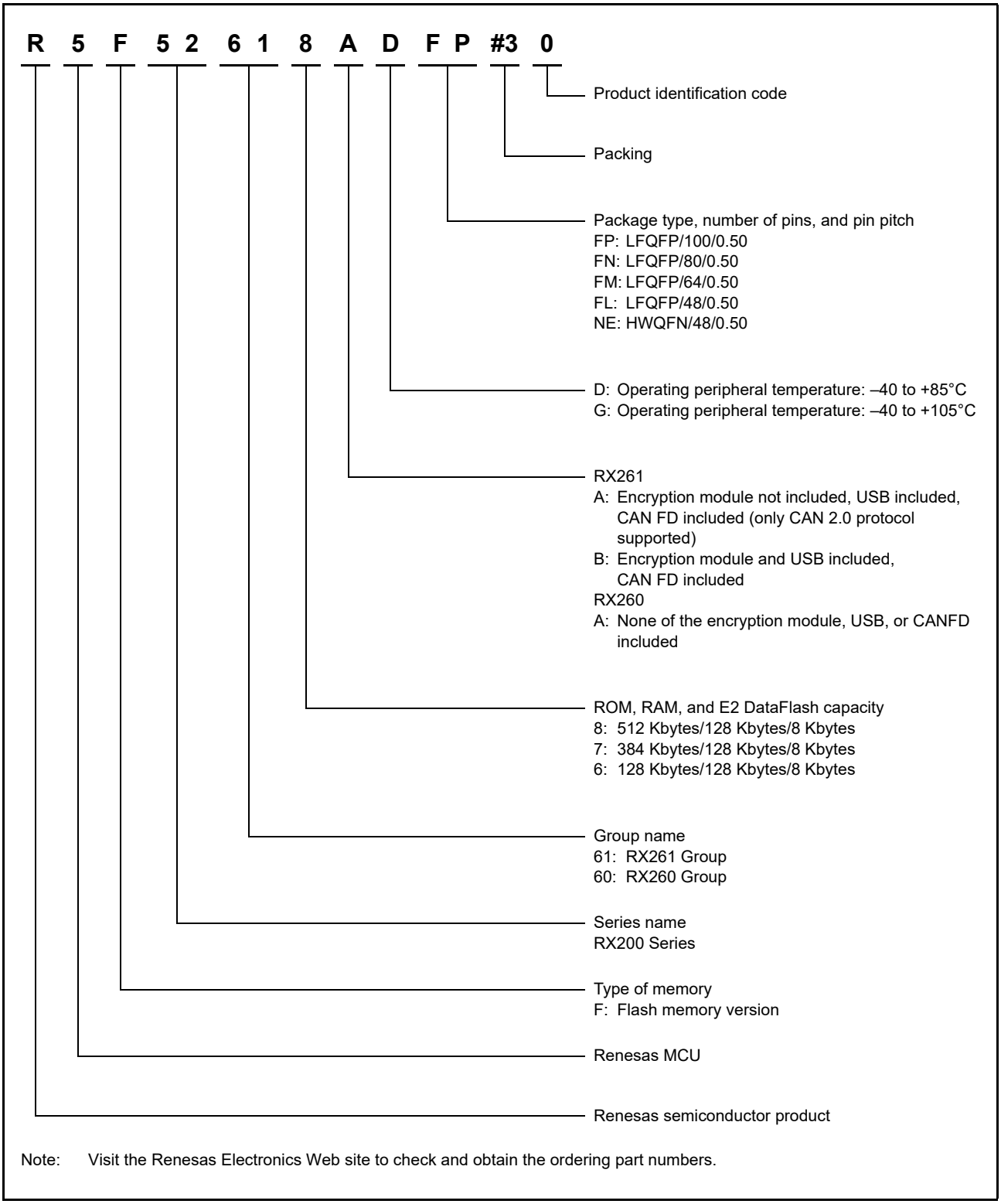
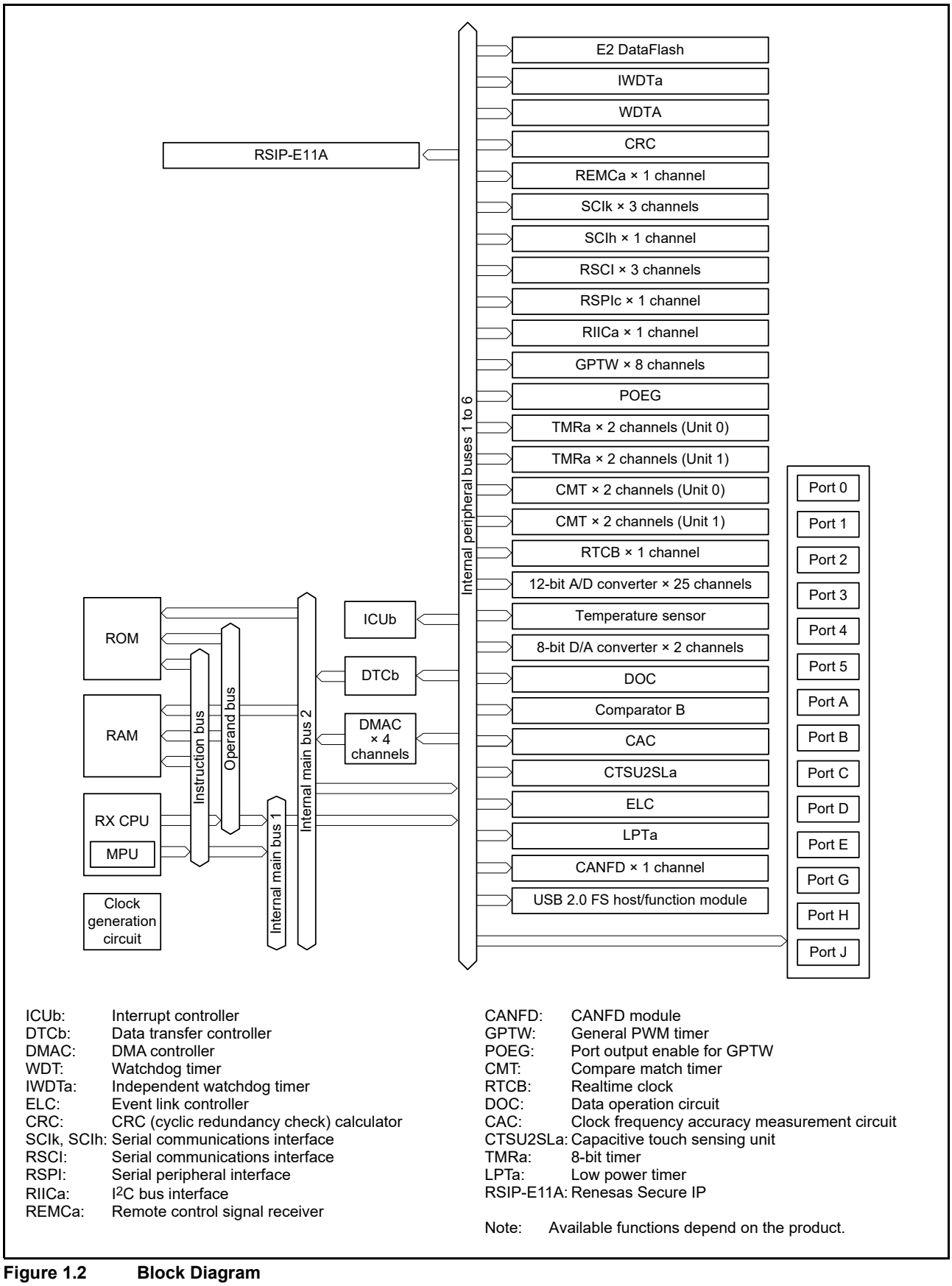


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ICUb: Interrupt controller
DTCb: Data transfer controller
DMAC: DMA controller
WDT: Watchdog timer
IWDTa: Independent watchdog timer
ELC: Event link controller
CRC: CRC (cyclic redundancy check) calculator
SCIk, SCIh: Serial communications interface
RSCI: Serial communications interface
RSPI: Serial peripheral interface
RIICa: I^2C bus interface
REMCa: Remote control signal receiver

CANFD: CANFD module
GPTW: General PWM timer
POEG: Port output enable for GPTW
CMT: Compare match timer
RTCB: Realtime clock
DOC: Data operation circuit
CAC: Clock frequency accuracy measurement circuit
CTSU2SLa: Capacitive touch sensing unit
TMRa: 8-bit timer
LPTa: Low power timer
RSIP-E11A: Renesas Secure IP

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 (1/4)

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	Input	Connect this pin to the VSS pin via the 4.7 µF smoothing capacitor used to stabilize the internal power supply. Place the capacitor 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 can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal between XCIN and XCOUT.
	XCOUT	Output	
	EXCIN	Input	External clock input pin for the sub-clock oscillator
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on this pin must not be changed during operation.
	UB	Input	This pin is used in boot mode (USB interface).
	UPSEL	Input	This pin is used in boot mode (USB interface).
System control	RES#	Input	Reset pin. This MCU enters the reset state when this signal goes low.
Voltage detection circuit	CMPA2	Input	Detection target voltage pin for voltage detection 2.
CAC	CACREF	Input	Input pin for the clock frequency accuracy measurement circuit
On-chip emulator	FINED	I/O	FINE interface pin
Interrupts	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ7	Input	Interrupt request pins
General PWM timer	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOC0A to GTIOC7A, GTIOC0B to GTIOC7B	I/O	Input capture input/output compare output/PWM output pins
	GTIOC0A# to GTIOC7A#, GTIOC0B# to GTIOC7B#	I/O	Input capture inverted input/output compare inverted output/PWM inverted output pins
	GTCPPO0	Output	Synchronized PWM output
	GTIU, GTIV, GTIW	Input	Hall sensor input pins
	GTOUUP	Output	A three-phase PWM output for controlling a brushless DC motor (positive U-phase)
	GTOULO	Output	A three-phase PWM output for controlling a brushless DC motor (negative U-phase)
	GTOVUP	Output	A three-phase PWM output for controlling a brushless DC motor (positive V-phase)
	GTOVLO	Output	A three-phase PWM output for controlling a brushless DC motor (negative V-phase)
	GTOWUP	Output	A three-phase PWM output for controlling a brushless DC motor (positive W-phase)
	GTOWLO	Output	A three-phase PWM output for controlling a brushless DC motor (negative W-phase)
Realtime clock	RTCOUT	Output	Output pin for the 1-Hz/64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins

Table 1.4 Pin Functions (2/4)

Classifications	Pin Name	I/O	Description
8-bit timer	TMO0 to TMO3	Output	Compare match output pins
	TMCI0 to TMCI3	Input	Input pins for the external clock to be input to the counter
	TMRI0 to TMRI3	Input	Counter reset input pins
Low power timer	LPTO	Output	PWM output pin
Serial communications interface (SCIk)	• Asynchronous mode/clock synchronous mode		
	SCK1, SCK5, SCK6	I/O	Input/output pins for the clock
	RXD1, RXD5, RXD6	Input	Input pins for received data
	TXD1, TXD5, TXD6	Output	Output pins for transmitted data
	CTS1#, CTS5#, CTS6#	Input	Input pins for controlling the start of transmission and reception
	RTS1#, RTS5#, RTS6#	Output	Output pins for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL1, SSCL5, SSCL6	I/O	Input/output pins for the I^2C clock
	SSDA1, SSDA5, SSDA6	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK1, SCK5, SCK6	I/O	Input/output pins for the clock
	SMISO1, SMISO5, SMISO6	I/O	Input/output pins for slave transmit data
	SMOSI1, SMOSI5, SMOSI6	I/O	Input/output pins for master transmit data
	SS1#, SS5#, SS6#	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 receiving data
	TXD12	Output	Output pin for transmitting data
	CTS12#	Input	Input pin for controlling the start of transmission and reception
	RTS12#	Output	Output pin for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pin for the I^2C clock
	SSDA12	I/O	Input/output pin for the I^2C data
	• Simple SPI mode		
	SCK12	I/O	Input/output pin for the clock
	SMISO12	I/O	Input/output pin for slave transmit data
	SMOSI12	I/O	Input/output pin for master transmit 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

Table 1.4 Pin Functions (3/4)

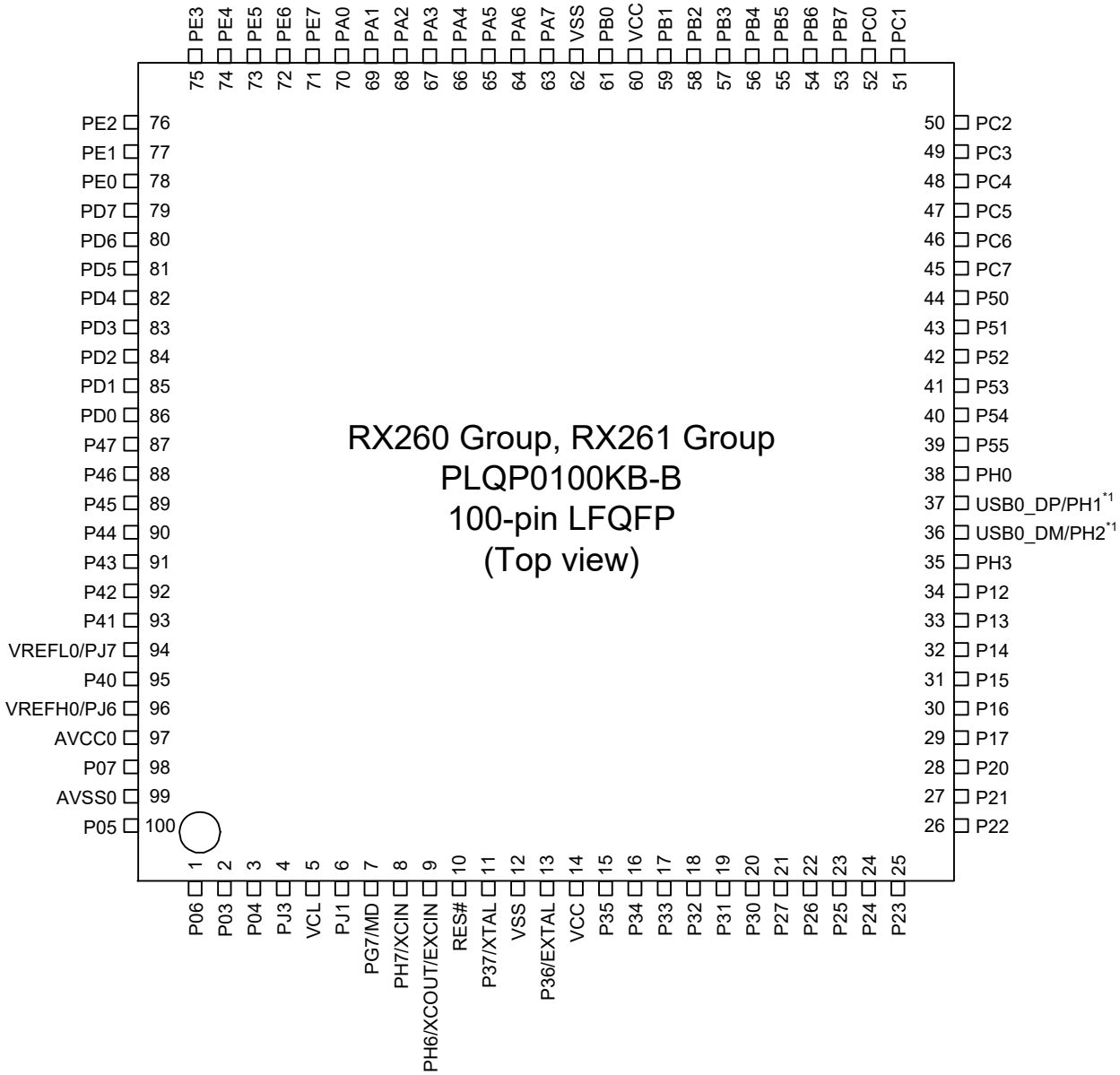
Classifications	Pin Name	I/O	Description
Serial communications interface (RSCI)	• Asynchronous mode/clock synchronous mode		
	SCK000, SCK008, SCK009	I/O	Input/output pins for the clock
	RXD000, RXD008, RXD009	Input	Input pins for received data
	TXD000, TXD008, TXD009	Output	Output pins for transmitted data
	CTS000#, CTS008#, CTS009#	Input	Input pins for controlling the start of transmission and reception
	RTS000#, RTS008#, RTS009#	Output	Output pins for controlling the start of transmission and reception
	DE000, DE008, DE009	Output	DriveEnable output pins
	• Simple I^2C mode		
	SSCL000, SSCL008, SSCL009	I/O	Input/output pins for the I^2C clock
	SSDA000, SSDA008, SSDA009	I/O	Input/output pins for the I^2C data
Serial communications interface (RSCI)	• Simple SPI mode		
	SCK000, SCK008, SCK009	I/O	Input/output pins for the clock
	SMISO000, SMISO008, SMISO009	I/O	Input/output pins for slave transmission of data
	SMOSI000, SMOSI008, SMOSI009	I/O	Input/output pins for master transmission of data
	SS000#, SS008#, SS009#	Input	Chip-select input pins
	• HBS support mode		
	RXD000, RXD008, RXD009	Input	Input pins for received data
	TXDA000, TXDA008, TXDA009	Output	Output pins for transmitted data
	TXDB000, TXDB008, TXDB009	Output	Output pins for transmitted data
Remote control signal receiver (REMC)	PMC0	Input	Input pin for external pulse signal
I^2C bus interface	SCL0	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open drain output.
	SDA0	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open drain output.
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
USB 2.0 FS host/function module	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver
	USB0_DM	I/O	D- I/O pin of the USB on-chip transceiver
	USB0_VBUS	Input	USB cable connection monitor pin
	USB0_EXICEN	Output	Low-power control signal for the OTG chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for the OTG chip
	USB0_OVRCURA, USB0_OVRCURB	Input	External overcurrent detection pins
	USB0_ID	Input	Mini-AB connector ID input pin during operation in OTG mode
CANFD module	CRX0	Input	Pin for receiving data
	CTX0	Output	Pin for transmitting data

Table 1.4 Pin Functions (4/4)

Classifications	Pin Name	I/O	Description
12-bit A/D converter	AN000 to AN007, AN016 to AN031	Input	Input pins for the analog signals to be processed by the A/D converter
	ADTRG0#	Input	Input pin for the external trigger signal that start the A/D conversion
8-bit D/A converter	DA0, DA1	Output	Analog output pins of the D/A converter
Comparator B	CMPB0, CMPB1	Input	Input pins for the analog signal to be processed by comparator B
	CVREFB0, CVREFB1	Input	Analog reference voltage supply pins for comparator B
	CMPOB0, CMPOB1	Output	Output pins for comparator B
Capacitive touch sensing unit (CTSU)	TS0 to TS35	I/O	Electrostatic capacitance measurement pins (touch pins)
	TSCAP	—	Connect to the VSS via a decoupling capacitor (0.01 µF) for stabilizing the internal voltage
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter and D/A converter. Connect this pin to VCC when not using the 12-bit A/D converter and D/A converter.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter and D/A converter. Connect this pin to VSS when not using the 12-bit A/D converter and D/A converter.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter.
I/O ports	P03 to P07	I/O	5-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 P55	I/O	6-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.
	PG7	I/O	1-bit input/output pin.
	PH0 to PH3, PH6, PH7	I/O	6-bit input/output pins (PH6, PH7: input pins).
	PJ1, PJ3, PJ6, PJ7	I/O	4-bit input/output pins.

1.5 Pin Assignments

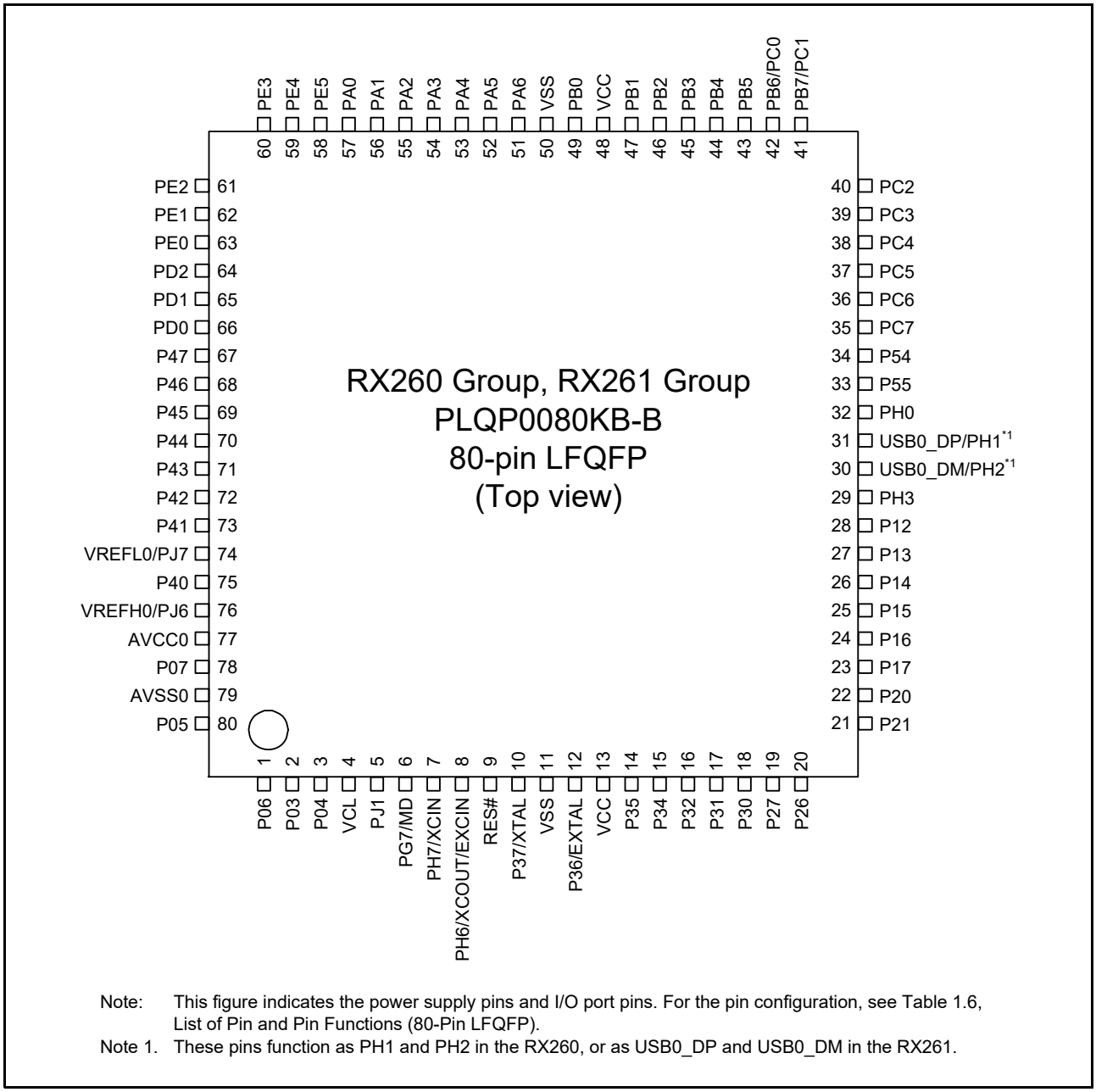
1.5.1 100-Pin LFQFP



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 (100-Pin LFQFP).
Note 1. These pins function as PH1 and PH2 in the RX260, or as USB0_DP and USB0_DM in the RX261.

Figure 1.3 Pin Assignments (100-Pin LFQFP)

1.5.2 80-Pin LFQFP

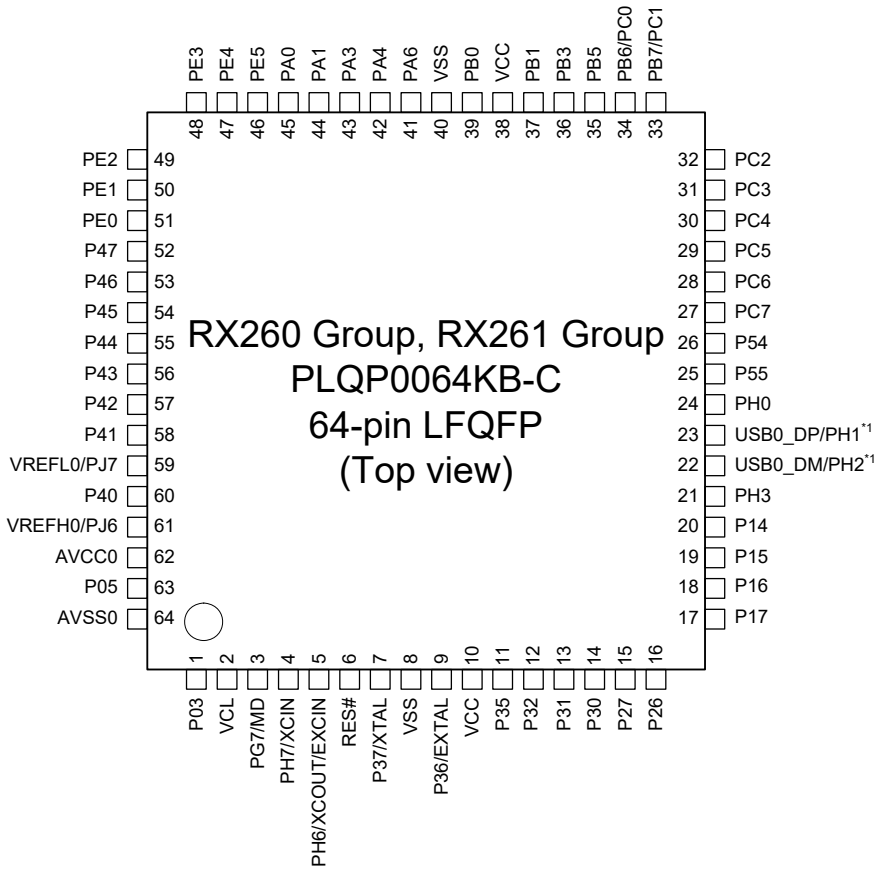


Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.6, List of Pin and Pin Functions (80-Pin LFQFP).

Note 1. These pins function as PH1 and PH2 in the RX260, or as USB0_DP and USB0_DM in the RX261.

Figure 1.4 Pin Assignments (80-Pin LFQFP)

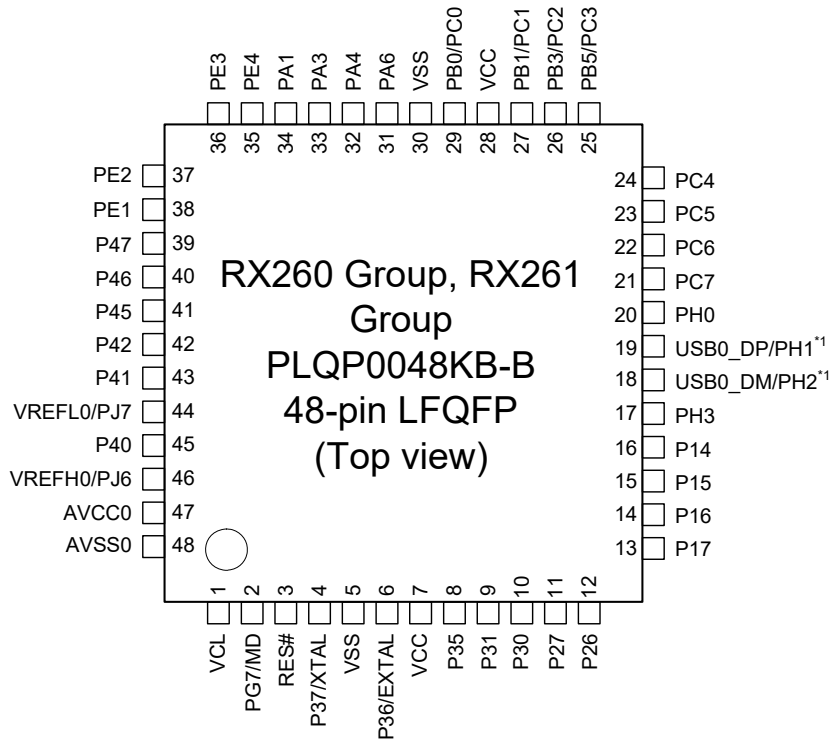
1.5.3 64-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.7, List of Pin and Pin Functions (64-Pin LFQFP).
Note 1. These pins function as PH1 and PH2 in the RX260, or as USB0_DP and USB0_DM in the RX261.

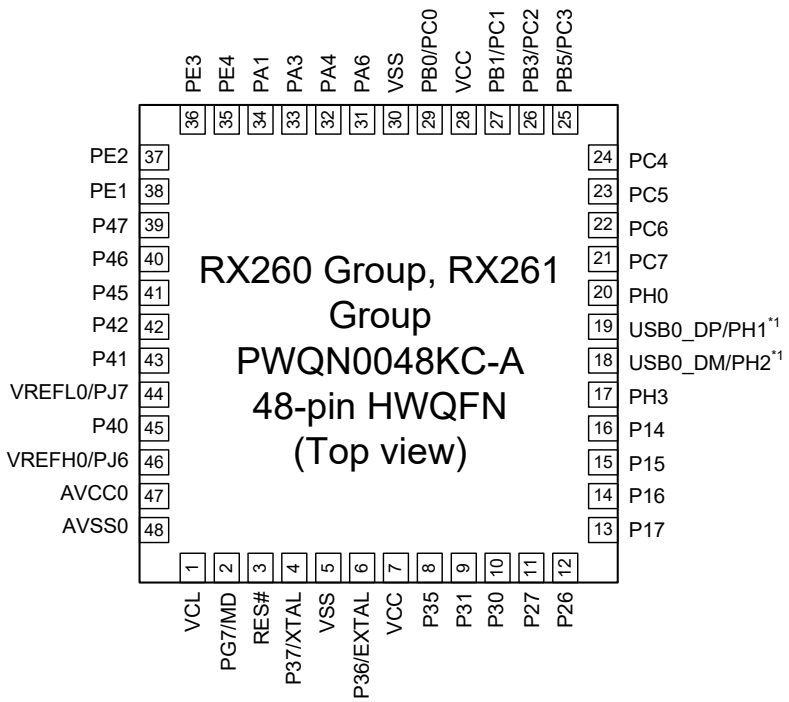
Figure 1.5 Pin Assignments (64-Pin LFQFP)

1.5.4 48-Pin LFQFP, 48-Pin HWQFN



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 (48-Pin LFQFP, 48-Pin HWQFN).
Note 1. These pins function as PH1 and PH2 in the RX260, or as USB0_DP and USB0_DM in the RX261.

Figure 1.6 Pin Assignments (48-Pin LFQFP)



Note: We recommend connecting the exposed die pad to VSS.
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.8, List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN).
Note 1. These pins function as PH1 and PH2 in the RX260, or as USB0_DP and USB0_DM in the RX261.

Figure 1.7 Pin Assignments (48-Pin HWQFN)

1.6 List of Pins and Pin Functions

1.6.1 100-Pin LFQFP

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (1/5)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
1		P06*1					
2		P03*1					DA0
3		P04*1					
4		PJ3	GTIOC6B/GTIOC6B#	CTS6#/RTS6#/SS6#			
5	VCL						
6		PJ1	GTIOC6A/GTIOC6A#/ GTCPPO0				
7	MD/FINED	PG7					
8	XCIN	PH7					
9	XCOUT/ EXCIN	PH6					
10	RES#						
11	XTAL	P37				IRQ4	
12	VSS						
13	EXTAL	P36				IRQ2	
14	VCC						
15	UPSEL	P35				NMI	
16		P34	GTIOC3A/GTIOC3A#/ GTIU/TMCI3	SCK6		IRQ4	
17		P33	GTIOC1B/GTIOC7B/ GTIOC1B#/GTIOC7B#/ TMRI3	RXD6/SMISO6/SSCL6/ CRX0*2		IRQ3	
18		P32	GTIOC1A/GTIOC7A/ GTIOC1A#/GTIOC7A#/ GTIW/TMO3/RTCOUT/ RTCIC2	TXD6/SMOSI6/SSDA6/ CTX0*2/ USB0_VBUSEN*2	TS0	IRQ2	
19		P31	GTIOC2B/GTIOC2B#/ GTOWLO/TMCI2/ RTCIC1	CTS1#/RTS1#/SS1#	TS1	IRQ1	
20		P30	GTIOC2A/GTIOC2A#/ GTOWUP/TMRI3/ RTCIC0	RXD1/SMISO1/SSCL1	TS2	IRQ0	
21		P27	GTIOC5B/GTIOC5B#/ TMCI3	SCK1	TS3		
22		P26	GTIOC5A/GTIOC5A#/ TMO1/LPTO	TXD1/SMOSI1/SSDA1/ USB0_VBUSEN*2	TS4		
23		P25	GTIOC1B/GTIOC6A/ GTIOC1B#/GTIOC6A#/ GTETRGB				ADTRG0#
24		P24	GTIOC1A/GTIOC6B/ GTIOC1A#/GTIOC6B#/ GTETRGA/TMRI1	USB0_VBUSEN*2			
25		P23	GTIOC0B/GTIOC3B/ GTIOC0B#/GTIOC3B#/ GTETRGD	CTS000#/RTS000#/ SS000#/DE000			

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (2/5)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
26		P22	GTIOC0A/GTIOC3A/ GTIOC0A#/GTIOC3A#/ GTETRGC/TMO0	SCK000/TXDB000/ USB0_OVRCURB*2			
27		P21	GTIOC2A/GTIOC4B/ GTIOC2A#/GTIOC4B#/ TMCI0	RXD000/SMISO000/ SSCL000/ USB0_EXICEN*2			
28		P20	GTIOC2B/GTIOC4A/ GTIOC2B#/GTIOC4A#/ TMRI0	TXD000/TXDA000/ SMOSI000/SSDA000/ USB0_ID*2			
29		P17	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC6A/GTIOC6A#/ GTETRGD/GTCPPO0/ GTOUUP/TMO1	SCK1/MISOA/SDA0		IRQ7	
30		P16	GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTIOC6B/GTIOC6B#/ GTETRGC/GTOULO/ TMO2/RTCOUT	TXD1/SMOSI1/SSDA1/ MOSIA/SCL0/ USB0_VBUS*2/ USB0_VBUSEN*2/ USB0_OVRCURB*2		IRQ6	ADTRG0#
31		P15	GTIOC3B/GTIOC5B/ GTIOC3B#/GTIOC5B#/ GTETRGB/GTIV/TMCI2	RXD1/SMISO1/SSCL1/ CRX0*2	TS5	IRQ5	
32		P14	GTIOC6A/GTIOC7B/ GTIOC6A#/GTIOC7B#/ GTETRGA/GTCPPO0/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0*2/ USB0_OVRCURA*2	TS6	IRQ4	
33		P13	GTIOC3B/GTIOC7A/ GTIOC3B#/GTIOC7A#/ GTIV/TMO3	SDA0		IRQ3	
34		P12	TMCI1	SCL0		IRQ2	
35		PH3	GTIOC2B/GTIOC2B#/ TMCI0		TS7		
36		PH2*3	GTIOC1B*3/ GTIOC1B#*3/TMRI0*3	USB0_DM*2	TS8*3	IRQ1*3	
37		PH1*3	GTIOC0B*3/ GTIOC0B#*3/GTOULO*3/ TMO0*3	USB0_DP*2	TS9*3	IRQ0*3	
38		PH0	GTIOC0A/GTIOC0A#/ GTOUUP/CACREF		TS10		
39		P55	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ TMO3	CRX0*2	TS11		
40		P54	GTIOC2A/GTIOC2A#/ TMCI1	CTX0*2	TS12		
41		P53		PMC0			
42		P52					
43		P51		PMC0			
44		P50					
45	UB	PC7	GTIOC6A/GTIOC6A#/ GTETRGB/GTCPPO0/ TMO2/LPTO/CACREF	TXD008/TXDA008/ SMOSI008/SSDA008/ MISOA	TS13		
46		PC6	GTIOC6B/GTIOC6B#/ GTETRGA/TMCI2	RXD008/SMISO008/ SSCL008/MOSIA/ USB0_EXICEN*2	TS14		

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (3/5)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
47		PC5	GTIOC0A/GTIOC7A/ GTIOC0A#/GTIOC7A#/ GTETRGD/GTOUUP/ GTIW/TMRI2	SCK008/TXDB008/ RSPCKA/USB0_ID*2/ PMC0	TS15		
48		PC4	GTIOC0B/GTIOC3A/ GTIOC0B#/GTIOC3A#/ GTETRGC/GTIU/ GTOULO/TMCI1	SCK5/CTS008#/ RTS008#/SS008#/ DE008/SSLA0/PMC0	TSCAP		
49		PC3	GTIOC2B/GTIOC2B#/ GTETRGB	TXD5/SMOSI5/SSDA5/ PMC0	TS16		
50		PC2	GTIOC2A/GTIOC2A#/ GTETRGA/GTOWUP	RXD5/SMISO5/SSCL5/ SSLA3	TS17		
51		PC1	GTIOC6A/GTIOC6A#/ GTETRGD/GTCPPO0	SCK5/SSLA2			
52		PC0	GTIOC6B/GTIOC6B#/ GTETRGC	CTS5#/RTS5#/SS5#/ SSLA1			
53		PB7	GTIOC0A/GTIOC7B/ GTIOC0A#/GTIOC7B#	TXD009/TXDA009/ SMOSI009/SSDA009	TS18		
54		PB6	GTIOC0B/GTIOC7A/ GTIOC0B#/GTIOC7A#	RXD009/SMISO009/ SSCL009	TS19		
55		PB5	GTIOC4B/GTIOC5A/ GTIOC4B#/GTIOC5A#/ GTIOC6B/GTIOC6B#/ TMRI1	SCK009/TXDB009/ USB0_VBUS*2	TS20		
56		PB4	GTIOC6A/GTIOC6A#	CTS009#/RTS009#/ SS009#/DE009	TS21		
57		PB3	GTIOC1A/GTIOC3A/ GTIOC1A#/GTIOC3A#/ GTIOC3B/GTIOC3B#/ GTETRGD/GTIU/ GTOVUP/TMO0/LPTO	SCK6/PMC0	TS22		
58		PB2	GTIOC3A/GTIOC3A#/ GTETRGC	CTS6#/RTS6#/SS6#	TS23		
59		PB1	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7A/GTIOC7A#/ GTOVLO/GTIW/ GTOWLO/TMCI0	TXD6/SMOSI6/SSDA6	TS24	IRQ4	CMPOB1
60	VCC						
61		PB0	GTIOC0B/GTIOC2A/ GTIOC0B#/GTIOC2A#/ GTOWUP	RXD6/SMISO6/SSCL6/ RSPCKA	TS25		
62	VSS						
63		PA7	GTIOC5B/GTIOC5B#	MISOA			
64		PA6	GTIOC0B/GTIOC5A/ GTIOC0B#/GTIOC5A#/ GTETRGB/GTOULO/ TMCI3	CTS5#/RTS5#/SS5#/ MOSIA	TS26		
65		PA5	GTIOC4B/GTIOC4B#	RSPCKA	TS27		
66		PA4	GTIOC1B/GTIOC4A/ GTIOC1B#/GTIOC4A#/ GTETRGA/GTOVLO/ TMRI0	TXD5/SMOSI5/SSDA5/ SSLA0	TS28	IRQ5	CVREFB1

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (4/5)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
67		PA3	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7B/GTIOC7B#/ GTETRGB/GTETRGD/ GTOVLO/GTOWLO	RXD5/SMISO5/SSCL5	TS29	IRQ6	CMPB1
68		PA2		RXD5/SMISO5/SSCL5/ SSLA3	TS30		
69		PA1	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC3B/GTIOC3B#/ GTETRGC/GTIV/ GTOUUP	SCK5/SSLA2	TS31		
70		PA0	GTIOC0A/GTIOC1A/ GTIOC0A#/GTIOC1A#/ GTOVUP/CACREF	SSLA1	TS32		
71		PE7				IRQ7	AN023
72		PE6				IRQ6	AN022
73		PE5	GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#			IRQ5	AN021/ CMPOB0
74	CLKOUT	PE4	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ GTIOC4A/GTIOC4A#/ GTOVUP/GTOWLO		TS33		AN020/ CMPA2
75	CLKOUT	PE3	GTIOC2A/GTIOC4B/ GTIOC2A#/GTIOC4B#/ GTOWUP	CTS12#/RTS12#/SS12#	TS34		AN019
76		PE2	GTIOC1A/GTIOC1A#/ GTOVUP	RXD12/SMISO12/ SSCL12/RXDX12	TS35	IRQ7	AN018/ CVREFB0
77		PE1	GTIOC1B/GTIOC1B#/ GTOVLO	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12			AN017/ CMPB0
78		PE0		SCK12			AN016
79		PD7				IRQ7	AN031
80		PD6				IRQ6	AN030
81		PD5				IRQ5	AN029
82		PD4				IRQ4	AN028
83		PD3				IRQ3	AN027
84		PD2	GTIOC2B/GTIOC2B#	SCK6/CRX0*2		IRQ2	AN026
85		PD1	GTIOC2A/GTIOC2A#	RXD6/SMISO6/SSCL6/ CTX0*2		IRQ1	AN025
86		PD0		TXD6/SMOSI6/SSDA6		IRQ0	AN024
87		P47*1					AN007
88		P46*1					AN006
89		P45*1					AN005
90		P44*1					AN004
91		P43*1					AN003
92		P42*1					AN002
93		P41*1					AN001
94	VREFL0	PJ7*1					
95		P40*1					AN000

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (5/5)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
96	VREFH0	PJ6*1					
97	AVCC0						
98		P07*1					ADTRG0#
99	AVSS0						
100		P05*1					DA1

Note 1. The power source of the I/O buffer for these pins is AVCC0.
Note 2. Not present in the RX260.
Note 3. Not present in the RX261.

1.6.2 80-Pin LFQFP

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (1/4)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
1		P06*1					
2		P03*1					DA0
3		P04*1					
4	VCL						
5		PJ1	GTIOC6A/GTIOC6A#/ GTCPPO0				
6	MD/FINED	PG7					
7	XCIN	PH7					
8	XCOUT/ EXCIN	PH6					
9	RES#						
10	XTAL	P37				IRQ4	
11	VSS						
12	EXTAL	P36				IRQ2	
13	VCC						
14	UPSEL	P35				NMI	
15		P34	GTIOC3A/GTIOC3A#/ GTIU/TMCI3	SCK6		IRQ4	
16		P32	GTIOC1A/GTIOC7A/ GTIOC1A#/GTIOC7A#/ GTIW/TMO3/RTCOUT/ RTCIC2	TXD6/SMOSI6/SSDA6/ CTX0*2/ USB0_VBUSEN*2	TS0	IRQ2	
17		P31	GTIOC2B/GTIOC2B#/ GTOWLO/TMCI2/ RTCIC1	CTS1#/RTS1#/SS1#	TS1	IRQ1	
18		P30	GTIOC2A/GTIOC2A#/ GTOWUP/TMRI3/ RTCIC0	RXD1/SMISO1/SSCL1	TS2	IRQ0	
19		P27	GTIOC5B/GTIOC5B#/ TMCI3	SCK1	TS3		
20		P26	GTIOC5A/GTIOC5A#/ TMO1/LPTO	TXD1/SMOSI1/SSDA1/ USB0_VBUSEN*2	TS4		
21		P21	GTIOC2A/GTIOC4B/ GTIOC2A#/GTIOC4B#/ TMCI0	RXD000/SMISO000/ SSCL000/ USB0_EXICEN*2			
22		P20	GTIOC2B/GTIOC4A/ GTIOC2B#/GTIOC4A#/ TMRI0	TXD000/SMOSI000/ SSDA000/USB0_ID*2			
23		P17	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC6A/GTIOC6A#/ GTETRGD/GTCPPO0/ GTOUUP/TMO1	SCK1/MISOA/SDA0		IRQ7	
24		P16	GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTIOC6B/GTIOC6B#/ GTETRGC/GTOULO/ TMO2/RTCOUT	TXD1/SMOSI1/SSDA1/ MOSIA/SCL0/ USB0_VBUS*2/ USB0_VBUSEN*2/ USB0_OVRCURB*2		IRQ6	ADTRG0#

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (2/4)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
25		P15	GTIOC3B/GTIOC5B/ GTIOC3B#/GTIOC5B#/ GTETRGB/GTIV/TMCI2	RXD1/SMISO1/SSCL1/ CRX0*2	TS5	IRQ5	
26		P14	GTIOC6A/GTIOC7B/ GTIOC6A#/GTIOC7B#/ GTETRGA/GTCPPO0/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0*2/ USB0_OVRCURA*2	TS6	IRQ4	
27		P13	GTIOC3B/GTIOC7A/ GTIOC3B#/GTIOC7A#/ GTIV/TMO3	SDA0		IRQ3	
28		P12	TMCI1	SCL0		IRQ2	
29		PH3	GTIOC2B/GTIOC2B#/ TMCI0		TS7		
30		PH2*3	GTIOC1B*3/ GTIOC1B#*3/TMRI0*3	USB0_DM*2	TS8*3	IRQ1*3	
31		PH1*3	GTIOC0B*3/ GTIOC0B#*3/GTOULO*3/ TMO0*3	USB0_DP*2	TS9*3	IRQ0*3	
32		PH0	GTIOC0A/GTIOC0A#/ GTOUUP/CACREF		TS10		
33		P55	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ TMO3	CRX0*2	TS11		
34		P54	GTIOC2A/GTIOC2A#/ TMCI1	CTX0*2	TS12		
35	UB	PC7	GTIOC6A/GTIOC6A#/ GTETRGB/GTCPPO0/ TMO2/LPTO/CACREF	TXD008/TXDA008/ SMOSI008/SSDA008/ MISOA	TS13		
36		PC6	GTIOC6B/GTIOC6B#/ GTETRGA/TMCI2	RXD008/SMISO008/ SSCL008/MOSIA/ USB0_EXICEN*2	TS14		
37		PC5	GTIOC0A/GTIOC7A/ GTIOC0A#/GTIOC7A#/ GTETRGD/GTOUUP/ GTIW/TMRI2	SCK008/TXDB008/ RSPCKA/USB0_ID*2/ PMC0	TS15		
38		PC4	GTIOC0B/GTIOC3A/ GTIOC0B#/GTIOC3A#/ GTETRGC/GTIU/ GTOULO/TMCI1	SCK5/CTS008#/ RTS008#/SS008#/ DE008/SSLA0/PMC0	TSCAP		
39		PC3	GTIOC2B/GTIOC2B#/ GTETRGB	TXD5/SMOSI5/SSDA5/ PMC0	TS16		
40		PC2	GTIOC2A/GTIOC2A#/ GTETRGA/GTOWUP	RXD5/SMISO5/SSCL5/ SSLA3	TS17		
41		PB7/PC1*4	GTIOC0A/GTIOC7B/ GTIOC0A#/GTIOC7B#	TXD009/TXDA009/ SMOSI009/SSDA009	TS18		
42		PB6/PC0*4	GTIOC0B/GTIOC7A/ GTIOC0B#/GTIOC7A#	RXD009/SMISO009/ SSCL009	TS19		
43		PB5	GTIOC4B/GTIOC5A/ GTIOC4B#/GTIOC5A#/ GTIOC6B/GTIOC6B#/ TMRI1	SCK009/TXDB009/ USB0_VBUS*2	TS20		
44		PB4	GTIOC6A/GTIOC6A#	CTS009#/RTS009#/ SS009#/DE009	TS21		

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (3/4)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
45		PB3	GTIOC1A/GTIOC3A/ GTIOC1A#/GTIOC3A#/ GTIOC3B/GTIOC3B#/ GTETRGD/GTIU/ GTOVUP/TMO0/LPTO	SCK6/PMC0	TS22		
46		PB2	GTIOC3A/GTIOC3A#/ GTETRGC	CTS6#/RTS6#/SS6#	TS23		
47		PB1	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7A/GTIOC7A#/ GTOVLO/GTIW/ GTOWLO/TMCI0	TXD6/SMOSI6/SSDA6	TS24	IRQ4	CMPOB1
48	VCC						
49		PB0	GTIOC0B/GTIOC2A/ GTIOC0B#/GTIOC2A#/ GTOWUP	RXD6/SMISO6/SSCL6/ RSPCKA	TS25		
50	VSS						
51		PA6	GTIOC0B/GTIOC5A/ GTIOC0B#/GTIOC5A#/ GTETRGB/GTOULO/ TMCI3	CTS5#/RTS5#/SS5#/ MOSIA	TS26		
52		PA5	GTIOC4B/GTIOC4B#	RSPCKA	TS27		
53		PA4	GTIOC1B/GTIOC4A/ GTIOC1B#/GTIOC4A#/ GTETRGA/GTOVLO/ TMRI0	TXD5/SMOSI5/SSDA5/ SSLA0	TS28	IRQ5	CVREFB1
54		PA3	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7B/GTIOC7B#/ GTETRGB/GTETRGD/ GTOVLO/GTOWLO	RXD5/SMISO5/SSCL5	TS29	IRQ6	CMPB1
55		PA2		RXD5/SMISO5/SSCL5/ SSLA3	TS30		
56		PA1	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC3B/GTIOC3B#/ GTETRGC/GTIV/ GTOUUP	SCK5/SSLA2	TS31		
57		PA0	GTIOC0A/GTIOC1A/ GTIOC0A#/GTIOC1A#/ GTOVUP/CACREF	SSLA1	TS32		
58		PE5	GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#			IRQ5	AN021/ CMPOB0
59	CLKOUT	PE4	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ GTIOC4A/GTIOC4A#/ GTOVUP/GTOWLO		TS33		AN020/ CMPA2
60	CLKOUT	PE3	GTIOC2A/GTIOC4B/ GTIOC2A#/GTIOC4B#/ GTOWUP	CTS12#/RTS12#/SS12#	TS34		AN019
61		PE2	GTIOC1A/GTIOC1A#/ GTOVUP	RXD12/SMISO12/ SSCL12/RXDX12	TS35	IRQ7	AN018/ CVREFB0
62		PE1	GTIOC1B/GTIOC1B#/ GTOVLO	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12			AN017/ CMPB0

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (4/4)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
63		PE0		SCK12			AN016
64		PD2	GTIOC2B/GTIOC2B#	SCK6/CRX0*2		IRQ2	AN026
65		PD1	GTIOC2A/GTIOC2A#	RXD6/SMISO6/SSCL6/ CTX0*2		IRQ1	AN025
66		PD0		TXD6/SMOSI6/SSDA6		IRQ0	AN024
67		P47*1					AN007
68		P46*1					AN006
69		P45*1					AN005
70		P44*1					AN004
71		P43*1					AN003
72		P42*1					AN002
73		P41*1					AN001
74	VREFL0	PJ7*1					
75		P40*1					AN000
76	VREFH0	PJ6*1					
77	AVCC0						
78		P07*1					ADTRG0#
79	AVSS0						
80		P05*1					DA1

Note 1. The power source of the I/O buffer for these pins is AVCC0.
Note 2. Not present in the RX260.
Note 3. Not present in the RX261.
Note 4. PC0 and PC1 are valid only when the port switching function is selected.

1.6.3 64-Pin LFQFP

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP) (1/3)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
1		P03*1					DA0
2	VCL						
3	MD/FINED	PG7					
4	XCIN	PH7					
5	XCOUT/ EXCIN	PH6					
6	RES#						
7	XTAL	P37				IRQ4	
8	VSS						
9	EXTAL	P36				IRQ2	
10	VCC						
11	UPSEL	P35				NMI	
12		P32	GTIOC1A/GTIOC7A/ GTIOC1A#/GTIOC7A#/ GTIW/TMO3/RTCOUT/ RTCIC2	TXD6/SMOSI6/SSDA6/ CTX0*2/ USB0_VBUSEN*2	TS0	IRQ2	
13		P31	GTIOC2B/GTIOC2B#/ GTOWLO/TMCI2/ RTCIC1	CTS1#/RTS1#/SS1#	TS1	IRQ1	
14		P30	GTIOC2A/GTIOC2A#/ GTOWUP/TMRI3/ RTCIC0	RXD1/SMISO1/SSCL1	TS2	IRQ0	
15		P27	GTIOC5B/GTIOC5B#/ TMCI3	SCK1	TS3		
16		P26	GTIOC5A/GTIOC5A#/ TMO1/LPTO	TXD1/SMOSI1/SSDA1/ USB0_VBUSEN*2	TS4		
17		P17	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC6A/GTIOC6A#/ GTETRGD/GTCPPO0/ GTOUUP/TMO1	SCK1/MISOA/SDA0		IRQ7	
18		P16	GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTIOC6B/GTIOC6B#/ GTETRGC/GTOULO/ TMO2/RTCOUT	TXD1/SMOSI1/SSDA1/ MOSIA/SCL0/ USB0_VBUS*2/ USB0_VBUSEN*2/ USB0_OVRCURB*2		IRQ6	ADTRG0#
19		P15	GTIOC3B/GTIOC5B/ GTIOC3B#/GTIOC5B#/ GTETRGB/GTIV/TMCI2	RXD1/SMISO1/SSCL1/ CRX0*2	TS5	IRQ5	
20		P14	GTIOC6A/GTIOC7B/ GTIOC6A#/GTIOC7B#/ GTETRGA/GTCPPO0/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0*2/ USB0_OVRCURA*2	TS6	IRQ4	
21		PH3	GTIOC2B/GTIOC2B#/ TMCI0		TS7		
22		PH2*3	GTIOC1B*3/ GTIOC1B#*3/ TMRI0*3	USB0_DM*2	TS8*3	IRQ1*3	

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP) (2/3)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
23		PH1*3	GTIOC0B*3/ GTIOC0B#*3/GTOULO*3/ TMO0*3	USB0_DP*2	TS9*3	IRQ0*3	
24		PH0	GTIOC0A/GTIOC0A#/ GTOUUP/CACREF		TS10		
25		P55	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ TMO3	CRX0*2	TS11		
26		P54	GTIOC2A/GTIOC2A#/ TMCI1	CTX0*2	TS12		
27	UB	PC7	GTIOC6A/GTIOC6A#/ GTETRGB/GTCPPO0/ TMO2/LPTO/CACREF	TXD008/TXDA008/ SMOSI008/SSDA008/ MISOA	TS13		
28		PC6	GTIOC6B/GTIOC6B#/ GTETRGA/TMCI2	RXD008/SMISO008/ SSCL008/MOSIA/ USB0_EXICEN*2	TS14		
29		PC5	GTIOC0A/GTIOC7A/ GTIOC0A#/GTIOC7A#/ GTETRGD/GTOUUP/ GTIW/TMRI2	SCK008/TXDB008/ RSPCKA/USB0_ID*2/ PMC0	TS15		
30		PC4	GTIOC0B/GTIOC3A/ GTIOC0B#/GTIOC3A#/ GTETRGC/GTIU/ GTOULO/TMCI1	SCK5/CTS008#/ RTS008#/SS008#/ DE008/SSLA0/PMC0	TSCAP		
31		PC3	GTIOC2B/GTIOC2B#/ GTETRGB	TXD5/SMOSI5/SSDA5/ PMC0	TS16		
32		PC2	GTIOC2A/GTIOC2A#/ GTETRGA/GTOWUP	RXD5/SMISO5/SSCL5/ SSLA3	TS17		
33		PB7/PC1*4	GTIOC0A/GTIOC7B/ GTIOC0A#/GTIOC7B#	TXD009/TXDA009/ SMOSI009/SSDA009	TS18		
34		PB6/PC0*4	GTIOC0B/GTIOC7A/ GTIOC0B#/GTIOC7A#	RXD009/SMISO009/ SSCL009	TS19		
35		PB5	GTIOC4B/GTIOC5A/ GTIOC4B#/GTIOC5A#/ GTIOC6B/GTIOC6B#/ TMRI1	SCK009/TXDB009/ USB0_VBUS*2	TS20		
36		PB3	GTIOC1A/GTIOC3A/ GTIOC1A#/GTIOC3A#/ GTIOC3B/GTIOC3B#/ GTETRGD/GTIU/ GTOVUP/TMO0/LPTO	SCK6/PMC0	TS22		
37		PB1	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7A/GTIOC7A#/ GTOVLO/GTIW/ GTOWLO/TMCI0	TXD6/SMOSI6/SSDA6	TS24	IRQ4	CMPOB1
38	VCC						
39		PB0	GTIOC0B/GTIOC2A/ GTIOC0B#/GTIOC2A#/ GTOWUP	RXD6/SMISO6/SSCL6/ RSPCKA	TS25		
40	VSS						
41		PA6	GTIOC0B/GTIOC5A/ GTIOC0B#/GTIOC5A#/ GTETRGB/GTOULO/ TMCI3	CTS5#/RTS5#/SS5#/ MOSIA	TS26		

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP) (3/3)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
42		PA4	GTIOC1B/GTIOC4A/ GTIOC1B#/GTIOC4A#/ GTETRGA/GTOVLO/ TMRI0	TXD5/SMOSI5/SSDA5/ SSLA0	TS28	IRQ5	CVREFB1
43		PA3	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7B/GTIOC7B#/ GTETRGB/GTETRGD/ GTOVLO/GTOWLO	RXD5/SMISO5/SSCL5	TS29	IRQ6	CMPB1
44		PA1	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC3B/GTIOC3B#/ GTETRGC/GTIV/ GTOUUP	SCK5/SSLA2	TS31		
45		PA0	GTIOC0A/GTIOC1A/ GTIOC0A#/GTIOC1A#/ GTOVUP/CACREF	SSLA1	TS32		
46		PE5	GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#			IRQ5	AN021/ CMPOB0
47	CLKOUT	PE4	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ GTIOC4A/GTIOC4A#/ GTOVUP/GTOWLO		TS33		AN020/ CMPA2
48	CLKOUT	PE3	GTIOC2A/GTIOC4B/ GTIOC2A#/GTIOC4B#/ GTOWUP	CTS12#/RTS12#/SS12#	TS34		AN019
49		PE2	GTIOC1A/GTIOC1A#/ GTOVUP	RXD12/SMISO12/ SSCL12/RXDX12	TS35	IRQ7	AN018/ CVREFB0
50		PE1	GTIOC1B/GTIOC1B#/ GTOVLO	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12			AN017/ CMPB0
51		PE0		SCK12			AN016
52		P47*1					AN007
53		P46*1					AN006
54		P45*1					AN005
55		P44*1					AN004
56		P43*1					AN003
57		P42*1					AN002
58		P41*1					AN001
59	VREFL0	PJ7*1					
60		P40*1					AN000
61	VREFH0	PJ6*1					
62	AVCC0						
63		P05*1					DA1
64	AVSS0						

Note 1. The power source of the I/O buffer for these pins is AVCC0.
Note 2. Not present in the RX260.
Note 3. Not present in the RX261.
Note 4. PC0 and PC1 are valid only when the port switching function is selected.

1.6.4 48-Pin LFQFP, 48-Pin HWQFN

Table 1.8 List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (1/3)

Pin Number 48-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
1	VCL						
2	MD/FINED	PG7					
3	RES#						
4	XTAL	P37				IRQ4	
5	VSS						
6	EXTAL	P36				IRQ2	
7	VCC						
8	UPSEL	P35				NMI	
9		P31	GTIOC2B/GTIOC2B#/ GTOWLO/TMCI2	CTS1#/RTS1#/SS1#	TS1	IRQ1	
10		P30	GTIOC2A/GTIOC2A#/ GTOWUP/TMRI3	RXD1/SMISO1/SSCL1	TS2	IRQ0	
11		P27	GTIOC5B/GTIOC5B#/ TMCI3	SCK1	TS3		
12		P26	GTIOC5A/GTIOC5A#/ TMO1/LPTO	TXD1/SMOSI1/SSDA1/ USB0_VBUSEN*1	TS4		
13		P17	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC6A/GTIOC6A#/ GTETRGD/GTCPPO0/ GTOUUP/TMO1	SCK1/MISOA/SDA0		IRQ7	
14		P16	GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTIOC6B/GTIOC6B#/ GTETRGC/GTOULO/ TMO2	TXD1/SMOSI1/SSDA1/ MOSIA/SCL0/ USB0_VBUS*1/ USB0_VBUSEN*1/ USB0_OVRCURB*1		IRQ6	ADTRG0#
15		P15	GTIOC3B/GTIOC5B/ GTIOC3B#/GTIOC5B#/ GTETRGB/GTIV/TMCI2	RXD1/SMISO1/SSCL1/ CRX0*1	TS5	IRQ5	
16		P14	GTIOC6A/GTIOC7B/ GTIOC6A#/GTIOC7B#/ GTETRGA/GTCPPO0/ TMRI2	CTS1#/RTS1#/SS1#/ CTX0*1/ USB0_OVRCURA*1	TS6	IRQ4	
17		PH3	GTIOC2B/GTIOC2B#/ TMCI0		TS7		
18		PH2*2	GTIOC1B*2/ GTIOC1B#*2/TMRI0*2	USB0_DM*1	TS8*2	IRQ1*2	
19		PH1*2	GTIOC0B*2/ GTIOC0B#*2/GTOULO*2/ TMO0*2	USB0_DP*1	TS9*2	IRQ0*2	
20		PH0	GTIOC0A/GTIOC0A#/ GTOUUP/CACREF		TS10		
21	UB	PC7	GTIOC6A/GTIOC6A#/ GTETRGB/GTCPPO0/ TMO2/LPTO/CACREF	TXD008/TXDA008/ SMOSI008/SSDA008/ MISOA	TS13		
22		PC6	GTIOC6B/GTIOC6B#/ GTETRGA/TMCI2	RXD008/SMISO008/ SSCL008/MOSIA/ USB0_EXICEN*1	TS14		

Table 1.8 List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (2/3)

Pin Number 48-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
23		PC5	GTIOC0A/GTIOC7A/ GTIOC0A#/GTIOC7A#/ GTETRGD/GTOUUP/ GTIW/TMRI2	SCK008/TXDB008/ RSPCKA/USB0_ID*1/ PMC0	TS15		
24		PC4	GTIOC0B/GTIOC3A/ GTIOC0B#/GTIOC3A#/ GTETRGC/GTIU/ GTOULO/TMCI1	SCK5/CTS008#/ RTS008#/SS008#/ DE008/SSLA0/PMC0	TSCAP		
25		PB5/PC3*3	GTIOC4B/GTIOC5A/ GTIOC4B#/GTIOC5A#/ GTIOC6B/GTIOC6B#/ TMRI1	USB0_VBUS*1	TS20		
26		PB3/PC2*3	GTIOC1A/GTIOC3A/ GTIOC1A#/GTIOC3A#/ GTIOC3B/GTIOC3B#/ GTETRGD/GTIU/ GTOVUP/TMO0/LPTO	SCK6/PMC0	TS22		
27		PB1/PC1*3	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7A/GTIOC7A#/ GTOVLO/GTIW/ GTOWLO/TMCI0	TXD6/SMOSI6/SSDA6	TS24	IRQ4	CMPOB1
28	VCC						
29		PB0/PC0*3	GTIOC0B/GTIOC2A/ GTIOC0B#/GTIOC2A#/ GTOWUP	RXD6/SMISO6/SSCL6/ RSPCKA	TS25		
30	VSS						
31		PA6	GTIOC0B/GTIOC5A/ GTIOC0B#/GTIOC5A#/ GTETRGB/GTOULO/ TMCI3	CTS5#/RTS5#/SS5#/ MOSIA	TS26		
32		PA4	GTIOC1B/GTIOC4A/ GTIOC1B#/GTIOC4A#/ GTETRGA/GTOVLO/ TMRI0	TXD5/SMOSI5/SSDA5/ SSLA0	TS28	IRQ5	CVREFB1
33		PA3	GTIOC1B/GTIOC2B/ GTIOC1B#/GTIOC2B#/ GTIOC7B/GTIOC7B#/ GTETRGB/GTETRGD/ GTOVLO/GTOWLO	RXD5/SMISO5/SSCL5	TS29	IRQ6	CMPB1
34		PA1	GTIOC0A/GTIOC0B/ GTIOC0A#/GTIOC0B#/ GTIOC3B/GTIOC3B#/ GTETRGC/GTIV/ GTOUUP	SCK5/SSLA2	TS31		
35	CLKOUT	PE4	GTIOC1A/GTIOC2B/ GTIOC1A#/GTIOC2B#/ GTIOC4A/GTIOC4A#/ GTOVUP/GTOWLO		TS33		AN020/ CMPA2
36	CLKOUT	PE3	GTIOC2A/GTIOC4B/ GTIOC2A#/GTIOC4B#/ GTOWUP	CTS12#/RTS12#	TS34		AN019
37		PE2	GTIOC1A/GTIOC1A#/ GTOVUP	RXD12/SSCL12/RXDX12	TS35	IRQ7	AN018/ CVREFB0

Table 1.8 List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (3/3)

Pin Number 48-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (GPTW, POEG, TMR, LPT, CAC)	Communications (SCI, RSCI, RSPI, RIIC, CANFD, USB, REMC)	Touch sensing	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPB)
38		PE1	GTIOC1B/GTIOC1B#/ GTOVLO	TXD12/SSDA12/ TXDX12/SIOX12			AN017/ CMPB0
39		P47*4					AN007
40		P46*4					AN006
41		P45*4					AN005
42		P42*4					AN002
43		P41*4					AN001
44	VREFL0	PJ7*4					
45		P40*4					AN000
46	VREFH0	PJ6*4					
47	AVCC0						
48	AVSS0						

Note 1. Not present in the RX260.
Note 2. Not present in the RX261.
Note 3. PC0 to PC3 are valid only when the port switching function is selected.
Note 4. The power source of the I/O buffer for these pins is AVCC0.

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)
- 111 instructions
  Standard provided instructions: 111
    Basic instructions: 77
    Single-precision floating point instructions: 11
    DSP instructions: 23
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table and interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

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

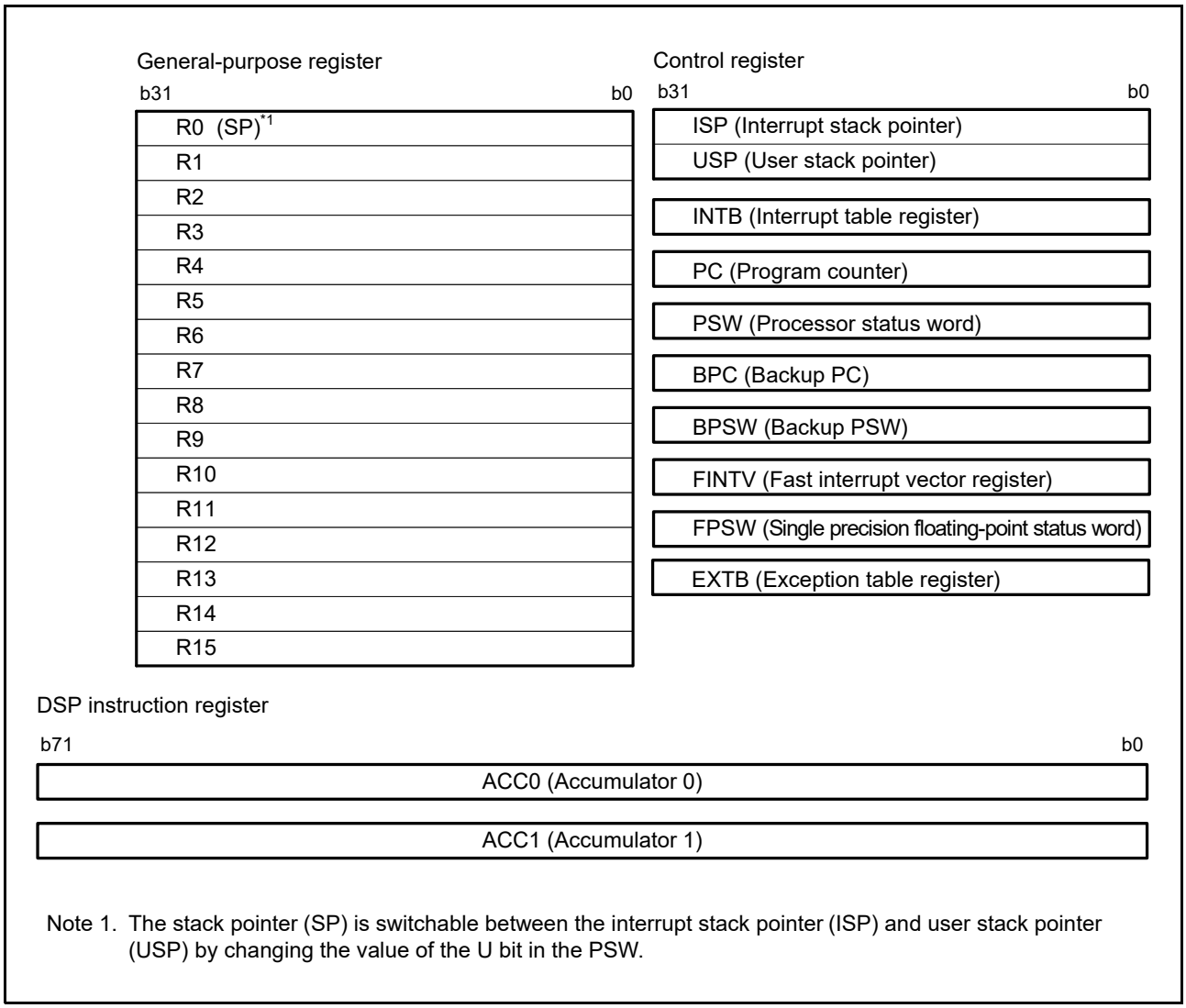


Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

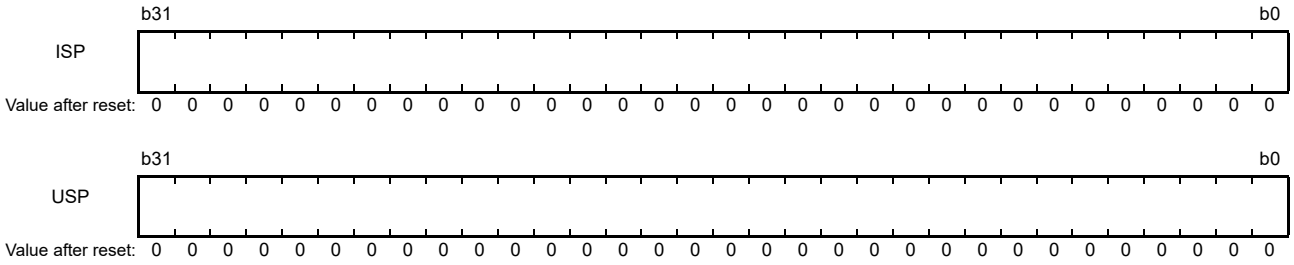
This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer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

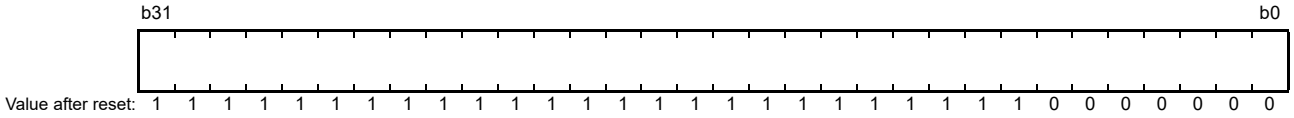
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Single-precision floating-point status word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

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



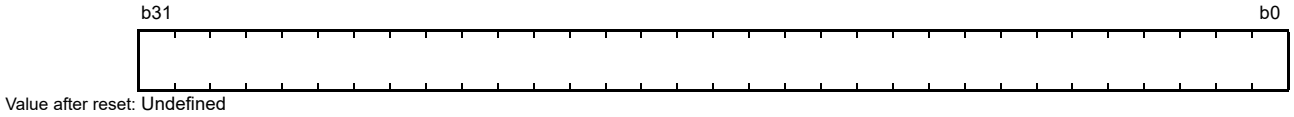
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2.2 Exception Table Register (EXTB)



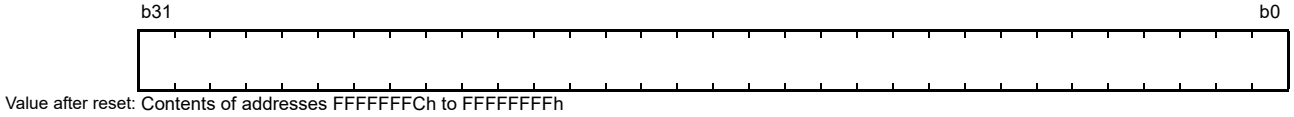
The exception table register (EXTB) specifies the address where the exception vector table starts.

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.

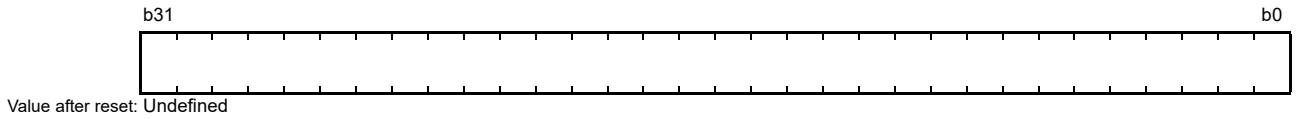
PM Bit (Processor Mode Select)

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)

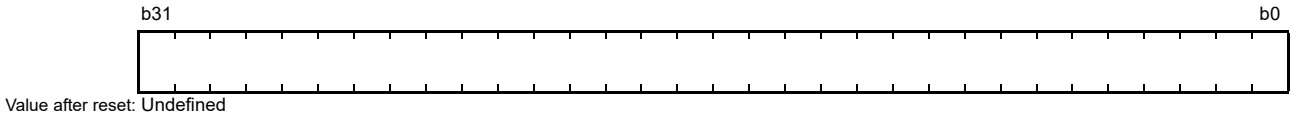
The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the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



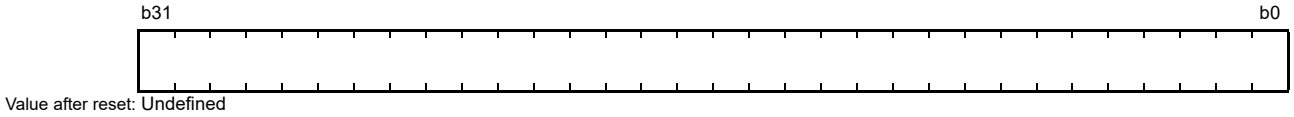
The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

2.2.2.7 Backup PSW (BPSW)



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

2.2.2.8 Fast Interrupt Vector Register (FINTV)



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

2.2.2.9 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)

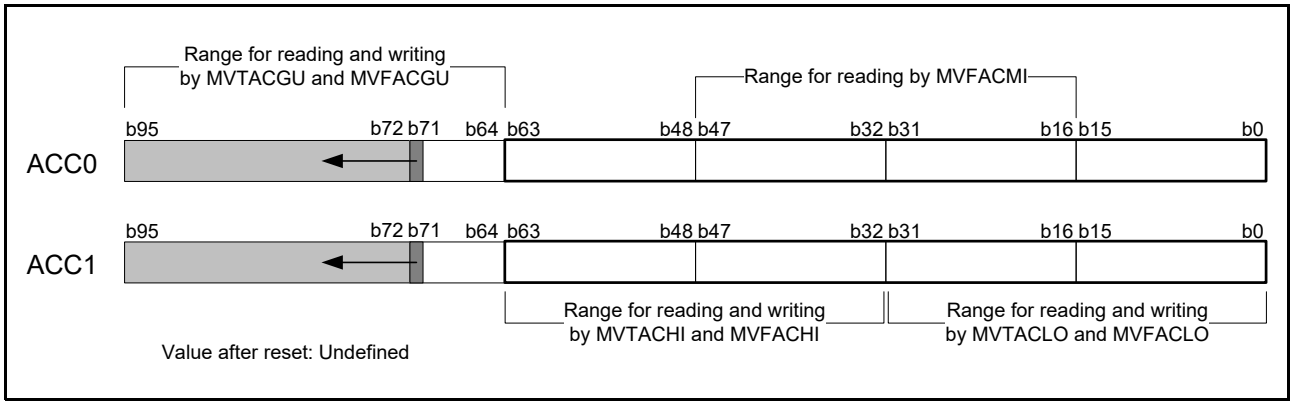
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

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, and WAIT instructions.

2.3.4 Switching Between Processor Modes

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

(1) Switching from user mode to supervisor mode

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

(2) Switching from supervisor mode to user mode

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

2.4 Data Types

The 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.

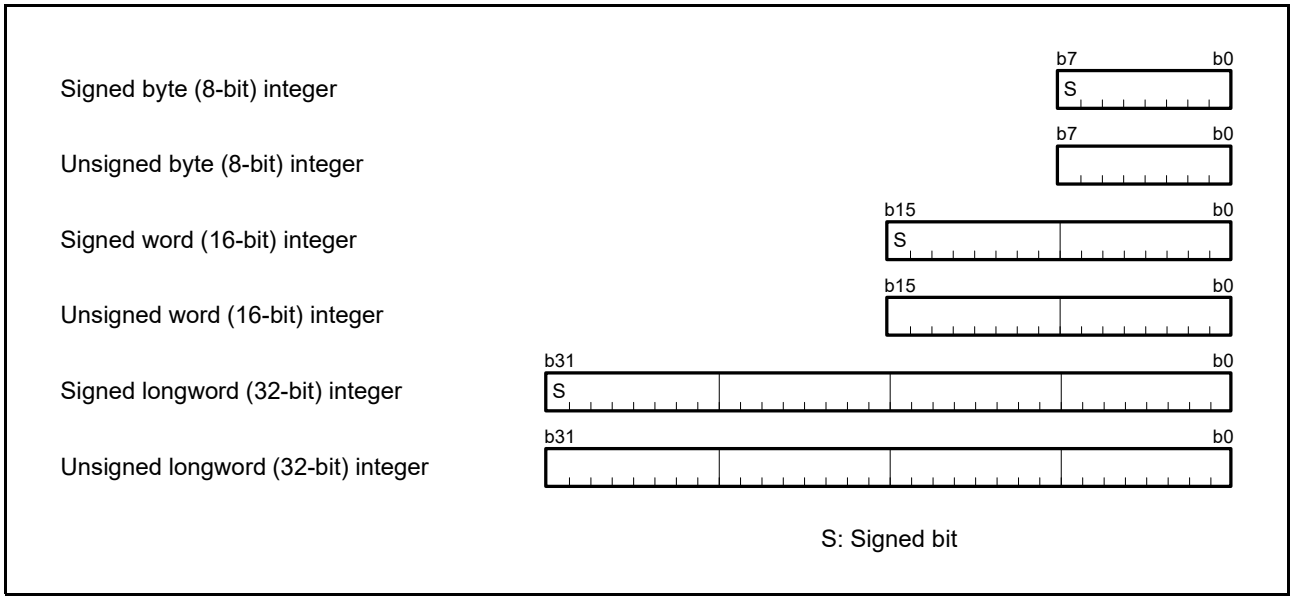


Figure 2.2 Integer

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.

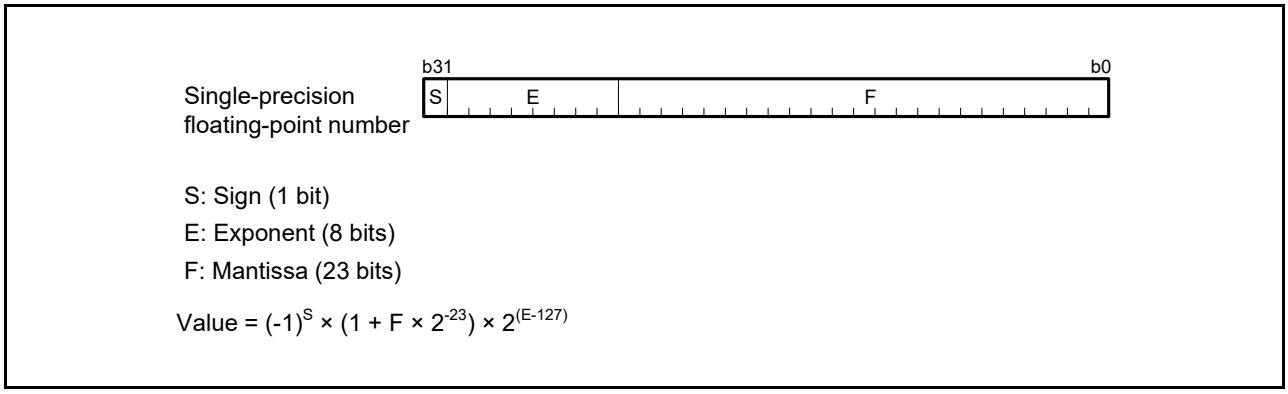


Figure 2.3 Single-Precision Floating-Point Number

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.

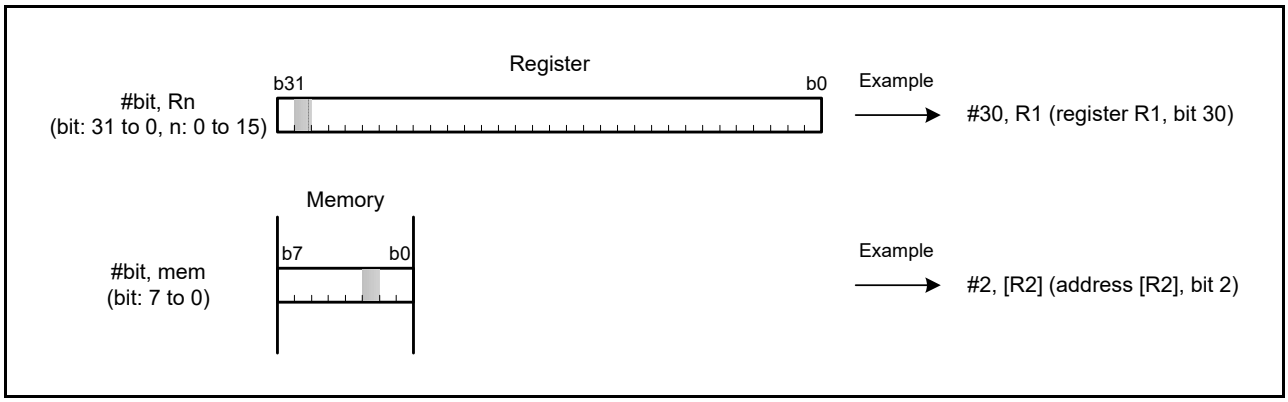


Figure 2.4 Bit

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.

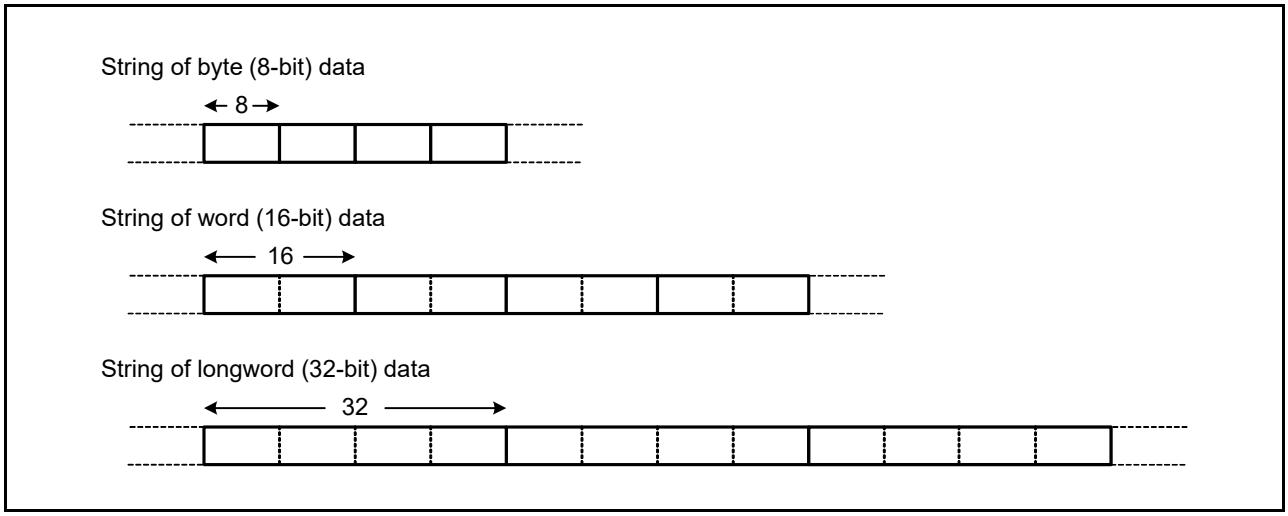


Figure 2.5 String

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.

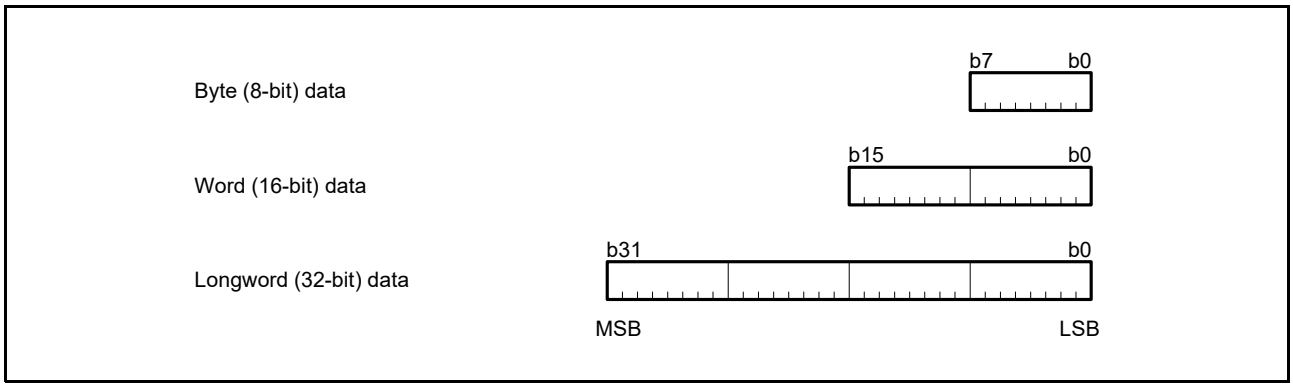


Figure 2.6 Data Arrangement in Registers

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.

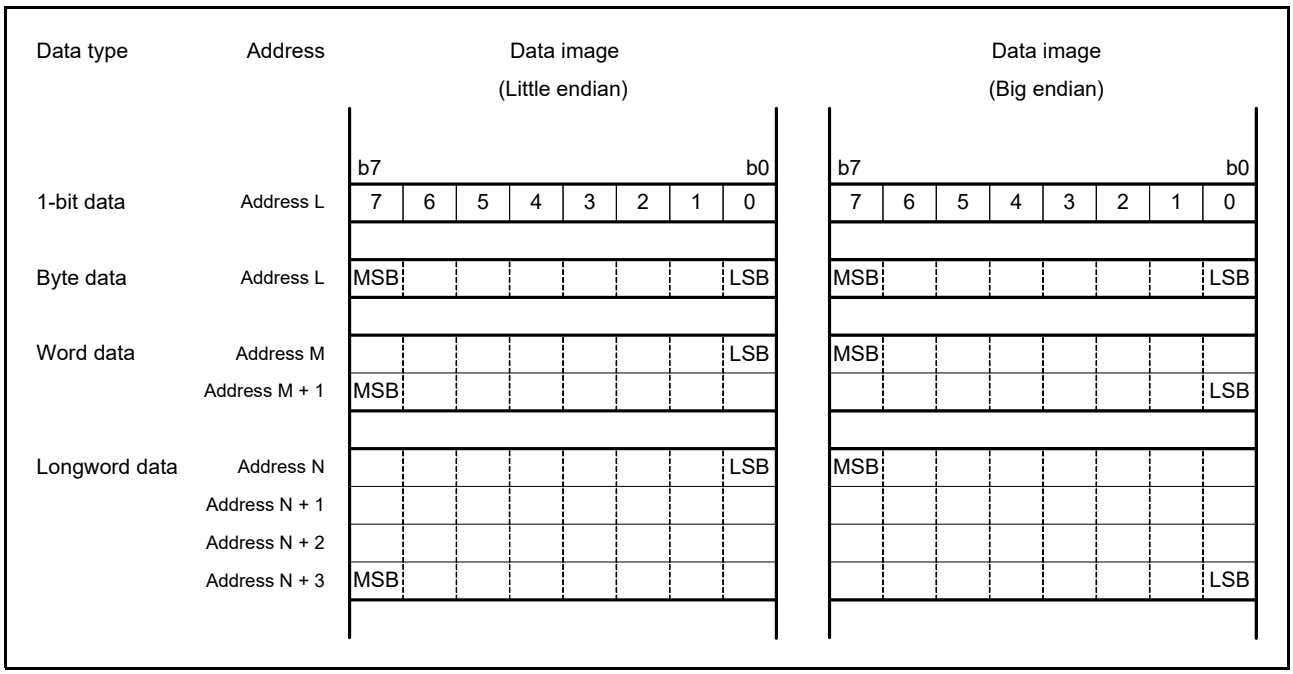


Figure 2.7 Data Arrangement in Memory

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.

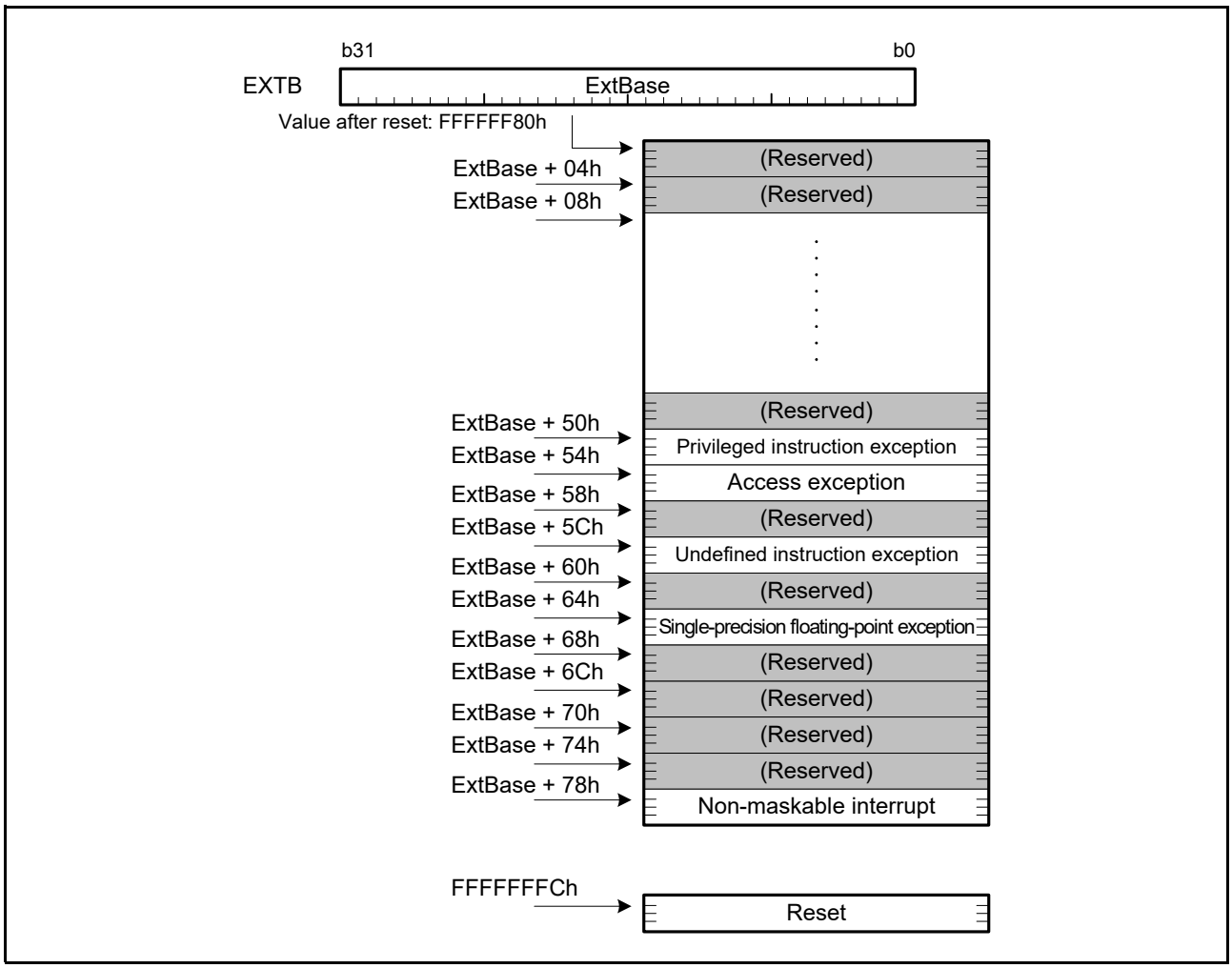


Figure 2.8 Exception Vector Table

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.

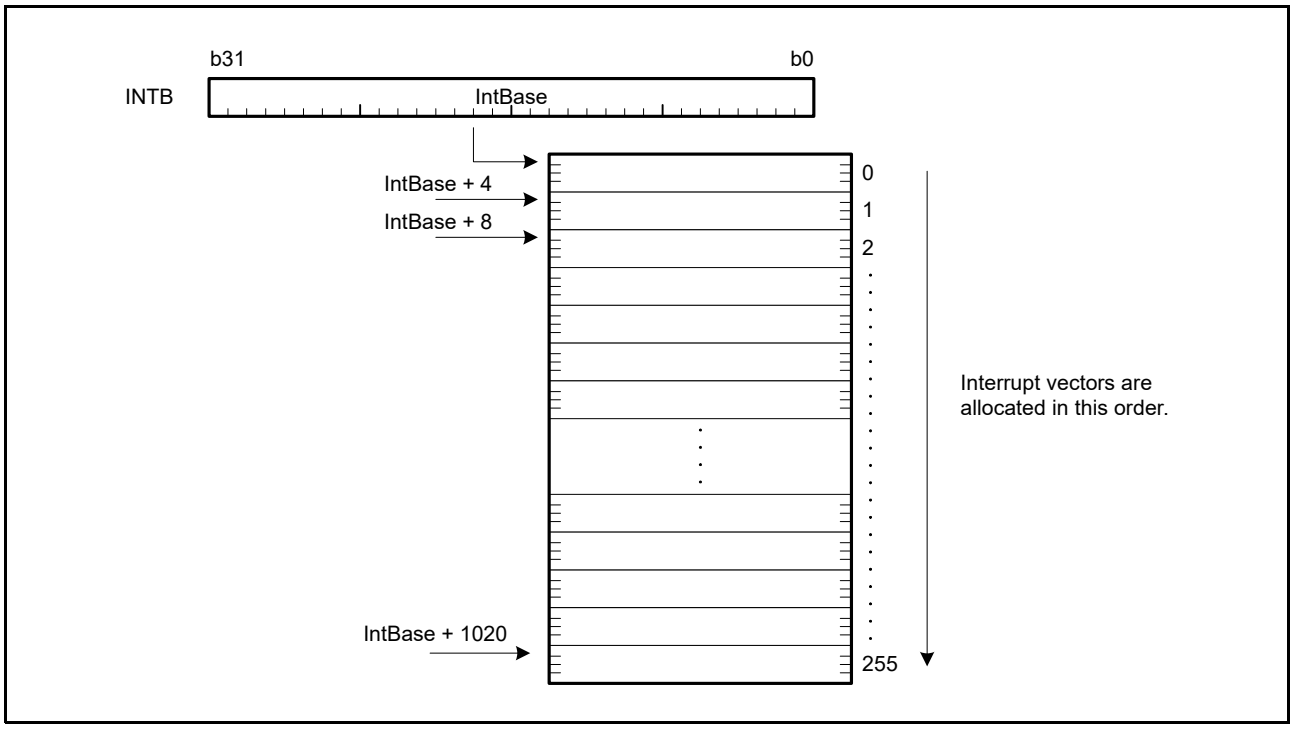


Figure 2.9 Interrupt Vector Table

2.7 Operation of Instructions

2.7.1 Restrictions on RMPA and String-Manipulation Instructions

2.7.1.1 Transfer Size and Data Prefetching

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

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

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

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

2.7.1.2 Access to the External Space

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

2.7.1.3 Access to I/O Registers

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

2.8 Numbers of Cycles

2.8.1 Instruction and Numbers of Cycles

Table 2.13 to Table 2.20 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

2.8.2 Numbers of Cycles for Response to Interrupts

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

Table 2.21 Numbers of Cycles for Response to Interrupts

Type of Interrupt Request/Details of Processing	Fast Interrupt	Other Interrupts
ICU Judgment of priority order	2 cycles	
CPU 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 2.21 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 2.20.
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.

3. Operating Modes

3.1 Operating Mode Types and Selection

There are two types of operating-mode selection: one is selection by the levels on pins on release from a reset (RES# pin reset or power-on reset), and the other is selection by software after release from a reset.

Table 3.1 shows the relationship between levels on the mode setting pins (MD, UB) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, refer to section 3.3, Details of Operating Modes.

Table 3.1 Selection of Operating Modes by the Mode Setting Pin

Mode Pin		Operating Mode
MD*1	UB*2	
High	—	Single-chip mode
Low	High	Boot mode (USB interface)
	Low	Boot mode (SCI interface)
Low→High*3	Low	Boot mode (FINE interface)

Note 1. Transition between operating modes is in progress during the wait time after RES# cancellation or power-on reset time. Accordingly, do not change the input level on the MD pin during these times. For details on the wait time after RES# cancellation and power-on reset time, see section 47, Electrical Characteristics. In boot mode, an input pull-up resistor is in the enabled state during the period from the resetting in progress to the completion of the operating mode transition. In single-chip mode, an input pull-up resistor is in the enabled state during the period from the resetting in progress to the setting of disabling the input pull-up resistor in the pull-up control register.

Note 2. The UB pin function is multiplexed on the same pin as the PC7 general-purpose port pin function and input and output pin functions for peripheral modules, so the pin can also be used in those roles.

Note 3. Place the low level on this pin when the reset is released, and change it to the high level within 20 to 100 msec.

The endian is selectable in single-chip mode. Endian is set by the MDE.MDE[2:0] bits in the option-setting memory. For the correspondence between the setting and endian, see Table 3.2.

Table 3.2 Endian Setting in Single-Chip Mode

MDE.MDE[2:0] Bits	Endian
000b	Big endian
111b	Little endian

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): SYSTEM.MDMONR 0008 0000h

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

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

Note 1. The initial value of the MD bit depends on the setting of the mode pin (MD). For the setting PORTG.PMR.b7, if the PG7/MD pin is set to the general purpose I/O port pin, this bit is read as 0 regardless of the state of the PG7/MD pin.

3.2.2 System Control Register 1 (SYSCR1)

Address(es): SYSTEM.SYSCR1 0008 0008h

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

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

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

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 (RAM disabled) to 1 (RAM enabled), 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. To retain the value in the RAM, keep the specified RAM standby voltage (VRAM). For details, refer to section 47, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, all I/O ports can be used as general input/output ports, peripheral function input/output, or interrupt input pins.
The chip starts up in single-chip mode if the high level is on the MD pin on release from the reset state.

3.3.2 Boot Mode (USB Interface)

In this mode, the flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM and E2 data flash memory) can be modified from outside the MCU by using the USB. For details, refer to section 46, Flash Memory (FLASH).
The chip starts up in boot mode (USB interface) when the MD pin is set to the low level and the UB pin is set to the high level at the time of release from the reset state. For details on boot mode (USB interface), refer to section 46.8.1, Boot Mode (USB Interface).

3.3.3 Boot Mode (SCI Interface)

In this mode, the flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM and E2 data flash memory) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, refer to section 46, 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. For details on boot mode (SCI interface), refer to section 46.8.2, Boot Mode (SCI Interface).

3.3.4 Boot Mode (FINE Interface)

In this mode, the flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM and E2 data flash memory) can be modified from outside the MCU by using the FINE. For details, refer to section 46, 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 msec. For details on boot mode (FINE interface), refer to section 46.8.3, Boot Mode (FINE Interface).

3.4 Transitions of Operating Modes

3.4.1 Mode Setting Pin Levels and Operating Mode Transitions

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

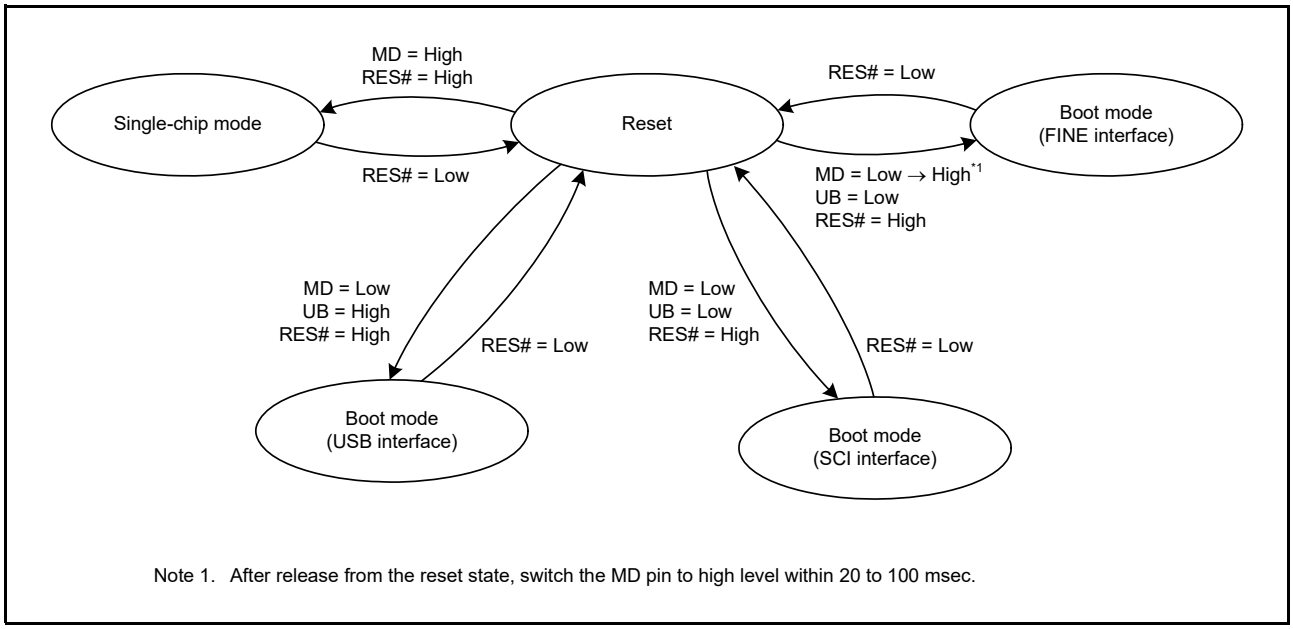
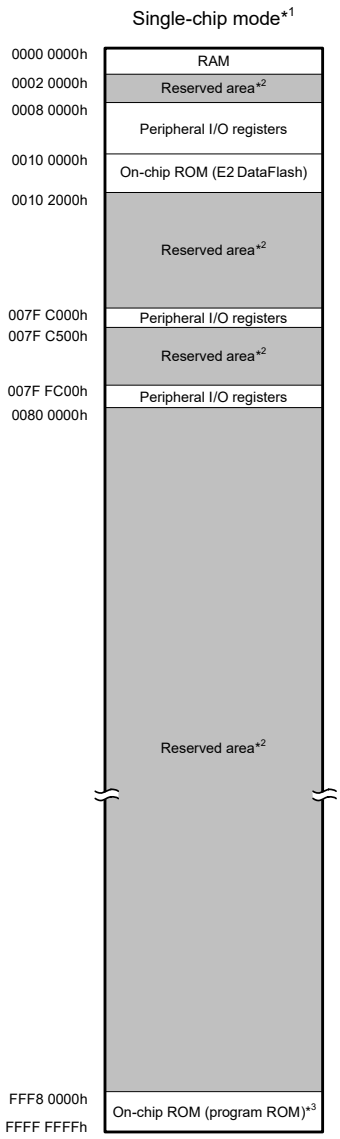


Figure 3.1 Mode Setting Pin Levels and Operating Modes

4. Address Space

4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
Figure 4.1 shows the memory maps.



Note 1. The address space in boot mode is the same as the address space in single-chip mode.
Note 2. Reserved areas should not be accessed.
Note 3. The capacity of ROM differs depending on the products.

ROM (bytes)		RAM (bytes)		E2 DataFlash (bytes)	
Capacity	**Address**	**Capacity**	**Address**	**Capacity**	**Address**
512 Kbytes	FFF8 0000h to FFFF FFFFh	128 Kbytes	0000 0000h to 0001 FFFFh	8 Kbytes	0010 0000h to 0010 1FFFh
384 Kbytes	FFFA 0000h to FFFF FFFFh				
256 Kbytes	FFFC 0000h to FFFF FFFFh				

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

Figure 4.1 Memory Map in Each Operating Mode

5. I/O Registers

This section provides information on the on-chip I/O register addresses and bit configuration. The information is given as shown below. Notes on writing to registers are also given below.

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

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- Numbers of cycles for access indicate numbers of cycles of the given base clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.

(2) Notes on writing to I/O registers

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

[Examples of cases requiring special care]

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

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

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

[Instruction examples]

- Byte-size I/O registers

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

- Word-size I/O registers

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

- Longword-size I/O registers

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

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

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

For numbers of clock cycles for access to I/O registers, see 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 3, and 6

The number of bus cycles of internal peripheral buses 1 to 3, and 6 differs according to the register to be accessed.
When the registers for peripheral functions connected to internal peripheral buses 2, 3, and 6 (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) 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 cycles 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.

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 (DTC).

(4) 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) Notes on Sleep Mode and Mode Transitions

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

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 32)

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 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 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 002Ch	SYSTEM	PLL2 Control Register	PLL2CR	16	16	3 ICLK		section 9
0008 002Eh	SYSTEM	PLL2 Control Register 2	PLL2CR2	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 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK		section 9
0008 003Eh	SYSTEM	CLKOUT Output Control Register	CKOCR	16	16	3 ICLK		section 9
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK		section 9
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK		section 9
0008 0043h	SYSTEM	Low-Speed On-Chip Oscillator Forced Oscillation Control Register	LOFCR	8	8	3 ICLK		section 9
0008 0061h	SYSTEM	Low-Speed On-Chip Oscillator Trimming Register 2	LOCOTRR2	8	8	3 ICLK		section 9
0008 0064h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Trimming Register	ILOCOTRR	8	8	3 ICLK		section 9
0008 0068h	SYSTEM	High-Speed On-Chip Oscillator Trimming Register 0	HOCOTRR0	8	8	3 ICLK		section 9
0008 006Eh	SYSTEM	CANFD Clock Frequency Division Control Register	CANFDCKDIVCR	8	8	3 ICLK		section 9
0008 0074h	SYSTEM	USB Clock Control Register	USBCKCR	8	8	3 ICLK		section 9
0008 0076h	SYSTEM	CANFD Clock Control Register	CANFDCKCR	8	8	3 ICLK		section 9
0008 0083h	SYSTEM	Sub-Clock Oscillator Mode Control Register	SOMCR	8	8	3 ICLK		section 9
0008 009Fh	SYSTEM	RAM Power Saving Control Register	RPSCR	8	8	3 ICLK		section 11
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK		section 11
0008 00A1h	SYSTEM	Sleep Mode Return Clock Source Switching Register	RSTCKCR	8	8	3 ICLK		section 11
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK		section 9
0008 00AAh	SYSTEM	Sub Operating Power Control Register	SOPCCR	8	8	3 ICLK		section 11
0008 00ACh	SYSTEM	Snooze Control Register 2	SNZCR2	16	16	3 ICLK		section 11
0008 00AEh	SYSTEM	Snooze Control Register	SNZCR	16	16	3 ICLK		section 11
0008 00B0h	LPT	Low-Power Timer Control Register 1	LPTCR1	8	8	3 ICLK		section 27
0008 00B1h	LPT	Low-Power Timer Control Register 2	LPTCR2	8	8	3 ICLK		section 27
0008 00B2h	LPT	Low-Power Timer Control Register 3	LPTCR3	8	8	3 ICLK		section 27
0008 00B4h	LPT	Low-Power Timer Period Setting Register	LPTPRD	16	16	3 ICLK		section 27
0008 00B8h	LPT	Low-Power Timer Compare Register 0	LPCMR0	16	16	3 ICLK		section 27
0008 00BAh	LPT	Low-Power Timer Compare Register 1	LPCMR1	16	16	3 ICLK		section 27
0008 00BCh	LPT	Low-Power Timer Standby Wakeup Enable Register	LPWUCR	16	16	3 ICLK		section 27

Table 5.1 List of I/O Registers (Address Order) (2 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
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 45
0008 1201h	RAM	RAM Error Status Register	RAMSTS	8	8	2 ICLK		section 45
0008 1204h	RAM	RAM Protection Register	RAMPRCR	8	8	2 ICLK		section 45
0008 1208h	RAM	RAM Error Address Capture Register	RAMECAD	32	32	2 ICLK		section 45
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
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

Table 5.1 List of I/O Registers (Address Order) (3 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
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 2200h	DMAC	DMAC Module Start Register	DMAST	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 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 711Bh to 0008 71FFh	ICU	DTC Transfer Request Enable Register 027 to DTC Transfer Request Enable Register 255	DTCER027 to DTCER255	8	8	2 ICLK		section 14

Table 5.1 List of I/O Registers (Address Order) (4 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
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
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	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 7500h to 0008 7507h	ICU	IRQ Control Register 0 to IRQ Control Register 7	IRQCR0 to IRQCR7	8	8	2 ICLK		section 14
0008 7510h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		section 14
0008 7514h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	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 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	2 ICLK	section 25
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2 or 3 PCLKB	2 ICLK	section 28
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2 or 3 PCLKB	2 ICLK	section 28
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2 or 3 PCLKB	2 ICLK	section 28
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2 or 3 PCLKB	2 ICLK	section 28
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2 or 3 PCLKB	2 ICLK	section 29
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2 or 3 PCLKB	2 ICLK	section 29
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2 or 3 PCLKB	2 ICLK	section 29
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2 or 3 PCLKB	2 ICLK	section 29
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2 or 3 PCLKB	2 ICLK	section 29
0008 80C0h	DA	D/A Data Register 0	DADR0	16	16	2 or 3 PCLKB	2 ICLK	section 41
0008 80C2h	DA	D/A Data Register 1	DADR1	16	16	2 or 3 PCLKB	2 ICLK	section 41
0008 80C4h	DA	D/A Control Register	DACR	8	8	2 or 3 PCLKB	2 ICLK	section 41
0008 80C5h	DA	Data Register Format Select Register	DADPR	8	8	2 or 3 PCLKB	2 ICLK	section 41
0008 80C6h	DA	D/A A/D Synchronous Start Control Register	DAADSCR	8	8	2 or 3 PCLKB	2 ICLK	section 41
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (5 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8204h	TMR01	Time Constant Register A	TCORA	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8206h	TMR01	Time Constant Register B	TCORB	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8208h	TMR01	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 8209h	TMR1	Timer Counter	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 820Ah	TMR01	Timer Counter Control Register	TCCR	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 820Ch	TMR0	Timer Counter Start Register	TCSTR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8214h	TMR23	Time Constant Register A	TCORA	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8216h	TMR23	Time Constant Register B	TCORB	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8218h	TMR23	Timer Counter	TCNT	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 8219h	TMR3	Timer Counter	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 821Ah	TMR23	Timer Counter Control Register	TCCR	16	16	2 or 3 PCLKB	2 ICLK	section 24
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 821Ch	TMR2	Timer Counter Start Register	TCSTR	8	8	2 or 3 PCLKB	2 ICLK	section 24
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2 or 3 PCLKB	2 ICLK	section 36
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2 or 3 PCLKB	2 ICLK	section 36
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2 or 3 PCLKB	2 ICLK	section 36
0008 8300h	RIIC0	I^2C-bus Control Register 1	ICCR1	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8301h	RIIC0	I^2C-bus Control Register 2	ICCR2	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8302h	RIIC0	I^2C-bus Mode Register 1	ICMR1	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8303h	RIIC0	I^2C-bus Mode Register 2	ICMR2	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8304h	RIIC0	I^2C-bus Mode Register 3	ICMR3	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8305h	RIIC0	I^2C-bus Function Enable Register	ICFER	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8306h	RIIC0	I^2C-bus Status Enable Register	ICSER	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8307h	RIIC0	I^2C-bus Interrupt Enable Register	ICIER	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8308h	RIIC0	I^2C-bus Status Register 1	ICSR1	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8309h	RIIC0	I^2C-bus Status Register 2	ICSR2	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2 or 3 PCLKB	2 ICLK	section 33

Table 5.1 List of I/O Registers (Address Order) (6 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8310h	RIIC0	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8311h	RIIC0	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8312h	RIIC0	I²C-bus Transmit Data Register	ICDRT	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8313h	RIIC0	I²C-bus Receive Data Register	ICDRR	8	8	2 or 3 PCLKB	2 ICLK	section 33
0008 8380h	RSPI0	RSPI Control Register	SPCR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 8381h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 8382h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 8383h	RSPI0	RSPI Status Register	SPSR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 8384h	RSPI0	RSPI Data Register	SPDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 35
0008 8388h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 8389h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 838Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 838Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 838Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 838Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 838Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 838Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 8390h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 8392h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 8394h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 8396h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 8398h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 839Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 839Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 839Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	2 or 3 PCLKB	2 ICLK	section 35
0008 83A0h	RSPI0	RSPI Data Control Register 2	SPDCR2	8	8	2 or 3 PCLKB	2 ICLK	section 35
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9006h	S12AD	A/D Channel Select Register A1	ADANSA1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Select Register 0	ADADS0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 900Ah	S12AD	A/D-Converted Value Addition/Average Function Select Register 1	ADADS1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9012h	S12AD	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9016h	S12AD	A/D Channel Select Register B1	ADANSB1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 901Ah	S12AD	A/D Temperature Sensor Data Register	ADTSDR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2 or 3 PCLKB	2 ICLK	section 40

Table 5.1 List of I/O Registers (Address Order) (7 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9030h	S12AD	A/D Data Register 8	ADDR8	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9040h	S12AD	A/D Data Register 16	ADDR16	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9042h	S12AD	A/D Data Register 17	ADDR17	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9044h	S12AD	A/D Data Register 18	ADDR18	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9046h	S12AD	A/D Data Register 19	ADDR19	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9048h	S12AD	A/D Data Register 20	ADDR20	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 904Ah	S12AD	A/D Data Register 21	ADDR21	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 904Ch	S12AD	A/D Data Register 22	ADDR22	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 904Eh	S12AD	A/D Data Register 23	ADDR23	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9050h	S12AD	A/D Data Register 24	ADDR24	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9052h	S12AD	A/D Data Register 25	ADDR25	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9054h	S12AD	A/D Data Register 26	ADDR26	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9056h	S12AD	A/D Data Register 27	ADDR27	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9058h	S12AD	A/D Data Register 28	ADDR28	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 905Ah	S12AD	A/D Data Register 29	ADDR29	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 905Ch	S12AD	A/D Data Register 30	ADDR30	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 905Eh	S12AD	A/D Data Register 31	ADDR31	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 907Dh	S12AD	A/D Event Link Control Register	ADELCCR	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 907Eh	S12AD	A/D Convert Cycle Control Register	ADCCR	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 908Ah	S12AD	A/D High-Potential/Low-Potential Reference Voltage Control Register	ADHVREFCNT	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 908Ch	S12AD	A/D Compare Function Window A/B Status Monitor Register	ADWINMON	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 9090h	S12AD	A/D Compare Function Control Register	ADCMPCR	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9092h	S12AD	A/D Compare Function Window A Extended Input Select Register	ADCMPANSER	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 9093h	S12AD	A/D Compare Function Window A Extended Input Comparison Condition Setting Register	ADCMPLER	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 9094h	S12AD	A/D Compare Function Window A Channel Select Register 0	ADCMPANSR0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9096h	S12AD	A/D Compare Function Window A Channel Select Register 1	ADCMPANSR1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 9098h	S12AD	A/D Compare Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 909Ah	S12AD	A/D Compare Function Window A Comparison Condition Setting Register 1	ADCMPLR1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 909Ch	S12AD	A/D Compare Function Window A Lower-Side Level Setting Register	ADCMPDR0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 909Eh	S12AD	A/D Compare Function Window A Upper-Side Level Setting Register	ADCMPDR1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90A0h	S12AD	A/D Compare Function Window A Channel Status Register 0	ADCMPSR0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90A2h	S12AD	A/D Compare Function Window A Channel Status Register 1	ADCMPSR1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90A4h	S12AD	A/D Compare Function Window A Extended Input Channel Status Register	ADCMPSER	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90A6h	S12AD	A/D Compare Function Window B Channel Select Register	ADCMPBNSR	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90A8h	S12AD	A/D Compare Function Window B Lower-Side Level Setting Register	ADWINLLB	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90AAh	S12AD	A/D Compare Function Window B Upper-Side Level Setting Register	ADWINULB	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90ACh	S12AD	A/D Compare Function Window B Channel Status Register	ADCMPBSR	8	8	2 or 3 PCLKB	2 ICLK	section 40

Table 5.1 List of I/O Registers (Address Order) (8 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 90B0h	S12AD	A/D Data Storage Buffer Register 0	ADBUF0	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90B2h	S12AD	A/D Data Storage Buffer Register 1	ADBUF1	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90B4h	S12AD	A/D Data Storage Buffer Register 2	ADBUF2	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90B6h	S12AD	A/D Data Storage Buffer Register 3	ADBUF3	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90B8h	S12AD	A/D Data Storage Buffer Register 4	ADBUF4	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90BAh	S12AD	A/D Data Storage Buffer Register 5	ADBUF5	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90BCh	S12AD	A/D Data Storage Buffer Register 6	ADBUF6	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90BEh	S12AD	A/D Data Storage Buffer Register 7	ADBUF7	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90C0h	S12AD	A/D Data Storage Buffer Register 8	ADBUF8	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90C2h	S12AD	A/D Data Storage Buffer Register 9	ADBUF9	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90C4h	S12AD	A/D Data Storage Buffer Register 10	ADBUF10	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90C6h	S12AD	A/D Data Storage Buffer Register 11	ADBUF11	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90C8h	S12AD	A/D Data Storage Buffer Register 12	ADBUF12	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90CAh	S12AD	A/D Data Storage Buffer Register 13	ADBUF13	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90CCh	S12AD	A/D Data Storage Buffer Register 14	ADBUF14	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90CEh	S12AD	A/D Data Storage Buffer Register 15	ADBUF15	16	16	2 or 3 PCLKB	2 ICLK	section 40
0008 90D0h	S12AD	A/D Data Storage Buffer Enable Register	ADBUFEN	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90D2h	S12AD	A/D Data Storage Buffer Pointer Register	ADBUFPTR	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90DDh	S12AD	A/D Sampling State Register L	ADSSTRL	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90DEh	S12AD	A/D Sampling State Register T	ADSSTRT	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90DFh	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E7h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 90E8h	S12AD	A/D Sampling State Register 8	ADSSTR8	8	8	2 or 3 PCLKB	2 ICLK	section 40
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A020h	SMCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A022h	SMCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A024h	SMCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31

Table 5.1 List of I/O Registers (Address Order) (9 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A033h	SCI1	Data Comparison Control Register	DCCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A03Ah	SCI1	Comparison Data Register	CDR	16	16	4 or 5 PCLKB	2 ICLK	section 31
0008 A03Ah	SCI1	Comparison Data Register H	CDR.H	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A03Bh	SCI1	Comparison Data Register L	CDR.L	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A03Ch	SCI1	Serial Port Register	SPTR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A03Dh	SCI1	Transmit/Receive Timing Select Register	TMGR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A0h	SMCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A2h	SMCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A4h	SMCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0B3h	SCI5	Data Comparison Control Register	DCCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0BAh	SCI5	Comparison Data Register	CDR	16	16	4 or 5 PCLKB	2 ICLK	section 31
0008 A0BAh	SCI5	Comparison Data Register H	CDR.H	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0BBh	SCI5	Comparison Data Register L	CDR.L	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0BCh	SCI5	Serial Port Register	SPTR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0BDh	SCI5	Transmit/Receive Timing Select Register	TMGR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C0h	SMCI6	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C2h	SMCI6	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C4h	SMCI6	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31

Table 5.1 List of I/O Registers (Address Order) (10 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C6h	SCI6	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C6h	SMCI6	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0C9h	SCI6	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CAh	SCI6	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CBh	SCI6	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CCh	SCI6	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CEh	SCI6	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CEh	SCI6	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0CFh	SCI6	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0D0h	SCI6	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 A0D0h	SCI6	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0D1h	SCI6	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0D2h	SCI6	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0D3h	SCI6	Data Comparison Control Register	DCCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0DAh	SCI6	Comparison Data Register H	CDR	16	16	4 or 5 PCLKB	2 ICLK	section 31
0008 A0DAh	SCI6	Comparison Data Register L	CDR.H	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0DBh	SCI6	Comparison Data Register	CDR.L	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0DCh	SCI6	Serial Port Register	SPTR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 A0DDh	SCI6	Transmit/Receive Timing Select Register	TMGR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2 or 3 PCLKB	2 ICLK	section 10
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2 or 3 PCLKB	2 ICLK	section 10
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2 or 3 PCLKB	2 ICLK	section 10
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2 or 3 PCLKB	2 ICLK	section 10
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2 or 3 PCLKB	2 ICLK	section 10
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2 or 3 PCLKB	2 ICLK	section 10
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2 or 3 PCLKB	2 ICLK	section 10
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2 or 3 PCLKB	2 ICLK	section 10
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2 or 3 PCLKB	2 ICLK	section 44
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2 or 3 PCLKB	2 ICLK	section 44
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2 or 3 PCLKB	2 ICLK	section 44
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B109h	ELC	Event Link Setting Register 8	ELSR8	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B10Fh	ELC	Event Link Setting Register 14	ELSR14	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B111h	ELC	Event Link Setting Register 16	ELSR16	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2 or 3 PCLKB	2 ICLK	section 19

Table 5.1 List of I/O Registers (Address Order) (11 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B146h	ELC	Event Link Setting Register 48	ELSR48	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B147h	ELC	Event Link Setting Register 49	ELSR49	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B148h	ELC	Event Link Setting Register 50	ELSR50	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B149h	ELC	Event Link Setting Register 51	ELSR51	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B14Ah	ELC	Event Link Setting Register 52	ELSR52	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B14Bh	ELC	Event Link Setting Register 53	ELSR53	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B14Ch	ELC	Event Link Setting Register 54	ELSR54	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B14Dh	ELC	Event Link Setting Register 55	ELSR55	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B14Eh	ELC	Event Link Setting Register 56	ELSR56	8	8	2 or 3 PCLKB	2 ICLK	section 19
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B300h	SMCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B302h	SMCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B304h	SMCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B309h	SCI12	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Ah	SCI12	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Bh	SCI12	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Ch	SCI12	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	2 ICLK	section 31
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	2 ICLK	section 31

Table 5.1 List of I/O Registers (Address Order) (12 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B321h	SCI12	Control Register 0	CR0	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B322h	SCI12	Control Register 1	CR1	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B323h	SCI12	Control Register 2	CR2	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B324h	SCI12	Control Register 3	CR3	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B325h	SCI12	Port Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B327h	SCI12	Status Register	STR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2 or 3 PCLKB	2 ICLK	section 31
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C010h	PORTG	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C011h	PORTH	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C030h	PORTG	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C031h	PORTH	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C040h	PORT0	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C041h	PORT1	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (13 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C042h	PORT2	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C043h	PORT3	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C044h	PORT4	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C045h	PORT5	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C04Ah	PORTA	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C04Bh	PORTB	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C04Ch	PORTC	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C04Dh	PORTD	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C04Eh	PORTE	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C050h	PORTG	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C051h	PORTH	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C052h	PORTJ	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB	3 ICLK	section 20
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C070h	PORTG	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C071h	PORTH	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C082h	PORT1	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C083h	PORT1	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C084h	PORT2	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C085h	PORT2	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C086h	PORT3	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C087h	PORT3	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C08Ah	PORT5	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C08Bh	PORT5	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C094h	PORTA	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C095h	PORTA	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C096h	PORTB	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C097h	PORTB	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C098h	PORTC	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C099h	PORTC	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C09Ah	PORTD	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C09Ch	PORTE	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C09Dh	PORTE	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0A1h	PORTG	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0C0h	PORT0	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0C1h	PORT1	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0C2h	PORT2	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0C3h	PORT3	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0C4h	PORT4	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0C5h	PORT5	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (14 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C0CAh	PORTA	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0CBh	PORTB	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0CCh	PORTC	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0CDh	PORTD	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0CEh	PORTE	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0D0h	PORTG	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0D1h	PORTH	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C0D2h	PORTJ	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C120h	PORT	Port Switching Register B	PSRB	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C121h	PORT	Port Switching Register A	PSRA	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C122h	PORT	Port Reading Wait Control Register	PRWCNTR	8	8	2 or 3 PCLKB	2 ICLK	section 20
0008 C143h	MPC	P03 Pin Function Control Register	P03PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C145h	MPC	P05 Pin Function Control Register	P05PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C14Ah	MPC	P12 Pin Function Control Register	P12PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C14Bh	MPC	P13 Pin Function Control Register	P13PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C14Ch	MPC	P14 Pin Function Control Register	P14PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C14Dh	MPC	P15 Pin Function Control Register	P15PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C14Eh	MPC	P16 Pin Function Control Register	P16PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C14Fh	MPC	P17 Pin Function Control Register	P17PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C15Bh	MPC	P33 Pin Function Control Register	P33PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C15Ch	MPC	P34 Pin Function Control Register	P34PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C15Eh	MPC	P36 Pin Function Control Register	P36PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C15Fh	MPC	P37 Pin Function Control Register	P37PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C169h	MPC	P51 Pin Function Control Register	P51PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C16Bh	MPC	P53 Pin Function Control Register	P53PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C16Ch	MPC	P54 Pin Function Control Register	P54PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C16Dh	MPC	P55 Pin Function Control Register	P55PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C190h	MPC	PA0 Pin Function Control Register	PA0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C191h	MPC	PA1 Pin Function Control Register	PA1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21

Table 5.1 List of I/O Registers (Address Order) (15 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C196h	MPC	PA6 Pin Function Control Register	PA6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C197h	MPC	PA7 Pin Function Control Register	PA7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A0h	MPC	PC0 Pin Function Control Register	PC0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A1h	MPC	PC1 Pin Function Control Register	PC1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A2h	MPC	PC2 Pin Function Control Register	PC2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A3h	MPC	PC3 Pin Function Control Register	PC3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A4h	MPC	PC4 Pin Function Control Register	PC4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A5h	MPC	PC5 Pin Function Control Register	PC5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A6h	MPC	PC6 Pin Function Control Register	PC6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A7h	MPC	PC7 Pin Function Control Register	PC7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A8h	MPC	PD0 Pin Function Control Register	PD0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1A9h	MPC	PD1 Pin Function Control Register	PD1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1AAh	MPC	PD2 Pin Function Control Register	PD2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1AFh	MPC	PD7 Pin Function Control Register	PD7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B5h	MPC	PE5 Pin Function Control Register	PE5PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B6h	MPC	PE6 Pin Function Control Register	PE6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1B7h	MPC	PE7 Pin Function Control Register	PE7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1C8h	MPC	PH0 Pin Function Control Register	PH0PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1C9h	MPC	PH1 Pin Function Control Register	PH1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1CAh	MPC	PH2 Pin Function Control Register	PH2PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1CBh	MPC	PH3 Pin Function Control Register	PH3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1D1h	MPC	PJ1 Pin Function Control Register	PJ1PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1D3h	MPC	PJ3 Pin Function Control Register	PJ3PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1D6h	MPC	PJ6 Pin Function Control Register	PJ6PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C1D7h	MPC	PJ7 Pin Function Control Register	PJ7PFS	8	8	2 or 3 PCLKB	2 ICLK	section 21
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 6
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 6
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 9

Table 5.1 List of I/O Registers (Address Order) (16 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4 or 5 PCLKB	2 or 3 ICLK	section 8
0008 C400h	RTC	64-Hz Counter	R64CNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C402h	RTC	Second Counter	RSECCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C402h	RTC	Binary Counter 0	BCNT0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C404h	RTC	Minute Counter	RMINCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C404h	RTC	Binary Counter 1	BCNT1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C406h	RTC	Hour Counter	RHRCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C406h	RTC	Binary Counter 2	BCNT2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C408h	RTC	Day-of-Week Counter	RWKCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C408h	RTC	Binary Counter 3	BCNT3	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C40Ah	RTC	Date Counter	RDAYCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C40Ch	RTC	Month Counter	RMONCNT	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C40Eh	RTC	Year Counter	RYRCNT	16	16	2 or 3 PCLKB	2 ICLK	section 26
0008 C410h	RTC	Second Alarm Register	RSECAR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C410h	RTC	Binary Counter 0 Alarm Register	BCNT0AR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C412h	RTC	Minute Alarm Register	RMINAR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C412h	RTC	Binary Counter 1 Alarm Register	BCNT1AR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C414h	RTC	Hour Alarm Register	RHRAR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C416h	RTC	Day-of-Week Alarm Register	RWKAR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C416h	RTC	Binary Counter 3 Alarm Register	BCNT3AR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C418h	RTC	Date Alarm Register	RDAYAR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C418h	RTC	Binary Counter 0 Alarm Enable Register	BCNT0AER	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C41Ah	RTC	Month Alarm Register	RMONAR	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C41Ah	RTC	Binary Counter 1 Alarm Enable Register	BCNT1AER	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C41Ch	RTC	Year Alarm Register	RYRAR	16	16	2 or 3 PCLKB	2 ICLK	section 26
0008 C41Ch	RTC	Binary Counter 2 Alarm Enable Register	BCNT2AER	16	16	2 or 3 PCLKB	2 ICLK	section 26
0008 C41Eh	RTC	Year Alarm Enable Register	RYRAREN	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C41Eh	RTC	Binary Counter 3 Alarm Enable Register	BCNT3AER	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C422h	RTC	RTC Control Register 1	RCR1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C424h	RTC	RTC Control Register 2	RCR2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C42Eh	RTC	Time Error Adjustment Register	RADJ	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C440h	RTC	Time Capture Control Register 0	RTCCR0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C442h	RTC	Time Capture Control Register 1	RTCCR1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C444h	RTC	Time Capture Control Register 2	RTCCR2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C452h	RTC	Second Capture Register 0	RSECCP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C452h	RTC	BCNT0 Capture Register 0	BCNT0CP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C454h	RTC	Minute Capture Register 0	RMINCP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C454h	RTC	BCNT1 Capture Register 0	BCNT1CP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C456h	RTC	Hour Capture Register 0	RHRCP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C456h	RTC	BCNT2 Capture Register 0	BCNT2CP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C45Ah	RTC	Date Capture Register 0	RDAYCP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C45Ah	RTC	BCNT3 Capture Register 0	BCNT3CP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C45Ch	RTC	Month Capture Register 0	RMONCP0	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C462h	RTC	Second Capture Register 1	RSECCP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C462h	RTC	BCNT0 Capture Register 1	BCNT0CP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C464h	RTC	Minute Capture Register 1	RMINCP1	8	8	2 or 3 PCLKB	2 ICLK	section 26

Table 5.1 List of I/O Registers (Address Order) (17 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C464h	RTC	BCNT1 Capture Register 1	BCNT1CP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C466h	RTC	Hour Capture Register 1	RHRCP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C466h	RTC	BCNT2 Capture Register 1	BCNT2CP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C46Ah	RTC	Date Capture Register 1	RDAYCP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C46Ah	RTC	BCNT3 Capture Register 1	BCNT3CP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C46Ch	RTC	Month Capture Register 1	RMONCP1	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C472h	RTC	Second Capture Register 2	RSECCP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C472h	RTC	BCNT0 Capture Register 2	BCNT0CP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C474h	RTC	Minute Capture Register 2	RMINCP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C474h	RTC	BCNT1 Capture Register 2	BCNT1CP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C476h	RTC	Hour Capture Register 2	RHRCP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C476h	RTC	BCNT2 Capture Register 2	BCNT2CP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C47Ah	RTC	Date Capture Register 2	RDAYCP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C47Ah	RTC	BCNT3 Capture Register 2	BCNT3CP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C47Ch	RTC	Month Capture Register 2	RMONCP2	8	8	2 or 3 PCLKB	2 ICLK	section 26
0008 C580h	CMPB	Comparator B Control Register 1	CPBCNT1	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C581h	CMPB	Comparator B Control Register 2	CPBCNT2	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C582h	CMPB	Comparator B Flag Register	CPBFLG	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C583h	CMPB	Comparator B Interrupt Control Register	CPBINT	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C584h	CMPB	Comparator B Filter Select Register	CPBF	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C585h	CMPB	Comparator B Mode Select Register	CPBMD	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C586h	CMPB	Comparator B Reference Input Voltage Select Register	CPBREF	8	8	2 or 3 PCLKB	2 ICLK	section 43
0008 C587h	CMPB	Comparator B Output Control Register	CPBOCR	8	8	2 or 3 PCLKB	2 ICLK	section 43
0009 E000h	POEG	POEG Group A Setting Register	POEGGA	32	32	2 or 3 PCLKB	2 ICLK	section 23
0009 E100h	POEG	POEG Group B Setting Register	POEGGB	32	32	2 or 3 PCLKB	2 ICLK	section 23
0009 E200h	POEG	POEG Group C Setting Register	POEGGC	32	32	2 or 3 PCLKB	2 ICLK	section 23
0009 E300h	POEG	POEG Group D Setting Register	POEGGD	32	32	2 or 3 PCLKB	2 ICLK	section 23
000A 0000h	USB0	System Configuration Control Register	SYSCFG	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 0004h	USB0	System Configuration Status Register 0	SYSSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0008h	USB0	Device State Control Register 0	DVSTCTR0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0014h	USB0	CFIFO Port Register	CFIFO	16	8, 16	3 or 4 PCLKB	2 ICLK	section 30
000A 0018h	USB0	D0FIFO Port Register	D0FIFO	16	8, 16	3 or 4 PCLKB	2 ICLK	section 30
000A 001Ch	USB0	D1FIFO Port Register	D1FIFO	16	8, 16	3 or 4 PCLKB	2 ICLK	section 30
000A 0020h	USB0	CFIFO Port Select Register	CFIFOSEL	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 0022h	USB0	CFIFO Port Control Register	CFIFOCTR	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 0028h	USB0	D0FIFO Port Select Register	D0FIFOSEL	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 002Ah	USB0	D0FIFO Port Control Register	D0FIFOCTR	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 002Ch	USB0	D1FIFO Port Select Register	D1FIFOSEL	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 002Eh	USB0	D1FIFO Port Control Register	D1FIFOCTR	16	16	3 or 4 PCLKB	2 ICLK	section 30
000A 0030h	USB0	Interrupt Enable Register 0	INTENB0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30

Table 5.1 List of I/O Registers (Address Order) (18 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0032h	USB0	Interrupt Enable Register 1	INTENB1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0036h	USB0	BRDY Interrupt Enable Register	BRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0038h	USB0	NRDY Interrupt Enable Register	NRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 003Ah	USB0	BEMP Interrupt Enable Register	BEMPENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 003Ch	USB0	SOF Output Configuration Register	SOFCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0040h	USB0	Interrupt Status Register 0	INTSTS0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0042h	USB0	Interrupt Status Register 1	INTSTS1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0046h	USB0	BRDY Interrupt Status Register	BRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0048h	USB0	NRDY Interrupt Status Register	NRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 004Ah	USB0	BEMP Interrupt Status Register	BEMPSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 004Ch	USB0	Frame Number Register	FRMNUM	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0054h	USB0	USB Request Type Register	USBREQ	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0056h	USB0	USB Request Value Register	USBVAL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 0058h	USB0	USB Request Index Register	USBINDX	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 005Ah	USB0	USB Request Length Register	USBLENG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 005Ch	USB0	DCP Configuration Register	DCPCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30
000A 005Eh	USB0	DCP Maximum Packet Size Register	DCPMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 30

Table 5.1 List of I/O Registers (Address Order) (19 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0060h	USB0	DCP Control Register	DCPCTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0064h	USB0	Pipe Window Select Register	PIPESEL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0068h	USB0	Pipe Configuration Register	PIPECFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 006Ch	USB0	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 006Eh	USB0	Pipe Cycle Control Register	PIPEPERI	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0070h	USB0	Pipe 1 Control Register	PIPE1CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0072h	USB0	Pipe 2 Control Register	PIPE2CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0074h	USB0	Pipe 3 Control Register	PIPE3CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0076h	USB0	Pipe 4 Control Register	PIPE4CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0078h	USB0	Pipe 5 Control Register	PIPE5CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 007Ah	USB0	Pipe 6 Control Register	PIPE6CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 007Ch	USB0	Pipe 7 Control Register	PIPE7CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 007Eh	USB0	Pipe 8 Control Register	PIPE8CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0080h	USB0	Pipe 9 Control Register	PIPE9CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0090h	USB0	Pipe 1 Transaction Counter Enable Register	PIPE1TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0092h	USB0	Pipe 1 Transaction Counter Register	PIPE1TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0094h	USB0	Pipe 2 Transaction Counter Enable Register	PIPE2TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30

Table 5.1 List of I/O Registers (Address Order) (20 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0096h	USB0	Pipe 2 Transaction Counter Register	PIPE2TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0098h	USB0	Pipe 3 Transaction Counter Enable Register	PIPE3TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 009Ah	USB0	Pipe 3 Transaction Counter Register	PIPE3TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 009Ch	USB0	Pipe 4 Transaction Counter Enable Register	PIPE4TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 009Eh	USB0	Pipe 4 Transaction Counter Register	PIPE4TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00A0h	USB0	Pipe 5 Transaction Counter Enable Register	PIPE5TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00A2h	USB0	Pipe 5 Transaction Counter Register	PIPE5TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00D0h	USB0	Device Address 0 Configuration Register	DEVADD0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00D2h	USB0	Device Address 1 Configuration Register	DEVADD1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00D4h	USB0	Device Address 2 Configuration Register	DEVADD2	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00D6h	USB0	Device Address 3 Configuration Register	DEVADD3	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00D8h	USB0	Device Address 4 Configuration Register	DEVADD4	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 00DAh	USB0	Device Address 5 Configuration Register	DEVADD5	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*1	section 30
000A 0700h	CTSU	CTSU A/D Converter Connection Control Register	CTSUADCC	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0900h	CTSU	CTSU Control Register A	CTSUCRA	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0904h	CTSU	CTSU Control Register B	CTSUCRB	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0908h	CTSU	CTSU Measurement Channel Register	CTSUMCH	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 090Ch	CTSU	CTSU Channel Enable Control Register A	CTSUCHACA	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0910h	CTSU	CTSU Channel Enable Control Register B	CTSUCHACB	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0914h	CTSU	CTSU Channel Transmit/Receive Control Register A	CTSUCHTRCA	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0918h	CTSU	CTSU Channel Transmit/Receive Control Register B	CTSUCHTRCB	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 091Ch	CTSU	CTSU Status Register	CTSUSR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0920h	CTSU	CTSU Sensor Offset Register	CTSUSO	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0924h	CTSU	CTSU Sensor Counter	CTSUSCNT	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0928h	CTSU	CTSU Calibration Register	CTSUCALIB	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39

Table 5.1 List of I/O Registers (Address Order) (21 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 092Ch	CTSU	CTSU Sensor Unit Clock Control Register A	CTSUSUCLKA	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0930h	CTSU	CTSU Sensor Unit Clock Control Register B	CTSUSUCLKB	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0940h	CTSU	CTSU Option Setting Register	CTSUOPT	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0944h	CTSU	CTSU Sensor Counter Automatic Correction Table Access Register	CTSUSCNTACT	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 094Ch	CTSU	CTSU Multi-Clock Automatic Correction Table 1	CTSUMCACT1	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0950h	CTSU	CTSU Multi-Clock Automatic Correction Table 2	CTSUMCACT2	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0954h	CTSU	CTSU Multi-Clock Automatic Correction Table 3	CTSUMCACT3	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0958h	CTSU	CTSU Automatic Judgment Control Register	CTSUAJCR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 095Ch	CTSU	CTSU Threshold Register	CTSUAJTHR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0960h	CTSU	CTSU Moving Average Result Register	CTSUAJMMAR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0964h	CTSU	CTSU Baseline Average Intermediate Result Register	CTSUAJBLACT	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0968h	CTSU	CTSU Baseline Average Result Register	CTSUAJBLAR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 096Ch	CTSU	CTSU Automatic Judgment Result Register	CTSUAJRR	32	16, 32	2 or 3 PCLKB	2 ICLK	section 39
000A 0B00h	REMC0	Function Select Register 0	REMCON0	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B01h	REMC0	Function Select Register 1	REMCON1	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B02h	REMC0	Status Register	REMSTS	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B03h	REMC0	Interrupt Control Register	REMINT	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B05h	REMC0	Compare Control Register	REMCPC	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B06h	REMC0	Compare Value Setting Register	REMCPD	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 0B08h	REMC0	Header Pattern Minimum Width Setting Register	HDPMIN	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 0B0Ah	REMC0	Header Pattern Maximum Width Setting Register	HDPMAX	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 0B0Ch	REMC0	Data ‘0’ Pattern Minimum Width Setting Register	D0PMIN	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B0Dh	REMC0	Data ‘0’ Pattern Maximum Width Setting Register	D0PMAX	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B0Eh	REMC0	Data ‘1’ Pattern Minimum Width Setting Register	D1PMIN	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B0Fh	REMC0	Data ‘1’ Pattern Maximum Width Setting Register	D1PMAX	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B10h	REMC0	Special Data Pattern Minimum Width Setting Register	SDPMIN	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 0B12h	REMC0	Special Data Pattern Maximum Width Setting Register	SDPMAX	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 0B14h	REMC0	Pattern End Setting Register	REMPE	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 0B16h	REMC0	Receiver Standby Control Register	REMSTC	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B17h	REMC0	Receive Bit Count Register	REMRBIT	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B18h	REMC0	Receive Data 0 Register	REMDAT0	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B19h	REMC0	Receive Data 1 Register	REMDAT1	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B1Ah	REMC0	Receive Data 2 Register	REMDAT2	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B1Bh	REMC0	Receive Data 3 Register	REMDAT3	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B1Ch	REMC0	Receive Data 4 Register	REMDAT4	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B1Dh	REMC0	Receive Data 5 Register	REMDAT5	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B1Eh	REMC0	Receive Data 6 Register	REMDAT6	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B1Fh	REMC0	Receive Data 7 Register	REMDAT7	8	8	2 or 3 PCLKB	2 ICLK	section 37
000A 0B20h	REMC0	Measurement Result Register	REMTIM	16	16	2 or 3 PCLKB	2 ICLK	section 37
000A 1000h	RSCI0	Receive Data Register	RDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 32
000A 1004h	RSCI0	Transmit Data Register	TDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 32
000A 1008h	RSCI0	Control Register 0	SCR0	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 100Ch	RSCI0	Control Register 1	SCR1	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1010h	RSCI0	Control Register 2	SCR2	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1014h	RSCI0	Control Register 3	SCR3	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1018h	RSCI0	Control Register 4	SCR4	32	32	2 or 3 PCLKB	2 ICLK	section 32

Table 5.1 List of I/O Registers (Address Order) (22 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 101Eh	RSCI0	HBS Support Mode Control Register	HBSCR	8	8	2 or 3 PCLKB	2 ICLK	section 32
000A 1020h	RSCI0	I²C Mode Register	SIMR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1030h	RSCI0	DE Signal Control Register	DECR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1048h	RSCI0	Status Register	SSR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 104Ch	RSCI0	I²C Status Register	SISR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1068h	RSCI0	Status Clear Register	SSCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 106Ch	RSCI0	I²C Status Clear Register	SISCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1400h	RSCI8	Receive Data Register	RDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 32
000A 1404h	RSCI8	Transmit Data Register	TDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 32
000A 1408h	RSCI8	Control Register 0	SCR0	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 140Ch	RSCI8	Control Register 1	SCR1	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1410h	RSCI8	Control Register 2	SCR2	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1414h	RSCI8	Control Register 3	SCR3	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1418h	RSCI8	Control Register 4	SCR4	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 141Eh	RSCI8	HBS Support Mode Control Register	HBSCR	8	8	2 or 3 PCLKB	2 ICLK	section 32
000A 1420h	RSCI8	I²C Mode Register	SIMR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1430h	RSCI8	DE Signal Control Register	DECR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1448h	RSCI8	Status Register	SSR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 144Ch	RSCI8	I²C Status Register	SISR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1468h	RSCI8	Status Clear Register	SSCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 146Ch	RSCI8	I²C Status Clear Register	SISCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1480h	RSCI9	Receive Data Register	RDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 32
000A 1484h	RSCI9	Transmit Data Register	TDR	32	8, 16, 32	2 or 3 PCLKB	2 ICLK	section 32
000A 1488h	RSCI9	Control Register 0	SCR0	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 148Ch	RSCI9	Control Register 1	SCR1	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1490h	RSCI9	Control Register 2	SCR2	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1494h	RSCI9	Control Register 3	SCR3	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 1498h	RSCI9	Control Register 4	SCR4	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 149Eh	RSCI9	HBS Support Mode Control Register	HBSCR	8	8	2 or 3 PCLKB	2 ICLK	section 32
000A 14A0h	RSCI9	I²C Mode Register	SIMR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14ACh	RSCI9	Manchester Mode Control Register	MMCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14B0h	RSCI9	DE Signal Control Register	DECR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14B4h	RSCI9	Extended Serial Mode Control Register 0	XCR0	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14B8h	RSCI9	Extended Serial Mode Control Register 1	XCR1	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14BCh	RSCI9	Extended Serial Mode Control Register 2	XCR2	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14C8h	RSCI9	Status Register	SSR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14CCh	RSCI9	I²C Status Register	SISR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14D8h	RSCI9	Manchester Mode Status Register	MMSR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14DCh	RSCI9	Extended Serial Mode Status Register 0	XSR0	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14E0h	RSCI9	Extended Serial Mode Status Register 1	XSR1	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14E8h	RSCI9	Status Clear Register	SSCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14ECh	RSCI9	I²C Status Clear Register	SISCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14F4h	RSCI9	Manchester Mode Status Clear Register	MMSCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 14F8h	RSCI9	Extended Serial Mode Status Clear Register	XSCR	32	32	2 or 3 PCLKB	2 ICLK	section 32
000A 8000h	CANFD0	Nominal Bit Rate Configuration Register	NBCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8004h	CANFD0	Channel Control Register	CHCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8008h	CANFD0	Channel Status Register	CHSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 800Ch	CANFD0	Channel Error Status Register	CHESR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34

Table 5.1 List of I/O Registers (Address Order) (23 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 8014h	CANFD	Global Configuration Register	GCFG	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8018h	CANFD	Global Control Register	GCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 801Ch	CANFD	Global Status Register	GSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8020h	CANFD	Global Error Status Register	GESR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8024h	CANFD	Timestamp Counter Register	TSCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8028h	CANFD	Acceptance Filter List Control Register	AFCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 802Ch	CANFD	Acceptance Filter List Configuration Register	AFCFG	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8030h	CANFD	Receive Message Buffer Configuration Register	RMCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8034h	CANFD	Receive Message Buffer New Data Register	RMNDR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8038h	CANFD	Receive Message Buffer Interrupt Enable Register	RMIER	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 803Ch	CANFD	Receive FIFO 0 Configuration Register	RFCR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8040h	CANFD	Receive FIFO 1 Configuration Register	RFCR1	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8044h	CANFD	Receive FIFO 0 Status Register	RFSR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8048h	CANFD	Receive FIFO 1 Status Register	RFSR1	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 804Ch	CANFD	Receive FIFO 0 Pointer Control Register	RFPCR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8050h	CANFD	Receive FIFO 1 Pointer Control Register	RFPCR1	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8054h	CANFD	Common FIFO 0 Configuration Register	CFCR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8058h	CANFD	Common FIFO 0 Status Register	CFSR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 805Ch	CANFD	Common FIFO 0 Pointer Control Register	CFPCR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8060h	CANFD	FIFO Empty Status Register	FESR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8064h	CANFD	FIFO Full Status Register	FFSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8068h	CANFD	FIFO Message Lost Status Register	FMLSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 806Ch	CANFD	Receive FIFO Interrupt Status Register	RFISR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8070h	CANFD	Transmit Message Buffer 0 Control Register	TMCR0	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8071h	CANFD	Transmit Message Buffer 1 Control Register	TMCR1	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8072h	CANFD	Transmit Message Buffer 2 Control Register	TMCR2	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8073h	CANFD	Transmit Message Buffer 3 Control Register	TMCR3	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8074h	CANFD	Transmit Message Buffer 0 Status Register	TMSR0	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8075h	CANFD	Transmit Message Buffer 1 Status Register	TMSR1	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8076h	CANFD	Transmit Message Buffer 2 Status Register	TMSR2	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8077h	CANFD	Transmit Message Buffer 3 Status Register	TMSR3	8	8	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8078h	CANFD	Transmit Message Buffer Transmission Request Status Register 0	TMTRSR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 807Ch	CANFD	Transmit Message Buffer Transmission Abort Request Status Register 0	TMARSR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8080h	CANFD	Transmit Message Buffer Transmission Completion Status Register 0	TMTCSR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8084h	CANFD	Transmit Message Buffer Transmission Abort Status Register 0	TMTASR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8088h	CANFD	Transmit Message Buffer Interrupt Enable Register	TMIER0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 808Ch	CANFD0	Transmit Queue 0 Configuration Register	TQCR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8090h	CANFD0	Transmit Queue 0 Status Register	TQSR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8094h	CANFD0	Transmit Queue 0 Pointer Control Register	TQPCR0	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8098h	CANFD0	Transmission History Configuration Register	THCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 809Ch	CANFD0	Transmission History Status Register	THSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80A0h	CANFD0	Transmission History Pointer Control Register	THPCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80A4h	CANFD	Transmit Interrupt Status Register	TISR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80A8h	CANFD	Global Test Mode Configuration Register	GTMCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80ACh	CANFD	Global Test Mode Enable Register	GTMER	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80B0h	CANFD	Global CAN FD Configuration Register	GFDCFG	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80B8h	CANFD	Global Test Mode Lock Key Register	GTMLKR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34

Table 5.1 List of I/O Registers (Address Order) (24 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 80C0h	CANFD	Acceptance Filter List Ignore Entry Setting Register	AFIGSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80C4h	CANFD	Acceptance Filter List Ignore Entry Enable Register	AFIGER	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80C8h	CANFD	DMA Transfer Control Register	DTCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80CCh	CANFD	DMA Transfer Status Register	DTSR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 80D8h	CANFD	Global Reset Control Register	GRCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8100h	CANFD0	Data Bit Rate Configuration Register	DBCR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8104h	CANFD0	CAN FD Configuration Register	FDCFG	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8108h	CANFD0	CAN FD Control Register	FDCTR	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 810Ch	CANFD0	CAN FD Status Register	FDSTS	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8110h	CANFD0	CAN FD CRC Register	FDCRC	32	32	2 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8120h	CANFD	Acceptance Filter List 0	AFL0	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8130h	CANFD	Acceptance Filter List 1	AFL1	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8140h	CANFD	Acceptance Filter List 2	AFL2	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8150h	CANFD	Acceptance Filter List 3	AFL3	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8160h	CANFD	Acceptance Filter List 4	AFL4	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8170h	CANFD	Acceptance Filter List 5	AFL5	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8180h	CANFD	Acceptance Filter List 6	AFL6	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8190h	CANFD	Acceptance Filter List 7	AFL7	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 81A0h	CANFD	Acceptance Filter List 8	AFL8	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 81B0h	CANFD	Acceptance Filter List 9	AFL9	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 81C0h	CANFD	Acceptance Filter List 10	AFL10	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 81D0h	CANFD	Acceptance Filter List 11	AFL11	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 81E0h	CANFD	Acceptance Filter List 12	AFL12	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 81F0h	CANFD	Acceptance Filter List 13	AFL13	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8200h	CANFD	Acceptance Filter List 14	AFL14	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8210h	CANFD	Acceptance Filter List 15	AFL15	128	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
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 or 3 PCLKB	1 or 2 ICLK	section 34
000A 8520h to 000A 856Bh	CANFD	Receive FIFO 0	RFB0	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 856Ch to 000A 85B7h	CANFD	Receive FIFO 1	RFB1	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 85B8h to 000A 8603h	CANFD	Common FIFO 0	CFB0	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8604h to 000A 864Fh	CANFD	Transmit Message Buffer 0	TMB0	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8650h to 000A 869Bh	CANFD	Transmit Message Buffer 1	TMB1	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 869Ch to 000A 86E7h	CANFD	Transmit Message Buffer 2	TMB2	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 86E8h to 000A 8733h	CANFD	Transmit Message Buffer 3	TMB3	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8740h	CANFD0	Transmission History Access Register 0	THACR0	32	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8744h	CANFD0	Transmission History Access Register 1	THACR1	32	32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8920h to 000A 896Bh	CANFD	Receive Message Buffer 0	RMB0	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 896Ch to 000A 89B7h	CANFD	Receive Message Buffer 1	RMB1	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 89B8h to 000A 8A03h	CANFD	Receive Message Buffer 2	RMB2	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8A04h to 000A 8A4Fh	CANFD	Receive Message Buffer 3	RMB3	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8A50h to 000A 8A9Bh	CANFD	Receive Message Buffer 4	RMB4	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8A9Ch to 000A 8AE7h	CANFD	Receive Message Buffer 5	RMB5	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34

Table 5.1 List of I/O Registers (Address Order) (25 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 8AE8h to 000A 8B33h	CANFD	Receive Message Buffer 6	RMB6	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8B34h to 000A 8B7Fh	CANFD	Receive Message Buffer 7	RMB7	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8D20h to 000A 8D6Bh	CANFD	Receive Message Buffer 8	RMB8	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8D6Ch to 000A 8DB7h	CANFD	Receive Message Buffer 9	RMB9	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8DB8h to 000A 8E03h	CANFD	Receive Message Buffer 10	RMB10	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8E04h to 000A 8E4Fh	CANFD	Receive Message Buffer 11	RMB11	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8E50h to 000A 8E9Bh	CANFD	Receive Message Buffer 12	RMB12	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8E9Ch to 000A 8EE7h	CANFD	Receive Message Buffer 13	RMB13	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8EE8h to 000A 8F33h	CANFD	Receive Message Buffer 14	RMB14	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 8F34h to 000A 8F7Fh	CANFD	Receive Message Buffer 15	RMB15	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9120h to 000A 916Bh	CANFD	Receive Message Buffer 16	RMB16	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 916Ch to 000A 91B7h	CANFD	Receive Message Buffer 17	RMB17	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 91B8h to 000A 9203h	CANFD	Receive Message Buffer 18	RMB18	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9204h to 000A 924Fh	CANFD	Receive Message Buffer 19	RMB19	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9250h to 000A 929Bh	CANFD	Receive Message Buffer 20	RMB20	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 929Ch to 000A 92E7h	CANFD	Receive Message Buffer 21	RMB21	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 92E8h to 000A 9333h	CANFD	Receive Message Buffer 22	RMB22	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9334h to 000A 937Fh	CANFD	Receive Message Buffer 23	RMB23	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9520h to 000A 956Bh	CANFD	Receive Message Buffer 24	RMB24	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 956Ch to 000A 95B7h	CANFD	Receive Message Buffer 25	RMB25	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 95B8h to 000A 9603h	CANFD	Receive Message Buffer 26	RMB26	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9604h to 000A 964Fh	CANFD	Receive Message Buffer 27	RMB27	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9650h to 000A 969Bh	CANFD	Receive Message Buffer 28	RMB28	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 969Ch to 000A 96E7h	CANFD	Receive Message Buffer 29	RMB29	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 96E8h to 000A 9733h	CANFD	Receive Message Buffer 30	RMB30	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000A 9734h to 000A 977Fh	CANFD	Receive Message Buffer 31	RMB31	608	8, 16, 32	3 or 4 PCLKB	1 or 2 ICLK	section 34
000C 2000h	GPTW0	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2004h	GPTW0	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2008h	GPTW0	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 200Ch	GPTW0	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2010h	GPTW0	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2014h	GPTW0	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2018h	GPTW0	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2024h	GPTW0	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (26 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2028h	GPTW0	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 202Ch	GPTW0	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2030h	GPTW0	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2034h	GPTW0	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2038h	GPTW0	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 203Ch	GPTW0	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2040h	GPTW0	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2044h	GPTW0	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2048h	GPTW0	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 204Ch	GPTW0	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2050h	GPTW0	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2054h	GPTW0	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2058h	GPTW0	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 205Ch	GPTW0	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2060h	GPTW0	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2064h	GPTW0	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2068h	GPTW0	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 206Ch	GPTW0	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2070h	GPTW0	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2074h	GPTW0	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2078h	GPTW0	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 207Ch	GPTW0	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2080h	GPTW0	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2084h	GPTW0	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2088h	GPTW0	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 208Ch	GPTW0	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 20D0h	GPTW0	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 20D4h	GPTW0	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 20ECh	GPTW0	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2100h	GPTW1	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2104h	GPTW1	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2108h	GPTW1	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 210Ch	GPTW1	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2110h	GPTW1	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2114h	GPTW1	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2118h	GPTW1	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2124h	GPTW1	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2128h	GPTW1	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 212Ch	GPTW1	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2130h	GPTW1	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2134h	GPTW1	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (27 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2138h	GPTW1	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 213Ch	GPTW1	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2140h	GPTW1	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2144h	GPTW1	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2148h	GPTW1	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 214Ch	GPTW1	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2150h	GPTW1	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2154h	GPTW1	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2158h	GPTW1	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 215Ch	GPTW1	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2160h	GPTW1	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2164h	GPTW1	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2168h	GPTW1	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 216Ch	GPTW1	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2170h	GPTW1	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2174h	GPTW1	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2178h	GPTW1	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 217Ch	GPTW1	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2180h	GPTW1	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2184h	GPTW1	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2188h	GPTW1	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 218Ch	GPTW1	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 21D0h	GPTW1	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 21D4h	GPTW1	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 21ECh	GPTW1	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2200h	GPTW2	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2204h	GPTW2	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2208h	GPTW2	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 220Ch	GPTW2	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2210h	GPTW2	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2214h	GPTW2	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2218h	GPTW2	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2224h	GPTW2	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2228h	GPTW2	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 222Ch	GPTW2	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2230h	GPTW2	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2234h	GPTW2	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2238h	GPTW2	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 223Ch	GPTW2	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2240h	GPTW2	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2244h	GPTW2	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (28 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2248h	GPTW2	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 224Ch	GPTW2	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2250h	GPTW2	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2254h	GPTW2	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2258h	GPTW2	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 225Ch	GPTW2	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2260h	GPTW2	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2264h	GPTW2	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2268h	GPTW2	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 226Ch	GPTW2	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2270h	GPTW2	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2274h	GPTW2	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2278h	GPTW2	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 227Ch	GPTW2	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2280h	GPTW2	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2284h	GPTW2	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2288h	GPTW2	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 228Ch	GPTW2	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 22D0h	GPTW2	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 22D4h	GPTW2	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 22ECh	GPTW2	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2300h	GPTW3	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2304h	GPTW3	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2308h	GPTW3	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 230Ch	GPTW3	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2310h	GPTW3	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2314h	GPTW3	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2318h	GPTW3	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 231Ch	GPTW3	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2320h	GPTW3	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2324h	GPTW3	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2328h	GPTW3	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 232Ch	GPTW3	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2330h	GPTW3	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2334h	GPTW3	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2338h	GPTW3	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 233Ch	GPTW3	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2340h	GPTW3	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2348h	GPTW3	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 234Ch	GPTW3	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2350h	GPTW3	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2354h	GPTW3	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (29 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2358h	GPTW3	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 235Ch	GPTW3	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2360h	GPTW3	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2364h	GPTW3	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2368h	GPTW3	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 23D0h	GPTW3	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 23D4h	GPTW3	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 23ECh	GPTW3	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2400h	GPTW4	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2404h	GPTW4	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2408h	GPTW4	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 240Ch	GPTW4	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2410h	GPTW4	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2414h	GPTW4	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2418h	GPTW4	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 241Ch	GPTW4	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2420h	GPTW4	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2424h	GPTW4	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2428h	GPTW4	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 242Ch	GPTW4	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2430h	GPTW4	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2434h	GPTW4	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2438h	GPTW4	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 243Ch	GPTW4	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2440h	GPTW4	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2448h	GPTW4	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 244Ch	GPTW4	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2450h	GPTW4	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2454h	GPTW4	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2458h	GPTW4	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 245Ch	GPTW4	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2460h	GPTW4	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2464h	GPTW4	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2468h	GPTW4	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 24D0h	GPTW4	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 24D4h	GPTW4	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 24ECh	GPTW4	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2500h	GPTW5	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2504h	GPTW5	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2508h	GPTW5	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 250Ch	GPTW5	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2510h	GPTW5	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2514h	GPTW5	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (30 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2518h	GPTW5	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 251Ch	GPTW5	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2520h	GPTW5	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2524h	GPTW5	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2528h	GPTW5	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 252Ch	GPTW5	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2530h	GPTW5	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2534h	GPTW5	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2538h	GPTW5	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 253Ch	GPTW5	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2540h	GPTW5	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2548h	GPTW5	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 254Ch	GPTW5	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2550h	GPTW5	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2554h	GPTW5	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2558h	GPTW5	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 255Ch	GPTW5	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2560h	GPTW5	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2564h	GPTW5	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2568h	GPTW5	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 25D0h	GPTW5	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 25D4h	GPTW5	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 25ECh	GPTW5	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2600h	GPTW6	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2604h	GPTW6	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2608h	GPTW6	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 260Ch	GPTW6	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2610h	GPTW6	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2614h	GPTW6	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2618h	GPTW6	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 261Ch	GPTW6	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2620h	GPTW6	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2624h	GPTW6	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2628h	GPTW6	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 262Ch	GPTW6	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2630h	GPTW6	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2634h	GPTW6	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2638h	GPTW6	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 263Ch	GPTW6	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2640h	GPTW6	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2648h	GPTW6	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (31 / 32)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 264Ch	GPTW6	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2650h	GPTW6	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2654h	GPTW6	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2658h	GPTW6	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 265Ch	GPTW6	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2660h	GPTW6	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2664h	GPTW6	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2668h	GPTW6	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 26D0h	GPTW6	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 26D4h	GPTW6	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 26ECh	GPTW6	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2700h	GPTW7	General PWM Timer Write-Protection Register	GTWP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2704h	GPTW7	General PWM Timer Software Start Register	GTSTR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2708h	GPTW7	General PWM Timer Software Stop Register	GTSTP	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 270Ch	GPTW7	General PWM Timer Software Clear Register	GTCLR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2710h	GPTW7	General PWM Timer Start Source Select Register	GTSSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2714h	GPTW7	General PWM Timer Stop Source Select Register	GTPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2718h	GPTW7	General PWM Timer Clear Source Select Register	GTCSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 271Ch	GPTW7	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2720h	GPTW7	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2724h	GPTW7	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2728h	GPTW7	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 272Ch	GPTW7	General PWM Timer Control Register	GTCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2730h	GPTW7	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2734h	GPTW7	General PWM Timer I/O Control Register	GTIOR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2738h	GPTW7	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 273Ch	GPTW7	General PWM Timer Status Register	GTST	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2740h	GPTW7	General PWM Timer Buffer Enable Register	GTBER	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2748h	GPTW7	General PWM Timer Counter	GTCNT	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 274Ch	GPTW7	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2750h	GPTW7	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2754h	GPTW7	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2758h	GPTW7	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 275Ch	GPTW7	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2760h	GPTW7	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2764h	GPTW7	General PWM Timer Period Setting Register	GTPR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2768h	GPTW7	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 27D0h	GPTW7	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 27D4h	GPTW7	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 27ECh	GPTW7	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000C 2B00h	GPTW	Output Phase Switching Control Register	OPSCR	32	32	4 or 5 PCLKA	2 or 3 ICLK	section 22
000E D000h	CANFD	ECC Control/Status Register	ECCSR	32	32	2 or 3 PCLKA	1 or 2 ICLK	section 34
000E D004h	CANFD	ECC Test Mode Register	ECTMR	16	16	2 or 3 PCLKA	1 or 2 ICLK	section 34

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E D00Ch	CANFD	ECC Decoder Test Data Register	ECTDR	32	32	2 or 3 PCLKA	1 or 2 ICLK	section 34
000E D010h	CANFD	ECC Error Address Register	ECEAR	32	32	2 or 3 PCLKA	1 or 2 ICLK	section 34
007F C090h	FLASH	E2 DataFlash Control Register	DFLCTL	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C100h	FLASH	Flash P/E Mode Control Register	FPMCR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C104h	FLASH	Flash Area Select Register	FASR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C108h	FLASH	Flash Processing Start Address Register L	FSARL	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C110h	FLASH	Flash Processing Start Address Register H	FSARH	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C114h	FLASH	Flash Control Register	FCR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C118h	FLASH	Flash Processing End Address Register L	FEARL	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C120h	FLASH	Flash Processing End Address Register H	FEARH	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C124h	FLASH	Flash Reset Register	FRESETR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C128h	FLASH	Flash Status Register 0	FSTATR0	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C12Ch	FLASH	Flash Status Register 1	FSTATR1	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C130h	FLASH	Flash Write Buffer Register 0	FWB0	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C138h	FLASH	Flash Write Buffer Register 1	FWB1	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C140h	FLASH	Flash Write Buffer Register 2	FWB2	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C144h	FLASH	Flash Write Buffer Register 3	FWB3	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C180h	FLASH	Protection Unlock Register	FPR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C184h	FLASH	Protection Unlock Status Register	FPSR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C1C0h	FLASH	Flash Start-Up Setting Monitor Register	FSCMR	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C1C8h	FLASH	Flash Access Window Start Address Monitor Register	FAWSMR	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C1D0h	FLASH	Flash Access Window End Address Monitor Register	FAWEMR	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C1D8h	FLASH	Flash Initial Setting Register	FISR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C1DCh	FLASH	Flash Extra Area Control Register	FEXCR	8	8	2 or 3 FCLK	2 ICLK	section 46
007F C1E0h	FLASH	Flash Error Address Monitor Register L	FEAML	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C1E8h	FLASH	Flash Error Address Monitor Register H	FEAMH	16	16	2 or 3 FCLK	2 ICLK	section 46
007F C228h	TEMPS	Temperature Sensor Calibration Data Register	TSCDR	16	16	2 or 3 FCLK	2 ICLK	section 42
007F C350h	FLASH	Unique ID Register 0	UIDR0	32	32	2 or 3 FCLK	2 ICLK	section 46
007F C354h	FLASH	Unique ID Register 1	UIDR1	32	32	2 or 3 FCLK	2 ICLK	section 46
007F C358h	FLASH	Unique ID Register 2	UIDR2	32	32	2 or 3 FCLK	2 ICLK	section 46
007F C35Ch	FLASH	Unique ID Register 3	UIDR3	32	32	2 or 3 FCLK	2 ICLK	section 46
007F C3A4h	CTSU	CTSU Trimming Register A	CTSUTRIMA	32	32	2 or 3 FCLK	2 ICLK	section 39
007F C3A8h	CTSU	CTSU Trimming Register B	CTSUTRIMB	32	32	2 or 3 FCLK	2 ICLK	section 39
007F FFB0h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 or 3 FCLK	2 ICLK	section 46
007F FFC0h	FLASH	Memory Wait Cycle Setting Register	MEMWAITR	16	16	2 or 3 FCLK	2 ICLK	section 46
FFFF FF80h	OFSM	Endian Select Register	MDE	32	32	1 ICLK		section 7
FFFF FF88h	OFSM	Option Function Select Register 1	OFS1	32	32	1 ICLK		section 7
FFFF FF8Ch	OFSM	Option Function Select Register 0	OFS0	32	32	1 ICLK		section 7

Note 1. When the register is accessed while the USB is operating, a delay may be generated in accessing.

6. Resets

6.1 Overview

There are seven types of resets: RES# pin reset, power-on reset, voltage monitoring 0 reset, voltage monitoring 1 reset, voltage monitoring 2 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 monitored: VPOR)*1
Voltage monitoring 0 reset	VCC falls (voltage monitored: Vdet0)*1
Voltage monitoring 1 reset	VCC falls (voltage monitored: Vdet1)*1
Voltage monitoring 2 reset	VCC falls (voltage monitored: Vdet2)*1
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Watchdog timer reset	The watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDAb) and section 47, 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 Initialized by Each Reset Source

Target 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	Software Reset
The power-on reset detect flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—*1	✓	—	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—	—
The independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	✓	✓	✓	—	—	—	—	—
Registers related to the 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)	✓	✓	✓	✓	—	—	—	—
The voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF)	✓	✓	✓	✓	✓	—	—	—
Registers related to voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0])	✓	✓	✓	✓	✓	—	—	—
(LVD1CR1, LVD1SR)	✓	✓	✓	✓	✓	—	—	—
The voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF)	✓	✓	✓	✓	✓	✓	—	—
Registers related to voltage monitor function 2 (LVD2CR0, LVCMPCR.EXVCCINP2, LVD2E, LVDLVLR.LVD2LVL[1:0])	✓	✓	✓	✓	✓	✓	—	—
(LVD2CR1, LVD2SR)	✓	✓	✓	✓	✓	✓	—	—
The software reset detect flag (RSTSR2.SWRF)	✓	✓	✓	✓	✓	✓	✓	—
Register related to the realtime clock*2	—	—	—	—	—	—	—	—
Register related to the sub-clock oscillator (SOSCCR, SOMCR)	—	✓	—	—	—	—	—	—
Operating mode*3	✓	✓	—	—	—	—	—	—
Registers other than the above, CPU, and internal state	✓	✓	✓	✓	✓	✓	✓	✓

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

Note 1. Initialized at a power-on.

Note 2. Some control bits are initialized by all types of reset or a power-on reset. For details on the target bits, refer to section 26, Realtime Clock (RTCBa).

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

When a reset is canceled, the reset exception handling starts. For the reset exception handling, see 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): SYSTEM.RSTSR0 0008 C290h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	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
b7 to b4	—	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.

PORF Flag (Power-On Reset Detect Flag)

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

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When resets shown in Table 6.2 occur.
- When PORF is read as 1 and then 0 is written to PORF.

LVD0RF Flag (Voltage Monitoring 0 Reset Detect Flag)

The LVD0RF flag indicates that VCC voltage has fallen below Vdet0.

[Setting condition]

- When Vdet0-level VCC voltage is detected.

[Clearing conditions]

- When resets listed in Table 6.2 occur.
- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that VCC voltage has fallen below Vdet1.

[Setting condition]

- When Vdet1-level VCC voltage is detected.

[Clearing conditions]

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

LVD2RF Flag (Voltage Monitoring 2 Reset Detect Flag)

The LVD2RF flag indicates that VCC voltage has fallen below Vdet2.
[Setting condition]

- When Vdet2-level VCC voltage is detected.

[Clearing conditions]

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

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): SYSTEM.RSTSR1 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 at a power-on.
[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): SYSTEM.RSTSR2 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): SYSTEM.SWRR 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

Writing A501h in the SWRR register resets the MCU. This register is read as 0000h. Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

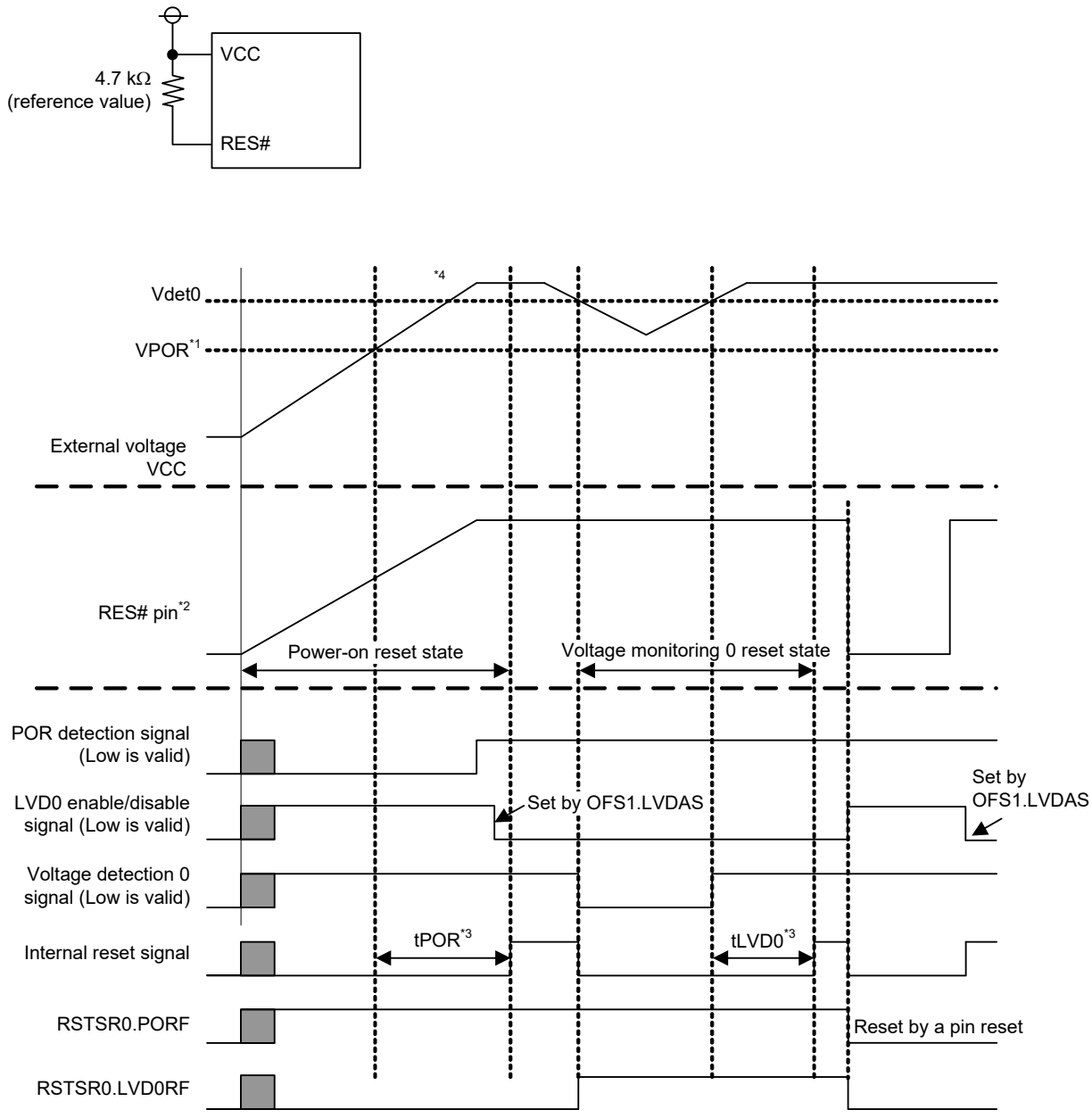
6.3 Operation

6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the LSI enters a reset state.
In order to unfailingly reset the LSI, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancellation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, see section 47, 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. A power-on reset is generated when power is supplied to the RES# pin while it is connected to VCC via a resistor. When connecting a capacitor to the RES# pin, also ensure that the voltage on the RES# pin is always at least VIH. For details on VIH, refer to section 47, Electrical Characteristics. After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is a stabilization period for the external power supply and the MCU circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection circuit 0 start bit (LVDAS) in 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 VDSEL2 and 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 (LVDAb).



Note: For details on the electrical characteristics, see section 47, Electrical Characteristics.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on reset.
Note 2. Ensure that the voltage on the RES# pin is always at least VIH.
Note 3. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.
Note 4. 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 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.

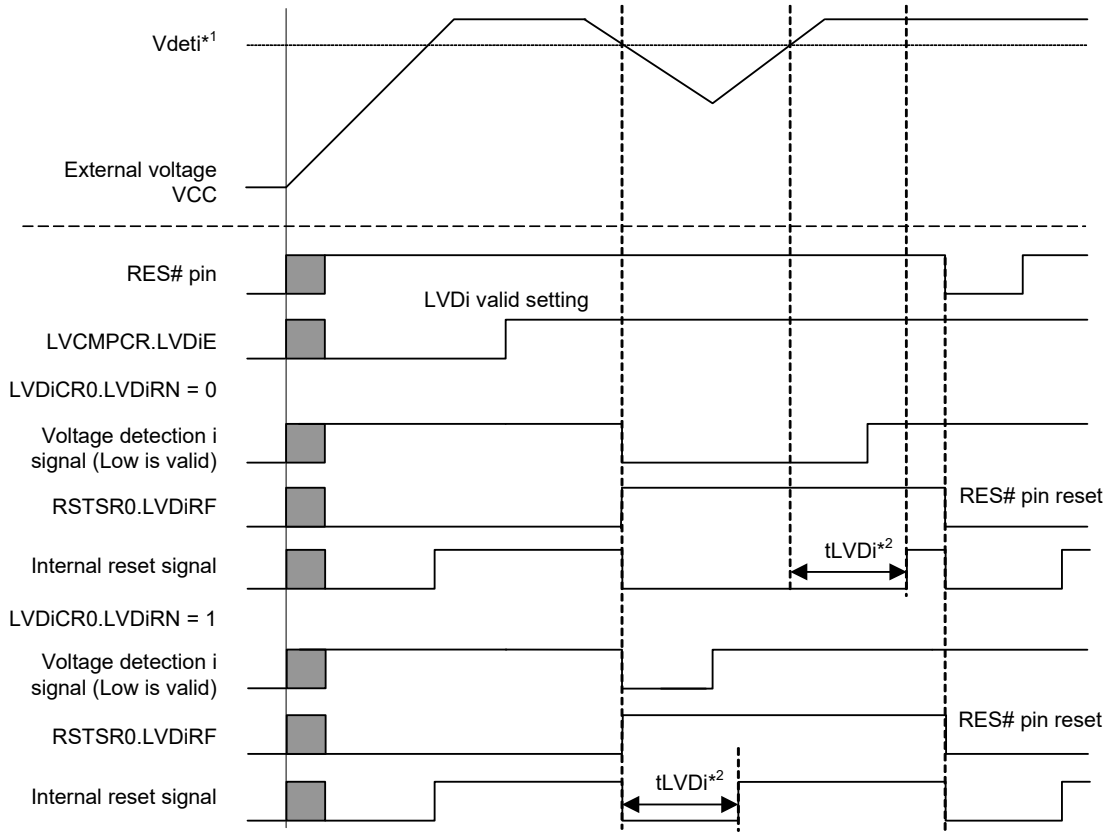
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of 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 negation select bit (LVD1RN) in the LVD1CR0 register. 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/ comparator A2 reset negation select bit (LVD2RN) in the LVD2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection level 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 (LVDAb).



Note: For details on the electrical characteristics, see section 47, 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 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 setting the IWDT reset control register (IWDTRCR) and 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 outside the refresh-permitted period. When the internal reset time (tRESW2) has elapsed after the independent watchdog timer reset has been generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the independent watchdog timer reset, see section 29, Independent Watchdog Timer (IWDTa).

6.3.5 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by settings in the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the watchdog timer reset, see section 28, Watchdog Timer (WDTA).

6.3.6 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.

6.3.7 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.

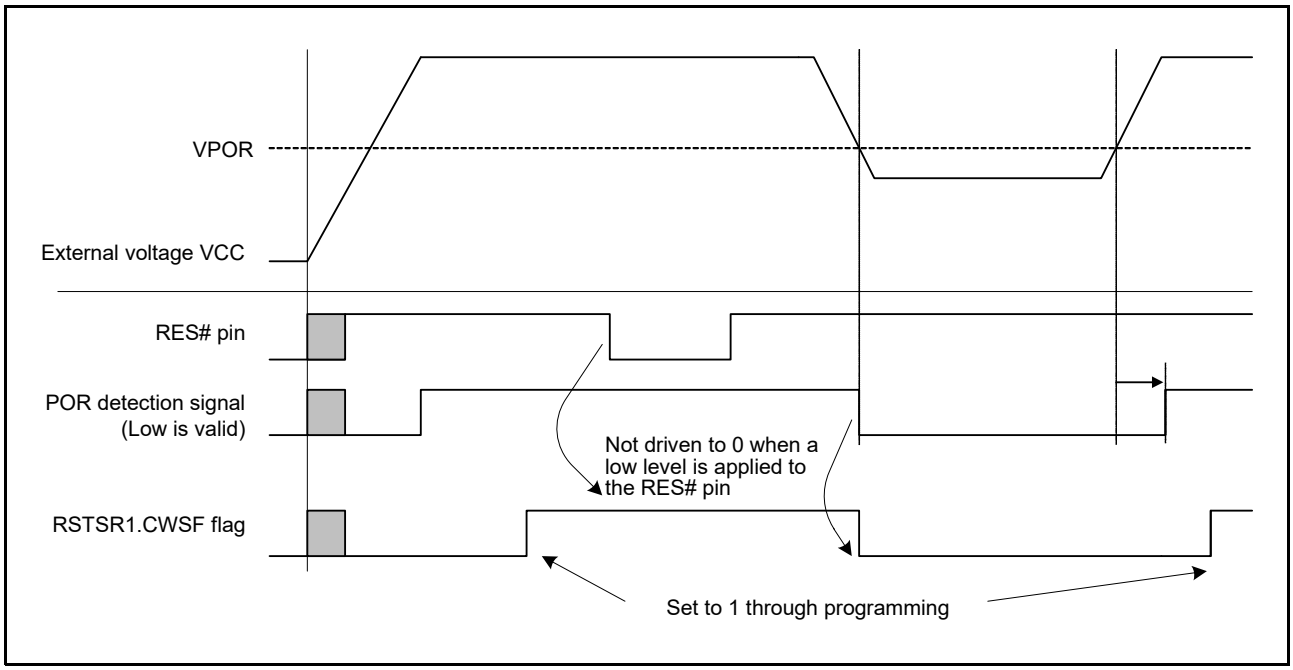


Figure 6.3 Example of Cold/Warm Start Determination Operation

6.3.8 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.

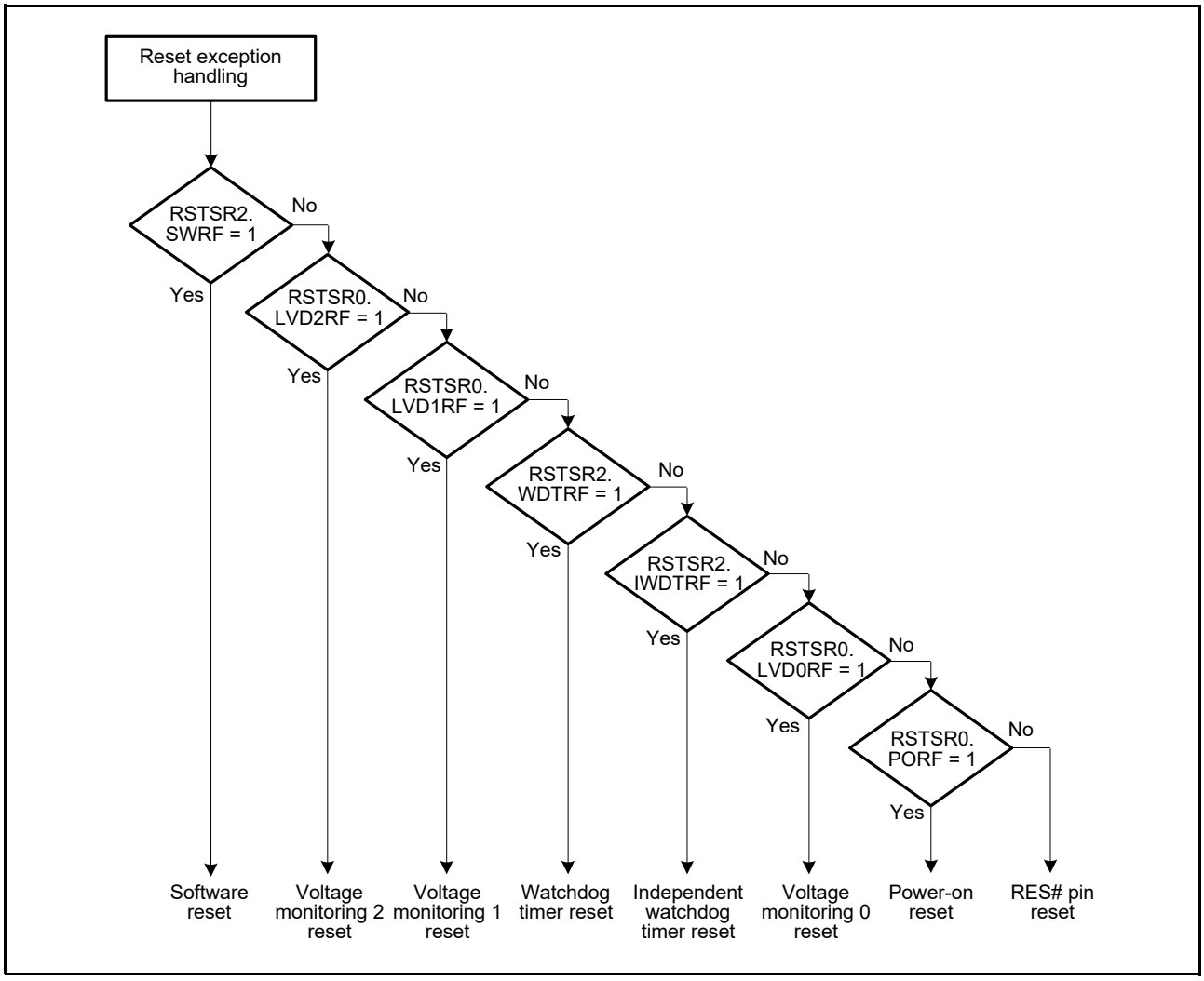


Figure 6.4 Example of Reset Generation Source Determination Flow

7. Option-Setting Memory (OFSM)

7.1 Overview

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

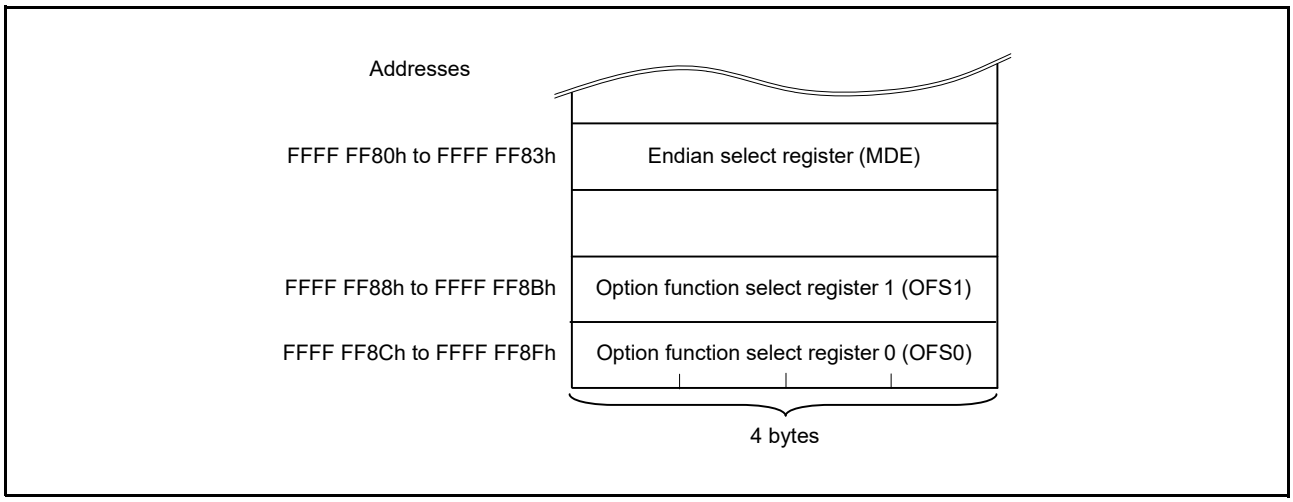


Figure 7.1 Option-Setting Memory Area

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 FFFF FF8Ch

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

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

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: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT 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 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, or deep sleep 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 is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The OFS0 register is allocated in the ROM. Set this register at the same time as writing the program. After writing to the OFS0 register once, do not write to it again.
When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.
The setting in the OFS0 register is ignored in 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 128, 512, 1024, or 2048 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of IWDT-dedicated clock cycles) it takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] bits and IWDTTOPS[1:0] bits.
For details, see section 29, Independent Watchdog Timer (IWDTa).

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

These bits select, from 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256, the division ratio of the prescaler to divide the frequency of the IWDT-dedicated clock. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 128 to 524288 IWDT-dedicated clock cycles.
For details, see 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. Either an independent watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 29, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects whether to stop counting when entering sleep, software standby, or deep sleep mode.
For details, see 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, see 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, see 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. Either a watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 28, Watchdog Timer (WDTA).

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): OFSM.OFS1 FFFF FF88h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	HOCOFRQ[1:0]		—	—	—	HOCOEN	—	—	—	VDSEL2	FASTSTUP	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: 3.85 V is selected 0 1: 2.85 V is selected 1 0: 2.53 V is selected 1 1: 1.90 V is selected	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitoring 0 reset is enabled after a reset 1: Voltage monitoring 0 reset is disabled after a reset	R
b3	FASTSTUP	Power-On Fast Startup Time	0: Fast startup time at power on 1: Normal startup	R
b4	VDSEL2	Voltage Detection 0 Level Select 2	0: 1.69 V is selected 1: The detection level is selected by the VDSEL[1:0] bits	R
b7 to b5	—	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
b11 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b13 to b12	HOCOFRQ[1:0]	HOCO Frequency Select	b13 b12 0 0: 64 MHz is selected 0 1: 48 MHz is selected 1 0: 24 MHz is selected 1 1: 32 MHz is selected	R
b31 to b14	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.
When erasing the block including the OFS1 register, the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ignored in 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 to be monitored by the voltage detection 0 circuit.

LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitoring 0 reset is enabled or disabled after a reset.
The Vdet0 voltage to be monitored by the voltage detection 0 circuit is selected by the VDSEL2 and VDSEL[1:0] bits.

FASTSTUP Bit (Power-On Fast Startup Time)

The startup time can be reduced by setting this bit to 0 (fast startup time at power on) when it is possible to meet the power-on VCC rising gradient (during fast startup time) shown in Electrical Characteristics. Do not set this bit to 0 when it is not possible to meet the power-on VCC rising gradient (during fast startup time).

VDSEL2 Bit (Voltage Detection 0 Level Select 2)

This bit selects the voltage detection level to be monitored by the voltage detection 0 circuit.
When this bit is 1, the settings of the VDSEL[1:0] bits are valid.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation 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 wait 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.
Also, when the HOCOEN bit is set to 0, the HOCO oscillation stabilization time (tHOCO) is secured by hardware, so the clock with the accuracy of the HOCO oscillation frequency (fHOCO) shown in Electrical Characteristics is supplied after release from the CPU reset state.

7.2.3 Endian Select Register (MDE)

Address(es): OFSM.MDE FFFF FF80h

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

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

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

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The MDE register selects the endian for the CPU.
MDE is allocated in the ROM. Set the register at the same time as writing the program. After writing to the register once, do not write to it again.
When erasing the block including the MDE register, the MDE register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

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

7.3 Usage Note

7.3.1 Setting Example of Option-Setting Memory

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

- To set FFFF FFF8h in the OFS0 register

```
.ORG    0FFFFFF8CH
.LWORD  0FFFFFFF8H
```

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

8. Voltage Detection Circuit (LVDAb)

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

8.1 Overview

In voltage detection 0, the detection voltage can be selected from five levels using option function select register 1 (OFS1).
In voltage detection 1, the detection voltage can be selected from 16 levels using the voltage detection level select register (LVDLVLR).
In voltage detection 2, the detection voltage can be selected from four levels using the LVDLVLR register by switching between input voltages to VCC and the CMPA2 pin.
Voltage monitoring 0 reset, voltage monitoring 1 reset/interrupt, and voltage monitoring 2 reset/interrupt 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 LVD Specifications

Item		Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet0	Vdet1	Vdet2
	Detection voltage	Voltage selectable from five levels using OFS1	Voltage selectable from 16 levels using the LVDLVLR.LVD1LVL[3:0] bits	Voltage selectable from four levels using the LVDLVLR.LVD2LVL[1:0] bits
	Monitoring flag	Not available	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 or the CMPA2 pin CPU restart timing selectable: after specified time with VCC or the CMPA2 pin > Vdet2 or after specified time with Vdet2 > VCC or the CMPA2 pin
	Interrupt	Not available	Voltage monitoring 1 interrupt	Voltage monitoring 2 interrupt
			Non-maskable or maskable interrupt is selectable	Non-maskable or maskable interrupt is selectable
			Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either	Interrupt request issued when Vdet2 > VCC or the CMPA2 pin and VCC or the CMPA2 pin > Vdet2 or either
Event link function		Not available	Available Vdet1 passage detection event output	Available Vdet2 passage detection event output

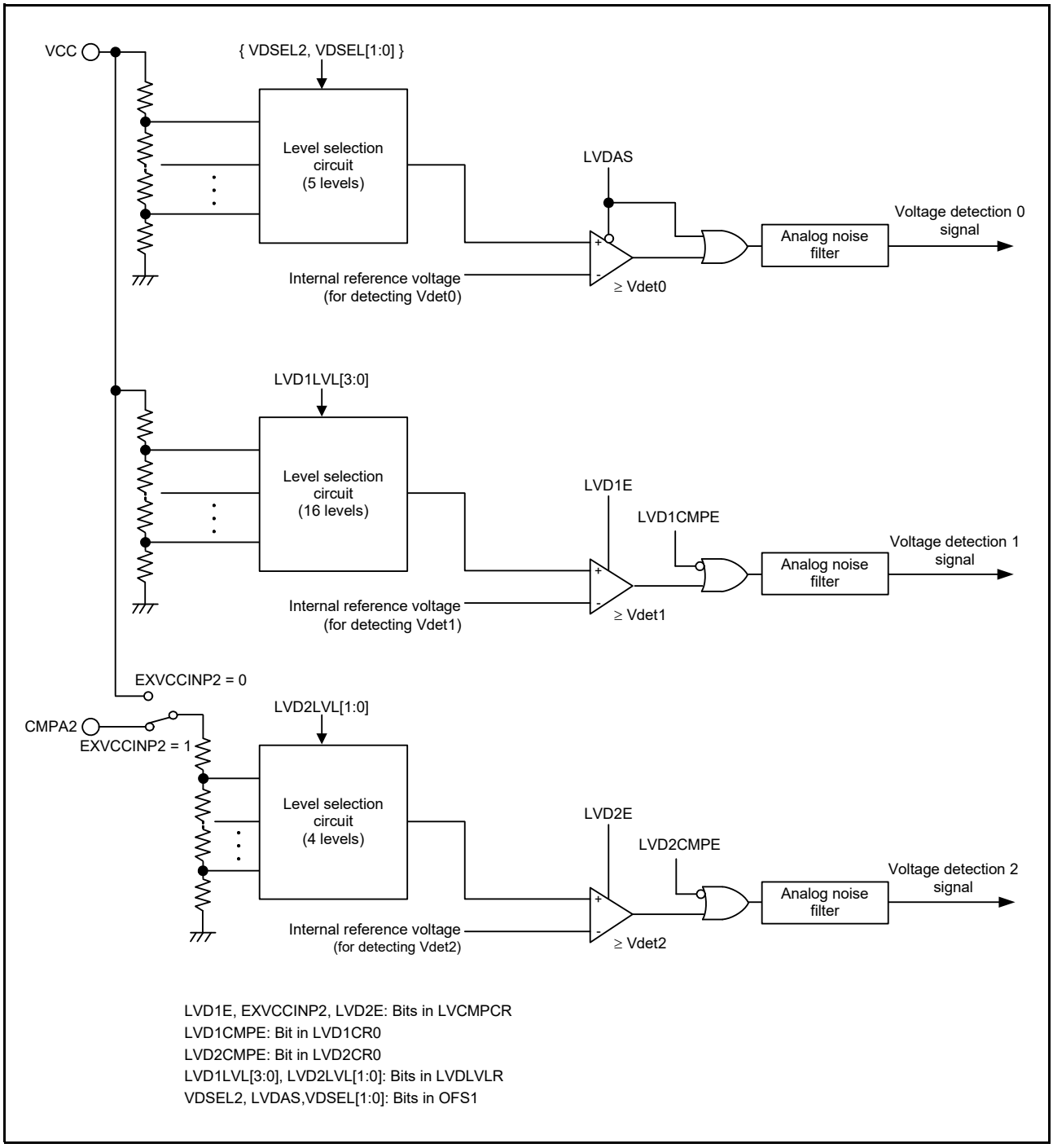


Figure 8.1 Block Diagram of the LVD

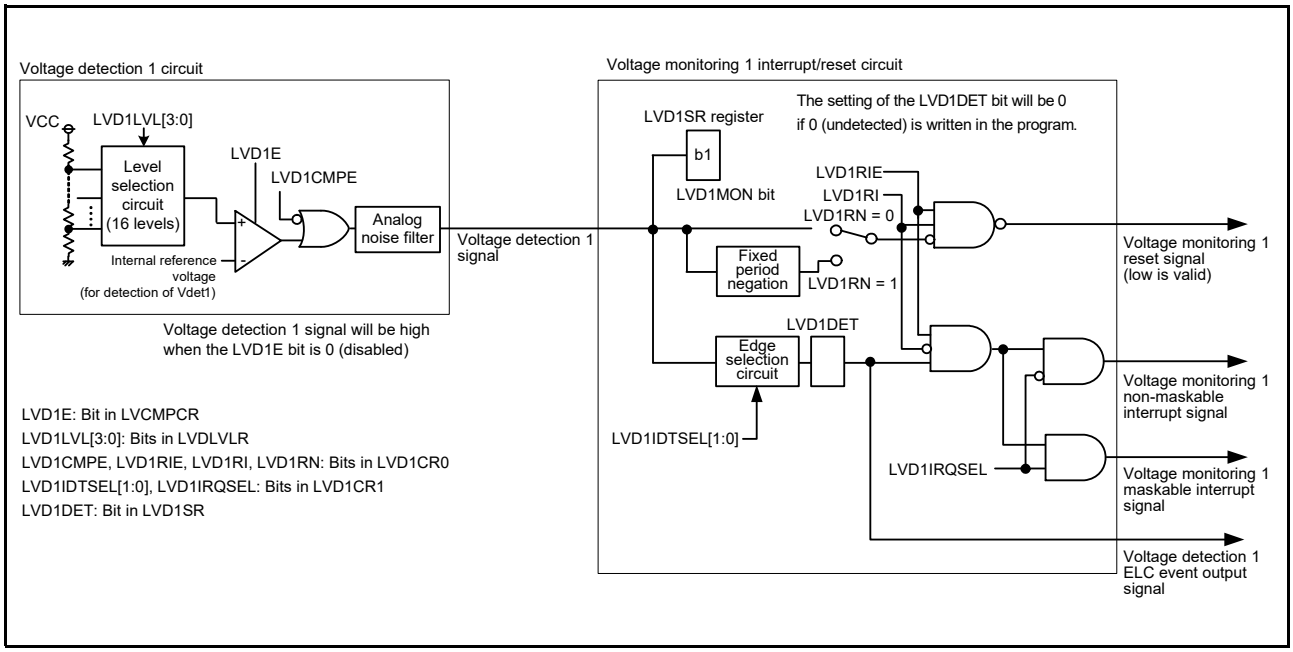


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

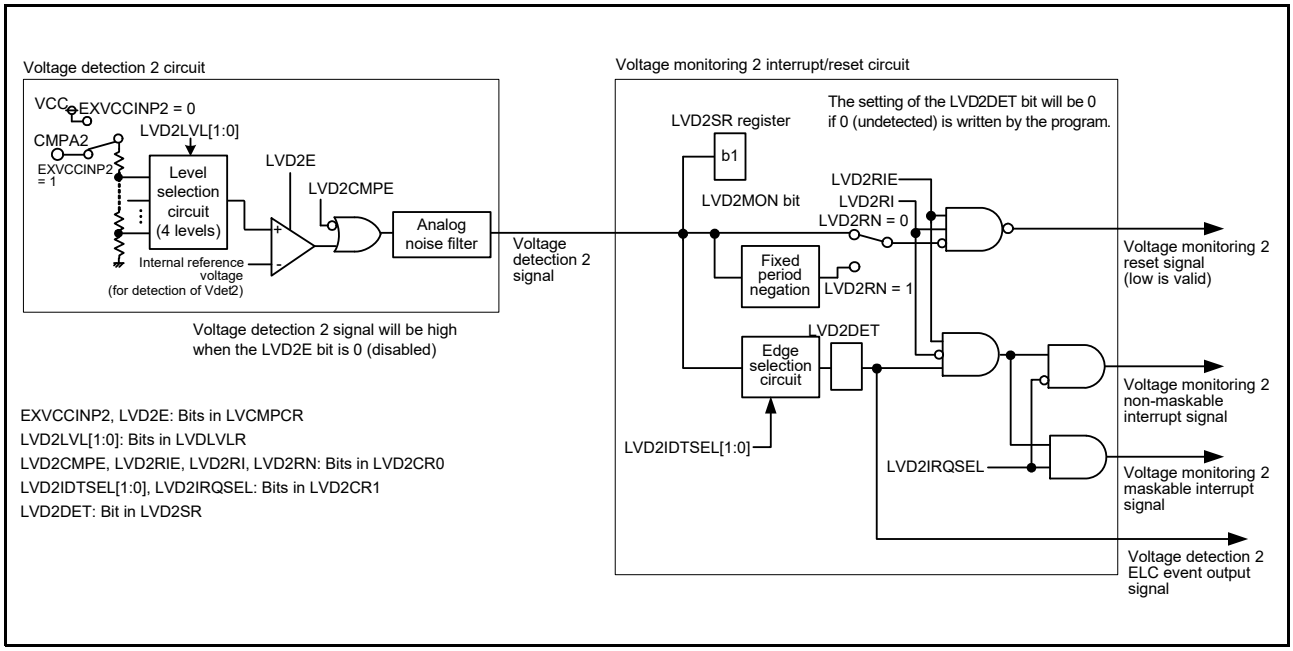


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

Table 8.2 lists the I/O pins relevant to the voltage detection circuit.

Table 8.2 I/O Pins of the Voltage Detection Circuit

Pin Name	I/O	Function
CMPA2	Input	Detection target voltage pin for voltage detection 2

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): SYSTEM.LVD1CR1 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/ELC Event 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: Setting prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): SYSTEM.LVD1SR 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 circuit is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two 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) again after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

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): SYSTEM.LVD2CR1 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/ELC Event Generation Condition Select	b1 b0 0 0: When VCC or the CMPA2 pin ≥ Vdet2 (rise) is detected 0 1: When VCC or the CMPA2 pin < Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): SYSTEM.LVD2SR 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 or the CMPA2 pin < Vdet2 1: VCC or the CMPA2 pin ≥ 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: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two 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 is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) again after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

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

8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): SYSTEM.LVCMPCR 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	EXVCCINP2	—	—	—
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	EXVCCINP2	Voltage Detection 2 Comparison Voltage External Input Select *1	0: Power supply voltage (VCC) 1: CMPA2 pin input voltage	R/W
b4	—	Reserved	This bit is 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

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. The EXVCCINP2 bit can be changed only when the LVD1E and LVD2E bits are both 0 (voltage detection 1 circuit and voltage detection 2 circuit disabled).

LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON flag, 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 flag, 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): SYSTEM.LVDLVLR 0008 C298h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2LVL[1:0]		LVD1LVL[3:0]			
Value after reset:	0	0	0	0	1	0	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 0 0 0: 4.29 V 0 0 0 1: 4.16 V 0 0 1 0: 4.03 V 0 0 1 1: 3.86 V 0 1 0 0: 3.10 V 0 1 0 1: 3.00 V 0 1 1 0: 2.90 V 0 1 1 1: 2.80 V 1 0 0 0: 2.68 V 1 0 0 1: 2.59 V 1 0 1 0: 2.48 V 1 0 1 1: 2.20 V 1 1 0 0: 1.96 V 1 1 0 1: 1.86 V 1 1 1 0: 1.75 V 1 1 1 1: 1.65 V Settings other than those listed above are prohibited.	R/W
b5, b4	LVD2LVL[1:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b5 b4 0 0: 4.32 V 0 1: 4.17 V 1 0: 4.03 V 1 1: 3.84 V	R/W
b7, b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

When changing the LVDLVLR register, first set the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits to 0 (voltage detection n circuit disabled) (n = 1, 2).

When a setting is made so that the voltage detection level range set by the LVD1LVL[3:0] bit overlaps with the range set by the LVD2LVL[1:0] bit register, it cannot be specified which of LVD1 and LVD2 is used for voltage detection. For details on the voltage detection level range, refer to section 47, Electrical Characteristics.

When LVD0 is enabled, set the detection level for LVD1 higher than that of LVD0.

While LVD0 is enabled, only execute changing of the voltage detection level using the LVD1LVL[3:0] bits one time after release from the reset state.

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): SYSTEM.LVD1CR0 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	—	—	—	LVD1CMPE	—	LVD1RIE
Value after reset:	1	0	0	0	X	0	0	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison results output disabled 1: Voltage monitoring 1 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt occurs when the voltage passes Vdet1 1: Voltage monitoring 1 reset occurs when the voltage falls below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 Reset Negation Select	0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected. 1: Negation follows a stabilization time (tLVD1) after assertion of the voltage monitoring 1 reset.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

The LVD1RIE bit is enabled when the LVCMPCR.LVD1E bit is set to 1 (voltage detection 1 circuit enabled) and the LVD1CMPE bit is set to 1 (voltage monitoring 1 circuit comparison results output enabled).

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

LVD1RN Bit (Voltage Monitoring 1 Reset Negation Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, 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 voltage monitoring 1 reset).

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): SYSTEM.LVD2CR0 0008 C29Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2RN	LVD2RI	—	—	—	LVD2CMPE	—	LVD2RIE
Value after reset:	1	0	0	0	X	0	0	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison results output disabled 1: Voltage monitoring 2 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 Negation Select	0: Negation follows a stabilization time (tLVD2) after VCC or the CMPA2 pin > Vdet2 is detected. 1: Negation follows a stabilization time (tLVD2) after assertion of the voltage monitoring 2 reset.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

The LVD2RIE bit is enabled when the LVCMPCR.LVD2E bit is set to 1 (voltage detection 2 circuit enabled) and the LVD2CMPE bit is set to 1 (voltage monitoring 2 circuit comparison results output enabled).

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

LVD2RN Bit (Voltage Monitoring 2 Reset Negation Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after VCC or the CMPA2 pin > Vdet2 is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset).

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

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

(1) Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits (voltage detection 1 level select).
(2) Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).
(3) After waiting for td(E-A), set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).

8.3.3 Monitoring Vdet2

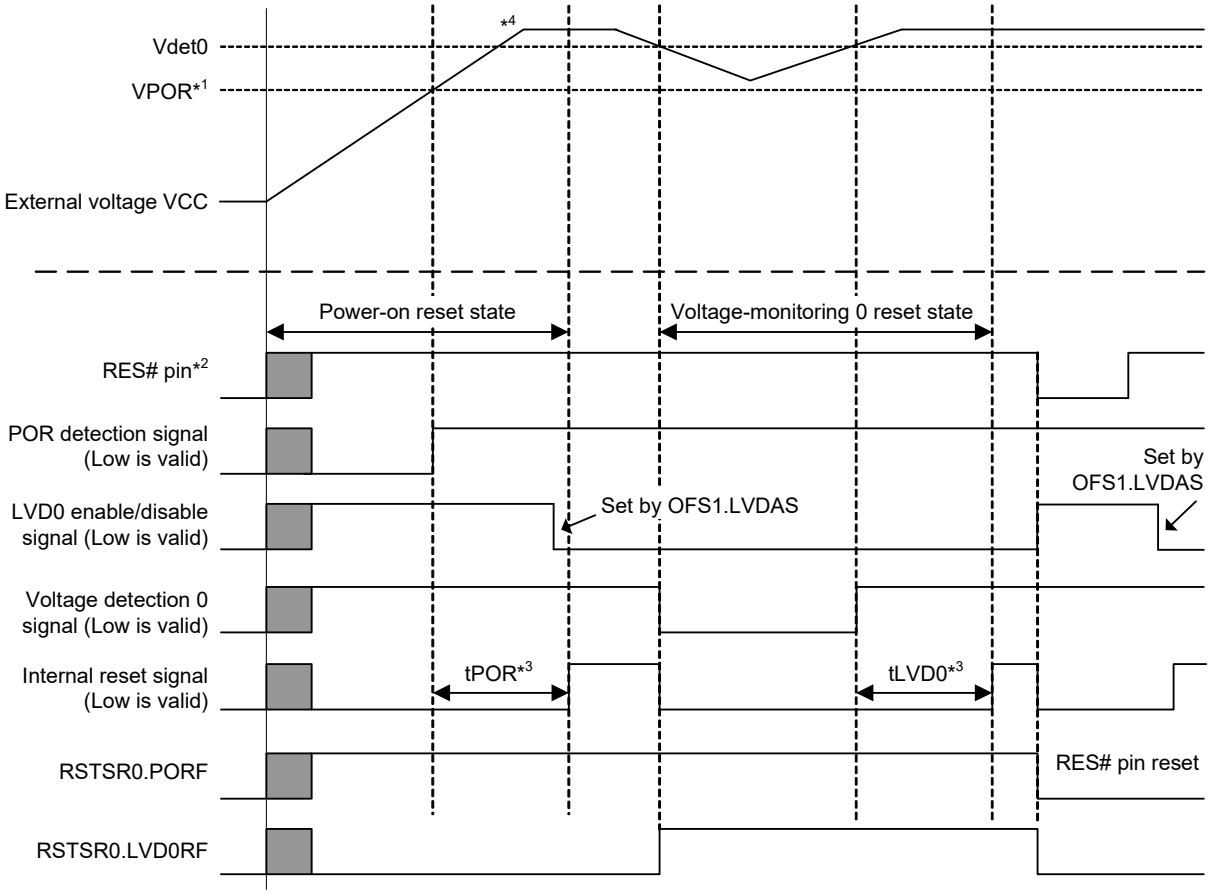
After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by voltage monitor 2.

(1) Specify the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits (voltage detection 2 level).
(2) Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or 1 (CMPA2 pin input voltage).
(3) Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).
(4) After waiting for td(E-A), set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).

8.4 Reset from Voltage Monitor 0

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

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



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on reset.
Note 2. Ensure that the voltage on the RES# pin is always at least VIH.
Note 3. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.
Note 4. 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 Monitoring 1

Table 8.3 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.4 shows the procedures for stopping bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Figure 8.5 shows an example of operations for a voltage monitoring 1 interrupt. For the operation of the voltage monitoring 1 reset, see Figure 6.2 in section 6, Resets.

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

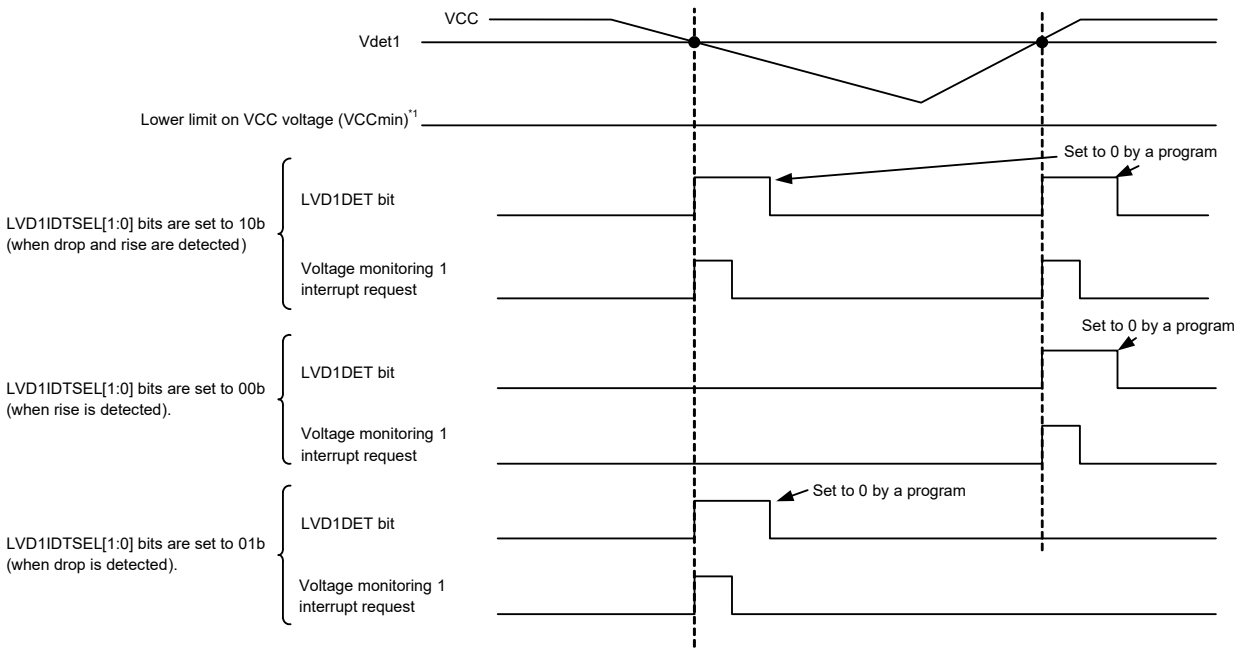
Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
2*1	Set the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.
3	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.	—
4	—	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).
5*1	Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).	
6*1	Wait for at least td(E-A).	
7	Set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).	
8	Wait for at least 2 µs.	—
9	Set the LVD1SR.LVD1DET bit to 0.	—
10	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).	—

Note 1. Steps 1, 2, 5, and 6 are not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 10.

Table 8.4 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).	—
2	Set the LVD1CR0.LVD1CMPE bit to 0 (voltage monitoring 1 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD1E bit to 0 (voltage detection 1 circuit disabled).	
4	—	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD1E, LVD1CR0.LVD1RIE, and LVD1CR0.LVD1CMPE.	

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.



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 Monitoring 2

Table 8.5 shows the procedures for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Table 8.6 shows the procedure for stopping bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Figure 8.6 shows an example of operations for a voltage monitoring 2 interrupt. For the operation of the voltage monitoring 2 reset, see Figure 6.2 in section 6, Resets.

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

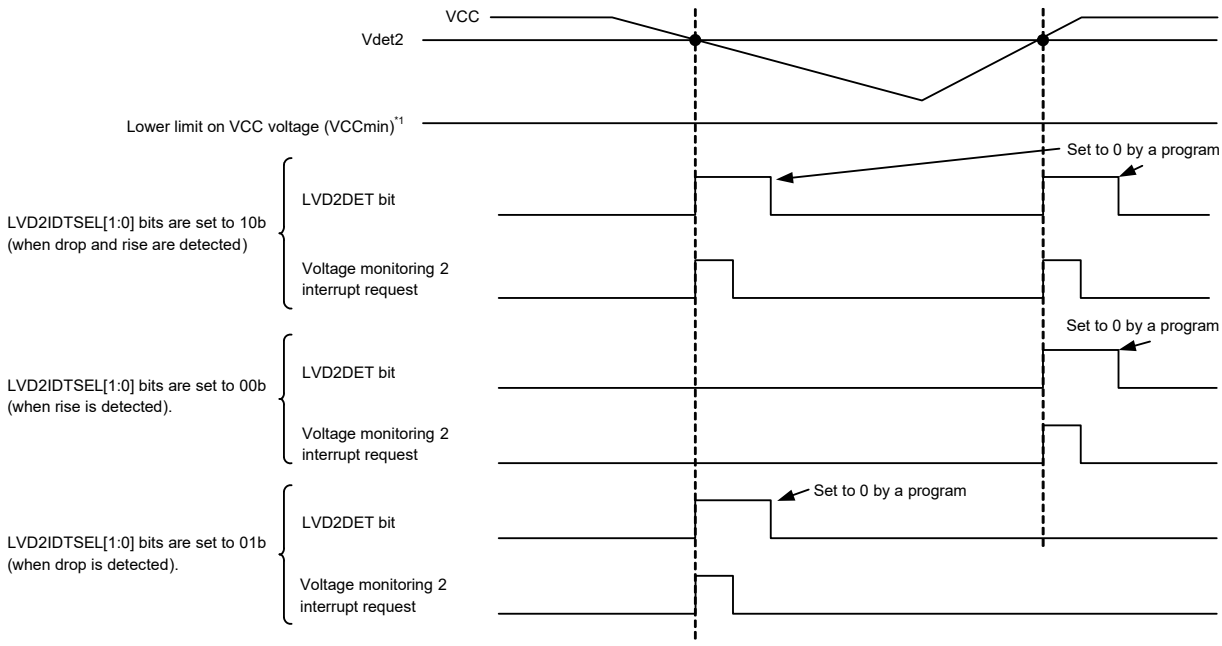
Step	Voltage Monitoring 2 Interrupt Voltage Monitoring 2 ELC Event Output	Voltage Monitoring 2 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits.	
2*1	Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or set it to 1 (CMPA2 pin input voltage).	
3*1	Set the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt).	Set the LVD2CR0.LVD2RI bit to 1 (voltage monitoring 2 reset). Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.
4	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—
5	—	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled).
6*1	Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).	
7*1	Wait for at least td(E-A).	
8	Set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).	
9	Wait for at least 2 µs.	—
10	Set the LVD2SR.LVD2DET bit to 0.	—
11	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled)	—

Note 1. Steps 1, 2, 3, 6, and 7 are not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 11.

Table 8.6 Procedures for Stopping Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset

Step	Voltage Monitoring 2 Interrupt Voltage Monitoring 2 ELC Event Output	Voltage Monitoring 2 Reset
1	Set the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled).	—
2	Set the LVD2CR0.LVD2CMPE bit to 0 (voltage monitoring 2 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD2E bit to 0 (voltage monitoring 2 circuit disabled).	
4	—	Set the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD2E, LVD2CR0.LVD2RIE, and LVD2CR0.LVD2CMPE.	

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 5.



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

Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

8.7 Event Link Output

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

(1) Vdet1 passage detection event output
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 1 circuit comparison result output are enabled.

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

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

8.7.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt request signal 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 1 and 2 interrupts in software standby. The event signals for the ELC, however, are output as follows:

- When the events of passing 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 the 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.

8.8 Usage Note

8.8.1 Note on Setting the Detection Level for LVD1 While LVD0 Is Enabled

While LVD0 is to be enabled, set the detection level for LVD1 to a value higher than that for LVD0. Moreover, once LVD0 has been enabled after release from the reset state, using the LVD1LVL[3:0] bits to change the voltage detection level for LVD1 is only allowed to proceed once.

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
Uses	• Generates the system clock (ICLK) to be supplied to the CPU, DMAC, DTC, ROM, and RAM. • Generates the peripheral module clock (PCLKA) to be supplied to the CANFD (message buffer RAM) and GPTW. • 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 FlashIF clock (FCLK) to be supplied to the FlashIF. • Generates the USB clock (UCLK) to be supplied to the USB. • 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 LPT clock (LPTCLK) to be supplied to the LPT. • Generates the REMC clock (REMCLK) to be supplied to the REMC. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the RTC clock (RTCSCLK) to be supplied to the RTC. • Generates the IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT.
Operating frequencies*1	• ICLK: 64 MHz (max)*2 • PCLKA: 64 MHz (max)*2, *3 • PCLKB: 32 MHz (max)*2, *3 • PCLKD: 64 MHz (max)*2 • FCLK*2: 1 to 64 MHz (for programming and erasing the ROM and E2 DataFlash) 64 MHz (max) (for reading from the E2 DataFlash) • UCLK: 48 MHz • CANFDCLK: 32 MHz (max)*4 • CANFDMCLK: 20 MHz (max)*4 • LPTCLK: Depends on the selected source clock. 32.768 kHz (sub-clock), 15 kHz (IWDT-dedicated clock, IWDTCLK), 1 MHz (LOCO clock frequency-divided by 4) • REMCLK: Same frequency as that of the selected oscillator • CACCLK: Same frequency as that of the selected oscillator • RTCSCLK: 32.768 kHz • IWDTCLK: 15 kHz
Main clock oscillator	• Resonator frequency: 1 to 20 MHz • External clock input frequency: 20 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal • Connection pins: EXTAL, XTAL • Oscillation stop detection function: When stopping of the main clock oscillation is detected, the system clock source is switched to the LOCO, and GPTW output can forcibly be driven to the high-impedance state. • Drive capacity switching function
Sub-clock oscillator	• Resonator frequency: 32.768 kHz • External clock input frequency: 32.768 kHz • Connectable resonator or additional circuit: crystal • Connection pin: XCIN, XCOUT • Sub-clock external input pin: EXCIN • Drive capacity switching function
PLL circuit	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 to 12.5 MHz • Frequency multiplication ratio: Selectable from 4 to 15.5 (increments of 0.5) • VCO oscillation frequency: 24 to 64 MHz (VCC ≥ 1.8 V)

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

Item	Specification
PLL2 circuit	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 to 12.5 MHz • Frequency multiplication ratio: Selectable from 4 to 15.5 (increments of 0.5) • VCO oscillation frequency: 24 to 64 MHz (VCC ≥ 1.8 V)
High-speed on-chip oscillator (HOCO)	Oscillation frequency: 24 MHz, 32 MHz, 48 MHz, 64 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 4 MHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 15 kHz

Note 1. The maximum operating frequency in high-speed operating mode. For the maximum operating frequency in the other operating modes, refer to section 11.2.6, Operating Power Control Register (OPCCR).
Note 2. The division ratio relationship of frequencies must be set as follows.
ICLK:FCLK = N:1 or 1:N, ICLK:PCLKA, PCLKB, PCLKD = N:1 or 1:N (N is an integer).
Note 3. For the peripheral module clocks when the CANFD module is not in use, ensure that the setting of the PCLKA frequency is no lower than that of PCLKB (PCLKA ≥ PCLKB).
Note 4. Ensure that the settings of the clock frequencies satisfy the following relationships when the CANFD module is in use.
PCLKA:PCLKB = 2:1
PCLKB ≥ CANFDCLK
PCLKB ≥ CANFDMCLK

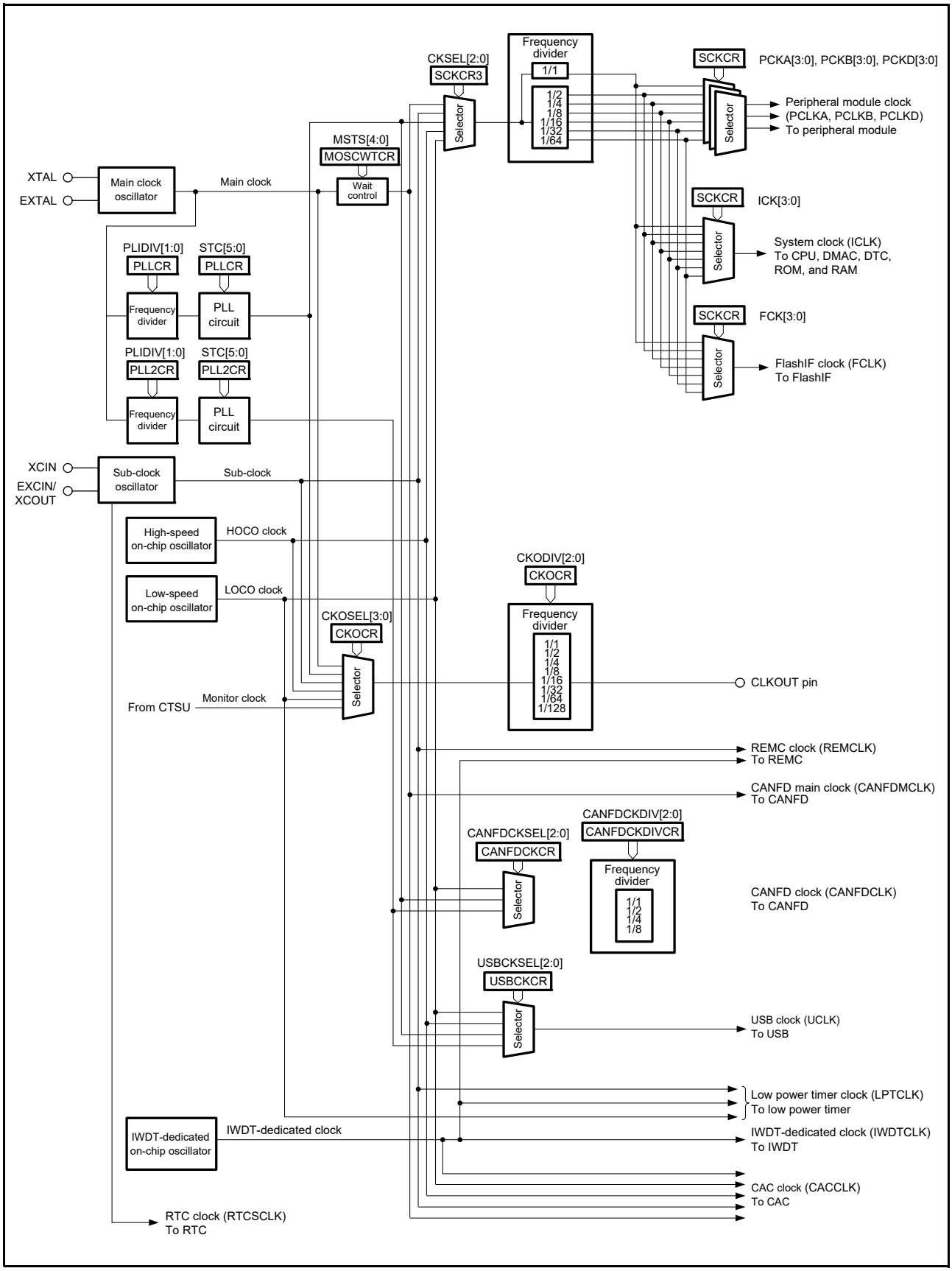


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.2 lists the I/O pins of the clock generation circuit.

Table 9.2 I/O Pins of Clock Generation Circuit

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal. The EXTAL pin can also be used to input an external clock. For details, refer to section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal. The EXCIN pin can also be used to input an external clock. For details, refer to section 9.4.2, External Clock Input.
XCOUT/EXCIN	I/O	
CLKOUT	Output	Clock output pin

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): SYSTEM.SCKCR 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FCK[3:0]				ICK[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	1	1	0	0	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
	PCKA[3:0]				PCKB[3:0]				—	—	—	—	PCKD[3:0]			
Value after reset:	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1

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 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	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select	b11 b8 0 0 0 0: ×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	b11 b8 0 0 0 0: ×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
b23 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select	b27 b24 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	FCK[3:0]	FlashIF Clock (FCLK) Select*1	b31 b28 0 0 0 0: ×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: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Selection of a clock with a frequency higher than 32 MHz for the system clock (ICLK) requires a setting for the insertion of cycles of waiting for ROM access. For details, refer to section 46, Flash Memory (FLASH).

This register cannot be rewritten while the flash memory is being programmed or erased.

PCKD[3:0] Bits (Peripheral Module Clock D (PCLKD) Select)

These bits select the frequency of peripheral module clock D (PCLKD).

PCKB[3:0] Bits (Peripheral Module Clock B (PCLKB) Select)

These bits select the frequency of peripheral module clock B (PCLKB).

PCKA[3:0] Bits (Peripheral Module Clock A (PCLKA) Select)

These bits select the frequency of peripheral module clock A (PCLKA).

ICK[3:0] Bits (System Clock (ICLK) Select)

These bits select the frequency of the system clock (ICLK).

FCK[3:0] Bits (FlashIF Clock (FCLK) Select*1)

These bits select the frequency of the FlashIF clock (FCLK).

9.2.2 System Clock Control Register 3 (SCKCR3)

Address(es): SYSTEM.SCKCR3 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

This register cannot be rewritten while the flash memory is being programmed or erased.

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

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) 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.

In middle-speed operating mode 2, do not select the main clock oscillator as the clock source.

In low-speed operating mode, only the sub-clock oscillator is selectable as the clock source.

9.2.3 PLL Control Register (PLLCR)

Address(es): SYSTEM.PLLCR 0008 0028h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 0 1 1 1: ×4 0 0 1 0 0 0: ×4.5 0 0 1 0 0 1: ×5 0 0 1 0 1 0: ×5.5 0 0 1 0 1 1: ×6 0 0 1 1 0 0: ×6.5 0 0 1 1 0 1: ×7 0 0 1 1 1 0: ×7.5 0 0 1 1 1 1: ×8 0 1 0 0 0 0: ×8.5 0 1 0 0 0 1: ×9 0 1 0 0 1 0: ×9.5 0 1 0 0 1 1: ×10 0 1 0 1 0 0: ×10.5 0 1 0 1 0 1: ×11 0 1 0 1 1 0: ×11.5 0 1 0 1 1 1: ×12 0 1 1 0 0 0: ×12.5 0 1 1 0 0 1: ×13 0 1 1 0 1 0: ×13.5 0 1 1 0 1 1: ×14 0 1 1 1 0 0: ×14.5 0 1 1 1 0 1: ×15 0 1 1 1 1 0: ×15.5 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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

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

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

Set these bits so that the frequency of PLL input signal is within the range of 4 MHz to 12.5 MHz.

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

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

Set these bits so that the PLL oscillation frequency is within the range of 24 MHz to 64 MHz.

9.2.4 PLL Control Register 2 (PLLCR2)

Address(es): SYSTEM.PLLCR2 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.
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.

- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 before restarting the PLL.
- Confirm that the PLL is operating and that the OSCOVFSR.PLOVF bit is 1 before stopping the PLL.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.PLOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

Setting the PLLEN bit to 1 (stopping the PLL circuit) is prohibited while the PLL clock is selected by the SCKCR3.CKSEL[2:0] bits.
Setting the PLLEN bit to 0 (operating the PLL circuit) is prohibited while VCC is less than 1.8 V, or the low-speed operating mode is selected by the SOPCCR.SOPCM bit.
Setting the PLLEN bit to 1 (stopping the PLL circuit) is prohibited while the PLL is selected for UCLK by the USBCKCR.USBCKSEL[2:0] bits, and the MSTPCRB.MSTPB19 bit is 0 (enabling the module clock).
Setting the PLLEN bit to 1 (stopping the PLL circuit) is prohibited while the PLL is selected for CANFDCLK by the CANFDCKCR.CANFDCKSEL[2:0] bits, and the MSTPCRD.MSTPD9 bit is 0 (enabling the module clock).

9.2.5 PLL2 Control Register (PLL2CR)

Address(es): SYSTEM.PLL2CR 0008 002Ch

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL2 Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 0 1 1 1: ×4 0 0 1 0 0 0: ×4.5 0 0 1 0 0 1: ×5 0 0 1 0 1 0: ×5.5 0 0 1 0 1 1: ×6 0 0 1 1 0 0: ×6.5 0 0 1 1 0 1: ×7 0 0 1 1 1 0: ×7.5 0 0 1 1 1 1: ×8 0 1 0 0 0 0: ×8.5 0 1 0 0 0 1: ×9 0 1 0 0 1 0: ×9.5 0 1 0 0 1 1: ×10 0 1 0 1 0 0: ×10.5 0 1 0 1 0 1: ×11 0 1 0 1 1 0: ×11.5 0 1 0 1 1 1: ×12 0 1 1 0 0 0: ×12.5 0 1 1 0 0 1: ×13 0 1 1 0 1 0: ×13.5 0 1 1 0 1 1: ×14 0 1 1 1 0 0: ×14.5 0 1 1 1 0 1: ×15 0 1 1 1 1 0: ×15.5 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Writing to the PLL2CR is prohibited when the PLL2CR2.PLL2EN bit is 0 (PLL2 is operating).

PLIDIV[1:0] Bits (PLL2 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL2 clock source.

Set these bits so that the frequency of PLL2 input signal is within the range of 4 MHz to 12.5 MHz.

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

These bits select the frequency multiplication factor of the PLL2 circuit.

Set these bits so that the PLL2 oscillation frequency is within the range of 24 MHz to 64 MHz.

9.2.6 PLL2 Control Register 2 (PLL2CR2)

Address(es): SYSTEM.PLL2CR2 0008 002Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLL2EN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLL2EN	PLL2 Stop Control	0: PLL2 is operating. 1: PLL2 is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

PLL2EN Bit (PLL2 Stop Control)

This bit runs or stops the PLL2 circuit.
After the setting of the PLL2EN bit has been changed to make the PLL2 run, only start using the PLL2 clock after confirming that the OSCOVFSR.PL2OVF flag has been set to 1.
That is, a fixed time for stabilization is required after the setting for PLL2 operation having been made. A fixed time is also required for oscillation to stop after the setting to stop PLL2 operation having been made. Accordingly, take note of the following limitations when starting and stopping PLL2 operation.

- After stopping the PLL2, confirm that the OSCOVFSR.PL2OVF bit is 0 before restarting the PLL2.
- Confirm that the PLL2 is operating and that the OSCOVFSR.PL2OVF bit is 1 before stopping the PLL2.
- Confirm that the OSCOVFSR.PL2OVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the PLL2, confirm that the OSCOVFSR.PL2OVF bit is 0 and execute a WAIT instruction before entering software standby mode.

Setting the PLL2EN bit to 0 (operating the PLL2 circuit) is prohibited while VCC is less than 1.8 V, or the low-speed operating mode is selected by the SOPCCR.SOPCM bit.
Setting the PLL2EN bit to 1 (stopping the PLL2 circuit) is prohibited while the PLL2 is selected for UCLK by the USBCKCR.USBCKSEL[2:0] bits, and the MSTPCRB.MSTPB19 bit is 0 (enabling the module clock).
Setting the PLL2EN bit to 1 (stopping the PLL2 circuit) is prohibited while the PLL2 is selected for CANFDCLK by the CANFDCKCR.CANFDCKSEL[2:0] bits, and the MSTPCRD.MSTPD9 bit is 0 (enabling the module clock).

9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): SYSTEM.MOSCCR 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Set this register after setting up the main clock oscillator wait control register.

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

After setting the MOSTP bit to 0 (main clock oscillator is operating), read the OSCOVFSR.MOOVF bit to confirm that is has become 1, and then use the main clock.

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.

- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 before restarting the main clock oscillator.
- Confirm that the main clock oscillator is operating and that the OSCOVFSR.MOOVF bit is 1 before stopping the main clock oscillator.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF bit is 1 and execute a WAIT instruction in order to operate the main clock oscillator and place the MCU in software standby mode.
- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

Do not set the MOSTP bit to 1 when any of the following conditions is met.

- When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)
- When the PLL clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 100b)
- When the PLL is operating (PLLCR2.PLLEN = 0)
- When the PLL2 is operating (PLL2CR2.PLL2EN = 0)

Do not set the MOSTP bit to 0 when the following condition is met.

- When low-speed operating mode is selected by the SOPCCR.SOPCM bit

9.2.8 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): SYSTEM.SOSCCR 0008 0033h

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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit will not be initialized by the source other than power-on resetting.

SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.
When changing the value of the SOSTP 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).
To use the sub-clock, the sub-clock oscillator mode control register (SOMCR) must be set before setting the SOSTP bit to 0. After the setting of the SOSTP bit has been changed so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization time (t_{SUBOSC}) has elapsed.
That is, a fixed time for stabilization is required after the setting for sub-clock oscillator operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

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

While the sub-clock oscillator is selected by the SCKCR3.CKSEL[2:0] bits, do not set the SOSTP bit to 1 (sub-clock oscillator is stopped).

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

Address(es): SYSTEM.LOCOCR 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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 clock after the LOCO clock oscillation stabilization time (t_{LOCO}) has elapsed.

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

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby mode 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.
- To switch between starting and stopping of the LOCO on the LCSTP bit, do so while the oscillators other than LOCO are stable or stopped.

While the LOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the LCSTP bit to 1 (LOCO is stopped).

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the LCSTP bit to 0 (enabling the LOCO operation).

When the clock source is to be switched by the ELC, set the LCSTP bit to 0 (enabling the LOCO operation) before enabling the ELC.

When the LOCO is selected for UCLK by the USBCKCR.USBCKSEL[2:0] bits, do not set the MSTPCRB.MSTPB19 bit to 0 (enabling the module clock).

Setting the LCSTP bit to 1 (stopping the LOCO circuit) is prohibited while the LOCO is selected for CANFDCLK by the CANFDCKCR.CANFDCKSEL[2:0] bits, and the MSTPCRD.MSTPD9 bit is 0 (enabling the module clock).

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

Address(es): SYSTEM.ILOCOCR 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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

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

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

After the setting of the ILCSTP bit has been changed so that the IWDT-dedicated on-chip oscillator operates, supply of the clock is started the MCU internally after a fixed time corresponding to the IWDT-dedicated clock oscillation stabilization time (t_{ILOCO}) has elapsed.

If the IWDT-dedicated clock is to be used, only start using the oscillator after this wait time has elapsed.

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

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

Address(es): SYSTEM.HOCOCR 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: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 1.

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, only start using the HOCO clock after confirming that the OSCOVFSR.HCOVF flag has been set to 1.
Ensure that the high-speed on-chip oscillator is operating while this MCU is placed in the programming and erasing mode for the flash memory.
That is, a fixed time for stabilization of oscillation is required after the setting for HOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF flag is 0 before restarting the HOCO.
- Confirm that the HOCO is operating and that the OSCOVFSR.HCOVF flag is 1 before stopping the HOCO.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF flag is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF flag is 0 and execute a WAIT instruction before entering software standby mode.
- Make sure that the high-speed on-chip oscillator is operating while this MCU is placed in the programming and erasing mode for the ROM and E2 DataFlash.

Setting the HCSTP bit to 1 (stopping the HOCO) is prohibited while the HOCO is selected by the SCKCR3.CKSEL[2:0] bits.
Setting the HCSTP bit to 0 (running the HOCO) is prohibited while low-speed operating mode is selected by the SOPCCR.SOPCM bit.
Setting the HCSTP bit to 1 (stopping the HOCO) is prohibited while the HOCO is selected for UCLK by the USBCKCR.USBCKSEL[2:0] bits, and the MSTPCRB.MSTPB19 bit is 0 (enabling the module clock).

9.2.12 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): SYSTEM.OSCOVFSR 0008 003Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PL2OVF	—	HCOVF	PLOVF	—	MOOVF
Value after reset:	0	0	0	0	0/1*1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: Main clock is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock*2	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock	R
b3	HCOVF	HOCO Clock Oscillation Stabilization Flag	0: HOCO is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock	R
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	PL2OVF	PLL2 Clock Oscillation Stabilization Flag	0: PLL2 is stopped or not stabilized 1: Oscillation is stable and the clock can be used	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The HCOVF flag value after a reset is 1 when the HOCO oscillation enable bit in option function selection register 1 (OFS1.HOCOEN) is 0. The HCOVF flag value after a reset is 1 when the OFS1.HOCOEN bit is 0.
Note 2. When an appropriate value is set in the wait control register for each oscillator. If a set value (wait time) is not adequate, clock supply starts before oscillation becomes stable.

The OSCOVFSR register monitors whether oscillation of each oscillator has become stable.
If a wait control register is provided for each oscillator, specify a wait time that is longer than or equal to the stabilization time of the corresponding oscillation circuit.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the main clock is stable.
[Setting condition]

- After the MOSCCR.MOSTP bit is set to 0 (main clock oscillator is operating) when the MOSTP bit is 1 (main clock oscillator is stopped), the corresponding time set in the MOSCWTCR register has elapsed and supply of the main clock is started to the MCU internally.

[Clearing condition]

- After the MOSCCR.MOSTP bit is set to 1, the processing to stop the oscillation of the main clock oscillator is completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the PLL clock is stable.
[Setting condition]

- After the PLLCR2.PLLEN is set to 0 (PLL is operating) when the PLLEN bit is 1 (PLL is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (tPLL) has elapsed, and supply of the PLL clock is started to the MCU internally.

[Clearing condition]

- After the PLLCR2.PLLEN bit is set to 1, the processing to stop the oscillation of the PLL is completed.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the HOCO clock is stable.

[Setting condition]

- After the HOCOCR.HCSTP bit is set to 0 (HOCO is operating) when the HCSTP bit is 1 (HOCO is stopped), supply of the HOCO clock is started to the MCU internally.

[Clearing condition]

- After the HOCOCR.HCSTP bit is set to 1, the processing to stop the oscillation of the HOCO is completed.

PL2OVF Flag (PLL2 Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the PLL2 clock is stable.

[Setting condition]

- After the PLL2CR2.PLL2EN is set to 0 (PLL2 is operating) when the PLL2EN bit is 1 (PLL2 is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (tPLL) has elapsed, and supply of the PLL2 clock is started to the MCU internally.

[Clearing condition]

- After the PLL2CR2.PLL2EN bit is set to 1, the processing to stop the oscillation of the PLL2 is completed.

9.2.13 CLKOUT Output Control Register (CKOCR)

Address(es): SYSTEM.CKOCR 0008 003Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CKOSTP	CKODIV[2:0]			CKOSEL[3:0]				—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CKOSEL[3:0]	CLKOUT Output Source Select	b11 b8 0 0 0 0: LOCO 0 0 0 1: HOCO 0 0 1 0: Main clock oscillator 0 0 1 1: Sub-clock oscillator 0 1 0 0: PLL circuit 1 0 0 0: CTSU monitor clock Settings other than above are prohibited.	R/W
b14 to b12	CKODIV[2:0]	CLKOUT Output Division Ratio Select	b14 b12 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64 1 1 1: ×1/128	R/W
b15	CKOSTP	CLKOUT Output Stop Control	0: CLKOUT pin output enabled*1 1: CLKOUT pin output disabled (fixed to the low level)	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It is also necessary to set the pin function control register and port mode register for the corresponding pin.

The CKOCR register sets the clock signal for output from the CLKOUT pin.

CKOSEL[3:0] Bits (CLKOUT Output Source Select)

The CKOSEL[3:0] bits select the clock source for output from the CLKOUT pin from among the LOCO clock, HOCO clock, main clock, sub-clock, PLL clock, and CTSU monitor clock.
To change the clock source, start by setting the CKOSTP bit to 1.

CKODIV[2:0] Bits (CLKOUT Output Division Ratio Select)

Set these bits to select the clock division ratio.
Set the CKOSTP bit to 1 when changing the division ratio.
For details on the characteristics of the clock output from the CLKOUT pin, see section 47.5.6.12, CLKOUT.
Set the division ratio of the output clock according to the CLKOUT pin output cycle specification.

CKOSTP Bit (CLKOUT Output Stop Control)

Set this bit to enable or disable output from the CLKOUT pin.
When this bit is set to 1, the selected clock is output. When this bit is set to 1, a low level is output.
If the CKOSTP bit is rewritten while the clock is still oscillating, a glitch may be generated in the output.

9.2.14 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): SYSTEM.OSTDCR 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 POEG. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POEG.	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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

To set the OSTDIE bit to 0 after the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) has been set to 1, wait for at least three cycles of PCLKB before doing so.

If the OSTDSR.OSTDF flag requires clearing, do this after setting 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 a given I/O register, wait 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 (LOCO is stopped) to the LOCOCR.LCSTP bit is invalid.

When the OSTDSR.OSTDF flag is 1 (main clock oscillation stop has been detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode. To make a transition to software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

9.2.15 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): SYSTEM.OSTDSR 0008 0041h

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

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: The main clock oscillation stop has not been detected. 1: The main clock oscillation stop has been detected.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0.

OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.

Once the main clock oscillation stop is detected, the OSTDF 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 three ICLK cycles of wait time is necessary between writing 0 to the OSTDF flag and reading the OSTDF flag as 0. If the OSTDF flag is set to 0 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.

When the main clock oscillator (010b) or PLL (100b) is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), the OSTDF flag cannot be modified to 0. The OSTDF flag should be set to 0 after switching the clock source to a source other than the main clock oscillator and the PLL.

[Setting condition]

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

[Clearing condition]

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

9.2.16 Low-Speed On-Chip Oscillator Forced Oscillation Control Register (LOFCR)

Address(es): SYSTEM.LOFCR 0008 0043h

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

Bit	Symbol	Bit Name	Description	R/W
b0	LOFXIN	Low-Speed On-Chip Oscillator Forced Oscillation	0: Stopping of the oscillation in the software standby mode (normal operation) 1: No stopping of the oscillation in the software standby mode (forced oscillation)	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

LOFXIN Bit (Low-Speed On-Chip Oscillator Forced Oscillation)

This bit enables or disables the forced oscillation of the low-speed on-chip oscillator (LOCO).

This bit is used to control the forced oscillation of the low-speed on-chip oscillator (LOCO) in the software standby mode. Only enable the LOCO forced oscillation function when the LOCO is selected as the clock source for the low-power timer (LPT) and counting is to be continued in the software standby mode.

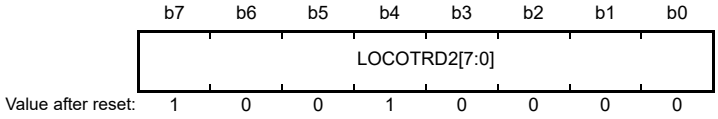
To write a new value to the LOFXIN bit, read it to confirm that the value has actually been updated, and then execute the subsequent instructions.

Change the value of the LOFXIN bit while the LOCOCR.LCSTP bit is 0 (LOCO in operation).

When setting the LOFXIN bit to 1, wait for at least five cycles of the LOCO after having set the LOCOCR.LCSTP bit to 0 (enabling the LOCO operation). When setting the LOCOCR.LCSTP bit to 1 (disabling the LOCO operation) after having set the LOFXIN bit to 0, wait for at least five cycles of the LOCO to do so.

9.2.17 Low-Speed On-Chip Oscillator Trimming Register 2 (LOCOTRR2)

Address(es): SYSTEM.LOCOTRR2 0008 0061h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOCOTRD2[7:0]	Low-Speed On-Chip Oscillator Frequency Adjustment 2	b7 b0 0 0 0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 0 0 0 1: 1 : : 1 1 1 1 1 1 1 0: 254 1 1 1 1 1 1 1 1: 255 (Frequency: High)	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

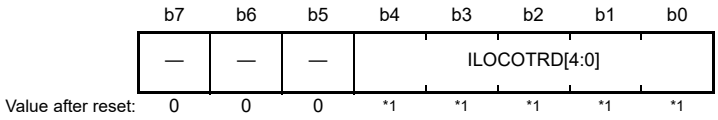
LOCOTRD2[7:0] Bits (Low-Speed On-Chip Oscillator Frequency Adjustment 2)

Set the frequency adjustment value for the low-speed on-chip oscillator.

The setting range is from 0 (00h) to 255 (FFh) by binary numbers. The greater the set value is, the higher the frequency is.

9.2.18 IWDT-Dedicated On-Chip Oscillator Trimming Register (ILOCOTRR)

Address(es): SYSTEM.ILOCOTRR 0008 0064h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ILOCOTRD[4:0]	IWDT-Dedicated On-Chip Oscillator Frequency Adjustment	b4 b0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 1: 1 : : 1 1 1 1 0: 30 1 1 1 1 1: 31 (Frequency: High)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

ILOCOTRD[4:0] Bits (IWDT-Dedicated On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the IWDT-dedicated on-chip oscillator.

The setting range is from 0 (00h) to 31 (1Fh) by binary numbers. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

9.2.19 High-Speed On-Chip Oscillator Trimming Register 0 (HOCOTRR0)

Address(es): SYSTEM.HOCOTRR0 0008 0068h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	HOCOTRD[5:0]					
Value after reset:	0	0	*1	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	HOCOTRD[5:0]	High-Speed On-Chip Oscillator Frequency Adjustment	b5 b0 0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 0 1: 1 : : 1 1 1 1 1 0: 62 1 1 1 1 1 1: 63 (Frequency: High)	R/W *2
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value
Note 2. Writing to the HOCOTRR0 register is prohibited when the HOCO is to be selected for the USB clock source by the setting of the USBCKCR register.

HOCOTRD[5:0] Bits (High-Speed On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the high-speed on-chip oscillator.

The setting range is from 0 (00h) to 63 (3Fh) by binary numbers. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

9.2.20 CANFD Clock Frequency Division Control Register (CANFDCKDIVCR)

Address(es): SYSTEM.CANFDCKDIVCR 0008 006Eh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CANFDCKDIV[2:0]	CANFDCLK Frequency Division Ratio Select	b2 b0 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 1 0 0: ×1/8 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CANFDCKDIV[2:0] Bits (CANFDCLK Frequency Division Ratio Select)

These bits select the frequency of the CANFD clock (CANFDCLK).

Set the MSTPCRD.MSTPD9 bit to 1 (disabling the module clock) and confirm that the CANFDCKCR.CANFDCKSRDY flag is set to 1 (supply of the clock signal being stopped) before rewriting this register.

9.2.21 USB Clock Control Register (USBCKCR)

Address(es): SYSTEM.USBCKCR 0008 0074h

	b7	b6	b5	b4	b3	b2	b1	b0
	USBCKSRDY	USBCKSREQ	—	—	—	USBCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	USBCKSEL[2:0]	USB Clock Source Select	b2 b0 0 0 0: HOCO 0 1 0: LOCO*1 1 0 1: PLL 1 1 0: PLL2 Settings other than above are prohibited.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	USBCKSREQ	Clock Supply Stop Request	0: Requests to resume supply of the USB clock from the clock source selection circuit 1: Requests to stop supply of the USB clock from the clock source selection circuit	R/W
b7	USBCKSRDY	Clock Supply Status Flag	0: The USB clock is being supplied from the clock source selection circuit. 1: Supply of the USB clock from the clock source selection circuit is stopped.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. When the LOCO is selected for UCLK by the USBCKCR.USBCKSEL[2:0] bits, do not set the MSTPCRB.MSTPB19 bit to 0 (enabling the module clock).

The USBCKCR register controls the USB clock (UCLK).
Modify the settings of the USBCKSEL[2:0] bits by following the procedure listed below to ensure stable output of the clock before and after switching its source.

1. Make the clock sources before and after being switched oscillate, and ensure their oscillations have become stable.
2. Write 1 to the MSTPCRB.MSTPB19 bit (disabling the module clock).
3. Write 1 to the USBCKSREQ bit.
4. Poll the USBCKSRDY flag until its value becomes 1.
   Supply of the USB clock (UCLK) is stopped while the USBCKSRDY flag is 1.
5. Rewrite the values of the USBCKSEL[2:0] bits.
6. Write 0 to the USBCKSREQ bit.
7. Poll the USBCKSRDY flag until its value becomes 0.
   The USBCKSRDY flag becoming 0 indicates that the source clock for the USB has been switched.
8. Write 0 to the MSTPCRB.MSTPB19 bit (enabling the module clock).

For a transition to software standby mode, do not execute the WAIT instruction while switching of the clock source is in progress, that is, while USBCKSREQ = 1 and USBCKSRDY = 0 or USBCKSREQ = 0 and USBCKSRDY = 1.

USBCKSEL[2:0] Bits (USB Clock Source Select)

These bits select the source of the USB clock (UCLK).
Set the frequency of the clock source so that the UCLK frequency will be 48 MHz.
Set the MSTPCRB.MSTPB19 bit to 1 (disabling the module clock) and confirm that the USBCKSRDY flag is set to 1 (supply of the clock signal being stopped) before rewriting these bits.
Setting the USBCKSEL[2:0] bits to 000b is possible, and the HOCO oscillation frequency is corrected to have an error within ±0.25%, but only when the USB module is in use as a function controller.

USBCKSREQ Bit (Clock Supply Stop Request)

This bit requests to stop or resume supply of the clock signal to the USB.

USBCKSRDY Flag (Clock Supply Status Flag)

This flag indicates the state of the clock supply to the USB. The clock signal is not supplied to the USB while the USBCKSRDY flag is 1.

9.2.22 CANFD Clock Control Register (CANFDCKCR)

Address(es): SYSTEM.CANFDCKCR 0008 0076h

	b7	b6	b5	b4	b3	b2	b1	b0
	CANFDCKSRDY	CANFDCKSREQ	—	—	—	CANFDCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CANFDCKSEL[2:0]	CANFD Clock Source Select	b2 b0 0 1 0: LOCO 1 0 1: PLL 1 1 0: PLL2 Settings other than above are prohibited.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CANFDCKSREQ	Clock Supply Stop Request	0: Requests to resume supply of the CANFD clock from the clock source selection circuit 1: Requests to stop supply of the CANFD clock from the clock source selection circuit	R/W
b7	CANFDCKSRDY	Clock Supply Status Flag	0: The CANFD clock is being supplied from the clock source selection circuit. 1: Supply of the CANFD clock from the clock source selection circuit is stopped.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The CANFDCKCR register controls the CANFD clock (CANFDCLK).
Modify the settings of the CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[2:0] bits by following the procedure listed below to ensure stable output of the clock before and after switching its source.

1. Make the clock sources before and after being switched oscillate, and ensure their oscillations have become stable.
2. Write 1 to the MSTPCRD.MSTPD9 bit (disabling the module clock).
3. Write 1 to the CANFDCKSREQ bit.
4. Poll the CANFDCKSRDY flag until its value becomes 1.
   Supply of CANFD clock (CANFDCLK) stops while the CANFDCKSRDY flag is 1.
5. Rewrite the values of the CANFDCKDIVCR. CANFDCKDIV[2:0] and CANFDCKSEL[2:0] bits.
6. Write 0 to the CANFDCKSREQ bit.
7. Poll the CANFDCKSRDY flag until its value becomes 0.
   The CANFDCKSRDY flag becoming 0 indicates that the source clock for the CANFD has been switched.
8. Write 0 to the MSTPCRD.MSTPD9 bit (enabling the module clock).

For a transition to software standby mode, do not execute the WAIT instruction while switching of the clock source is in progress, that is, while CANFDCKSREQ = 1 and CANFDCKSRDY = 0 or CANFDCKSREQ = 0 and CANFDCKSRDY = 1.

CANFDCKSEL[2:0] Bits (CANFD Clock Source Select)

These bits select the source of the CANFD clock (CANFDCLK).
Set the MSTPCRD.MSTPD9 bit to 1 (disabling the module clock) and confirm that the CANFDCKSRDY flag is set to 1 (supply of the clock signal being stopped) before rewriting these bits.

CANFDCKSREQ Bit (Clock Supply Stop Request)

This bit requests to stop or resume supply of the clock signal to the CANFD.

CANFDCKSRDY Flag (Clock Supply Status Flag)

This flag indicates the state of the clock supply to the CANFD. The clock signal is not supplied to the CANFD while the CANFDCKSRDY flag is 1.

9.2.23 Sub-Clock Oscillator Mode Control Register (SOMCR)

Address(es): SYSTEM.SOMCR 0008 0083h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SODRV[1:0]	SOSC Drive Capacity Control	b1 b0 0 0: Drive capacity for standard CL 0 1: High-drive output for the low CL 1 0: Middle-drive output for the low CL 1 1: Low-drive output for the low CL	R/W
b5-b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SOSEL	SOSC Oscillation 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit will not be initialized by the source other than power-on resetting.

SODRV[1:0] Bits (SOSC Drive Capacity Control)

These bits control the drive capacity of the sub-clock oscillator. Set these bits while the SOSCCR.SOSTP bit is 1.

SOSEL Bit (SOSC Oscillation Switching)

This bit switches the source for the sub-clock oscillator. Set the SOSEL bit while the SOSCCR.SOSTP bit is 1.

9.2.24 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): SYSTEM.MOSCWTCR 0008 00A2h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Wait Time	b4 b0 0 0 0 0 0: Wait time = 0 cycles (0 µs) 0 0 0 0 1: Wait time = 1024 cycles (256 µs) 0 0 0 1 0: Wait time = 2048 cycles (512 µs) 0 0 0 1 1: Wait time = 4096 cycles (1.024 ms) 0 0 1 0 0: Wait time = 8192 cycles (2.048 ms) 0 0 1 0 1: Wait time = 16384 cycles (4.096 ms) 0 0 1 1 0: Wait time = 32768 cycles (8.192 ms) 0 0 1 1 1: Wait time = 65536 cycles (16.384 ms) 0 1 0 0 0: Wait time = 131072 cycles (32.768 ms) Settings other than above are prohibited. Wait time when LOCO = 4.0 MHz (0.25 µs, TYP.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

MSTS[4:0] Bits (Main Clock Oscillator Wait Time)

Set these bits to select the oscillation stabilization wait time of the main clock oscillator.

Set the main clock oscillation stabilization time to longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is externally input, set these bits to 00000b because the oscillation stabilization time is not required.

The wait time set by the MSTS[4:0] bits is counted using the LOCO clock. The LOCO automatically oscillates when necessary, regardless of the value of the LOCOCR.LCSTP bit.

After the set wait time has elapsed, supply of the main clock is started to the MCU internally and the OSCOVFSR.MOOVF flag becomes 1. If the set wait time is short, supply of the main clock is started before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCOVFSR.MOOVF flag is 0. Do not rewrite this register under any other conditions.

9.2.25 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): SYSTEM.MOFCR 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV21	—	—	—	—	—
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	MODRV21	Main Clock Oscillator Drive Capability Switch	0: 1 MHz to 10 MHz 1: 10 MHz to 20 MHz	R/W
b6	MOSEL	Main Clock Oscillator Switch	0: Resonator 1: External oscillator input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The EXTAL/XTAL pin is also used as a port. In the initial setting state, the pin is set as a port.
Only write new values to the MOFCR register when the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 1 (stopping the main clock oscillator).

MODRV21 Bit (Main Clock Oscillator Drive Capability Switch)

These bits select the drive capability of the main clock oscillator.

MOSEL Bit (Main Clock Oscillator Switch)

This bit selects the oscillation source of the main clock oscillator.

9.3 Main Clock Oscillator

There are two ways of supplying the clock signal from the main clock oscillator: connecting an oscillator or the input of an external clock signal.

9.3.1 Connecting a Crystal

Figure 9.2 shows an example of connecting a crystal.
A damping resistor (Rd) should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

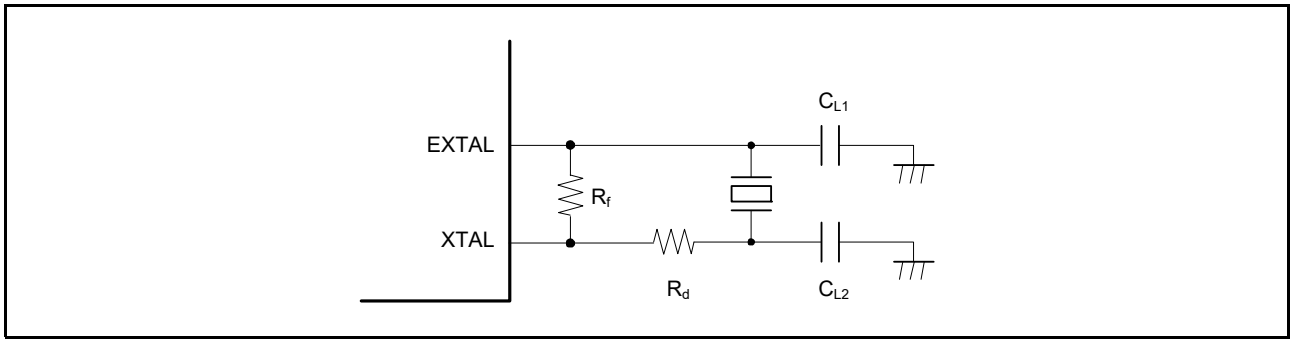


Figure 9.2 Example of Crystal Connection

9.3.2 External Clock Input

Figure 9.3 shows connection of an external clock. Set the MOFCR.MOSEL bit to 1 to operate the oscillator by inputting an external clock signal. The XTAL pin becomes high impedance and can be used as a general input port.

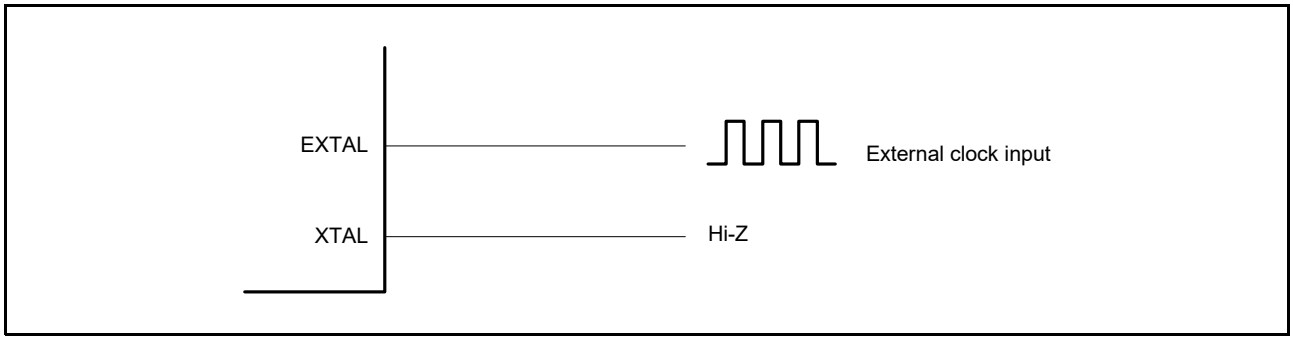


Figure 9.3 Connection Example of External Clock

9.3.3 Handling of Pins When the Main Clock is Not Used

For details on pin handling when the main clock is not used, refer to section 20.5, Handling of Unused Pins.

9.3.4 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 (main clock oscillator is operating).

9.4 Sub-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.

9.4.1 Connecting 32.768-kHz Crystal

Figure 9.4 shows an example of connecting a 32.768-kHz crystal.
A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between XCIN and XCOUT by following the instruction. When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the sub-clock oscillator described in Table 9.1.

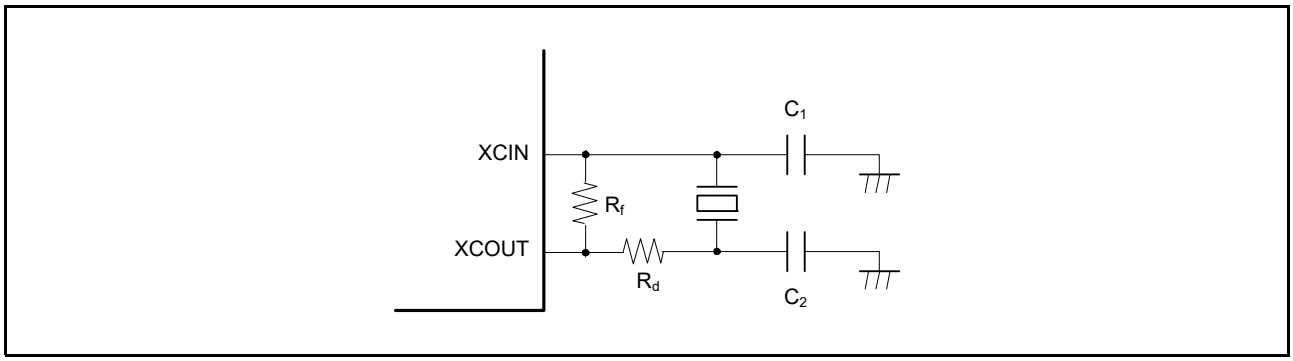


Figure 9.4 Connection Example of 32.768-kHz Crystal

9.4.2 External Clock Input

Figure 9.5 shows connection of an external clock. Set the SOMCR.SOSEL bit to 1 to operate the oscillator by inputting an external clock signal. The XTAL pin becomes high impedance and can be used as a general input port.

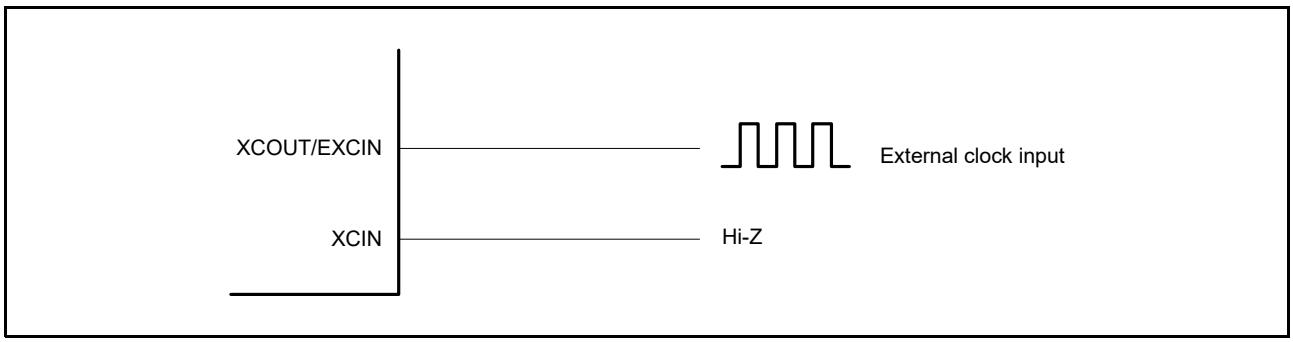


Figure 9.5 Connection Example of External Clock

9.4.3 Handling of Pins When Sub-Clock is Not Used

For details on pin handling when the sub-clock is not in use, refer to section 20.5, Handling of Unused Pins.

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.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the GPTW output can be forcibly driven to the high-impedance on the detection. For details, refer to section 23, GPTW Port Output Enable (POEGc).

In the MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator. For details on the detection period, refer to section 47, Electrical Characteristics.

When an oscillation stop is detected, the main clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage. Therefore, if an oscillation stop is detected with the main clock selected as the system clock source, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].

If an oscillation stop is detected while the PLL clock is selected by the clock source select bits (SCKCR3.CKSEL[2:0]) in system clock control register 3, the SCKCR3.CKSEL[2:0] bits value does not change and the PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.

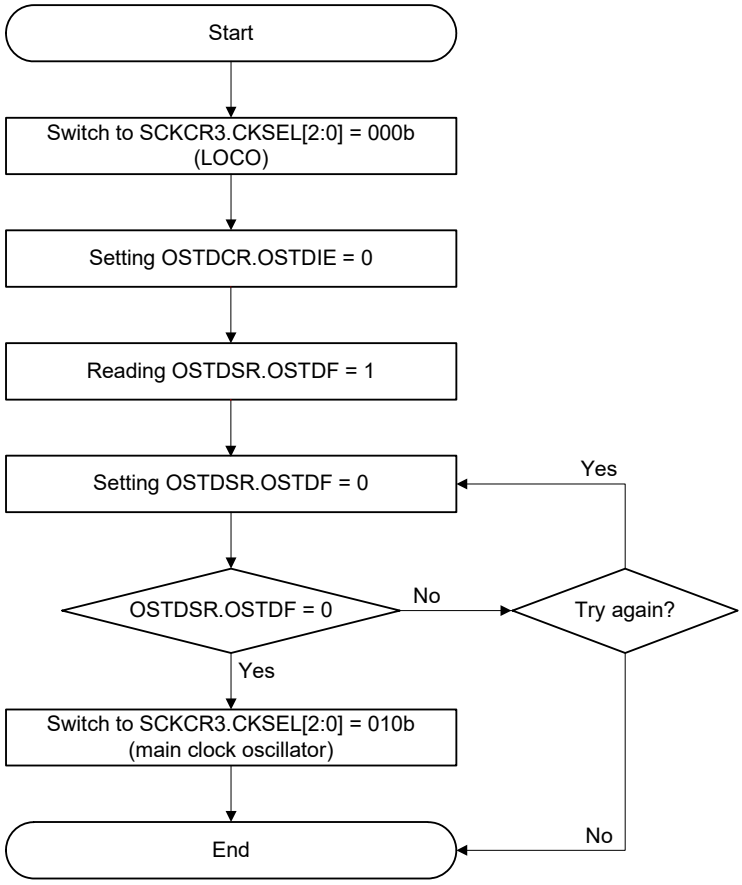
Switching between the main clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and set the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation stabilization time has elapsed.

After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time has elapsed.

The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode.

The clock signals that are switched to the LOCO clock on detection of oscillation by the main clock stopping when it was selected as the system clock source are the system clock, CAC main clock, and CANFD main clock (CANFDMCLK). The system clock (ICLK) frequency during the LOCO clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock (ICLK) select bits (SCKCR.ICK[3:0]).

When the system clock with the PLL clock selected as the system clock source, CANFD clock (CANFDCLK) with the PLL and PLL2 clock signals are selected as the source, and UCLK clock with the PLL and PLL2 clock signals are selected as the source, these clock signals run with the PLL free-running frequency by the oscillation stop detection.



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 9.6 Flow of Recovery from Detection of Oscillator Stop

9.5.2 Oscillation Stop Detection Interrupts

An oscillation-stop detection interrupt (OSTDI) will be generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit (OSTDCR.OSTDIE) is 1 (oscillation stop detection interrupt enabled). At this time, the main clock oscillator stop is notified to GPTW port output enable (POEG). On accepting the notification of the oscillation stop, the POEG sets the oscillation stop detection flag in POEG group n setting register (POEGGn.OSTPF (n = A to D)) to 1.

Wait at least three cycles of PCLKB before clearing the oscillation stop detection interrupt enable bit (OSTDIE) to 0 after the oscillation stop detection flag (OSTDF) has been set to 1.

When the OSTDSR.OSTDF flag requires clearing, do so setting the oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE) to 0. 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 a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after a reset release, enable the non-maskable interrupts by the software before using oscillation stop detection interrupts. For details, refer to section 14, Interrupt Controller (ICUb).

9.6 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.7 Internal Clock

Clock sources of internal clock signals are the main clock, sub-clock, HOCO clock, LOCO clock, PLL clock, PLL2 clock, and dedicated low-speed clock for the IWDT. The internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DMAC, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clocks of peripheral modules: Peripheral module clocks (PCLKA, PCLKB, and PCLKD)
(3) Operating clock of the FlashIF: FlashIF clock (FCLK)
(4) Operating clock of the USB: USB clock (UCLK)
(5) Operating clock for the REMC: REMC clock (REMCLK)
(6) Operating clock for the CAC: CAC clock (CACCLK)
(7) Operating clock for the RTC: RTC clock (RTCSCLK)
(8) Operating clock for the IWDT: IWDT-dedicated low-speed clock (IWDTCLK)
(9) Operating clock for the low-power timer: LPT clock (LPTCLK)
(10) Operating clock for the CANFD: CANFD clock (CANFDMCLK and CANFDCLK)

The frequencies of the internal clock signals are set by the combination of the following bits.

- The FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits in SCKCR select the divisors.
- The SCKCR3.CKSEL[2:0] bits select the clock source.
- The STC[5:0] and PLIDIV[1:0] bits in PLLCR select the frequency of the PLL circuit.
- The STC[5:0] and PLIDIV[1:0] bits in PLL2CR select the frequency of the PLL2 circuit.
- The OFS1.HOCOFRQ[1:0] bits select the frequency of the HOCO circuit.

If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value. If the clock with frequency higher than 32 MHz is to be selected as the system clock (ICLK), the number of wait cycles to insert must be set for the ROM access. section 46, Flash Memory (FLASH).

9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, ROM, and RAM.
The ICLK frequency is specified by the following bits.

- SCKCR.ICK[3:0]
- SCKCR3.CKSEL[2:0]
- PLLCR.STC[5:0] and PLLCR.PLIDIV[1:0]
- OFS1.HOCOFRQ[1:0]

9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, and PCLKD) are the operating clocks for use by peripheral modules.
The PCLKA, PCLKB, and PCLKD frequencies are specified by the following bits.

- SCKCR.PCKA[3:0], SCKCR.PCKB[3:0], and SCKCR.PCKD[3:0]
- SCKCR3.CKSEL[2:0]
- PLLCR.STC[5:0] and PLLCR.PLIDIV[1:0]
- OFS1.HOCOFRQ[1:0]

PCLKD is the operating clock for analog-to-digital conversion by the S12AD, PCLKA is for the CANFD (message buffer RAM) and GPTW, and PCLKB for the other peripheral modules.

9.7.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.
The FCLK frequency is specified by the following bits.

- SCKCR.FCK[3:0]
- SCKCR3.CKSEL[2:0]
- PLLCR.STC[5:0] and PLLCR.PLIDIV[1:0]
- OFS1.HOCOFRQ[1:0]

9.7.4 USB Clock

The USB clock (UCLK) is used as the operating clock for the USB.
Use the following bits to set the UCLK frequency.

- PLLCR.STC[5:0] and PLLCR.PLIDIV[1:0]
- PLL2CR.STC[5:0] and PLL2CR.PLIDIV[1:0]
- OFS1.HOCOFRQ[1:0]

When the USB module is used, setting must be made so that UCLK is 48 MHz.

9.7.5 REMC Clock

The REMC clock (REMCLK) is an operating clock for the REMC module.
REMCLK is generated by either the sub-clock oscillator or the IWDT-dedicated on-chip oscillator.

9.7.6 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC module.
CACCLK is generated by any of the following oscillators.

- Main clock oscillator
- Sub-clock oscillator
- High-speed on-chip oscillator
- Low-speed on-chip oscillator
- IWDT-dedicated on-chip oscillator

9.7.7 RTC Clock

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

9.7.8 IWDT-Dedicated Clock

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

9.7.9 Low-Power Timer Clock

The low-power timer clock (LPTCLK) is an operating clock for the low-power timer. LPTCLK can be generated by any of the following.

- Sub-clock oscillator
- IWDT-dedicated on-chip oscillator
- Frequency-dividing the signal from the low-speed on-chip oscillator by 4

9.7.10 CANFD Clock

The CANFD main clock (CANFDMCLK) and CANFD clock (CANFDCLK) are the operating clock signals for the CANFD module. CANFDMCLK is generated by the main-clock oscillator. CANFDCLK is generated by the LOCO, PLL, or PLL2. The frequency of CANFDCLK can be set by the following bits.

- PLLCR.STC[5:0]
- PLLCR.PLIDIV[1:0]
- PLL2CR.STC[5:0]
- PLL2CR.PLIDIV[1:0]
- CANFDCKCR.CANFDCKSEL[2:0]
- CANFDCKDIVCR.CANFDCKDIV[2:0]

When the CANFD module is to be used, the frequencies of CANFDMCLK and CANFDCLK must be no higher than PCLKB.

9.8 Usage Notes

9.8.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) supplied to each module can be selected by the SCKCR register. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKA, PCLKB, and PCLKD. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The relationship of frequencies of the system clock (ICLK), peripheral module clocks B and D (PCLKB and PCLKD), and FlashIF clock (FCLK) must be set as follows.
ICLK:FCLK = N:1 or 1:N (N is an integer)
ICLK:PCLKA, PCLKB, PCLKD = N:1 or 1:N (N is an integer)
PCLKA:PCLKB = 2:1 when the CANFD is in use
PCLKA ≥ PCLKB when the CANFD is not in use

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

9.8.2 Note on Rewriting the SCKCR3 Register

When the SCKCR3.CKSEL[2:0] bits have been rewritten, the clock output is temporarily stopped to prevent the switch of the clock source from generating a clock pulse of short duration (glitch). The interrupt controller may not detect the following signals that were input during this period.

(1) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 4 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/1 of the clock source (the SCKCR.PCKB[3:0] bits are 0000b).

(2) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 2.5 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/2 of the clock source (the SCKCR.PCKB[3:0] bits are 0001b).

(3) An RTC periodic interrupt when the SCKCR3.CKSEL[2:0] bits are changed to “011b” (sub-clock).

When the external pin interrupt or NMI pin interrupt is in use, input the signal with enough pulse width to exceed the time condition described in (1) and (2). When the RTC periodic interrupt is in use, switch the clock source after a periodic interrupt is generated and before the next periodic interrupt is generated.

9.8.3 Notes on Resonator

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

9.8.4 Notes on Board Design

When using a crystal, 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 9.7 to prevent electromagnetic induction from interfering with correct oscillation.

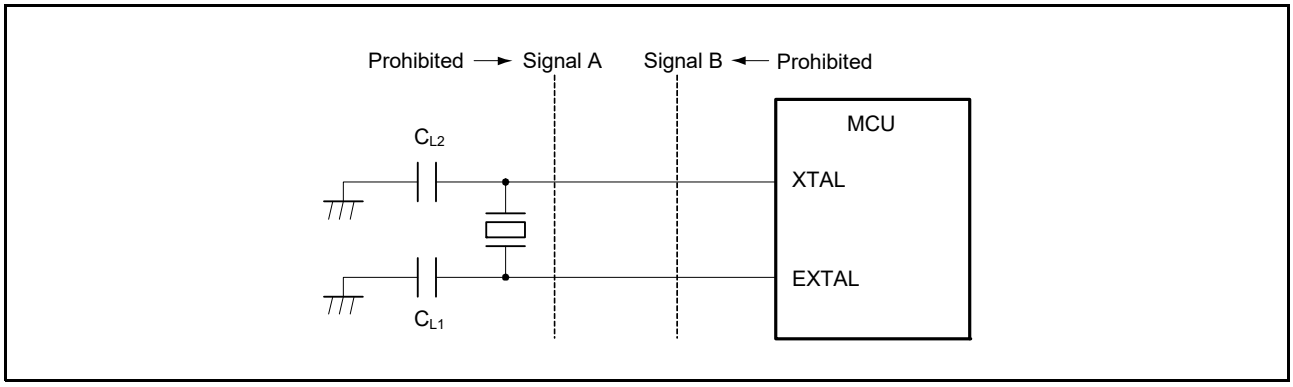


Figure 9.7 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

9.8.5 Notes on Resonator Connection Pins

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When using these pins as general ports, be sure to stop the main clock (MOSCCR.MOSTP = 1). However, do not use the EXTAL and XTAL pins as general ports P36 and P37 in a system that uses the main clock.
When the main clock is used, do not set P36 and P37 to output.
Note that the XTAL pin can be used as the P37 general input port pin when the settings of the MOFCR.MOSEL bit and MOSCCR.MOSTP bit are respectively 1 and 0, selecting operation with the external clock being input to the EXTAL pin.
The XCIN and XCOUT pins can be used as general input port pins PH7 and PH6 when the sub-clock is not used. If they are in use as the general port pins, make sure that the sub-clock is stopped (SOSCCR.SOSTP = 1).
In a system that uses the sub-clock, do not use the XCIN and XCOUT pins as general input port pins.
Note that the XCIN pin can be used as the general input port pin PH7 when the settings of the SOMCR.SOSEL bit and SOSCCR.SOSTP bit are respectively 1 and 0 for operation with the external clock being input to the EXCIN pin.

9.8.6 Notes on using a low CL crystal unit

When the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Main Clock Circuit and Sub-Clock Circuit” (R01AN7202EJ) to reduce the influence from noise.

9.8.7 Notes on Sub-Clock

With regard to making the sub-clock oscillator run or stop, setting the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) will make the oscillator run.

To use the sub-clock, perform initial settings according to the flowchart example shown in Figure 9.8. To use the sub-clock as the count source of the realtime clock, after that, perform the clock setting procedure shown in section 26.3.2, Clock and Count Mode Setting Procedure.

To use the sub-clock, do not use the PG7 pin as general I/O port pins.

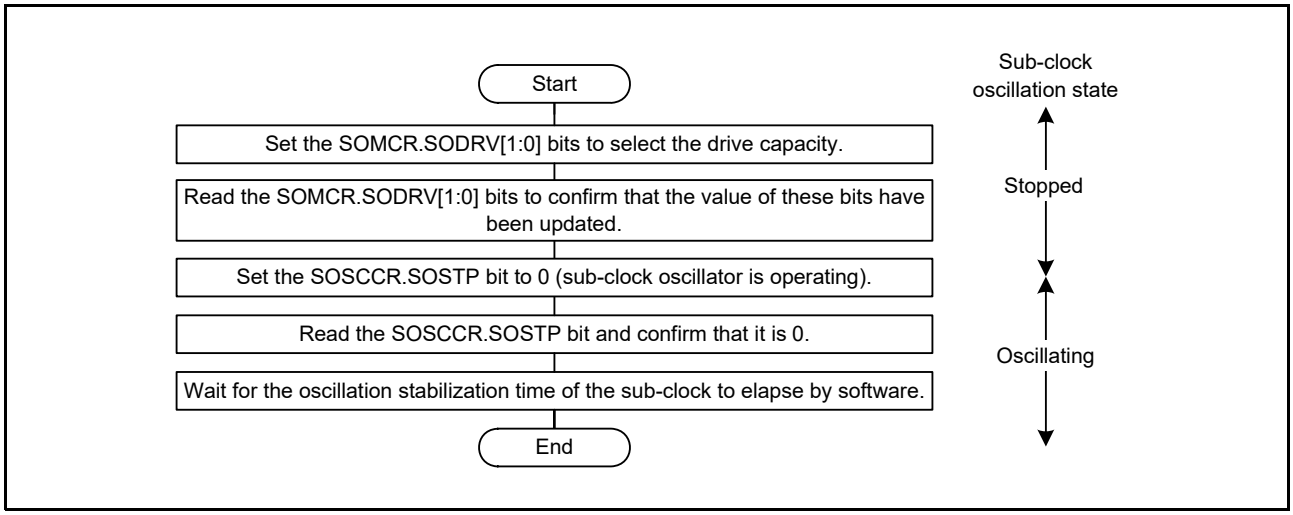


Figure 9.8 Example of Initialization Flowchart When Sub-Clock is Used as Count Source of Realtime Clock

10. Clock Frequency Accuracy Measurement Circuit (CAC)

This MCU incorporates a/ clock frequency accuracy measurement circuit (CAC).
The 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.

10.1 Overview

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.

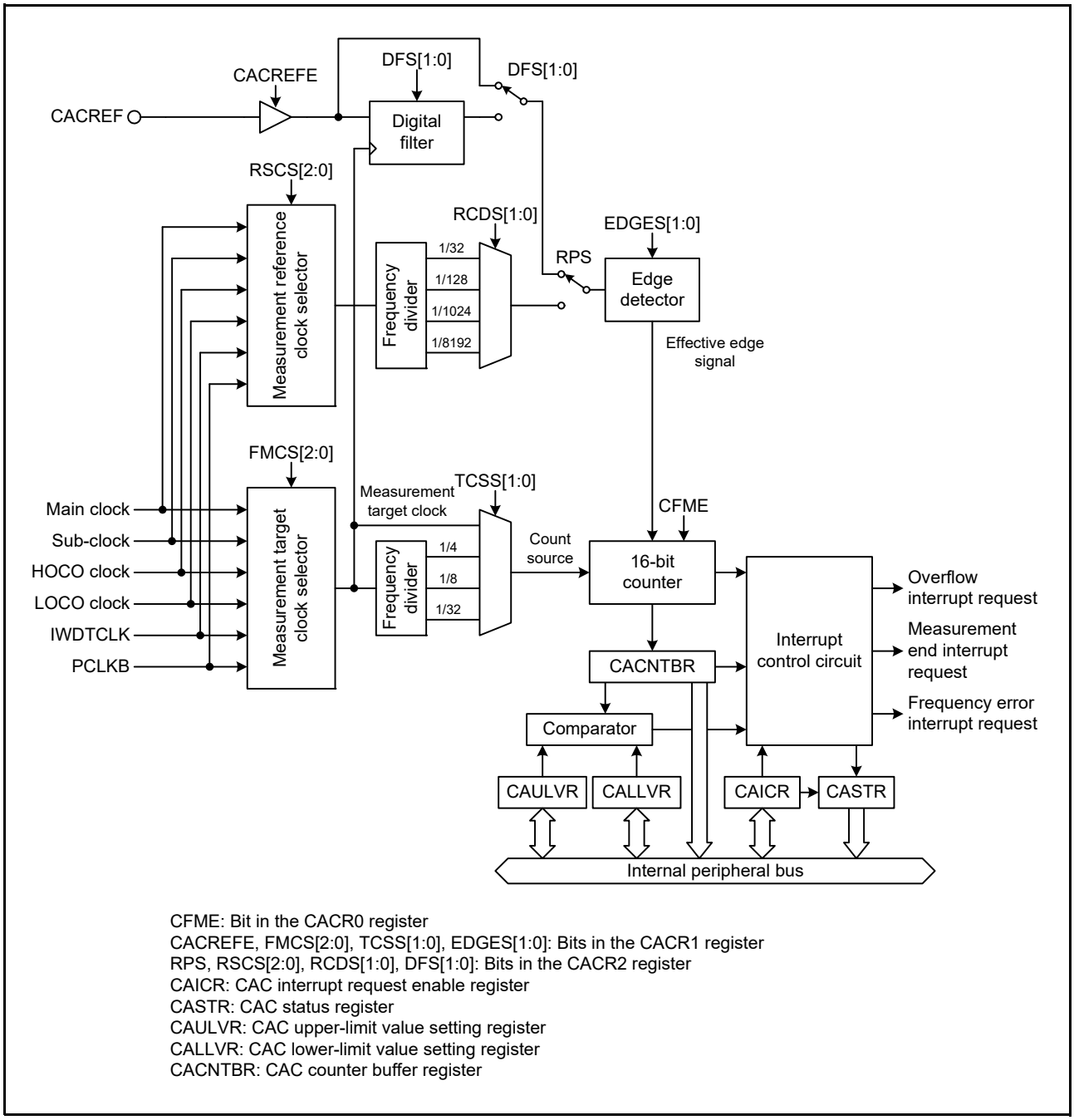


Figure 10.1 CAC Block Diagram

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

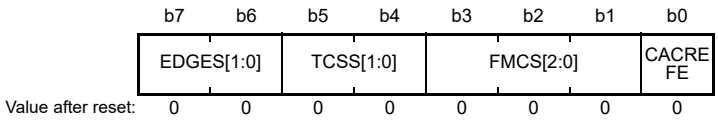
CFME Bit (Clock Frequency Measurement Enable)

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



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 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 Source Select)

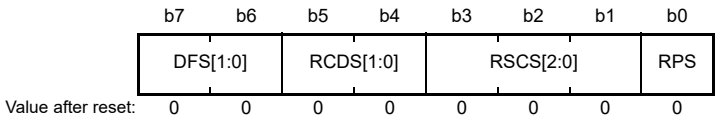
These bits select the count source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC.CACR2 0008 B002h



Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the measurement target clock. 1 0: The sampling clock for the digital filter is the measurement target clock divided by 4. 1 1: The sampling clock for the digital filter is the measurement target clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): CAC.CAICR 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC.CASTR 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 0008 B006h

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

The CAULVR register is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC.CALLVR 0008 B008h

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

The CALLVR register is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC.CACNTBR 0008 B00Ah

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

The CACNTBR register is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

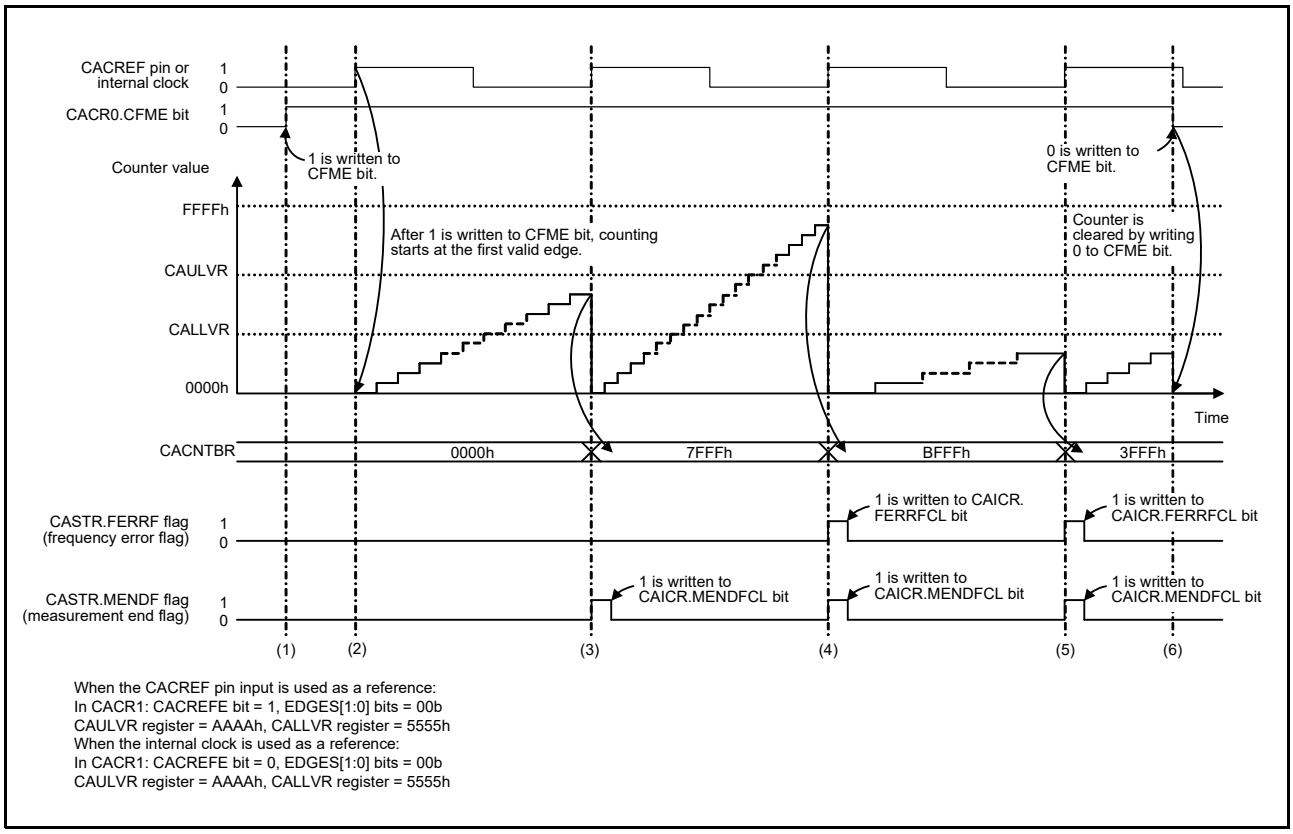


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (the CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (the CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. If the formula CALLVR ≤ CACNTBR ≤ CAULVR is satisfied, only the CASTR.MENDF flag is set to 1 because the clock frequency is correct. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR > CAULVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is

generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR < CALLVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(6) While the CACR0.CFME bit is 1, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers every time a valid edge is input. Writing 0 to the CACR0.CFME bit clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in the CACNTBR register may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source) / (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, by setting clock dividers, stopping modules, changing to low power consumption mode in normal operation, and changing to operating power control mode.
Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to change to low power consumption modes, states of the CPU and peripheral modules, and the method for exiting each mode.
After a reset, this MCU returns to normal mode, but modules except the DMAC, DTC and RAM are stopped.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Clock divider functions	Frequency division ratios can be independently set for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK).*1
Module stop function	Each peripheral module can be stopped independently by the module stop control register.
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 • Deep sleep mode • Software standby mode • Snooze mode
Operating power control modes	• Power consumption can be reduced in normal operation, sleep mode, deep sleep mode, and snooze mode by selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage range. • Four operating power control modes are available High-speed operating mode Middle-speed operating mode Middle-speed operating mode 2 Low-speed operating mode

Note 1. For details, refer to section 9, Clock Generation Circuit.

Table 11.2 Operating Conditions of Each Power Consumption Mode (1/2)

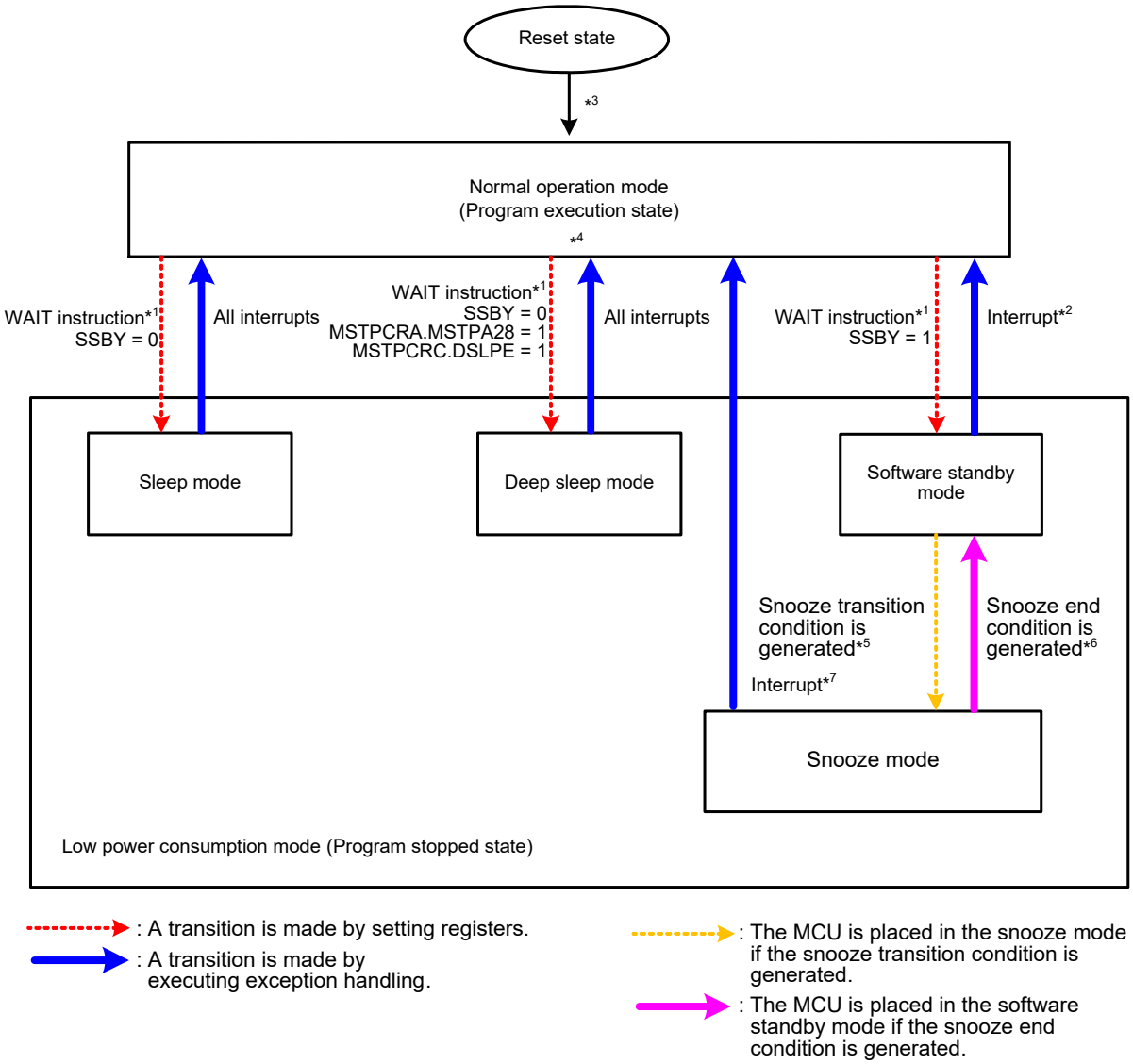
Operating Conditions	Sleep Mode	Deep Sleep Mode	Software Standby Mode	Snooze Mode*1
Entry trigger	Control register + instruction	Control register + instruction	Control register + instruction	Generation of the snooze transition in software standby mode
Triggers for exit other than a reset	Interrupt	Interrupt	Interrupt*2	Generation of interrupt*3 or snooze end conditions
State after exit from the given mode*4	Program in progress (interrupt handling)	Program in progress (interrupt handling)	Program in progress (interrupt handling)	Program in progress (interrupt handling) or software standby mode
Main clock oscillator	Operating possible	Operating possible	Stopped	Operating possible
Sub-clock oscillator	Operating possible	Operating possible	Operating possible	Operating possible
High-speed on-chip oscillator (HOCO)	Operating possible	Operating possible	Stopped	Operating possible
Low-speed on-chip oscillator (LOCO)	Operating possible	Operating possible	Operating possible*5	Operating possible
IWDT-dedicated on-chip oscillator (ILOCO)	Operating possible*6	Operating possible*6	Operating possible*6	Operating possible*6
PLL	Operating possible	Operating possible	Stopped	Operating possible
PLL2	Operating possible	Operating possible	Stopped	Operating possible
CLKOUT	Operating possible	Operating possible	Operating possible*7	Operating possible
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM0 (0000 0000h to 0001 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Operating possible (Retained)*8
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
DMAC	Operating possible*9	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)

Table 11.2 Operating Conditions of Each Power Consumption Mode (2/2)

Operating Conditions	Sleep Mode	Deep Sleep Mode	Software Standby Mode	Snooze Mode*1
DTC	Operating possible*10	Stopped (Retained)	Stopped (Retained)	Operating possible*8, *10
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*6	Operating possible*6	Operating possible*6	Operating possible*6
Remote control signal receiver (REMC)	Operating possible	Operating possible	Operating possible*11	Operating possible
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible	Operating possible
RTCOUT	Operating possible	Operating possible	Operating possible	Operating possible
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating	Operating
Comparator B	Operating possible	Operating possible	Operating possible*12	Operating possible*12
Low power timer (LPT)	Operating possible	Operating possible	Operating possible	Operating possible
Peripheral modules	Operating possible	Operating possible	Stopped (Retained)*13	Operating possible
I/O ports	Operating	Operating	Retained*14	Operating

“Operating possible” means that operating or stopped can be controlled by the register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.

Note 1. The operating clock is supplied to the on-chip peripheral modules for which supply of the clock signal is enabled (set to 0) by the MSTPCRm.MSTPmi bits (m = A to D, i = 31 to 0) once they have been placed in the snooze mode, and their operation is restarted. Accordingly, if operation of any on-chip modules is not required in the snooze mode, set the corresponding MSTPmi bit to 1 to place the modules in the module-stop state before placing them in the software standby mode.
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (interrupts on the RTC alarm, RTC periodic, IWDT, voltage monitoring, USB, REMC, ELC (LPT dedicated interrupt)).
Note 3. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7), any of peripheral interrupts (interrupts on the RTC alarm, RTC periodic, IWDT, voltage monitoring, USB, REMC), an interrupt for release from the snooze mode, or a non-maskable RAM error interrupt.
Note 4. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
Note 5. In this state, operation or stopping is selectable on the low-speed on-chip oscillator forced oscillation bit (LOFCR.LOFXIN) in the low-speed on-chip oscillator forced oscillation control register.
Note 6. Operating or stopping is selected by setting the IWDT sleep mode count stop control bit (IWDTSLCSTP) in option function select register 0 (OFS0) in IWDT auto-start mode. In any mode other than IWDT auto-start mode, operating or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDT count stop control register (IWDTCSTPR).
Note 7. The clock signal is not output when the clock output select bits in the CLKOUT output control register (CKOCR.CKOSEL[3:0]) are set to a value other than 0011b (sub clock oscillator). When the LOCO forced oscillation is enabled in software standby mode, do not set the CKOCR.CKOSEL[3:0] bits to 0000b (LOCO clock).
Note 8. In this state, operation or stopping is selectable by setting the DTC enable in the snooze mode bit in the snooze control register (SNZCR.SNZDTCE).
Note 9. Writing to the system control registers in the sleep mode is prohibited. For the system control registers, see Table 5.1, List of I/O Registers (Address Order), in which they are indicated as “SYSTEM” in the Module System column.
Note 10. Writing to the system control registers in the sleep mode or snooze mode is prohibited. For the system control registers, see Table 5.1, List of I/O Registers (Address Order), in which they are indicated as “SYSTEM” in the Module System column.
Note 11. Operation is possible when the clock source for the operating clock is the sub clock or IWDT-dedicated on-chip oscillator clock.
Note 12. Using the digital filter function is prohibited. Operation for outputting the comparison result to the CMPOBn pin is possible.
Note 13. The peripheral logic states are retained.
Note 14. When the remote control signal receiver (REMC) is operated, related pins continue operating.



Note 1. If an interrupt is accepted while entering low power consumption mode, the transition to the program stop state is exited and the interrupt exception handling is executed.
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring interrupts, and ELC (LPT dedicated interrupt), REMC, and USB).
Note 3. The LOCO is the clock source following a transition from the reset state to normal mode.
Note 4. Makes a transition from sleep mode, deep sleep mode, or software standby mode and snooze mode to normal operating mode by an interrupt. In the case of exiting sleep mode, the clock source after exiting is selectable. For details, refer to the description of the RSTCKCR register.
For deep sleep mode, software standby mode and snooze mode, the clock source after exiting is the same as that of the clock before entering software standby mode.
Note 5. Snooze transition condition can be set in the SNZCR register. For details, refer to the SNZCR register descriptions.
Note 6. Snooze end condition can be set in the SNZCR register. For details, refer to the SNZCR register descriptions.
Note 7. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7), any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring, REMC, and USB), an interrupt for release from the snooze mode, or a non-maskable RAM error interrupt.

Transition to the reset state is made at any state instantly after a RES# pin reset, power on reset, voltage monitoring reset, WDT reset, IWDT reset, or software reset is generated.

Figure 11.1 Mode Transitions

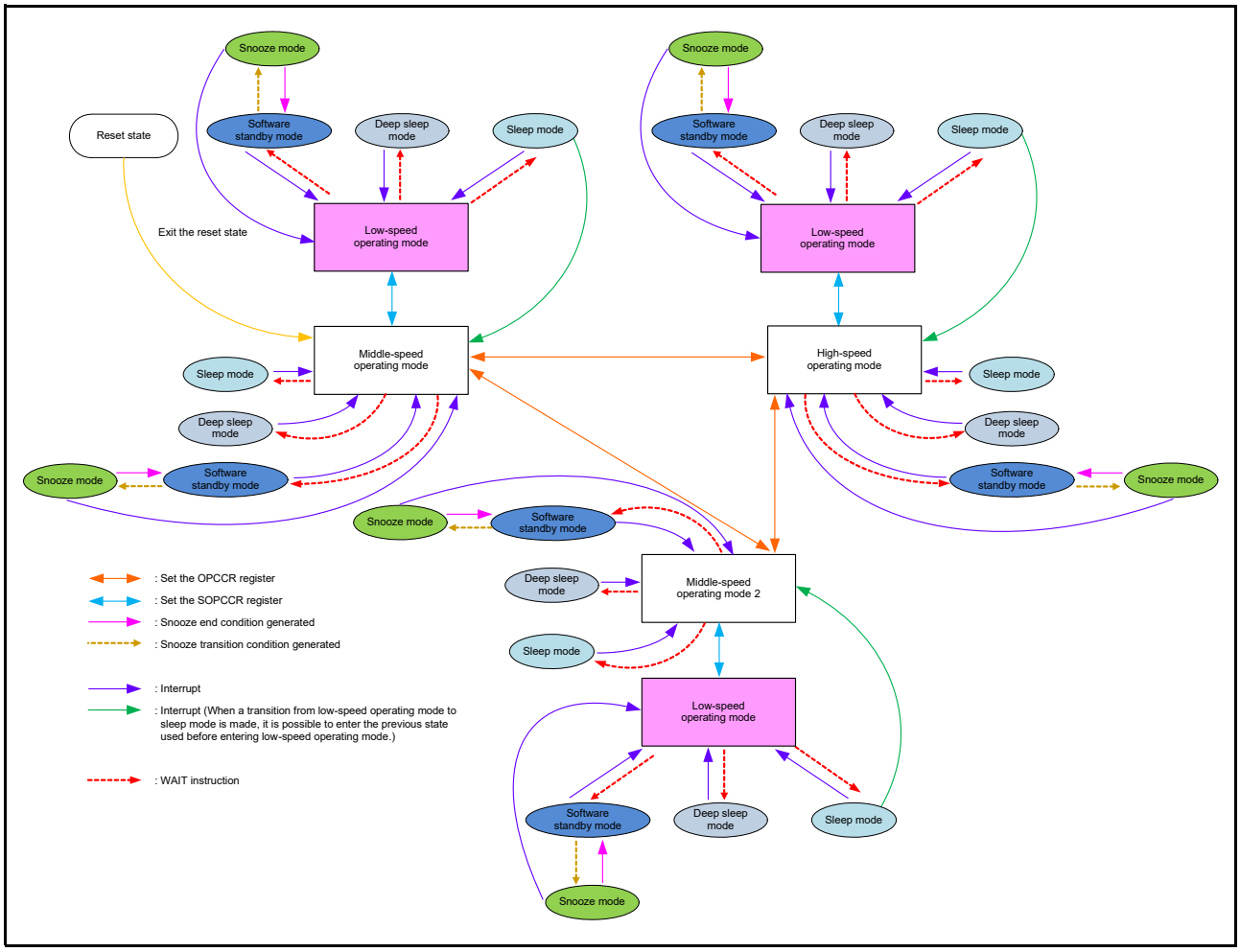


Figure 11.2 Operating Modes

- The sub-clock oscillator does not stop when entering software standby mode.
- It is possible to return from sleep mode to the previous operating state used before entering sleep mode.
  When the MCU has been placed in the sleep mode by releasing from the low-speed operating mode, it can be placed in the high-speed operating mode, middle-speed operating mode, or middle-speed operating mode 2 that was before it was placed in the low-speed operating mode.
- After releasing from the reset state, the MCU starts operating in the middle-speed operating mode.
- In sleep mode, in snooze mode or when the mode transition in progress, writing to the registers related to the system controlling (registers for which SYSTEM is shown in the module symbol column in the I/O register list) is prohibited.

Table 11.3 Oscillator Usability in Each Mode

	PLL	PLL2	HOCO	LOCO	IWDT dedicated On-chip Oscillator	Main Clock Oscillator	Sub-Clock Oscillator
High-speed operating mode	Usable	Usable	Usable	Usable	Usable	Usable	Usable
Middle-speed operating mode	Usable*1	Usable*1	Usable	Usable	Usable	Usable	Usable
Middle-speed operating mode 2	Usable*1	Usable*1	Usable	Usable	Usable	Usable	Usable
Low-speed operating mode	Not usable	Not usable	Not usable	Not usable	Usable	Not usable	Usable

Note 1. The PLL and PLL2 are only usable if VCC is no less than 1.8 V.

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): SYSTEM.SBYCR 0008 000Ch

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

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSBY	Software Standby	0: Set entry to sleep mode or deep sleep mode after the WAIT instruction is executed 1: Set entry to software standby mode after the WAIT instruction is executed	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

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 mode after an interrupt has triggered and exits from software standby mode, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to the SSBY bit.

When the oscillation stop detection function enable bit (OSTDCR.OSTDE) in the oscillation stop detection control register is 1, the set value of the SSBY bit is invalid. Even if the SSBY bit is 1, the MCU will enter sleep mode or deep sleep mode after execution of the WAIT instruction.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): SYSTEM.MSTPCRA 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MSTPA28	—	—	—	—	—	—	—	—	MSTPA19	—	MSTPA17	—
Value after reset:	1	1	1	0	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
	MSTPA15	MSTPA14	—	—	—	—	—	—	MSTPA7	—	MSTPA5	MSTPA4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPA4	8-bit Timer 3 and 2 (Unit 1) Module Stop	Target module: TMR3, TMR2 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	MSTPA5	8-bit Timer 1 and 0 (Unit 0) Module Stop	Target module: TMR1, TMR0 0: This module clock is enabled 1: This module clock is disabled	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	MSTPA7	General PWM Timer Module Stop	Target module: GPTW (GPTW0 to GPTW7), POEG 0: This module clock is enabled 1: This module clock is disabled	R/W
b13 to b8	—	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: This module clock is enabled 1: This module clock is disabled	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: This module clock is enabled 1: This module clock is disabled	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPA17	12-Bit A/D Converter Module Stop	Target module: S12AD 0: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	D/A Converter Module Stop	Target module: DA 0: This module clock is enabled 1: This module clock is disabled	R/W
b27 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: This module clock is enabled 1: This module clock is disabled	R/W
b31 to b29	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): SYSTEM.MSTPCRB 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB31	MSTPB30	—	—	—	MSTPB26	MSTPB25	—	MSTPB23	—	MSTPB21	—	MSTPB19	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	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: This module clock is enabled 1: This module clock is disabled	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	DOC Module Stop	Target module: DOC 0: This module clock is enabled 1: This module clock is disabled	R/W
b8, b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPB9	ELC Module Stop	Target module: ELC 0: This module clock is enabled 1: This module clock is disabled	R/W
b10	MSTPB10	Comparator B Module Stop	Target module: CMPB 0: This module clock is enabled 1: This module clock is disabled	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: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPB19	USB 2.0 FS Host/Function Module Stop*1	Target module: USB 0: This module clock is enabled 1: This module clock is disabled	R/W
b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b21	MSTPB21	I²C Bus Interface 0 Module Stop	Target module: RIIC0 0: This module clock is enabled 1: This module clock is disabled	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: This module clock is enabled 1: This module clock is disabled	R/W
b24	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: This module clock is enabled 1: This module clock is disabled	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: This module clock is enabled 1: This module clock is disabled	R/W
b29 to b27	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: This module clock is enabled 1: This module clock is disabled	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: RSCI0 0: This module clock is enabled 1: This module clock is disabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. This bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. The clock should be set for oscillation when this bit is 0 (this module clock is enabled). When entering software standby mode after rewriting this bit, wait for two cycles of the UCLK after rewriting, and execute a WAIT instruction. When stopping the clock after rewriting this bit to 1 (this module clock is disabled), wait for two cycles of the UCLK after rewriting, and stop the clock.

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): SYSTEM.MSTPCRC 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DSLPE	—	MSTPC 29	—	MSTPC 27	MSTPC 26	—	—	—	—	—	—	MSTPC 19	—	—	—
Value after reset:	0	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0001 FFFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19	Clock Frequency Accuracy Measurement Circuit Module Stop*2	Target module: CAC 0: This module clock is enabled 1: This module clock is disabled	R/W
b25 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b26	MSTPC26	Serial Communication Interface 9 Module Stop	Target module: RSCI9 0: This module clock is enabled 1: This module clock is disabled	R/W
b27	MSTPC27	Serial Communication Interface 8 Module Stop	Target module: RSCI8 0: This module clock is enabled 1: This module clock is disabled	R/W
b28	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b29	MSTPC29	Remote Control Signal Receiver Module Stop*3	Target module: REMC0 0: This module clock is enabled 1: This module clock is disabled	R/W
b30	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31	DSLPE	Deep Sleep Mode Enable	0: Deep sleep mode is disabled 1: Deep sleep mode is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. The corresponding MSTPC0 bit should not be set to 1 during access to the RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is 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 rewriting this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating and execute the WAIT instruction.

Note 3. Rewrite the MSTPC29 bit while the oscillation of the clock whose source is REMC0 is stable. For entering software standby mode after rewriting this bit, wait for two cycles of this clock and execute the WAIT instruction.

DSLPE Bit (Deep Sleep Mode Enable)

The DSLPE bit enables or disables a transition to deep sleep mode.

When the CPU executes the WAIT instruction with the DSLPE bit set to 1 and the SBYCR.SSBY and MSTPCRA.MSTPA28 bits meet specified conditions, the MCU enters deep sleep mode. For details, refer to section 11.6.2, Deep Sleep Mode.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): SYSTEM.MSTPCRD 0008 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPD31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	MSTPD9	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b9	MSTPD9	CANFD0 Module Stop*1	Target module: CANFD0 0: This module clock is enabled 1: This module clock is disabled	R/W
b10	MSTPD10	Touch Sensor Control Unit Module Stop	Target module: CTSU 0: This module clock is enabled 1: This module clock is disabled	R/W
b30 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	MSTPD31	RSIP Module Stop*2	Target module: RSIP 0: This module clock is enabled 1: This module clock is disabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. The MSTPD9 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. Also, make the setting for clock oscillation while the setting of the MSTPD9 bit is 0 (releasing supply of the module clock from the stopped state). 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. To stop the clock after having changed the value of the MSTPD9 bit to 1 (stopping supply of the module clock), wait for two cycles of CANFDMCLK and CANFDCLK before stopping the clock.

Note 2. Set the MSTPCRD.MSTPD31 bit to 1 (stopping supply of the module clock) before placing the MCU in the software standby mode. Wait for 30 µs after return from the software standby mode before setting the MSTPCRD.MSTPD31 bit to 0 (resuming supply of the module clock).

11.2.6 Operating Power Control Register (OPCCR)

Address(es): SYSTEM.OPCCR 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	OPCM[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b0 0 0 0: High-speed operating mode 0 1 0: Middle-speed operating mode 1 0 0: Middle-speed operating mode 2 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	0: Transition completed 1: During transition	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce the power consumption in the normal operating mode, sleep mode, deep sleep mode, and snooze mode.
Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

The OPCCR register cannot be rewritten under the following conditions:

- When the OPCCR.OPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation
- When the SOPCCR.SOPCM bit is 1 (low-speed operating mode)

The OPCCR register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures of changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During a transition to an operating power control mode (while the OPCCR.OPCMTSF flag is 1), a correct value cannot be read from the E2 DataFlash. If a setting is made so that the E2 DataFlash is read using a DTC/DMAC transfer, stop the DTC or DMAC module before rewriting the OPCCR.OPCM[2:0] bits.
During sleep mode, snooze 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)).

OPCM[2:0] Bits (Operating Power Control Mode Select)

The operating power control mode in the normal operating mode, sleep mode, deep sleep mode, and snooze mode is selectable.
Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.
In middle-speed operating mode 2, do not select the main clock oscillator as the clock source.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

This flag indicates the switching control state during and after operating power mode transition.
This flag becomes 1 when the value of the OPCM[2:0] bits is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the OPCM[2:0] bits when this flag is 0.

11.2.7 Sub Operating Power Control Register (SOPCCR)

Address(es): SYSTEM.SOPCCR 0008 00AAh

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

Bit	Symbol	Bit Name	Description	R/W
b0	SOPCM	Sub Operating Power Control Mode Select	0: High-speed operating mode, middle-speed operating mode or middle-speed operating mode 2*1 1: Low-speed operating mode	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SOPCMTSF	Sub Operating Power Control Mode Transition Status Flag	0: Transition completed 1: During transition	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Depends on the setting of OPCCR.OPCM[2:0].

The SOPCCR register is used to control the transition of the MCU to the low-speed operating mode, and to reduce the power consumption in the normal operating mode, sleep mode, deep sleep mode, and snooze mode.
Setting this register initiates entry to/exit from low-speed operating mode.
Low-speed operating mode is used for the sub-clock oscillator only.
The OPCCR register cannot be rewritten when the SOPCM bit is 1 (low-speed operating mode).

The SOPCCR register cannot be rewritten under the following conditions:

- When the SOPCCR.SOPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation

This register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures for changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During a transition to a sub operating power control mode (while the SOPCCR.SOPCMTSF flag is 1), a correct value cannot be read from the E2 DataFlash. If a setting is made so that the E2 DataFlash is read using a DTC/DMAC transfer, stop the DTC or DMAC module before rewriting the SOPCCR.SOPCM bit.
During sleep mode, snooze 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)).

SOPCM Bit (Sub Operating Power Control Mode Select)

This bit is used to select the operating power control mode in the normal operating mode, sleep mode, deep sleep mode, and snooze mode.
Setting this bit to 1 allows a transition to low-speed operating mode. Setting this bit to 0 allows a return to the operating mode (operating mode set by OPCCR.OPCM[2:0]) before the transition to low-speed operating mode.
Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

SOPCMTSF Flag (Sub Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the sub operating power control mode is switched. This flag becomes 1 when the value of the SOPCM bit is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the SOPCM bit when this flag is 0.

Table 11.4 Operating Frequency and Voltage Ranges in Operating Power Control Modes

Operating Power Control Mode	OPCM[2:0] Bits	SOPCM Bit	Operating Voltage Range	Operating Frequency Range					
				Flash Memory Read Frequency					Flash Memory Programming/ Erasure Frequency
				ICLK	FCLK	PCLKD	PCLKB	PCLKA	FCLK
High-speed operating mode	000b	0	2.4 to 5.5 V	Up to 64 MHz*1	Up to 64 MHz	Up to 64 MHz	Up to 32 MHz	Up to 64 MHz	1 to 64 MHz
			1.8 to 2.4 V	Up to 48 MHz*1, *2 (up to 16 MHz*3)	Up to 48 MHz*2 (up to 16 MHz*3)	Up to 48 MHz*2 (up to 16 MHz*3)	Up to 32 MHz*2 (up to 16 MHz*3)	Up to 48 MHz*2 (up to 16 MHz*3)	1 to 48 MHz*2 (1 to 16 MHz*3)
Middle-speed operating mode	010b	0	2.4 to 5.5 V	Up to 24 MHz	Up to 24 MHz	Up to 24 MHz	Up to 24 MHz	Up to 24 MHz	1 to 24 MHz
			1.8 to 2.4 V	Up to 24 MHz *2 (up to 16 MHz*3)	Up to 24 MHz *2 (up to 16 MHz*3)	Up to 24 MHz *2 (up to 16 MHz*3)	Up to 24 MHz *2 (up to 16 MHz*3)	Up to 24 MHz *2 (up to 16 MHz*3)	1 to 24 MHz*2 (1 to 16 MHz*3)
			1.6 to 1.8 V	Up to 4 MHz*2	Up to 4 MHz*2	Up to 4 MHz*2	Up to 4 MHz*2	Up to 4 MHz*2	1 to 4 MHz*2
Middle-speed operating mode 2	100b	0	1.8 to 5.5 V	Up to 1 MHz	Up to 1 MHz	Up to 1 MHz	Up to 1 MHz	Up to 1 MHz	1 MHz
			1.6 to 1.8 V	Up to 1 MHz*2	Up to 1 MHz*2	Up to 1 MHz*2	Up to 1 MHz*2	Up to 1 MHz*2	1 MHz*2
Low-speed operating mode	000b	1	1.6 to 5.5 V	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	—
	010b	1							
	100b	1							

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.
Note: When the CANFD module is to be used, set the frequency ratios for the PCLKA and PCLKB clocks such that PCLKA:PCLKB = 2:1. See section 9, Clock Generation Circuit for details.
Note: For details on the operating frequency and voltage ranges, also refer to section 47, Electrical Characteristics.
Note 1. Selecting a clock with a frequency higher than 32 MHz requires setting the insertion of a wait state for ROM access. For details, refer to section 46, Flash Memory (FLASH).
Note 2. The RSIP module cannot be used. Do not make the settings for release from the module-stop state.
Note 3. This is applicable when the RSIP module is to be used.

Each operating power control mode is described below.

- High-Speed Operating Mode

[When the operating voltage is in the range from 2.4 to 5.5 V]

The maximum operating frequency during reading of flash memory is 64 MHz for ICLK, PCLKA, PCLKD, and FCLK; 32 MHz for PCLKB. During programming/erasure of flash memory, the operating frequency is in the range from 1 to 64 MHz.

[When the operating voltage is in the range from 1.8 to 2.4 V]

When the RSIP module is not in use, the maximum operating frequency during reading of flash memory is 48 MHz for ICLK, FCLK, PCLKA, and PCLKD; 32 MHz for PCLKB. The operating frequency during programming/erasure of flash memory is in the range from 1 to 48 MHz.

When the RSIP module is in use, the maximum operating frequency during reading of flash memory is 16 MHz for ICLK, FCLK, PCLKA, PCLKB, and PCLKD. The operating frequency during programming/erasure of flash memory is in the range from 1 to 16 MHz.

Figure 11.3 shows the relation between the operating voltage and operating frequency in the high-speed operating mode and Figure 11.4 shows the relation between the operating voltage and operating frequency in the high-speed operating mode when the RSIP is not in use.

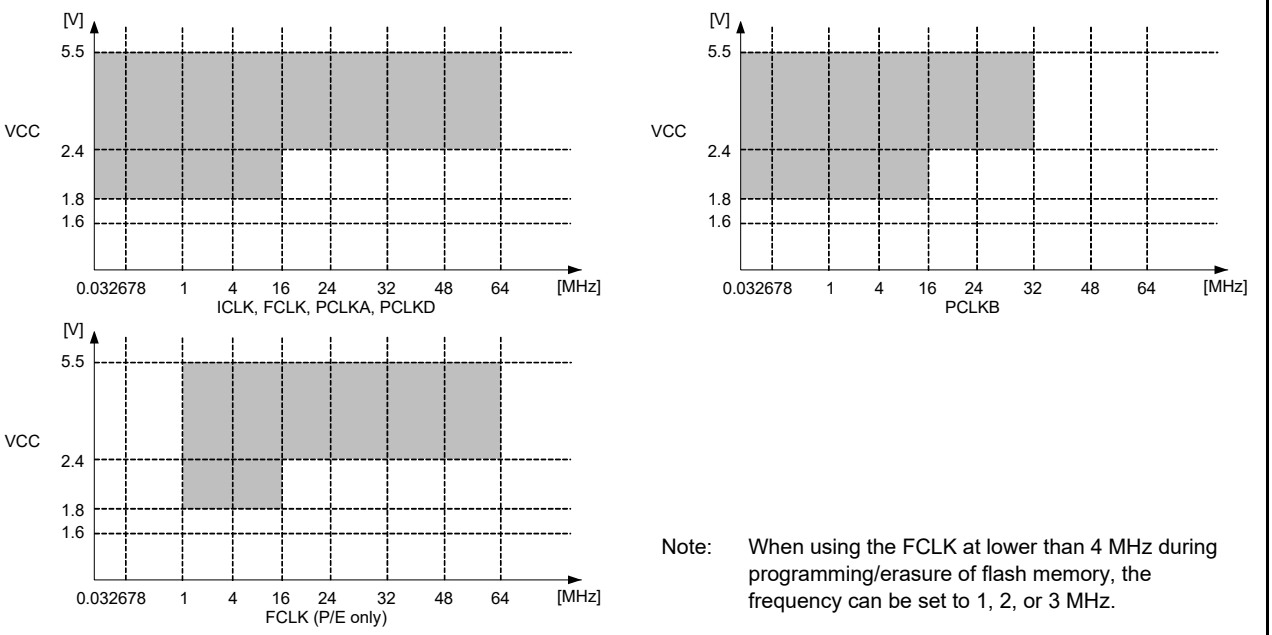


Figure 11.3 Operating Voltages and Frequencies in High-Speed Operating Mode

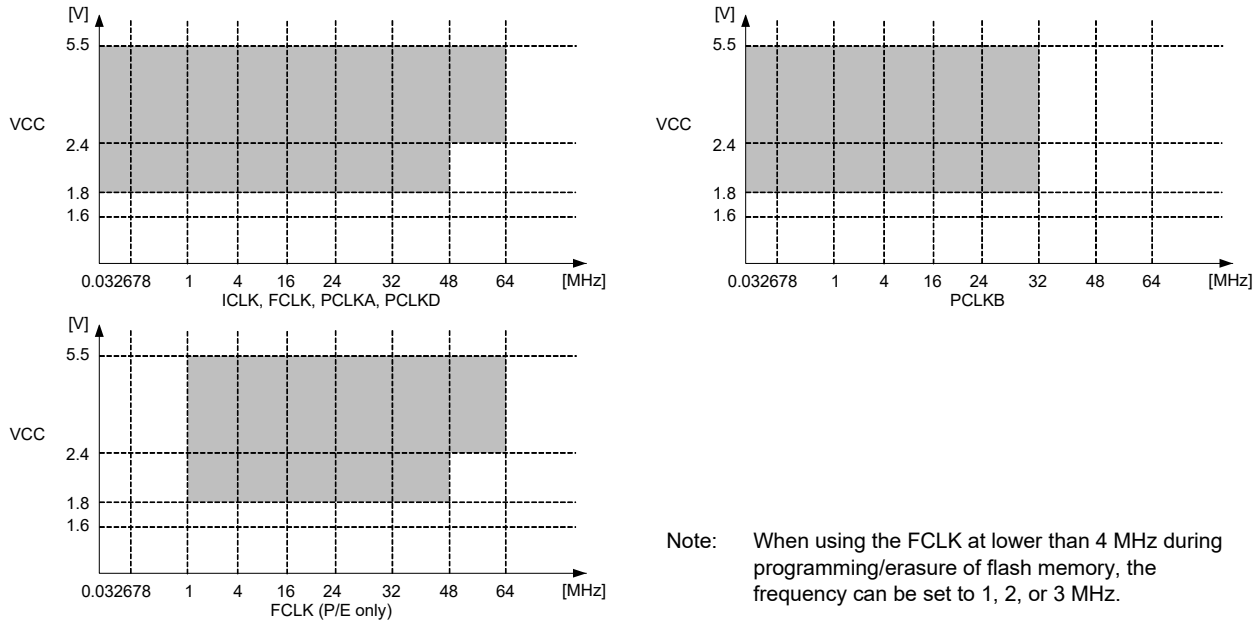


Figure 11.4 Operating Voltages and Frequencies in High-Speed Operating Mode (When the RSIP Is Not in Use)

- Middle-Speed Operating Mode

As compared to high-speed operating mode, this mode reduces power consumption for low-speed operation.

[When the operating voltage is in the range from 2.4 to 5.5 V]

The maximum operating frequency for ICLK, FCLK, PCLKA, PCLKB, and PCLKD during reading of flash memory is 24 MHz. During programming/erasure of flash memory, the operating frequency is in the range from 1 to 24 MHz.

[When the operating voltage is in the range from 1.8 to 2.4 V]

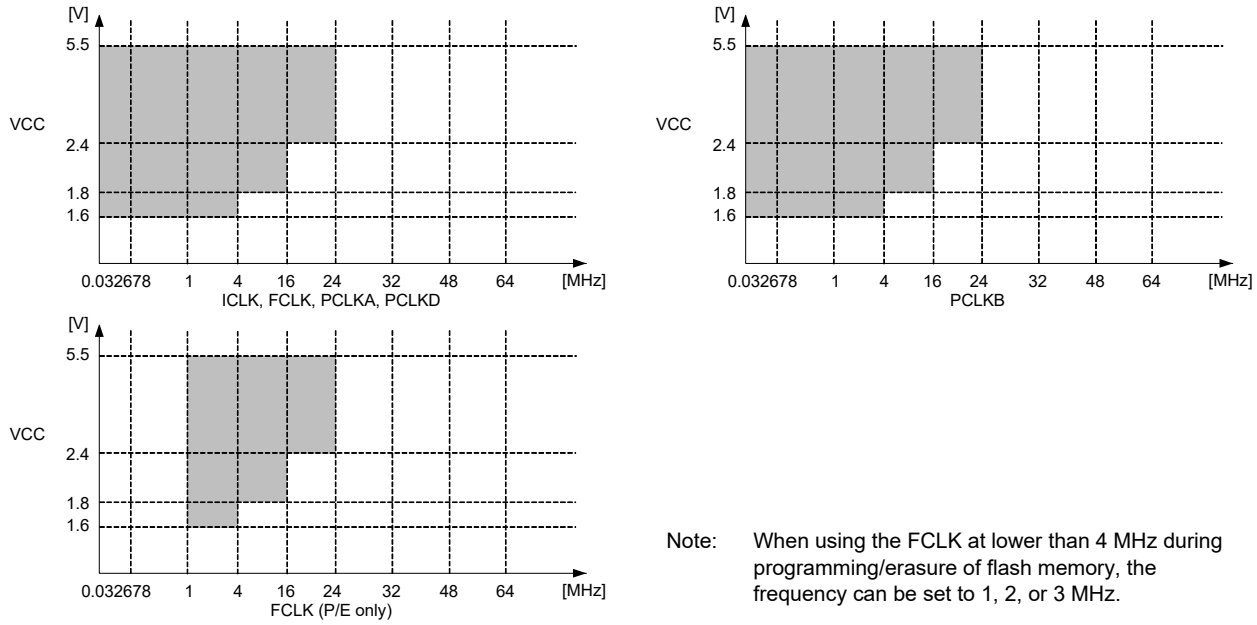
When the RSIP module is not in use, the maximum operating frequency during reading of flash memory is 24 MHz for ICLK, FCLK, PCLKA, PCLKB, and PCLKD. The operating frequency during programming/erasure of flash memory is in the range from 1 to 24 MHz.

When the RSIP module is in use, the maximum operating frequency during reading of flash memory is 16 MHz for ICLK, FCLK, PCLKA, PCLKB, and PCLKD. The operating frequency during programming/erasure of flash memory is in the range from 1 to 16 MHz.

[When the operating voltage is in the range from 1.6 to 1.8 V]

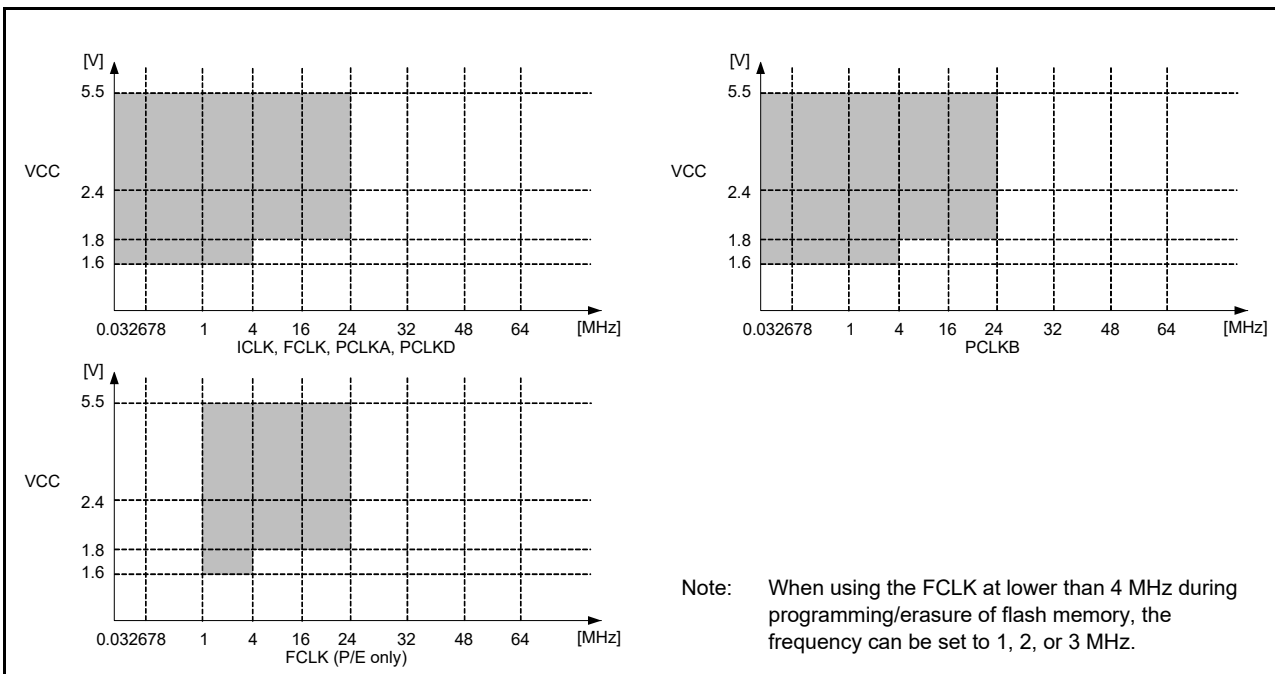
The maximum operating frequency for ICLK, FCLK, PCLKA, PCLKB, and PCLKD during reading of flash memory is 4 MHz. During programming/erasure of flash memory, the operating frequency is in the range from 1 to 4 MHz. The RSIP module cannot be used.

Figure 11.5 shows the relation between the operating voltage and operating frequency in the middle-speed operating mode and Figure 11.6 shows the relation between the operating voltage and operating frequency in the middle-speed operating mode when the RSIP is not in use.



Note: When using the FCLK at lower than 4 MHz during programming/erasure of flash memory, the frequency can be set to 1, 2, or 3 MHz.

Figure 11.5 Operating Voltages and Frequencies in Middle-Speed Operating Mode



Note: When using the FCLK at lower than 4 MHz during programming/erasure of flash memory, the frequency can be set to 1, 2, or 3 MHz.

Figure 11.6 Operating Voltages and Frequencies in Middle-Speed Operating Mode (When the RSIP Is Not in Use)

- Middle-Speed Operating Mode 2

As compared to middle-speed operating mode, this mode reduces power consumption for low-speed operation. In middle-speed operating mode 2, do not select the main clock oscillator as the clock source.

[When the operating voltage is in the range from 1.8 to 5.5 V]

The maximum operating frequency for ICLK, FCLK, PCLKA, PCLKB, and PCLKD during reading of flash memory is 1 MHz. During programming/erasure of flash memory, the operating frequency is 1 MHz.

[When the operating voltage is in the range from 1.6 to 1.8 V]

The maximum operating frequency for ICLK, FCLK, PCLKA, PCLKB, and PCLKD during reading of flash memory is 1 MHz. During programming/erasure of flash memory, the operating frequency is 1 MHz. The RSIP module cannot be used.

Figure 11.7 shows the relation between the operating voltage and operating frequency in the middle-speed operating mode 2.

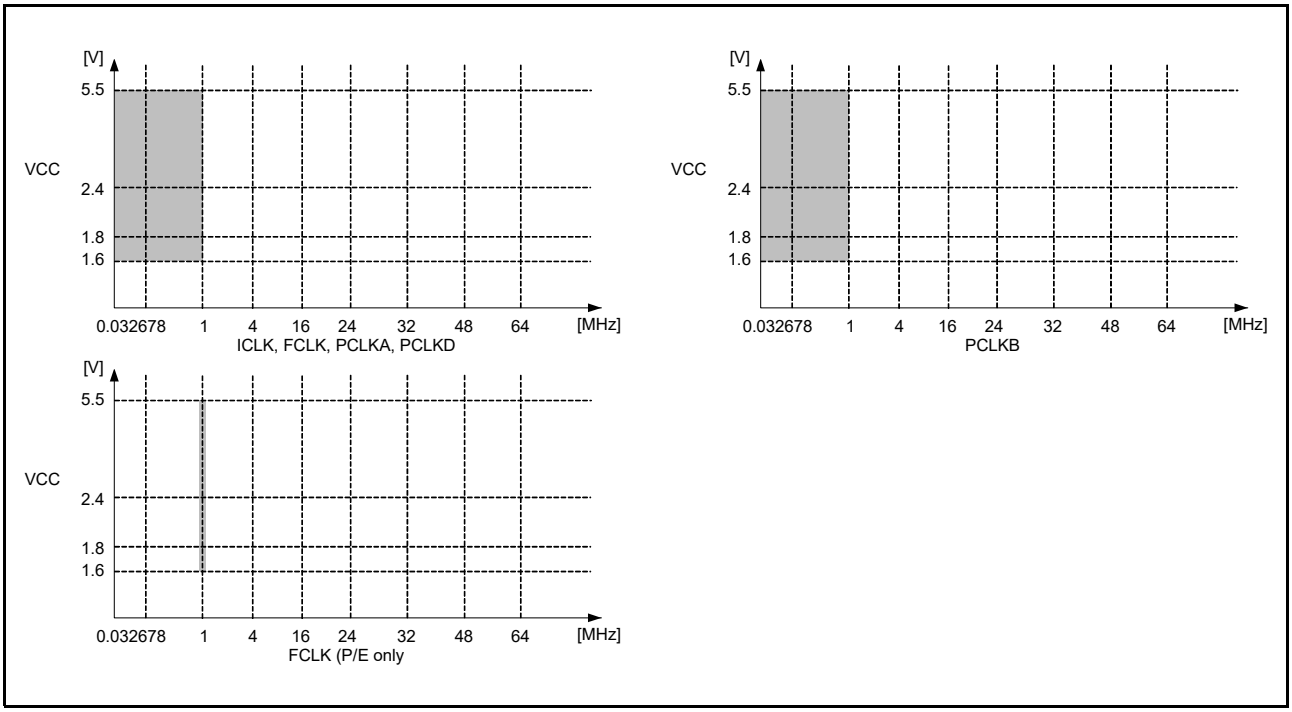


Figure 11.7 Operating Voltages and Frequencies in Middle-Speed Operating Mode 2

- Low-Speed Operating Mode

A transition to low-speed operating mode is set by writing 1 to the SOPCM bit in the SOPCCR register. The setting of the OPCM[2:0] bits cannot be modified during low-speed operating mode. This mode is used only for the sub oscillator of 32.768 kHz.

The maximum operating frequency for ICLK, FCLK, PCLKA, PCLKB, and PCLKD during reading of flash memory is 32.768 kHz. The operating voltage is in the range from 1.6 to 5.5 V.

The following restrictions apply when low-speed operating mode is selected:

- P/E operations for flash memory are prohibited.
- The PLL/PLL2, main clock oscillator, LOCO, and HOCO cannot be used.

Note: The SOPCM bit cannot be set to 1 when the PLLCR2.PLLEN bit is 0 (PLL is operating).
The SOPCM bit cannot be set to 1 when the PLL2CR2.PLL2EN bit is 0 (PLL2 is operating).
The SOPCM bit cannot be set to 1 when the HOCOCR.HCSTP bit is 0 (HOCO is operating).
The SOPCM bit cannot be set to 1 when the MOSCCR.MOSTP bit is 0 (Main clock oscillator is operating).
The SOPCM bit cannot be set to 1 when the LOCOCR.LCSTP bit is 0 (LOCO is operating).

Figure 11.8 shows the relation between the operating voltage and operating frequency in the low-speed operating mode.

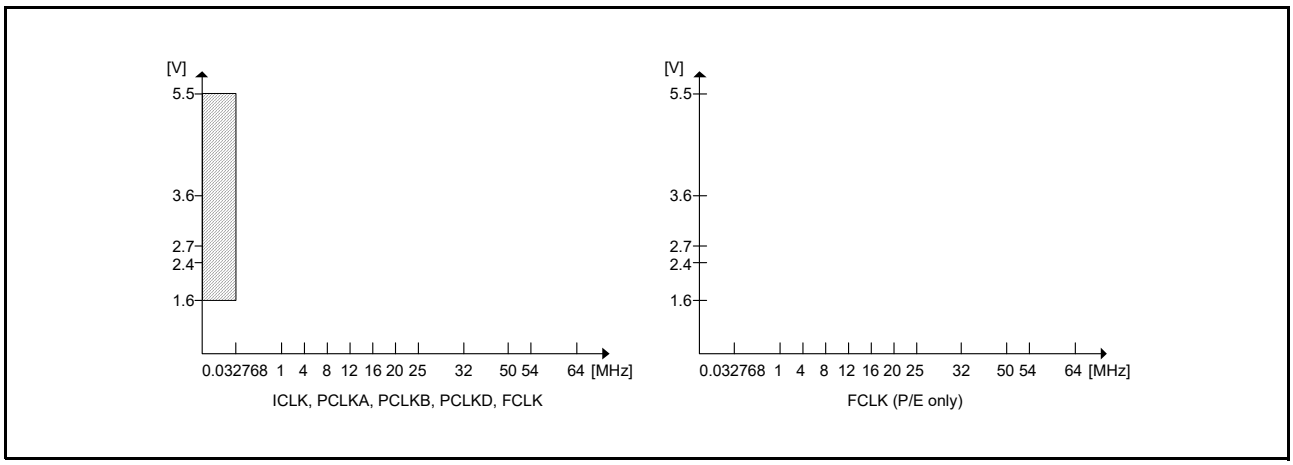


Figure 11.8 Operating Voltages and Frequencies in Low-Speed Operating Mode

11.2.8 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): SYSTEM.RSTCKCR 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 0: LOCO is selected 0 0 1: HOCO is selected*1 0 1 0: Main clock oscillator is selected*2 Settings other than above are prohibited when the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching at exit from sleep mode is disabled 1: Clock source switching at exit from sleep mode is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. The HOCO is only selectable when the high-speed operating mode is to be entered.
Note 2. The main clock oscillator is only selectable when the high-speed or middle-speed operating mode is to be entered.

RSTCKCR is used to control clock source switching at exit from sleep mode.
When exit from sleep mode is initiated by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP), the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP), and the LOCO stop bit in the low-speed on-chip oscillator control register (LOCOCR.LCSTP) are automatically modified to the operating state corresponding to the clock source to be used after transition. The value of the RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used at exit from sleep mode.
The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.
In transitions between operating modes shown in Figure 11.2, when the transition from the sleep mode will be to the high-speed operating mode, any of the LOCO, HOCO, and main clock oscillator is selectable.
When the transition from the sleep mode will be to the middle-speed operating mode, either the LOCO or main clock oscillator is selectable. When the transition from the sleep mode will be to middle-speed operating mode 2, the LOCO is selectable.
Note that when the transition from the sleep mode will be to middle-speed operating mode 2, set the frequencies of each clock (ICLK, FCLK, PCLKA, PCLKB, and PCLKD) to 1 MHz or less.

Table 11.5 When Exiting Sleep Mode to High-Speed Operating Mode and Middle-Speed Operating Mode

Operating Mode during Sleep	Clock Source during Sleep	RSTCKSEL[2:0]	Operating Mode after Exiting	Clock Source after Exiting
High-speed operating mode or low-speed operating mode after exit from high-speed operating mode	Sub-clock oscillator	000b (LOCO)	High-speed operating mode	LOCO
		001b (HOCO)		HOCO
		010b (main clock oscillator)		Main clock oscillator
Middle-speed operating mode or low-speed operating mode after exit from middle-speed operating mode	Sub-clock oscillator	000b (LOCO)	Middle-speed operating mode	LOCO
		010b (main clock oscillator)		Main clock oscillator
Middle-speed operating mode 2 or low-speed operating mode after exit from middle-speed operating mode 2	Sub-clock oscillator	000b (LOCO)	Middle-speed operating mode 2	LOCO*1

Note 1. The frequency of each clock (ICLK, FCLK, PCLKA, PCLKB, and PCLKD) must be lower than 1 MHz.

RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching when sleep mode is exited.

The clock source can be switched when exiting sleep mode only while the sub-clock oscillator is selected as the clock for entering sleep mode. Do not enable this bit when entering sleep mode while the HOCO, LOCO, main clock oscillator, or PLL is selected as the clock source.

When returning from sleep mode while this bit is enabled, the SOPCM bit in the SOPCCR register is automatically rewritten to 0 (middle-speed operating mode 2, middle-speed operating mode or high-speed operating mode).

The value of the frequency division setting (in the SCKCR register) is retained.

When the MCU is placed in the middle-speed operating mode 2 by release from the sleep mode, set the frequencies of each clock to 1 MHz or less.

11.2.9 Snooze Control Register (SNZCR)

Address(es): SYSTEM.SNZCR 0008 00AEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SNZDTCE	—	ADCSNZ3SEL [1:0]		ADCSNZ2SEL [1:0]		REMCSNZSEL [1:0]		CTSUSNZSEL [1:0]		ADCSNZSEL [1:0]		LPTSNZSEL [1:0]		SCISNZSEL [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	SCISNZSEL [1:0]	SCI Snooze Operation Select	Setting of the conditions regarding SCI5 for the MCU to be placed in the snooze mode and placed back in the software standby mode b1 b0 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode on detection of the falling edge of RXD5. It is placed back in the software standby mode when the received data do not match the value of the CDR register of SCI5. 1 1: The MCU is placed in the snooze mode on detection of the falling edge of RXD5. It is placed back in the software standby mode when the received data do not match the value of the CDR register of SCI5, or once transfer of the received data by the DTC has been completed.	R/W
b3, b2	LPTSNZSEL [1:0]	LPT Snooze Operation Select	Setting of the conditions regarding the LPT for the MCU to be placed in the snooze mode and placed back in the software standby mode b3 b2 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will not be placed back in the software standby mode. 1 1: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will be placed back in the software standby mode once a single round of the DTC transfer triggered by the LPT compare match 1 has been completed.	R/W
b5, b4	ADCSNZSEL [1:0]	S12AD Snooze Operation Select	Setting of the conditions regarding the S12AD for the MCU to be placed in the snooze mode and placed back in the software standby mode b5 b4 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will not be placed back in the software standby mode. 1 1: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will be placed back in the software standby mode once a single round of the DTC transfer triggered by the end of A/D conversion has been completed.	R/W
b7, b6	CTSUSNZSEL [1:0]	CTSU Snooze Operation Select	Setting of the conditions regarding the CTSU for the MCU to be placed in the snooze mode and placed back in the software standby mode b7 b6 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will be placed back in the software standby mode on request of the end of the snooze mode. 1 1: Setting prohibited	R/W

Bit	Symbol	Bit Name	Description	R/W
b9, b8	REMCSNZSEL [1:0]	REMC Snooze Operation Select	Setting of the conditions regarding the REMC for the MCU to be placed in the snooze mode and placed back in the software standby mode b9 b8 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode in response to start of a clock request by the REMC, and will be placed back in the software standby mode once completion of the clock request by the REMC has been detected. 1 1: Setting prohibited	R/W
b11, b10	ADCSNZ2SEL [1:0]	S12AD Snooze Operation 2 Select	Setting of the conditions regarding the S12AD for the MCU to be placed in the snooze mode and placed back in the software standby mode b11 b10 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will be placed back in the software standby mode if the S12AD comparison conditions were not being met. 1 1: Setting prohibited	R/W
b13, b12	ADCSNZ3SEL [1:0]	S12AD Snooze Operation 3 Select	Setting of the conditions regarding the S12AD for the MCU to be placed in the snooze mode and placed back in the software standby mode b13 b12 0 x: The MCU is not placed in the snooze mode by the setting of these bits. 1 0: The MCU is placed in the snooze mode in response to an LPT compare match 1, and will be placed back in the software standby mode if the S12AD comparison conditions were being met. 1 1: Setting prohibited	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	SNZDTCE	DTC Enable in the Snooze Mode	Setting of the enabling of the DTC in the snooze mode. 0: The DTC transfer is disabled in the snooze mode. 1: The DTC transfer is enabled in the snooze mode.	R/W

x: Don't care

Note: Change the value of this register after setting the PRCR.PRC1 bit to 1 (writing enabled).

The SNZCR register is used to specify the conditions for transitions from the software standby mode to the snooze mode and for release from the snooze mode. Set this register in accordance with the type of peripheral modules operating in the snooze mode before making a transition to the software standby mode. Also set the peripheral modules, interrupt controller, and the DTC that are operating in the snooze mode before making a transition to the software standby mode. In addition, when this register is to be used to specify the snooze transition and end conditions, set the low-power timer standby wakeup enable bit of the low-power timer standby wakeup enable register (LPWUCR.LPWKUPEN) to 0. Setting of this register does not affect the operation in the operating mode other than the software standby.

SCISNZSEL [1:0] Bits (SCI Snooze Operation Select)

These bits are used to select waking up the MCU from the software standby mode to enable data reception by SCI5. Setting these bits to 10b selects placing the MCU in the snooze mode for data reception on detection of a start bit on the RXD5 pin while the MCU is in the software standby mode. If the received data do not match the value of the CDR register of SCI5, the MCU is placed back in the software standby mode.

Setting these bits to 11b selects placing the MCU in the snooze mode for data reception on detection of a start bit on the RXD5 pin while the MCU is in the software standby mode. The MCU is placed back in the software standby mode when the received data do not match the value of the CDR register of SCI5, or when they match and transfer of the received data by the DTC is completed. Once the received data match the value of the CDR register, the MCU is placed in the snooze mode on detection of a start bit, and is placed back in the software standby mode every time a DTC transfer is completed.

LPTSNZSEL [1:0] Bits (LPT Snooze Operation Select)

These bits are used to select waking up the MCU from the software standby mode and placing it in the snooze mode in

response to an LPT compare match 1.
Setting these bits to 10b or 11b selects waking up the MCU from the software standby mode and placing it in the snooze mode on detection of an LPT compare match 1.
When the setting of these bits is 10b, the MCU remains in the snooze mode until an interrupt signal is asserted to release the snooze mode.
When the setting of these bits is 11b, the MCU is placed back in the software standby mode once a single round of the DTC transfer triggered by an LPT compare match 1 has been completed.

ADCSNZSEL [1:0] Bits (S12AD Snooze Operation Select)

These bits are used to select waking up the MCU from the software standby mode to enable A/D conversion by the S12AD. When you use these bits, specify the LPT compare match 1 as a trigger to start a conversion by using the event link controller (ELC).
Setting these bits to 10b or 11b selects waking up the MCU from the software standby mode and placing it in the snooze mode to enable A/D conversion on detection of an LPT compare match 1.
When the setting of these bits is 10b, the MCU remains in the snooze mode until an interrupt signal is asserted to release the snooze mode.
When the setting of these bits is 11b, the MCU is placed back in the software standby mode once a single round of the DTC transfer triggered by the end of A/D conversion has been completed.

CTSUSNZSEL [1:0] Bits (CTSU Snooze Operation Select)

These bits are used to select waking up the MCU from the software standby mode to enable CTSU operations. When you use these bits, specify an external trigger to start measurement by the CTSU and select the LPT compare match 1 as an event signal for the CTSU by using the ELC.
When the setting of these bits is 10b, the MCU is placed in the snooze mode to enable measurement on detection of an LPT compare match 1. Once the measurement has been completed and a snooze end request is generated by the CTSU, the MCU is placed back in the software standby mode.

REMCSNZSEL [1:0] Bits (REMC Snooze Operation Select)

These bits are used to select waking up the MCU from the software standby mode to enable remote control signal reception.
Setting these bits to 10b selects waking up the MCU from the software standby mode and placing it in the snooze mode on detection of start of a clock request by the REMC. Once completion of the clock request by the REMC has been detected, the MCU is placed back in the software standby mode.

ADCSNZ2SEL [1:0] Bits (S12AD Snooze Operation 2 Select)

These bits are used to select waking up the MCU from the software standby mode to enable A/D conversion. When you use these bits, specify the LPT compare match 1 as a trigger to start a conversion by using the event link controller (ELC).
Setting these bits to 10b selects waking up the MCU from the software standby mode and placing it in the snooze mode to enable A/D conversion on detection of an LPT compare match 1. Once the comparison conditions on completion of S12AD conversion in the snooze mode are not being met, the MCU is placed back in the software standby mode.

ADCSNZ3SEL [1:0] Bits (S12AD Snooze Operation 3 Select)

These bits are used to select waking up the MCU from the software standby mode to enable A/D conversion. When you use these bits, specify the LPT compare match 1 as a trigger to start a conversion by using the event link controller (ELC).
Setting these bits to 10b selects waking up the MCU from the software standby mode and placing it in the snooze mode to enable A/D conversion on detection of an LPT compare match 1. Once the comparison conditions on completion of S12AD conversion in the snooze mode are being met, the MCU is placed back in the software standby mode.

SNZDTCE Bit (DTC Enable in the Snooze Mode)

This bit is used to enable the DTC and RAM in the snooze mode. To operate the DTC and RAM in the snooze mode, set this bit to 1 before making a transition to the software standby mode.

11.2.10 Snooze Control Register 2 (SNZCR2)

Address(es): SYSTEM.SNZCR2 0008 00ACh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
		—	—	ADE3	—	ADE2	CTSUFNE[1:0]		ADE[1:0]		LPTCM1E[1:0]		SCIRXE[1:0]		—	SCIERE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SCIERE	SCI Reception Error Select	0: The SCI5 reception error is not selected as the condition for generating the interrupt for release from the snooze mode. 1: The SCI5 reception error is selected as the source for generating the interrupt for release from the snooze mode.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	SCIRXE[1:0]	SCI Reception Data Full Select	b3 b2 0 0: The SCI5 reception data full is not selected as the condition for generating the interrupt for release from the snooze mode. 0 1: The SCI5 reception data full is selected as the source for generating the interrupt for release from the snooze mode. 1 x: The DTC transfer completion event in response to the SCI5 reception data full is selected for generating the interrupt for release from the snooze mode.	R/W
b5, b4	LPTCM1E[1:0]	LPT Compare Match 1 Select	b5 b4 0 0: The LPT compare match 1 is not selected as the condition for generating the interrupt for release from the snooze mode. 0 1: The LPT compare match 1 is selected as the source for generating the interrupt for release from the snooze mode. 1 x: The DTC transfer completion event in response to the LPT compare match 1 is selected for generating the interrupt for release from the snooze mode.	R/W
b7, b6	ADE[1:0]	Completion of the S12AD Conversion Select	b7 b6 0 0: The completion of the S12AD conversion is not selected as the condition for generating the interrupt for release from the snooze mode. 0 1: The completion of the S12AD conversion is selected as the source for generating the interrupt for release from the snooze mode. 1 x: The DTC transfer completion event in response to the completion of the S12AD conversion is selected for generating the interrupt for release from the snooze mode.	R/W
b9, b8	CTSUFNE[1:0]	CTSU Measurement Finish Select	b9 b8 0 0: The completion of the CTSU measurement is not selected as the condition for generating the interrupt for release from the snooze mode. 0 1: The completion of the CTSU measurement is selected as the source for generating the interrupt for release from the snooze mode. 1 x: Setting prohibited	R/W
b10	ADE2	S12AD Comparison Conditions Non-Satisfaction Select	0: The S12AD comparison conditions non-satisfaction event is not selected for release from the snooze mode. 1: The S12AD comparison conditions non-satisfaction event is selected for release from the snooze mode.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADE3	S12AD Comparison Conditions Satisfaction Select	0: The S12AD comparison conditions satisfaction event is not selected for release from the snooze mode. 1: The S12AD comparison conditions satisfaction event is selected for release from the snooze mode.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

Note: Write a new value to this register after setting the PRCR.PRC1 bit to 1 (writing is enabled).

The SNZCR2 register is used to select the interrupt sources for release from the snooze mode. If the selected source or event is generated, the interrupt for release from the snooze mode is generated. Set this register before the MCU is placed in the software standby mode according to the type of the peripheral modules operating in the snooze mode. If the multiple sources or events are selected, an interrupt for release from the snooze mode is generated whenever one of the selected sources or events is generated.
This register setting is effective in the operation mode other than the software standby.

SCIERE Bit (SCI Reception Error Select)

This bit is used to set the SCI5 reception error as the condition for generating the interrupt for release from the snooze mode. If the SCIERE bit is set to 1, the SCI5 reception error is selected as the source for generating the interrupt for release from the snooze mode.

SCIRXE[1:0] Bits (SCI Reception Data Full Select)

These bits are used to set the SCI5 reception data full as the condition for generating the interrupt for release from the snooze mode. If the SCIRXE[1:0] bits are set to 01b, the SCI5 reception data full is selected as the source for generating the interrupt for release from the snooze mode.
If the SCIRXE[1:0] bits are set to 10b or 11b, the DTC transfer completion event in response to the SCI5 reception data full is selected as the source for generating the interrupt for release from the snooze mode.

LPTCM1E[1:0] Bits (LPT Compare Match 1 Select)

These bits are used to set the LPT compare match 1 as the condition for generating the interrupt for release from the snooze mode. If the LPTCM1E[1:0] bits are set to 01b, the LPT compare match 1 is selected as the source for generating the interrupt for release from the snooze mode.
If the LPTCM1E[1:0] bits are set to 10b or 11b, the DTC transfer completion event in response to the LPT compare match 1 is selected as the source for generating the interrupt for release from the snooze mode.

ADE[1:0] Bits (Completion of the S12AD Conversion Select)

These bits are used to set the completion of the S12AD conversion as the condition for generating the interrupt for release from the snooze mode. If the ADE[1:0] bits are set to 01b, the completion of the S12AD conversion is selected as the source for generating the interrupt for release from the snooze mode.
If the ADE[1:0] bits are set to 10b or 11b, the DTC transfer completion event in response to the completion of the S12AD conversion is selected as the source for generating the interrupt for release from the snooze mode.

CTSUFNE[1:0] Bits (CTSU Measurement Finish Select)

These bits are used to set the completion of the CTSU measurement as the condition for generating the interrupt for release from the snooze mode. If the CTSUFNE[1:0] bits are set to 01b, the completion of the CTSU measurement is selected as the source for generating the interrupt for release from the snooze mode.

ADE2 Bit (S12AD Comparison Conditions Non-Satisfaction Select)

This bit is used to set the S12AD comparison conditions non-satisfaction event for release from the snooze mode. If the ADE2 bit is set to 1, the S12AD comparison conditions non-satisfaction event is selected for release from the snooze mode.

ADE3 Bit (S12AD Comparison Conditions Satisfaction Select)

This bit is used to set the S12AD comparison conditions satisfaction event for release from the snooze mode. If the ADE3 bit is set to 1, the S12AD comparison conditions satisfaction event is selected for release from the snooze mode.

11.2.11 RAM Power Saving Control Register (RPSCR)

Address(es): SYSTEM.RPSCR 0008 009Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PSA3	PSA2	PSA1	PSA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSA0	Power Supply Shutoff Area 0 Select	0: The area from 0000 0000h to 0000 7FFFh is not set as a power-supply shutoff area. 1: The area from 0000 0000h to 0000 7FFFh is set as a power-supply shutoff area.	R/W
b3	PSA1	Power Supply Shutoff Area 1 Select	0: The area from 0000 8000h to 0000 FFFFh is not set as a power-supply shutoff area. 1: The area from 0000 8000h to 0000 FFFFh is set as a power-supply shutoff area.	R/W
b2	PSA2	Power Supply Shutoff Area 2 Select	0: The area from 0001 0000h to 0001 7FFFh is not set as a power-supply shutoff area. 1: The area from 0001 0000h to 0001 7FFFh is set as a power-supply shutoff area.	R/W
b3	PSA3	Power Supply Shutoff Area 3 Select	0: The area from 0001 8000h to 0001 FFFFh is not set as a power-supply shutoff area. 1: The area from 0001 8000h to 0001 FFFFh is set as a power-supply shutoff area.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Don’t care

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

PSAn Bit (Power Supply Shutoff Area n Select) (n = 0 to 3)

This bit is used to set the area to which the supply of power is shut off in the software standby mode. Use of this bit produces a further reduction in the current drawn in the software standby mode.

Set the bit before a transition to the software standby mode. Wait for at least 1.2 µs after return from the software standby mode before using the area of RAM to which the supply of power had been shut off.

Note that data in the area set by the bit become undefined after a transition to the software standby mode.

11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency can change by setting the SCKCR.FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits. The CPU, DMAC, DTC, ROM, and RAM clocks can be set by the ICK[3:0] bits.
The peripheral module clocks can be set by the PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits. The flash memory clock can be set by the FCK[3:0] bits.
For details, refer to section 9, Clock Generation Circuit.

11.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to D; i = 0 to 31) in 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 exits the module state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module stop state.
After a reset is canceled, all modules other than the DMAC, DTC, and on-chip RAM are in the module stop state. Basically the registers in the module stop state cannot be read or written. However, note that data may be written to these registers if write access is made immediately after the setting of the module stop state. To avoid this, always write to the module stop registers after confirming that the last register setting is done.

11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal mode, sleep mode, deep sleep mode, and snooze mode.

11.5.1 Setting Operating Power Control Mode

Ensure that the frequencies of the clock signals are within the range of operating frequencies both before and after switching the operating power control mode.
Examples of the procedures for switching operating power control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

- Example 1: From high-speed operating mode to middle-speed operating mode

(High-speed operation in high-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bits to 010b (middle-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
(Middle-speed operation in middle-speed operating mode)

- Example 2: From high-speed/middle-speed operating mode to low-speed operating mode

(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for low-speed operating mode
↓
Confirm that all clock sources but the sub-clock oscillator are stopped
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the SOPCCR.SOPCM bit to 1 (low-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
(Low-speed operation in low-speed operating mode)

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

- Example 1: From low-speed operating mode to high-speed/middle-speed operating mode

(Low-speed operation in low-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the SOPCCR.SOPCM bit to 0 (high-speed operating mode or middle-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed/middle-speed operating mode
↓
(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)

- Example 2: From middle-speed operating mode to high-speed operating mode

(Middle-speed operation in middle-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bit to 0 (high-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed operating mode
↓
(High-speed operation in high-speed operating mode)

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 Entry to Sleep Mode

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

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for exit from sleep mode.
(3) Set the priority*3 of the interrupt to be used for exit 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 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.1.2 Exit from Sleep Mode

Exit from sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), sleep mode is not exited.
- Initiated by a RES# pin reset
  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 handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in sleep mode and sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 14, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

11.6.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used for exit from sleep mode, set the sleep mode return clock source switching register (RSTCKCR) and the wait control register for each clock. 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 exits sleep mode. At this time, the registers related to clock source switching are automatically rewritten.
For details, refer to section 11.2.8, Sleep Mode Return Clock Source Switching Register (RSTCKCR).
For details on settings the oscillation stabilization wait time, refer to section 9.2.24, Main Clock Oscillator Wait Control Register (MOSCWTCR).

11.6.2 Deep Sleep Mode

11.6.2.1 Entry to Deep Sleep Mode

When a WAIT instruction is executed with the MSTPCRC.DSLPE bit set to 1, the MSTPCRA.MSTPA28 bit set to 1, and the SBYCR.SSBY bit cleared to 0, a transition to deep sleep mode is made.
In deep sleep mode, the CPU and the DMAC, DTC, ROM, and RAM clocks stop. Peripheral functions do not stop.
When the WDT is used, the WDT stops counting when deep sleep mode is entered.
Counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to deep 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 deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.

To use deep sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for exit from deep sleep mode.
(3) Set the priority*3 of the interrupt to be used for exit from deep 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 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.2.2 Exit from Deep Sleep Mode

Exit from deep sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from deep sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), deep sleep mode is not exited.
- Initiated by the RES# pin reset
  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 handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by the independent watchdog timer
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in deep sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in deep sleep mode and deep sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 14, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

11.6.3 Software Standby Mode

11.6.3.1 Entry to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions*1, and all the other functions except the sub-clock oscillator stop. However, the contents of the CPU internal registers, RAM data, the states of on-chip peripheral functions, the I/O ports, and the sub-clock oscillator are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.
Set the DMAST.DMST bit of the DMAC to 0 before executing the WAIT instruction. When the DTC is not to be started in the snooze mode, set the DTCST.DTCST bit in the DTC to 0 before executing the WAIT instruction.
When the RSIP module is in use, set the MSTPCRD.MSTPD31 bit to 1 (stopping supply of the RSIP module clock) before executing the WAIT instruction.
When the WDT is used, the WDT stops counting when software standby mode is entered.
Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit 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 IWDTCSTPR.SLCSTP bit is 0.
When the oscillation stop detection function is enabled, software standby mode cannot be entered. To make a transition to software standby mode, execute a WAIT instruction after setting the OSTDE bit to 0.

To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*2 of the CPU to 0.
(2) Set the interrupt request destination*3 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*4 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*2 of the CPU.
(4) Set the IERm.IENj bit*4 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*2 of the CPU to 1).

Note 1. Operation of the REMC is possible when the clock source of the operating clock is the sub clock or IWDT-dedicated on-chip oscillator clock.
Note 2. For details, refer to section 2, CPU.
Note 3. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 4. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.3.2 Exit from Software Standby Mode

Release from the software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ7), peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring, ELC (LPT-dedicated), REMC, and USB), an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When any trigger which initiates exit from software standby mode is asserted, the oscillators which were operating before entry to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

When the RSIP is to be used after return from the software standby mode, wait for 30 µs before setting the MSTPCRD.MSTPD31 bit to 0 (releasing supply of the RSIP module clock from the stopped state).

- Initiated by an interrupt
  When an interrupt request from the NMI, IRQ0 to IRQ7, RTC alarm, RTC periodic, IWDT, voltage monitoring, ELC (LPT-dedicated), REMC, USB, each oscillator which was operating before a transition to software standby mode resumes oscillation. After the oscillation stabilization wait time of each oscillator set by the MOSCWTCR.MSTS[4:0] bits has elapsed, the MCU exits software standby mode and interrupt exception processing starts.
- Initiated by a RES# pin reset
  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.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU.
  Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

11.6.3.3 Example of Software Standby Mode Application

Figure 11.9 shows an example of entry to software standby mode by the falling edge of the IRQn pin, and exit from software standby mode by 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 entry to software standby mode is completed. After that, exit from software standby mode is initiated by the rising edge of the IRQn pin.
To exit software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, refer to section 14, Interrupt Controller (ICUb).

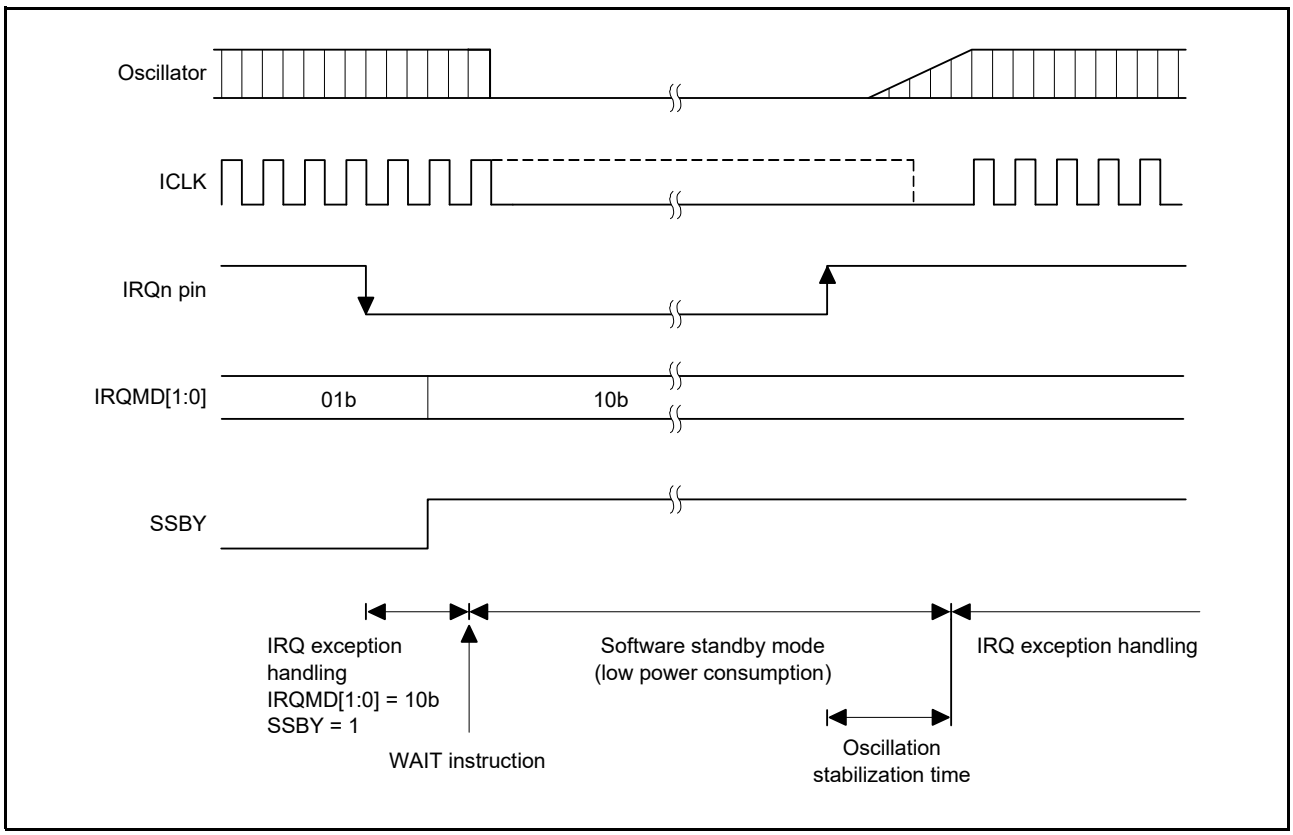


Figure 11.9 Example of Software Standby Mode Application

11.6.4 Snooze Mode

In the snooze mode, the peripheral modules temporarily resume their operations while in the software standby mode.

11.6.4.1 Placing the MCU in the Snooze Mode

When the MCU is placed in the software standby mode while the snooze transition condition is specified in the SNZCR register and then the corresponding condition is detected, the MCU is placed in the snooze mode.
In the snooze mode, peripheral modules other than the CPU and ROM, oscillators, and on-chip oscillators resume operations.
As for modules not to be used in the snooze mode, place them in the module stop or stopped state before the MCU is placed in the software standby mode. As for modules to be used in the snooze mode, set their operations before the MCU is placed in the software standby mode.
If the DTC is not to be used in the snooze mode, set the DTCST.DTCST bit in the DTC to 0, set SNZCR.SNZDTCE to 0 and then place the MCU to the software standby mode. If the DTC is to be used in the snooze mode, set the DTCST.DTCST bit in the DTC to 1, set SNZCR.SNZDTCE to 1 and then place the MCU to the software standby mode.

11.6.4.2 Return to the Software Standby Mode from the Snooze Mode

When the MCU is placed in the snooze mode while the snooze end condition is specified in the SNZCR register and then the corresponding condition is detected, the oscillators, on-chip oscillators, and peripheral modules stop operating and the MCU is returned to the software standby mode.
The snooze end condition is only enabled in the snooze mode and ignored in the state other than the snooze mode.
If the multiple combinations of snooze transition and snooze end conditions are set in the SNZCR register, some of them may be in the state that the snooze transition generated and the snooze end condition not generated. In that case, the MCU will not be placed in the software standby mode in response to the snooze end condition generated in other combinations.

11.6.4.3 Exit from Snooze Mode

Release from the snooze mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ7), peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring, REMC, and USB), an interrupt for release from the snooze mode, a non-maskable RAM error interrupt, an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset.

- Initiated by an interrupt
  Any signal among the NMI, IRQ0 to IRQ7, RTC alarm, RTC periodic, IWDT, voltage monitoring, REMC, USB, release from the snooze mode, and non-maskable RAM error interrupts serves as a trigger for release from the snooze mode and software standby mode, after which the CPU starts the interrupt exception processing. The interrupt for release from the snooze mode is only used to return to the normal operating mode from the snooze mode. Setting the SNZCR2 register allows selection of a source for the interrupt for release from the snooze mode. Before placing the MCU in the software standby mode, set the SNZCR2 register to enable the source for the interrupt for release from the snooze mode. Once the MCU has been returned to the normal operating mode, disable the source for the interrupt.
- Initiated by a RES# pin reset
  Driving the RES# pin low resets the CPU. Driving the RES# pin high allows the CPU to start the reset exception handling.
- Initiated by a power-on reset
  When a power-on reset is generated by a fall in the supply voltage, the snooze mode and software standby mode are released.
- Initiated by a voltage monitoring reset

When a voltage monitoring reset is generated by a fall in the supply voltage, the snooze mode and software standby mode are released.

- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow releases the MCU from the snooze mode or software standby mode. Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

11.6.4.4 Interrupt for Release from the Snooze Mode

The interrupt for release from the snooze mode is only used to return to the normal operating mode from the snooze mode. Generation of the interrupt with the source selected by the setting of the SNZCR2 register triggers release from the snooze mode (SNZI). For the relation between the setting of the SNZCR2 register and the selected source, see section 11.2.10, Snooze Control Register 2 (SNZCR2). Before placing the MCU in the software standby mode, set the SNZCR2 register to enable the given interrupt to serve as the source for release from the snooze mode. Once the MCU has been returned to the normal operating mode, disable the source for that interrupt. Note that when the interrupt request corresponding to the interrupt source selected by the setting of the SNZCR2 register has been enabled by setting the interrupt controller, the selected interrupt signal will be generated as well as the interrupt signal for release from the snooze mode after the MCU has been returned to the normal operating mode.

Table 11.6 Interrupt Source for Release from the Snooze Mode

Name	Interrupt Source	DTC Startup
SNZI	Source selected by the setting of the SNZCR2 register	Not possible

11.6.4.5 Example of Operations for Data Reception by SCI5 in the Snooze Mode

In the snooze mode, data can be received in the asynchronous mode of SCI5.
If the setting of the SNZCR.SCISNZSEL[1:0] bits is 10b or 11b, the MCU is placed in the snooze mode on detection of a start bit by SCI5 while the MCU is in the software standby mode. After the transition to the snooze mode in response to detection of the start bit, operation of the oscillators and on-chip oscillators that had been operating before the transition to the software standby mode is restarted, after which SCI5 restarts data reception once the oscillation stabilization time has elapsed. Sampling on the basic clock starts after the SCI5 restarted operating, that delays the sampling for the period equivalent to the oscillation stabilization time. Select the high-speed on-chip oscillator (HOCO) as a clock source and stop operating the low-speed on-chip oscillator (LOCO), main clock oscillator, PLL, and PLL2.
By setting the SCI5 reception data full as the condition for generating the interrupt for release from the snooze mode in the SNZCR2 register, the MCU can be placed in the normal operating mode on the completion of the data reception.
If the setting of the SNZCR.SCISNZSEL[1:0] bits is 10b, the MCU is returned to the software standby mode from the snooze mode unless the received data matches the value of the CDR register of SCI5. If the setting of the SNZCR.SCISNZSEL[1:0] bits is 11b, the MCU is returned to the software standby mode from the snooze mode unless the received data matches the value of the CDR register of SCI5 or when transfer of the received data by the DTC is completed if the received data matched the CDR value. Once the received data have matched the value of the CDR register of SCI5, the MCU is again placed in the snooze mode on detection of a start bit and is returned to the software standby mode from the snooze mode every time DTC transfer of the received data is completed.
If the continuous data is to be received by using the DTC in the snooze mode, set the DTC in the normal operation mode as well as the SNZCR.SNZDTCE bit to 1. As for the condition for generating the interrupt for release from the snooze mode in the SNZCR2 register, select the DTC transfer completion in response to the reception data full.
Figure 11.10 shows the flow of the setting of the data reception by the SCI5, and Figure 11.11 to Figure 11.13 show the timing of operations of the SCI5 to receive the data in the snooze mode.

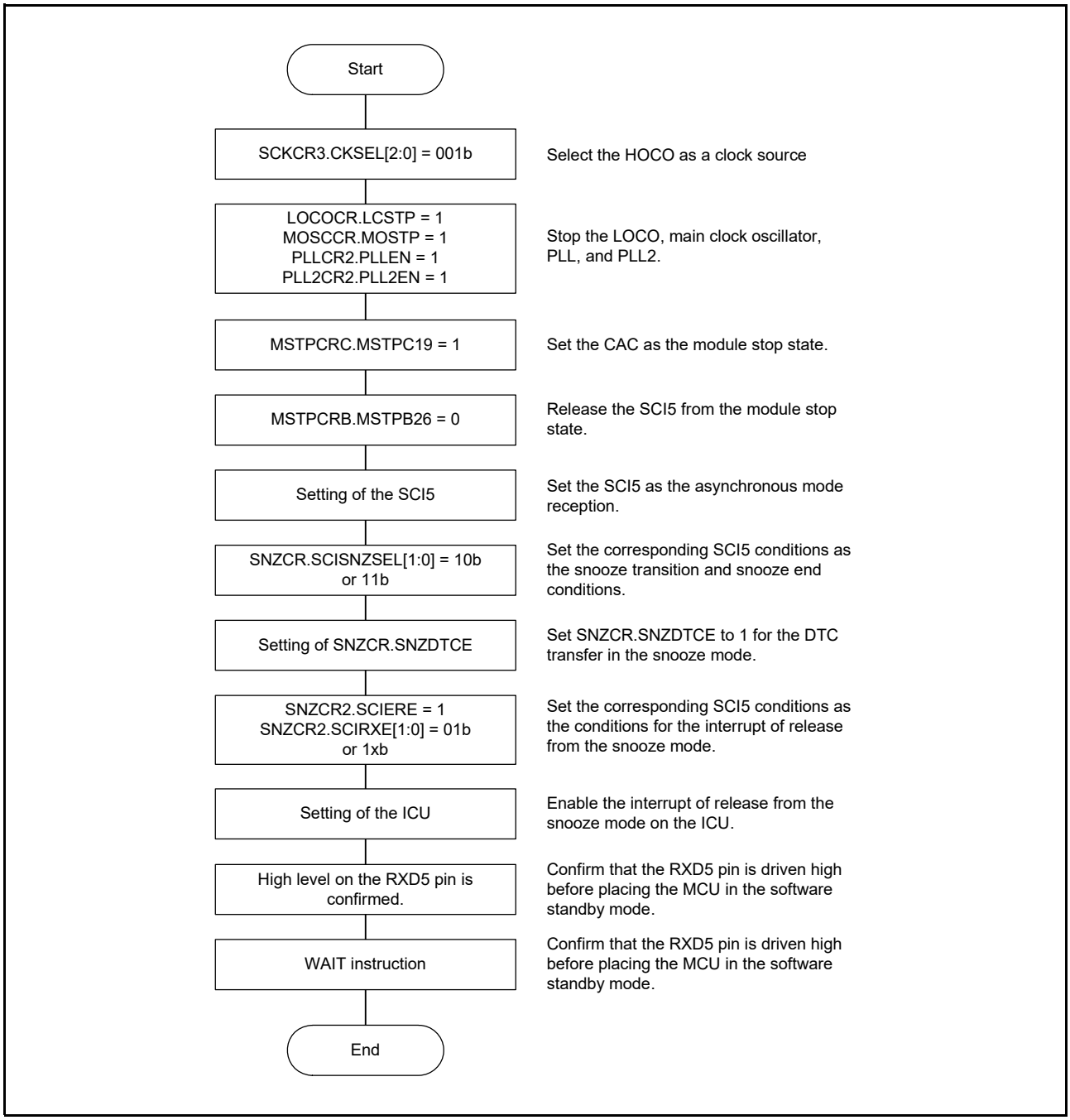


Figure 11.10 Example of Flow of the setting of the data reception by the SCI5

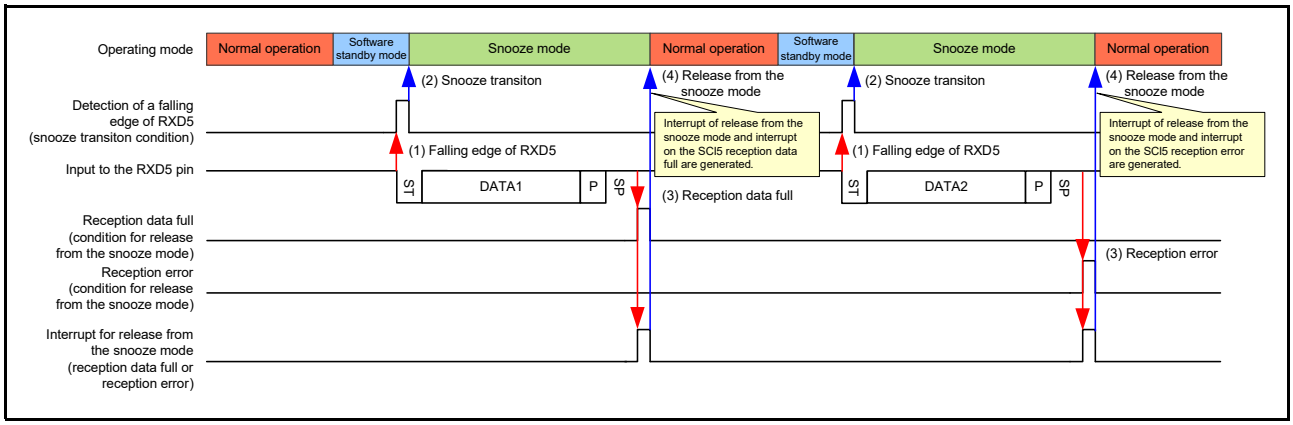


Figure 11.11 Reception of the SCI5 Data in the Snooze Mode (Snooze Mode to be Released by Reception Data Full or Reception Error)

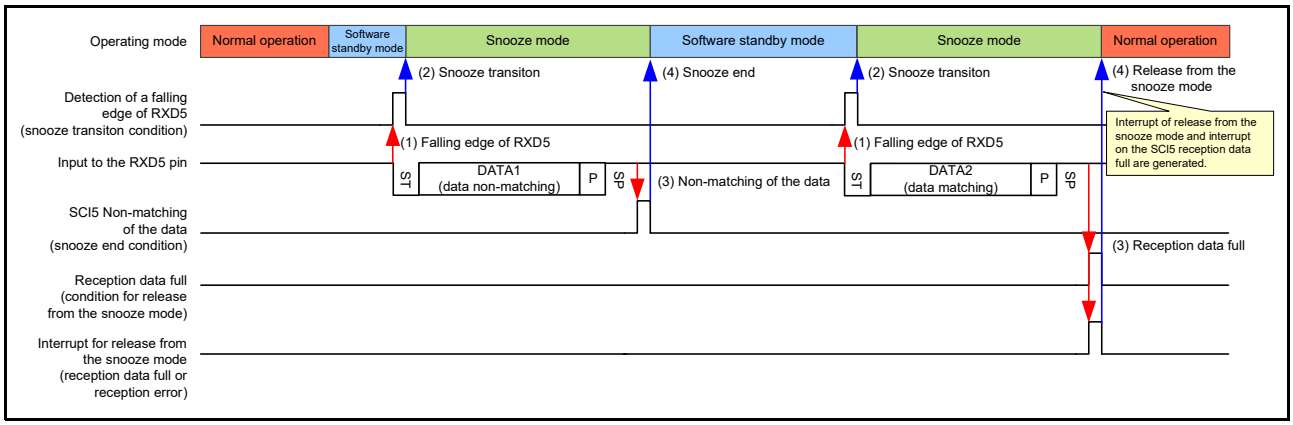


Figure 11.12 Reception of the SCI5 Data in the Snooze Mode (Snooze Mode to be Released by Non-Matching of Data or Reception Data Full)

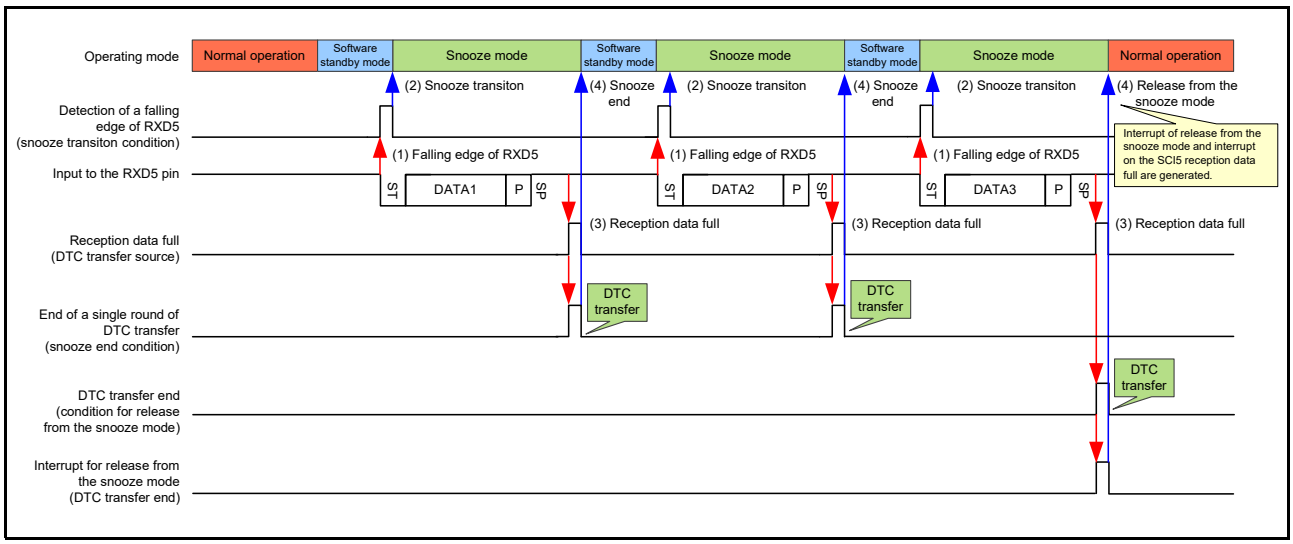


Figure 11.13 Reception of the SCI5 Data in the Snooze Mode (Snooze Mode to be Released by DTC Transfer End or a DTC Transfer End Interrupt)

11.6.4.6 Example of Operations for A/D Conversion in the Snooze Mode

The A/D conversion by S12AD can be driven in the snooze mode.
Periodic A/D conversion can be made to proceed by using the LPT compare match 1 from the ELC operated in the software standby mode as a trigger.
If the setting of the SNZCR.ADCSNZSEL[1:0] bits is 10b or 11b, the MCU is placed in the snooze mode on detection of an LPT compare match 1 while the MCU is in the software standby mode. When the transition to the snooze mode is made, the oscillators and on-chip oscillators that were operating before the transition to the software standby mode start operating again, and operation of the ELC and S12AD is then restarted after the oscillation stabilization time has elapsed.
If a trigger from the ELC is selected as the condition for starting A/D conversion by the S12AD and the LPT compare match 1 has been selected as the event to be linked to the S12AD on the ELC, A/D conversion by the S12AD can be made to proceed after the MCU has been placed in the snooze mode. Use the S12AD in the single scan mode. Do not attempt to convert the temperature sensor output or the internal reference voltage.
The MCU can be placed in the normal operating mode in response to the end of the A/D conversion by selecting the completion of the S12AD conversion as the condition for generating the interrupt for release from the snooze mode in the SNZCR2 register.
If the setting of the SNZCR.ADCSNZSEL[1:0] bits is 10b, the MCU remains in the snooze mode after it has entered the snooze mode until an interrupt signal causes release from the snooze mode. If the setting of the SNZCR.ADCSNZSEL[1:0] bits is 11b, the MCU is returned to the software standby mode from the snooze mode once a single round of the DTC transfer triggered by the end of A/D conversion has been completed while the MCU was in the snooze mode. After return to the software standby mode, if a further LPT compare match 1 occurs, the MCU is again placed in the snooze mode and starts A/D conversion.
To transfer the A/D conversion results to the RAM using the DTC in the snooze mode, set the DTC in the normal operating mode and set the SNZCR.SNZDTCE bit to 1. As for the condition for generating the interrupt for release from the snooze mode in the SNZCR2 register, select the event of the DTC transfer completion in response to the completion of the conversion by S12AD.
Figure 11.14 shows an example of flow of the A/D conversion setting by the S12AD in the snooze mode, and Figure 11.15 and Figure 11.16 show the timing of operations of the S12AD conversion in the snooze mode.

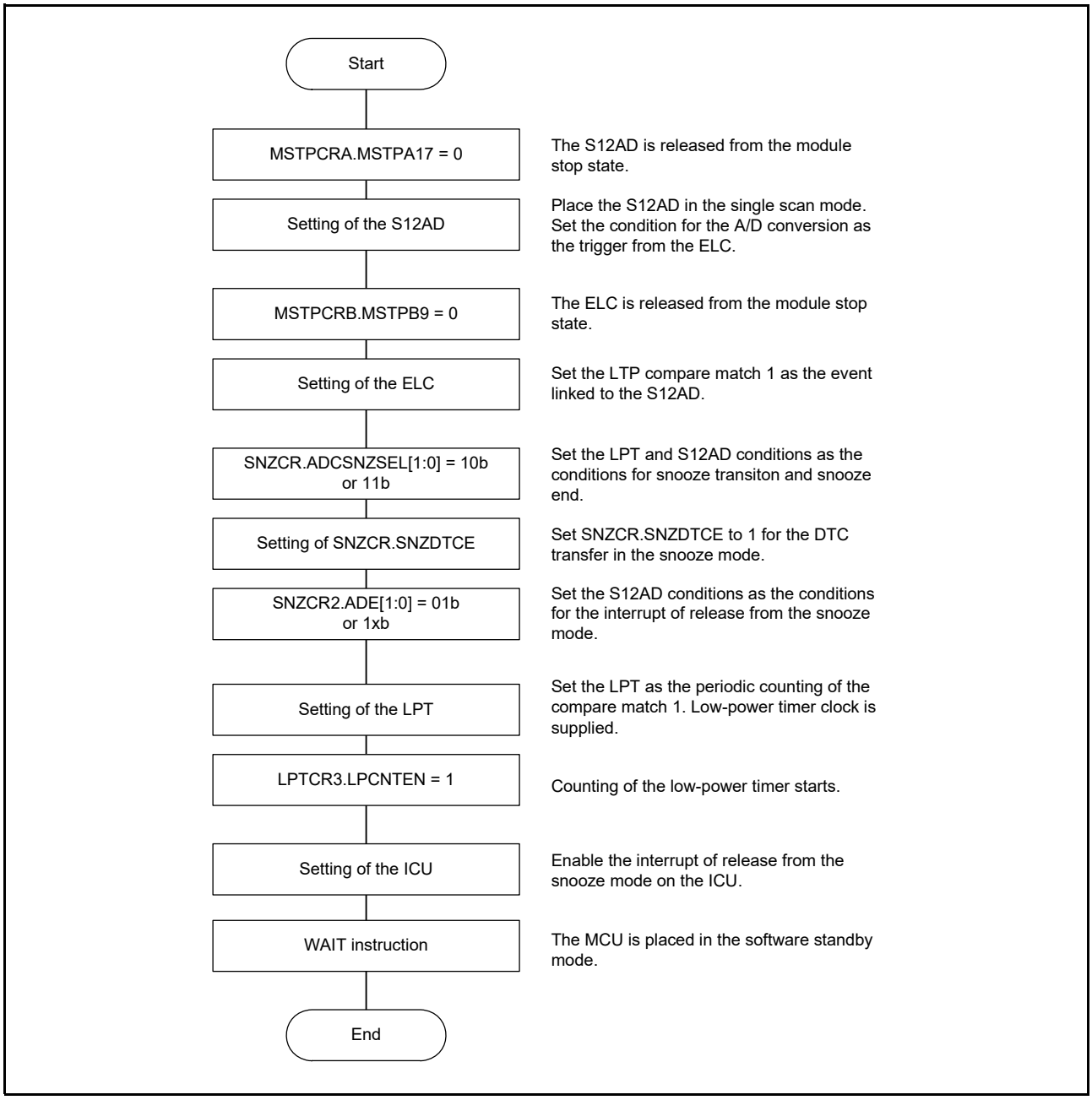


Figure 11.14 Example of Flow of the A/D Conversion Setting by the S12AD in the Snooze Mode

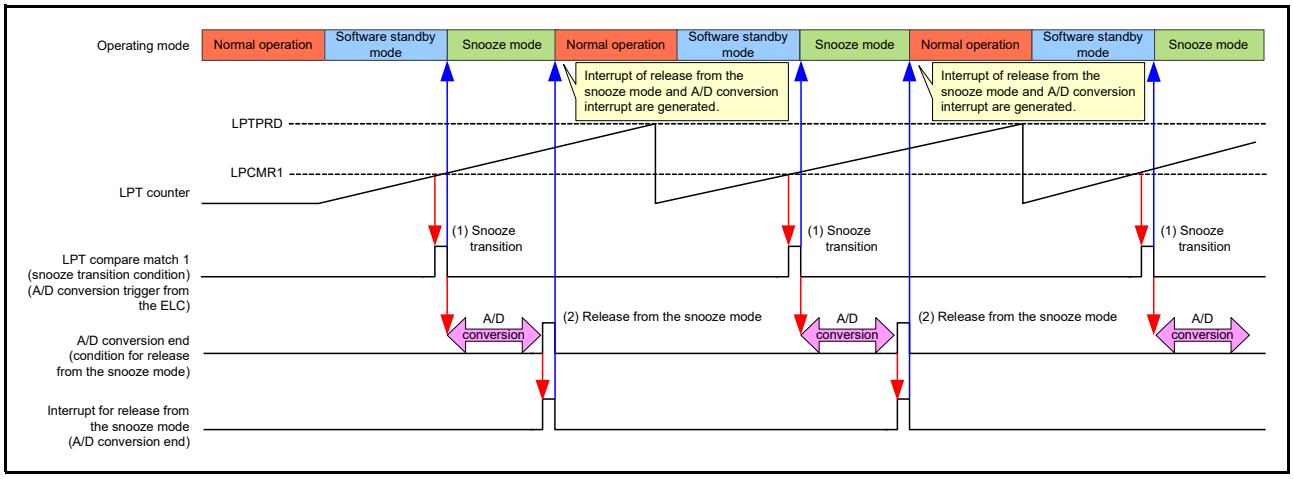


Figure 11.15 Timing of Operations of the S12AD Conversion in the Snooze Mode (No DTC Transfers)

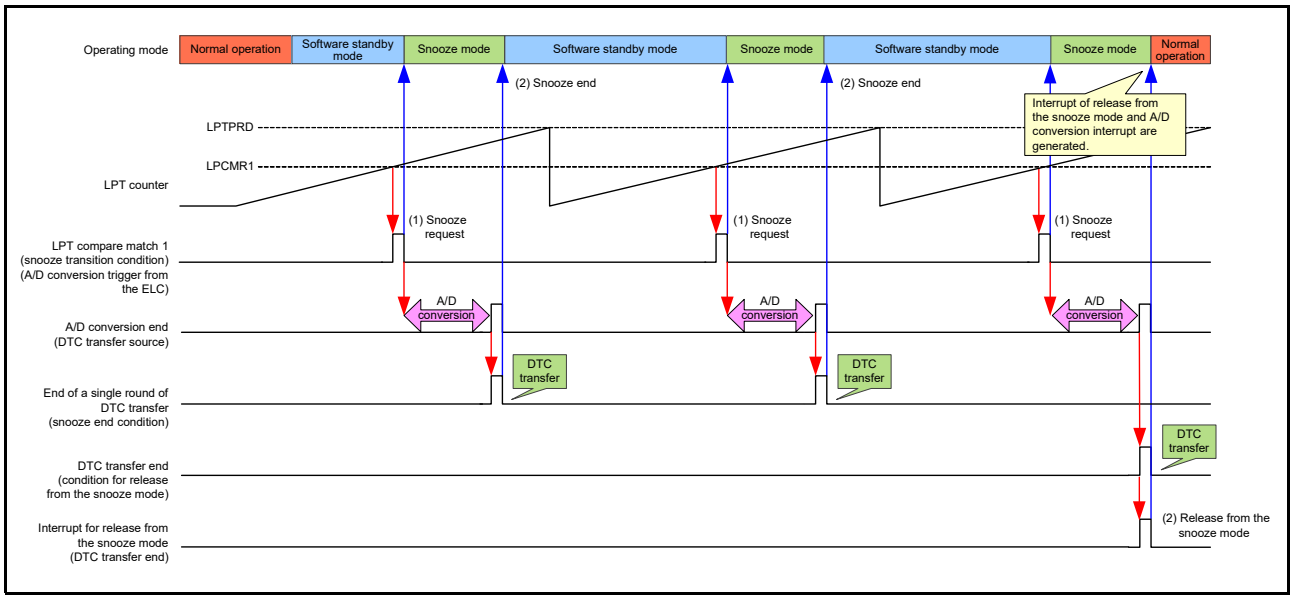


Figure 11.16 Timing of Operations of the S12AD Conversion in the Snooze Mode (with the DTC Transfer)

11.6.4.7 Example of Operations for CTSU Measurement in the Snooze Mode

The CTSU can measure the electrostatic capacitance of the touch sensor in the snooze mode. The CTSU periodically starts measuring the electrostatic capacitance by using the LPT compare match 1 from the ELC operated in the software standby mode as a trigger.

If the setting of the SNZCR.CTSUSNZSEL[1:0] bits is 10b, the MCU is placed in the snooze mode on detection of an LPT compare match 1 while the MCU is in the software standby mode. When the transition to the snooze mode has been made, the oscillators and on-chip oscillators that were in operation before the transition to the software standby mode restart operating, and then the ELC and CTSU restart operating after the oscillation stabilization time has elapsed. If the starting of measuring the electrostatic capacitance of the touch sensor by the CTSU is selected as a trigger from the external input (an event input from the ELC), and the LPT compare match 1 is selected as the event to be linked to the CTSU on the ELC, the CTSU starts measuring the electrostatic capacitance after the MCU is placed in the snooze mode. When measurement by the CTSU has been completed, the CTSU generates a snooze end request and the MCU is returned to the software standby mode from the snooze mode.

The MCU can be placed in the normal operating mode when the CTSU measurement is completed by selecting the completion of the CTSU measurement as the condition for generating the interrupt for release from the snooze mode in the SNZCR2 register.

To write to the register per channel using the DTC or to transfer the measured data per channel in the snooze mode, set the DTC in the normal operating mode and set the SNZCR.SNZDTCE bit to 1.

Figure 11.17 shows an example of flow of the measuring by the CTSU in the snooze mode, and Figure 11.18 shows the timing of measuring by the CTSU in the snooze mode.

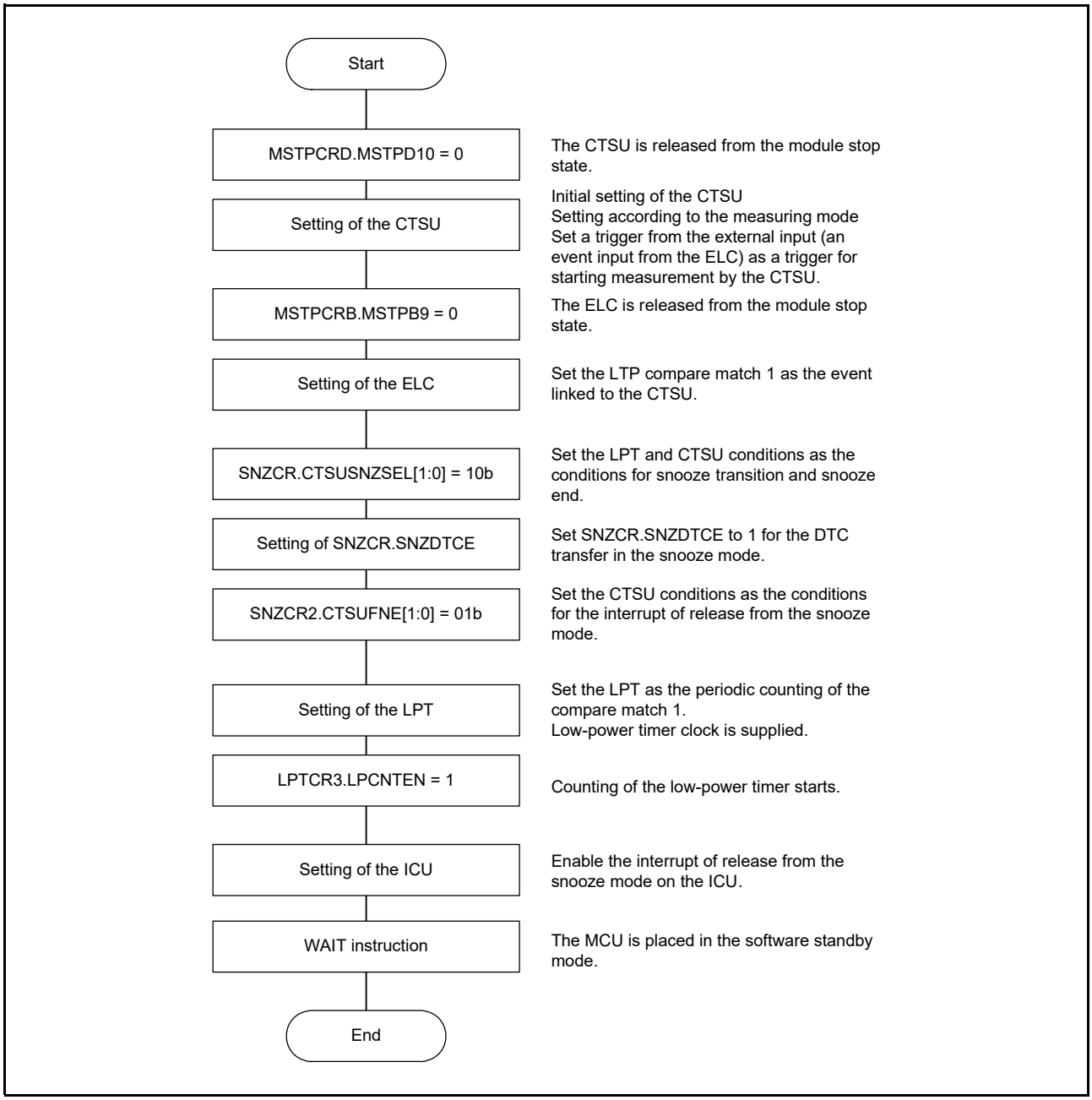


Figure 11.17 Example of Flow of the Measuring by the CTSU in the Snooze Mode

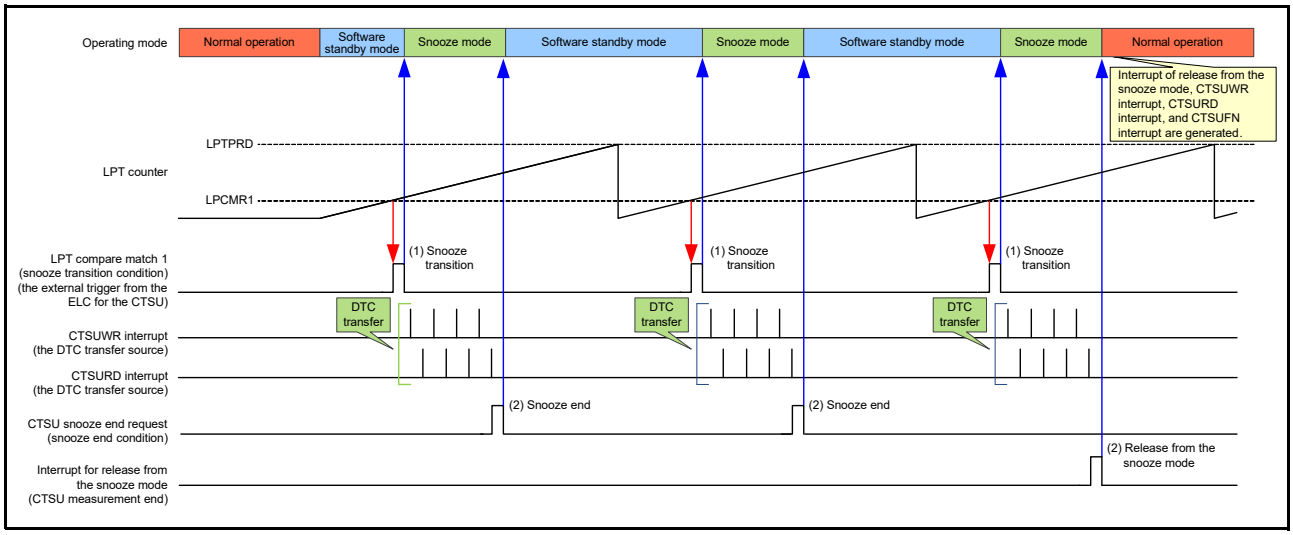


Figure 11.18 Timing of Measuring by the CTSU in the Snooze Mode

11.6.4.8 Example of Operations for Using the REMC in the Snooze Mode

The reception of remote control signals is possible in the snooze mode.
If the setting of the SNZCR.REMCSNZSEL[1:0] bits is 10b, the MCU is placed in the snooze mode on start of a request for a clock signal by the REMC while the MCU is in the software standby mode. After the transition to the snooze mode, operation of the oscillators and on-chip oscillators that had been operating before the transition to the software standby mode is restarted, after which the REMC restarts the reception of remote control signals once the oscillation stabilization time has elapsed. After the reception of remote control signals proceeds and the request for a clock signal by the REMC is completed, the MCU returns to the software standby mode from the snooze mode.
After exit from the snooze mode, if a further start of a request for a clock signal by the REMC is generated, the MCU is again placed in the snooze mode. For details on the reception of remote control signals in the snooze mode, see section 37.3.13.2, Data Reception in Software Standby Mode.

11.6.4.9 Example of Operations for Comparison with the Results of A/D Conversion in the Snooze Mode

The A/D conversion by the S12AD can be driven in the snooze mode. Periodic A/D conversion and comparison with the results of A/D conversion can be made to proceed by using the LPT compare match 1 from the ELC during operation in the software standby mode as a trigger.
If the setting of the SNZCR.ADCSNZ2SEL[1:0] or SNZCR.ADCSNZ3SEL[1:0] bits is 10b, the MCU is placed in the snooze mode on detection of an LPT compare match 1 while the MCU is in the software standby mode. When the transition to the snooze mode is made, the oscillators and on-chip oscillators that were operating before the transition to the software standby mode start operating again, and operation of the ELC and S12AD is then restarted after the oscillation stabilization time has elapsed. If a trigger from the ELC is selected as the condition for starting A/D conversion by the S12AD and the LPT compare match 1 has been selected as the event to be linked to the S12AD on the ELC, A/D conversion by the S12AD can be made to proceed after the MCU has been placed in the snooze mode. Use the S12AD in the single scan mode. Do not attempt to convert the temperature sensor output or the internal reference voltage.
The MCU can be placed in the normal operating mode by using the SNZCR2 register to select the comparison conditions satisfaction or non-satisfaction event with the results of conversion by the S12AD as the condition for the interrupt for release from the snooze mode.
If the setting of the SNZCR.ADCSNZ2SEL[1:0] bits is 10b, the MCU is returned to the software standby mode on generation of a comparison conditions non-satisfaction event with the result of A/D conversion in the snooze mode. If the setting of the SNZCR.ADCSNZ3SEL[1:0] bits is 10b, the MCU is returned to the software standby mode on generation of a comparison conditions satisfaction event with the result of A/D conversion in the snooze mode. If a further LPT compare match 1 is generated after return to the software standby mode, the MCU is again placed in the snooze mode to enable A/D conversion.

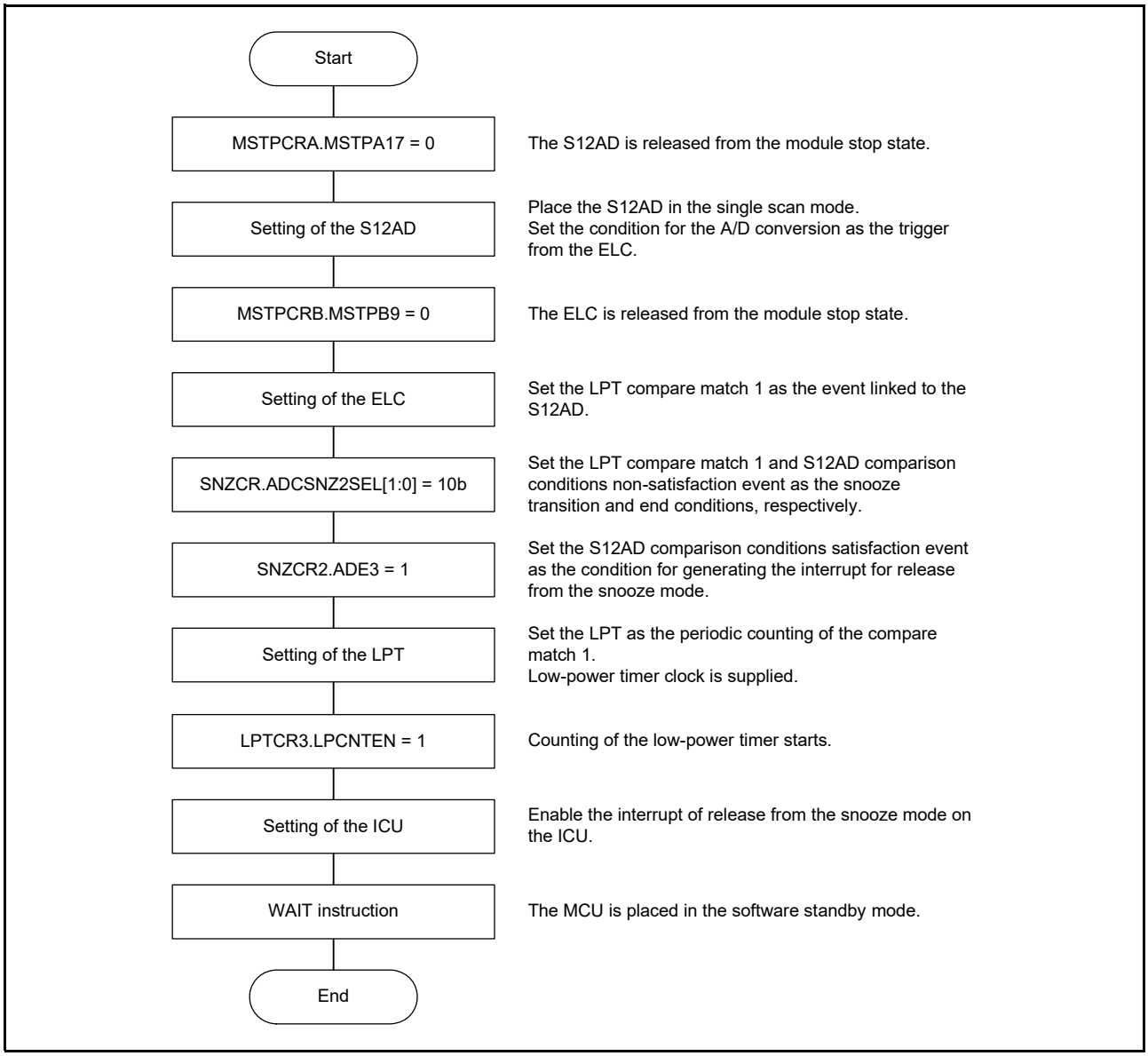


Figure 11.19 Example of the Flow of Settings for a Comparison Operation by the S12AD in the Snooze Mode

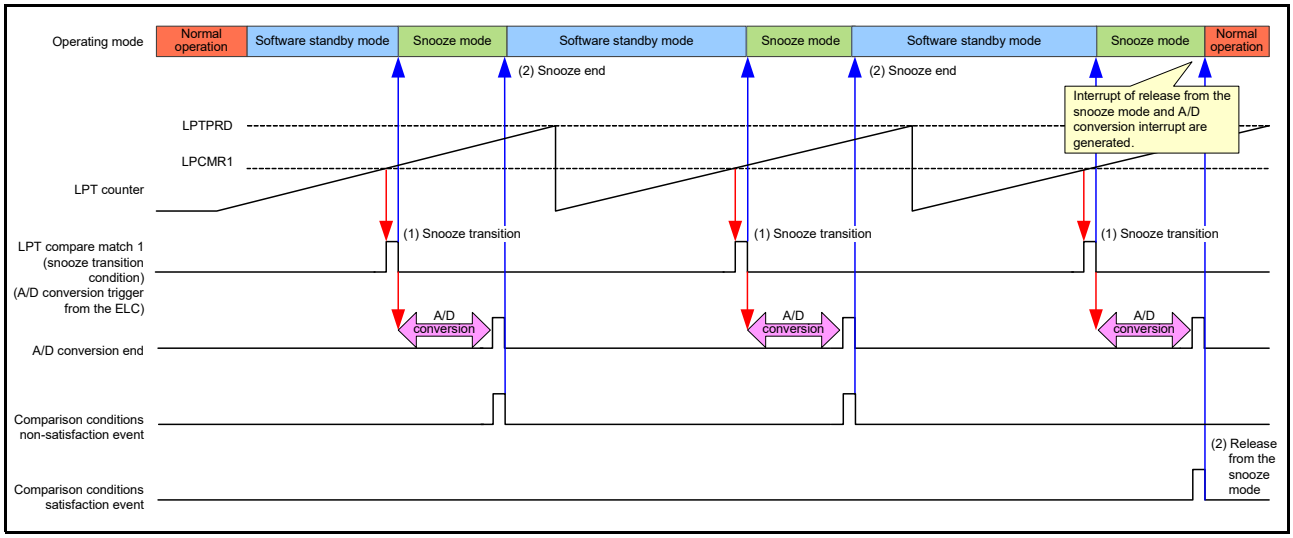


Figure 11.20 Timing of a Comparison Operation by the S12AD in the Snooze Mode

11.7 Usage Notes

11.7.1 I/O Port States

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

11.7.2 Module Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 to avoid activating the DMAC and DTC.
For details, refer to section 17, DMA Controller (DMACA) and section 18, Data Transfer Controller (DTCb).

11.7.3 Changing the Operating Frequency of the RSIP

To change the RSIP operating frequency to a higher one, set the MSTPD31 bit to 1 (stopping supply of the module clock). After having changed the operating frequency, wait for 30 µs before setting the MSTPD31 bit to 0 (resuming supply of the module clock).

11.7.4 On-Chip Peripheral Module Interrupts

Interrupts do not operate in the module stop state. Therefore, if the module stop state is made after an interrupt request is generated, a CPU interrupt source or a DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module stop state.

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

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

11.7.6 Timing of WAIT Instructions

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction being executed before the register setting is modified may cause unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last register setting is done.

11.7.7 Rewrite the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMAC and DTC to rewrite any registers related to the WDT while the chip is in sleep mode.
Depending on the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. To avoid this, do not set up the DMAC and DTC to rewrite any registers related to the IWDT in sleep mode.
The RSTCKCR register is a register that switches the clock source at exit from sleep mode. Changing the RSTCKCR register in sleep mode causes unintended operation, so do not write to this register in sleep mode.

11.7.8 DTC Transfer in the Snooze Mode

When using the DTC in the snooze mode, do not set the MRA.WBDIS bit to 1.

11.7.9 Data Reception by the SCI5 in the Snooze Mode

To receive the data by the SCI5 in the snooze mode, the following conditions must be satisfied.

- Use the HOCO as a clock source,
- Stop the LOCO, main oscillator, PLL, and PLL2 before the MCU is placed in the software standby mode.
- Set the MSTPCRC.MSTPC19 bit to 1 and set the CAC as the module stop state before the MCU is placed in the software standby mode.
- The RXD5 pin must retain the high level before the MCU is placed in the software standby mode.
- The MCU should not be placed in the software standby mode when the communication using the SCI5 is in progress.

11.7.10 LPT Operations in the Snooze Mode

When using the LPT in the snooze mode, set the MSTPCRB.MSTPB9 bit to 0, and release the ELC from the module stop state.

11.7.11 A/D Conversion in the Snooze Mode

To perform the A/D conversion in the snooze mode, set the MSTPCRB.MSTPB9 bit to 0, and release the ELC from the module stop state. Select the trigger from the ELC as the trigger for starting the A/D conversion and operate in the single scan mode. Do not convert the temperature sensor output and the internal reference voltage.

11.7.12 Measuring by the CTSU in the Snooze Mode

To measure the electrostatic capacitance of the touch sensor in the snooze mode, set the MSTPCRB.MSTPB9 bit to 0, and release the ELC from the module stop state. Select the external trigger from the ELC as the trigger for starting the measuring by the CTSU.

11.7.13 REMC Operations in the Snooze Mode

To receive remote control signals in the snooze mode, the following conditions must be satisfied.

- The HOCO is in use as the source of the system clock.
- Stopping of the LOCO, main oscillator, PLL, and PLL2 before the MCU is placed in the software standby mode.
- The MSTPCRC.MSTPC19 bit is set to 1 to disable the module clock for the CAC before the MCU is placed in the software standby mode.
- The MCU should not be placed in the software standby mode while the remote control signal receiver (REMC) is operating.

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, SCKCR3, PLLCR, PLLCR2, PLL2CR, PLL2CR2, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, LOFCR, OSTDCR, OSTDSR, CKOCR, LOCOTRR2, ILOCOTRR, HOCOTRR0, SOMCR, CANFDCKCR, CANFDCKDIVCR, USBCKCR
PRC1	• Register related to the operating modes: SYSCR1 • Registers related to low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, OPCCR, RSTCKCR, SOPCCR, RPSCR, SNZCR, SNZCR2 • Registers related to the clock generation circuit: MOFCR, MOSCWTCR • Software reset register: SWRR
PRC2	• Registers related to the low power timer: LPTCR1, LPTCR2, LPTCR3, LPTPRD, LPCMR0, LPCMR1, LPWUCR
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	PRC2	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 register related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption functions, the clock generation circuit, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	PRC2	Protect Bit 2	Enables writing to the registers related to the low power timer. 0: Write disabled 1: Write enabled	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 8 higher-order bits and the desired value to the 8 lower-order bits as a 16-bit unit.	R/W*1

Note 1. Write data is not retained.

PRCi Bits (Protect Bit i)

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

Setting the PRCi bits to 1 and 0 enable and disable 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.

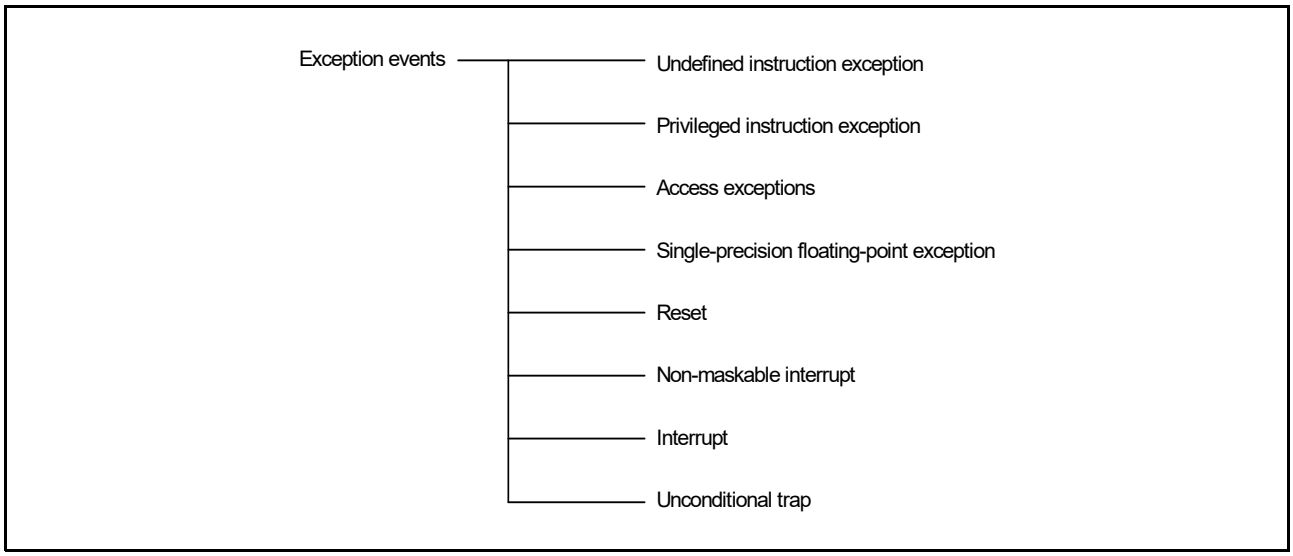


Figure 13.1 Types of Exception Events

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.

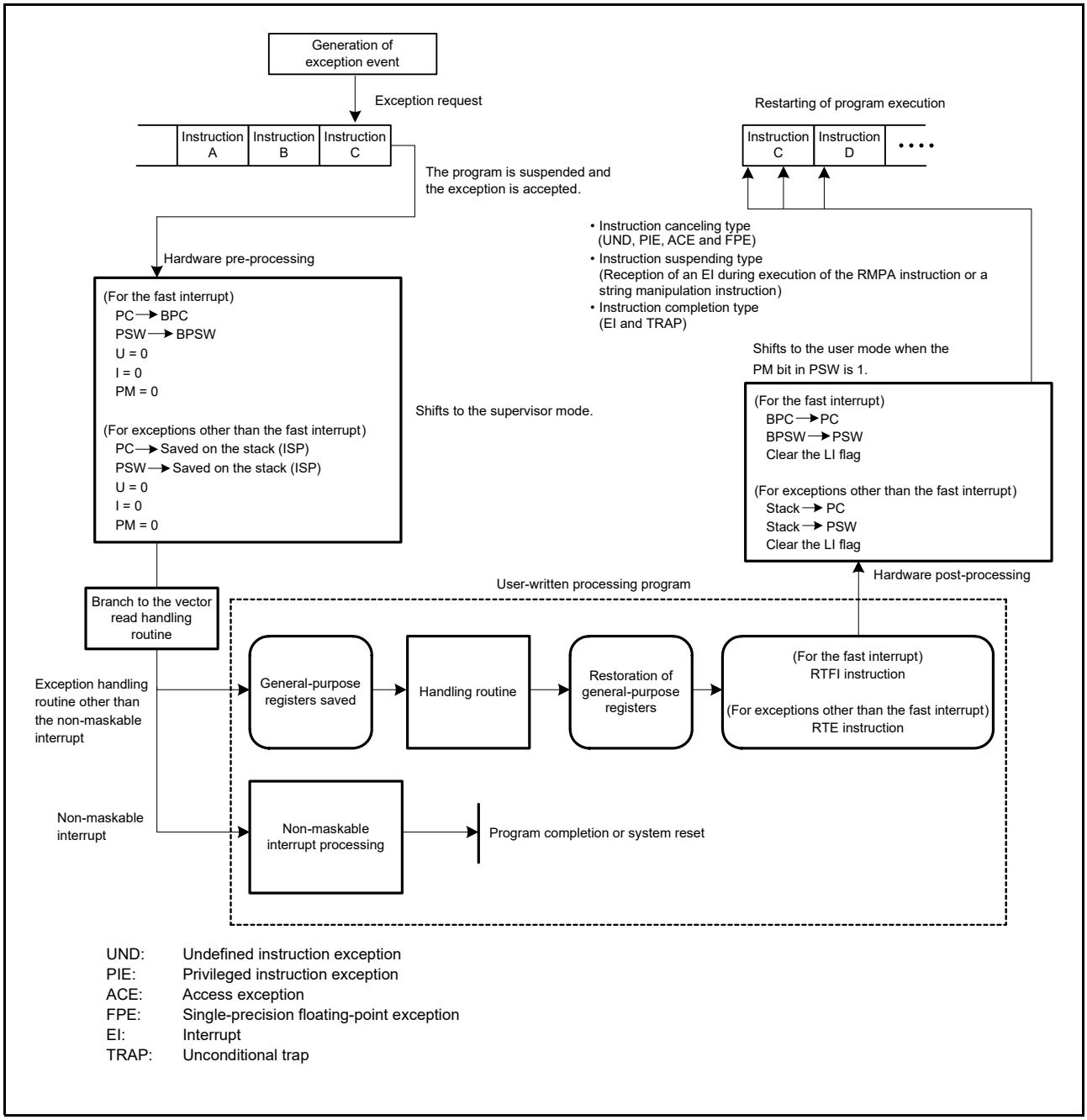


Figure 13.2 Outline of Exception Handling Procedure

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 can be used to save and restore the general-purpose and other registers at the start and end of an exception handling routine. Use the PUSH and POP instructions for saving to and restoring from the stack.

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 (ICUb)

14.1 Overview

The interrupt controller receives interrupt requests from peripheral modules and external pins, and generates an interrupt request to the CPU and a transfer request to the DTC and DMAC.

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

Table 14.1 Specifications of Interrupt Controller

Item		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge detection/level detection Edge detection or level detection is fixed for each source of connected peripheral modules.
	External pin interrupts	• Interrupts from pins IRQ0 to IRQ7 • Number of sources: 8 • Interrupt detection: Low level/falling edge/rising edge/rising and falling edges One of these detection methods can be set for each source. • Digital filter function: Supported
	Software interrupt	• Interrupt generated by writing to a register • One interrupt source
	Event link interrupt	The ELSR8I, ELSR18I, or ELSR19I interrupt is generated by an ELC event
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC/DMAC control	Interrupt sources can be used to start the DTC and DMAC.*1
Non-maskable interrupts	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter function: Supported
	Oscillation stop detection interrupt	Interrupt on detection of oscillation having stopped
	WDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	IWDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	Voltage monitoring 1 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 1 (LVD1)
	Voltage monitoring 2 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 2 (LVD2)
	RAM error interrupt	This interrupt occurs when a parity check error is detected in the RAM.
Return from low power consumption states	Sleep mode Deep sleep mode	Return is initiated by any non-maskable interrupt or any interrupt.
	Software standby mode	Return is initiated by the non-maskable interrupt, external pin interrupts (IRQ0 to IRQ7), peripheral interrupts (voltage monitoring 1, voltage monitoring 2, RTC alarm/periodic, REMC, USB0 resume), or ELSR8I interrupt (LPT dedicated interrupt).
	Snooze mode	Return is initiated by the non-maskable interrupt, external pin interrupts (IRQ0 to IRQ7), peripheral interrupts (voltage monitoring 1, voltage monitoring 2, RTC alarm/periodic, REMC, USB0 resume), or SNZI interrupt (snooze release interrupt).

Note 1. For the DTC and DMAC triggers, refer to Table 14.3, Interrupt Vector Table.

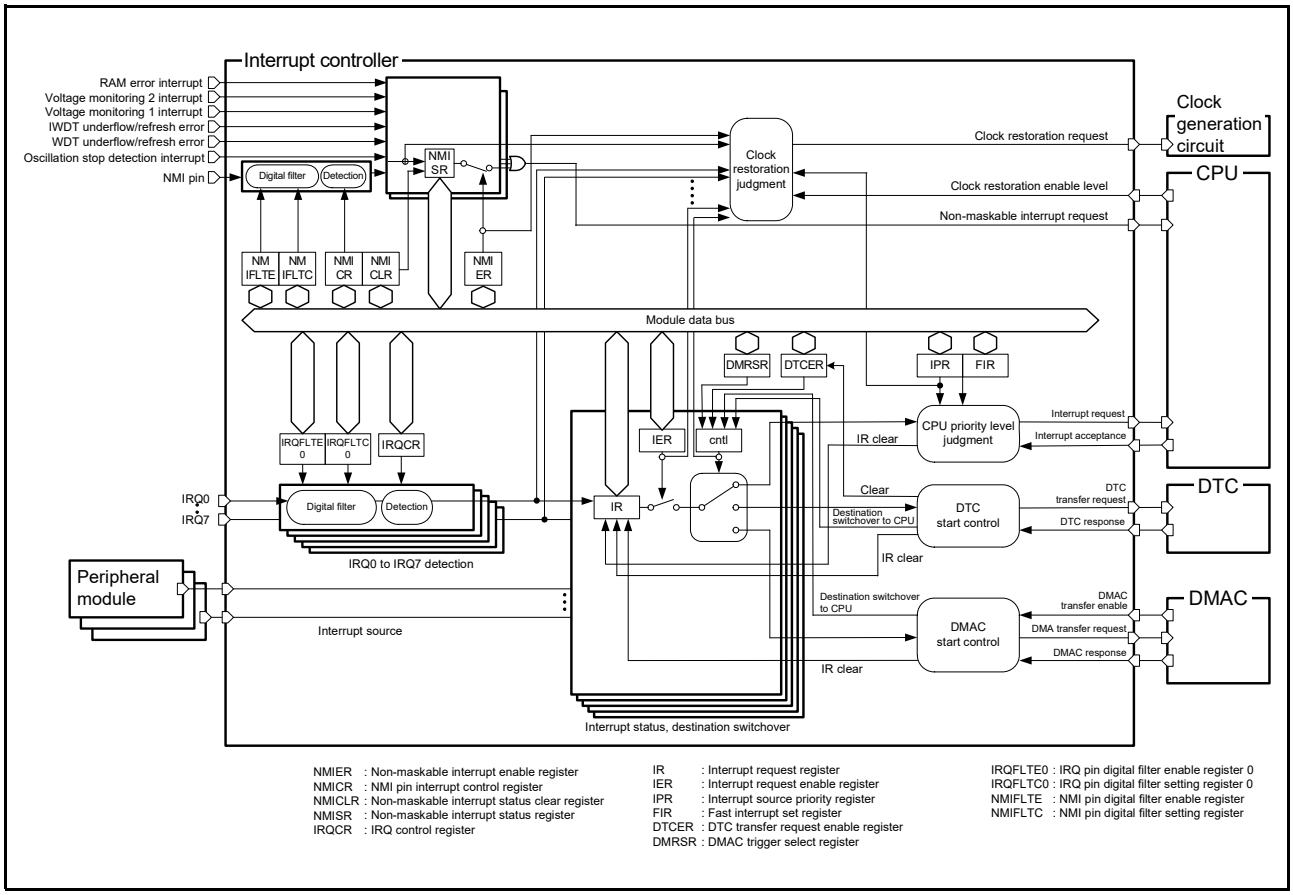


Figure 14.1 Block Diagram of Interrupt Controller

Table 14.2 lists the input/output pins of the interrupt controller.

Table 14.2 Pin Configuration of Interrupt Controller

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ7	Input	External interrupt request pins

14.2 Register Descriptions

14.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

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

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

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

This flag is the status flag of an individual interrupt request. This flag is set to 1 when the corresponding interrupt request is generated. To detect an interrupt request, the interrupt request output should be enabled by the corresponding peripheral module interrupt enable bit.
There are two interrupt request detection methods: edge detection and level detection. For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source. For interrupts from IRQi (i = 0 to 7) pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits. For detection of the various interrupt sources, see Table 14.3, Interrupt Vector Table.

(1) Edge detection

[Setting condition]

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

[Clearing conditions]

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

(2) Level detection

[Setting condition]

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

[Clearing condition]

- The flag is cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

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

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

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

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request.
When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request.
The setting of an IENj bit does not affect the IRn.IR flag (n = interrupt vector number). Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 14.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).
The IERm.IENj bit is set for each request source (vector number).
For the correspondence between interrupt sources and IERm.IENj bits, see Table 14.3, Interrupt Vector Table.
For the procedure for setting IERm.IENj bits during the selection of destinations for interrupt requests, refer to section 14.4.3, Selecting Interrupt Request Destinations.

14.2.3 Interrupt Source Priority Register n (IPRn) (n = interrupt vector number)

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

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

For the correspondence between interrupt sources and IPRn registers, see Table 14.3, Interrupt Vector Table.

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

These bits specify the priority level of the corresponding interrupt source.

Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect transfer requests to the DTC or DMAC.

The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.

If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.

These bits should be written to while an interrupt request is disabled (IERm.IENj bit = 0 (m = 02h to 1Fh; j = 0 to 7)).

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	Specify the vector number of an interrupt source to be a fast interrupt.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC or DMAC.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0 (m = 02h to 1Fh; j = 0 to 7)).

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

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

FIEN Bit (Fast Interrupt Enable)

This bit enables the fast interrupt.
Setting this bit to 1 makes the interrupt request of the vector number specified by the FVCT[7:0] bits a fast interrupt. When an interrupt request of the vector number specified by the FVCT[7:0] bits is generated and the interrupt request destination is the CPU while the FIEN bit is 1, the interrupt request is output to the CPU as a fast interrupt regardless of the setting of the IPRn register (n = interrupt vector number). When using the fast interrupt for returning from the software standby mode, see section 14.6.2, Return from Software Standby Mode.
If the setting of the IERm.IENj bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.
For settable vector numbers, see Table 14.3, Interrupt Vector Table.
Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, see section 13, Exception Handling, and section 14.4.6, Fast Interrupt.

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

Note 1. Only 1 can be written.

SWINT Bit (Software Interrupt Generation)

When 1 is written to the SWINT bit, the interrupt request register 027 (IR027) is set to 1.

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

If 1 is written to the SWINT bit when the DTC transfer request enable register 027 (DTCER027) is set to 1, a DTC transfer request is issued.

14.2.6 DTC Transfer Request Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): ICU.DTCER027 0008 711Bh 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 not selected as the DTC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

An interrupt source that has been selected as a DMAC trigger should not be specified as a DTC trigger. See Table 14.3, Interrupt Vector Table, for the interrupt sources that are selectable 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.

[Setting condition]

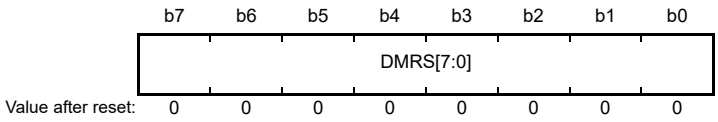
- When 1 is written to the DTCE bit

[Clearing conditions]

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

14.2.7 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



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

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

DMRS[7:0] Bits (DMAC Trigger Select)

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

For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

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

14.2.8 IRQ Control Register i (IRQCRi) (i = 0 to 7)

Address(es): ICU.IRQCR0 0008 7500h to ICU.IRQCR7 0008 7507h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Sense Select	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IENj bit in IERm (m = 02h to 1Fh; j = 0 to 7) is 0). After changing the setting, clear the IR flag in IRn before setting the interrupt enable bit. However, when the change is to the low level, the IR flag does not require clearing.

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

These bits select the interrupt detection sensing method of IRQi pin.
For the external pin interrupt detection setting, see section 14.4.8, External Pin Interrupts.

14.2.9 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): ICU.IRQFLTE0 0008 7510h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN0	IRQ0 Digital Filter Enable	0: Digital filter is disabled 1: Digital filter is enabled	R/W
b1	FLTEN1	IRQ1 Digital Filter Enable		R/W
b2	FLTEN2	IRQ2 Digital Filter Enable		R/W
b3	FLTEN3	IRQ3 Digital Filter Enable		R/W
b4	FLTEN4	IRQ4 Digital Filter Enable		R/W
b5	FLTEN5	IRQ5 Digital Filter Enable		R/W
b6	FLTEN6	IRQ6 Digital Filter Enable		R/W
b7	FLTEN7	IRQ7 Digital Filter Enable		R/W

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

This bit enables the digital filter used for the IRQi pin.

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

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

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.10 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): ICU.IRQFLTC0 0008 7514h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL7[1:0]		FCLKSEL6[1:0]		FCLKSEL5[1:0]		FCLKSEL4[1:0]		FCLKSEL3[1:0]		FCLKSEL2[1:0]		FCLKSEL1[1:0]		FCLKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL2[1:0]	IRQ2 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL3[1:0]	IRQ3 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL4[1:0]	IRQ4 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL5[1:0]	IRQ5 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL6[1:0]	IRQ6 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL7[1:0]	IRQ7 Digital Filter Sampling Clock		R/W

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

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

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

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.11 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. Writing to this bit has no effect.	R

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

NMIST Flag (NMI Status Flag)

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

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

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

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

- When the oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

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

- When the WDT underflow/refresh error interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

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

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

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

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

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

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

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

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

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit

RAMST Flag (RAM Error Interrupt Status Flag)

The RAMST flag indicates whether a RAM error interrupt request is generated from the RAM.
The RAMST flag is read-only. To set the RAMST flag to 0, clear all of the error status flags of the RAM. Refer to section 45.3.2, RAM Error Interrupt Function for details.
[Setting condition]

- A parity check error interrupt occurs (When the RAM.RAMSTS.RAMERR flag becomes 1)

[Clearing condition]

- When all sources which set the RAMST flag to 1 are cleared.

14.2.12 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	read as 0. The write value should be 0.	R/W

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

NMIEN Bit (NMI Pin Interrupt Enable)

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

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

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

WDTEN Bit (WDT Underflow/Refresh Error Enable)

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

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

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

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

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

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

This bit enables the voltage monitoring 2 interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

RAMEN Bit (RAM Error Interrupt Enable)

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

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

Note 1. Only 1 can be written to this bit.

NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

WDTCLR Bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

LVD1CLR Bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

LVD2CLR Bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

14.2.14 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	These bits are read as 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

NMIMD Bit (NMI Detection Set)

This bit specifies the detection edge of the NMI pin interrupt.

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

NFLTEN Bit (NMI Digital Filter Enable)

This bit enables the digital filter used for the NMI pin interrupt.

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

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

For details of the digital filter, see section 14.4.7, Digital Filter.

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

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

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

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

For details of the digital filter, see section 14.4.7, Digital Filter.

14.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

14.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes × 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.

Executing an INT instruction or BRK instruction leads to the generation of an unconditional trap. The same range of memory as shown in Table 14.3, Interrupt Vector Table, is used for the vectors for unconditional traps. The vector for BRK instructions is vector 0 while the vector numbers for INT instructions are specifiable as numbers in the range from 0 to 255.

Table 14.3 lists details of the interrupt vectors. Details of the headings in Table 14.3 are listed below.

Item	Description
Source of interrupt request generation	Name of the source for generation of the interrupt request
Name	Name of the interrupt
Vector no.	Vector number for the interrupt
Vector address offset	Value of the offset from the base address for the vector table
Interrupt detection method	“Edge” or “level” as the method for detection of the interrupt
CPU interrupt	“✓” in this column indicates usability as a CPU interrupt.
Start DTC	“✓” in this column indicates usability as a request for DTC transfer.
Start DMAC	“✓” in this column indicates usability as a request for DMA transfer.
SSBY return	“✓” in this column indicates usability as a request for return from software-standby mode.
IER	Name of the IER register and bit corresponding to the 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 14.3 Interrupt Vector Table (1/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	N/A	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	N/A	N/A	N/A	N/A	—	—	—
RAM	RAMERR	18	0048h	Level	✓	N/A	N/A	N/A	IER02.IEN2	IPR000	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	21	0054h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	26	0068h	—	N/A	N/A	N/A	N/A	—	—	—
ICU	SWINT	27	006Ch	Edge	✓	✓	N/A	N/A	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	✓	✓	✓	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	✓	N/A	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	✓	✓	✓	N/A	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	✓	✓	✓	N/A	IER03.IEN7	IPR007	DTCER031
CAC	FERRF	32	0080h	Level	✓	N/A	N/A	N/A	IER04.IEN0	IPR032	—
	MENDF	33	0084h	Level	✓	N/A	N/A	N/A	IER04.IEN1	IPR033	—
	OVFF	34	0088h	Level	✓	N/A	N/A	N/A	IER04.IEN2	IPR034	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	N/A	—	—	—
USB0	D0FIFO0	36	0090h	Edge	✓	✓	✓	N/A	IER04.IEN4	IPR036	DTCER036
	D1FIFO0	37	0094h	Edge	✓	✓	✓	N/A	IER04.IEN5	IPR037	DTCER037
	USBI0	38	0098h	Edge	✓	N/A	N/A	N/A	IER04.IEN6	IPR038	—
—	Reserved	39	009Ch	—	N/A	N/A	N/A	N/A	—	—	—
POEG	POEGGAI	40	00A0h	Level	✓	N/A	N/A	N/A	IER05.IEN0	IPR040	—
	POEGGBI	41	00A4h	Level	✓	N/A	N/A	N/A	IER05.IEN1	IPR041	—
	POEGGCI	42	00A8h	Level	✓	N/A	N/A	N/A	IER05.IEN2	IPR042	—
	POEGGDI	43	00ACh	Level	✓	N/A	N/A	N/A	IER05.IEN3	IPR043	—

Table 14.3 Interrupt Vector Table (2/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
RSPI0	SPEI0	44	00B0h	Level	✓	N/A	N/A	N/A	IER05.IEN4	IPR044	—
	SPRI0	45	00B4h	Edge	✓	✓	✓	N/A	IER05.IEN5		DTCER045
	SPTI0	46	00B8h	Edge	✓	✓	✓	N/A	IER05.IEN6		DTCER046
	SPII0	47	00BCh	Level	✓	N/A	N/A	N/A	IER05.IEN7		—
—	Reserved	48	00C0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	49	00C4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	50	00C8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	51	00CCh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	52	00D0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	53	00D4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	54	00D8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	55	00DCh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	56	00E0h	—	N/A	N/A	N/A	N/A	—	—	—
DOC	DOPCF	57	00E4h	Level	✓	N/A	N/A	N/A	IER07.IEN1	IPR057	—
CMPB	CMPB0	58	00E8h	Edge	✓	✓	✓	N/A	IER07.IEN2	IPR058	DTCER058
	CMPB1	59	00ECh	Edge	✓	✓	✓	N/A	IER07.IEN3	IPR059	DTCER059
CTSU	CTSUWR	60	00F0h	Edge	✓	✓	✓	N/A	IER07.IEN4	IPR060	DTCER060
	CTSURD	61	00F4h	Edge	✓	✓	✓	N/A	IER07.IEN5		DTCER061
	CTSUFN	62	00F8h	Edge	✓	N/A	N/A	N/A	IER07.IEN6		—
RTC	CUP	63	00FCh	Edge	✓	N/A	N/A	N/A	IER07.IEN7	IPR063	—
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	✓	✓	N/A	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/Level	✓	✓	N/A	✓	IER08.IEN5	IPR069	DTCER069
	IRQ6	70	0118h	Edge/Level	✓	✓	N/A	✓	IER08.IEN6	IPR070	DTCER070
	IRQ7	71	011Ch	Edge/Level	✓	✓	N/A	✓	IER08.IEN7	IPR071	DTCER071
—	Reserved	72	0120h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	73	0124h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	74	0128h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	75	012Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	76	0130h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	77	0134h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	78	0138h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	79	013Ch	—	N/A	N/A	N/A	N/A	—	—	—
ELC	ELSR8I	80	0140h	Edge	✓	N/A	N/A	✓	IER0A.IEN0	IPR080	—
SYSTEM	SNZI	81	0144h	Edge	✓	N/A	N/A	N/A	IER0A.IEN1	IPR081	
—	Reserved	82	0148h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	83	014Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	84	0150h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	85	0154h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	86	0158h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	87	015Ch	—	N/A	N/A	N/A	N/A	—	—	—
LVD/CMPA	LVD1	88	0160h	Edge	✓	N/A	N/A	✓	IER0B.IEN0	IPR088	—
	LVD2/CMPA2	89	0164h	Edge	✓	N/A	N/A	✓	IER0B.IEN1	IPR089	—

Table 14.3 Interrupt Vector Table (3/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
USB0	USBR0	90	0168h	Level	✓	N/A	N/A	✓	IER0B.IEN2	IPR090	—
—	Reserved	91	016Ch	—	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	—
—	Reserved	95	017Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	96	0180h	—	N/A	N/A	N/A	N/A	—	—	—
	Reserved	97	0184h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	98	0188h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	99	018Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	100	0190h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	101	0194h	—	N/A	N/A	N/A	N/A	—	—	—
S12AD	S12ADI0	102	0198h	Edge	✓	✓	✓	N/A	IER0C.IEN6	IPR102	DTCER102
	GBADI	103	019Ch	Edge	✓	✓	✓	N/A	IER0C.IEN7	IPR103	DTCER103
—	Reserved	104	01A0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	105	01A4h	—	N/A	N/A	N/A	N/A	—	—	—
ELC	ELSR18I	106	01A8h	Edge	✓	✓	✓	N/A	IER0D.IEN2	IPR106	DTCER106
	ELSR19I	107	01ACh	Edge	✓	✓	✓	N/A	IER0D.IEN3	IPR107	DTCER107
GPTW0	GTCIA0	108	01B0h	Edge	✓	✓	✓	N/A	IER0D.IEN4	IPR108	DTCER108
	GTCIB0	109	01B4h	Edge	✓	✓	✓	N/A	IER0D.IEN5		DTCER109
	GTCIC0	110	01B8h	Edge	✓	✓	✓	N/A	IER0D.IEN6	IPR110	DTCER110
	GTCID0	111	01BCh	Edge	✓	✓	✓	N/A	IER0D.IEN7		DTCER111
	GTCIE0	112	01C0h	Edge	✓	✓	✓	N/A	IER0E.IEN0		DTCER112
	GTCIF0	113	01C4h	Edge	✓	✓	✓	N/A	IER0E.IEN1		DTCER113
	GTCIV0	114	01C8h	Edge	✓	✓	✓	N/A	IER0E.IEN2	IPR114	DTCER114
	GTCIU0	115	01CCh	Edge	✓	✓	✓	N/A	IER0E.IEN3		DTCER115
GPTW1	GTCIA1	116	01D0h	Edge	✓	✓	✓	N/A	IER0E.IEN4	IPR116	DTCER116
	GTCIB1	117	01D4h	Edge	✓	✓	✓	N/A	IER0E.IEN5		DTCER117
	GTCIC1	118	01D8h	Edge	✓	✓	✓	N/A	IER0E.IEN6	IPR118	DTCER118
	GTCID1	119	01DCh	Edge	✓	✓	✓	N/A	IER0E.IEN7		DTCER119
	GTCIE1	120	01E0h	Edge	✓	✓	✓	N/A	IER0F.IEN0		DTCER120
	GTCIF1	121	01E4h	Edge	✓	✓	✓	N/A	IER0F.IEN1		DTCER121
	GTCIV1	122	01E8h	Edge	✓	✓	✓	N/A	IER0F.IEN2	IPR122	DTCER122
	GTCIU1	123	01ECh	Edge	✓	✓	✓	N/A	IER0F.IEN3		DTCER123
GPTW2	GTCIA2	124	01F0h	Edge	✓	✓	✓	N/A	IER0F.IEN4	IPR124	DTCER124
	GTCIB2	125	01F4h	Edge	✓	✓	✓	N/A	IER0F.IEN5		DTCER125
	GTCIC2	126	01F8h	Edge	✓	✓	✓	N/A	IER0F.IEN6	IPR126	DTCER126
	GTCID2	127	01FCh	Edge	✓	✓	✓	N/A	IER0F.IEN7		DTCER127
	GTCIE2	128	0200h	Edge	✓	✓	✓	N/A	IER10.IEN0		DTCER128
	GTCIF2	129	0204h	Edge	✓	✓	✓	N/A	IER10.IEN1		DTCER129
	GTCIV2	130	0208h	Edge	✓	✓	✓	N/A	IER10.IEN2	IPR130	DTCER130
	GTCIU2	131	020Ch	Edge	✓	✓	✓	N/A	IER10.IEN3		DTCER131

Table 14.3 Interrupt Vector Table (4/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
GPTW3	GTCIA3	132	0210h	Edge	✓	✓	✓	N/A	IER10.IEN4	IPR132	DTCER132
	GTCIB3	133	0214h	Edge	✓	✓	✓	N/A	IER10.IEN5		DTCER133
	GTCIC3	134	0218h	Edge	✓	✓	✓	N/A	IER10.IEN6	IPR134	DTCER134
	GTCID3	135	021Ch	Edge	✓	✓	✓	N/A	IER10.IEN7		DTCER135
	GTCIE3	136	0220h	Edge	✓	✓	✓	N/A	IER11.IEN0		DTCER136
	GTCIF3	137	0224h	Edge	✓	✓	✓	N/A	IER11.IEN1		DTCER137
	GTCIV3	138	0228h	Edge	✓	✓	✓	N/A	IER11.IEN2	IPR138	DTCER138
	GTCIU3	139	022Ch	Edge	✓	✓	✓	N/A	IER11.IEN3		DTCER139
GPTW4	GTCIA4	140	0230h	Edge	✓	✓	✓	N/A	IER11.IEN4	IPR140	DTCER140
	GTCIB4	141	0234h	Edge	✓	✓	✓	N/A	IER11.IEN5		DTCER141
	GTCIC4	142	0238h	Edge	✓	✓	✓	N/A	IER11.IEN6	IPR142	DTCER142
	GTCID4	143	023Ch	Edge	✓	✓	✓	N/A	IER11.IEN7		DTCER143
	GTCIE4	144	0240h	Edge	✓	✓	✓	N/A	IER12.IEN0		DTCER144
	GTCIF4	145	0244h	Edge	✓	✓	✓	N/A	IER12.IEN1		DTCER145
	GTCIV4	146	0248h	Edge	✓	✓	✓	N/A	IER12.IEN2	IPR146	DTCER146
	GTCIU4	147	024Ch	Edge	✓	✓	✓	N/A	IER12.IEN3		DTCER147
GPTW5	GTCIA5	148	0250h	Edge	✓	✓	✓	N/A	IER12.IEN4	IPR148	DTCER148
	GTCIB5	149	0254h	Edge	✓	✓	✓	N/A	IER12.IEN5		DTCER149
	GTCIC5	150	0258h	Edge	✓	✓	✓	N/A	IER12.IEN6	IPR150	DTCER150
	GTCID5	151	025Ch	Edge	✓	✓	✓	N/A	IER12.IEN7		DTCER151
	GTCIE5	152	0260h	Edge	✓	✓	✓	N/A	IER13.IEN0		DTCER152
	GTCIF5	153	0264h	Edge	✓	✓	✓	N/A	IER13.IEN1		DTCER153
	GTCIV5	154	0268h	Edge	✓	✓	✓	N/A	IER13.IEN2	IPR154	DTCER154
	GTCIU5	155	026Ch	Edge	✓	✓	✓	N/A	IER13.IEN3		DTCER155
GPTW6	GTCIA6	156	0270h	Edge	✓	✓	✓	N/A	IER13.IEN4	IPR156	DTCER156
	GTCIB6	157	0274h	Edge	✓	✓	✓	N/A	IER13.IEN5		DTCER157
	GTCIC6	158	0278h	Edge	✓	✓	✓	N/A	IER13.IEN6	IPR158	DTCER158
	GTCID6	159	027Ch	Edge	✓	✓	✓	N/A	IER13.IEN7		DTCER159
	GTCIE6	160	0280h	Edge	✓	✓	✓	N/A	IER14.IEN0		DTCER160
	GTCIF6	161	0284h	Edge	✓	✓	✓	N/A	IER14.IEN1		DTCER161
	GTCIV6	162	0288h	Edge	✓	✓	✓	N/A	IER14.IEN2	IPR162	DTCER162
	GTCIU6	163	028Ch	Edge	✓	✓	✓	N/A	IER14.IEN3		DTCER163
GPTW7	GTCIA7	164	0290h	Edge	✓	✓	✓	N/A	IER14.IEN4	IPR164	DTCER164
	GTCIB7	165	0294h	Edge	✓	✓	✓	N/A	IER14.IEN5		DTCER165
	GTCIC7	166	0298h	Edge	✓	✓	✓	N/A	IER14.IEN6	IPR166	DTCER166
	GTCID7	167	029Ch	Edge	✓	✓	✓	N/A	IER14.IEN7		DTCER167
	GTCIE7	168	02A0h	Edge	✓	✓	✓	N/A	IER15.IEN0		DTCER168
	GTCIF7	169	02A4h	Edge	✓	✓	✓	N/A	IER15.IEN1		DTCER169
	GTCIV7	170	02A8h	Edge	✓	✓	✓	N/A	IER15.IEN2	IPR170	DTCER170
	GTCIU7	171	02ACh	Edge	✓	✓	✓	N/A	IER15.IEN3		DTCER171
—	Reserved	172	02B0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	173	02B4h	—	N/A	N/A	N/A	N/A	—	—	—
TMR0	CMIA0	174	02B8h	Edge	✓	✓	N/A	N/A	IER15.IEN6	IPR174	DTCER174
	CMIB0	175	02BCh	Edge	✓	✓	N/A	N/A	IER15.IEN7		DTCER175
	OVI0	176	02C0h	Edge	✓	N/A	N/A	N/A	IER16.IEN0		—

Table 14.3 Interrupt Vector Table (5/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
TMR1	CMIA1	177	02C4h	Edge	✓	✓	N/A	N/A	IER16.IEN1	IPR177	DTCER177
	CMIB1	178	02C8h	Edge	✓	✓	N/A	N/A	IER16.IEN2		DTCER178
	OVI1	179	02CCh	Edge	✓	N/A	N/A	N/A	IER16.IEN3		—
TMR2	CMIA2	180	02D0h	Edge	✓	✓	N/A	N/A	IER16.IEN4	IPR180	DTCER180
	CMIB2	181	02D4h	Edge	✓	✓	N/A	N/A	IER16.IEN5		DTCER181
	OVI2	182	02D8h	Edge	✓	N/A	N/A	N/A	IER16.IEN6		—
TMR3	CMIA3	183	02DCh	Edge	✓	✓	N/A	N/A	IER16.IEN7	IPR183	DTCER183
	CMIB3	184	02E0h	Edge	✓	✓	N/A	N/A	IER17.IEN0		DTCER184
	OVI3	185	02E4h	Edge	✓	N/A	N/A	N/A	IER17.IEN1		—
CANFD	RFRI	186	02E8h	Level	✓	N/A	N/A	N/A	IER17.IEN2	IPR186	—
	GLEI	187	02ECh	Level	✓	N/A	N/A	N/A	IER17.IEN3	IPR187	—
	RMRI	188	02F0h	Level	✓	N/A	N/A	N/A	IER17.IEN4	IPR188	—
	RFDREQ0	189	02F4h	Edge	✓	✓	✓	N/A	IER17.IEN5	IPR189	DTCER189
	RFDREQ1	190	02F8h	Edge	✓	✓	✓	N/A	IER17.IEN6	IPR190	DTCER190
	EC1EI	191	02FCh	Edge	✓	N/A	N/A	N/A	IER17.IEN7	IPR191	—
	EC2EI	192	0300h	Edge	✓	N/A	N/A	N/A	IER18.IEN0	IPR192	—
	ECOVI	193	0304h	Edge	✓	N/A	N/A	N/A	IER18.IEN1	IPR193	—
CANFD0	CHTI	194	0308h	Level	✓	N/A	N/A	N/A	IER18.IEN2	IPR194	—
	CHEI	195	030Ch	Level	✓	N/A	N/A	N/A	IER18.IEN3	IPR195	—
	CFRI	196	0310h	Level	✓	N/A	N/A	N/A	IER18.IEN4	IPR196	—
	CFDREQ0	197	0314h	Edge	✓	✓	✓	N/A	IER18.IEN5	IPR197	DTCER197
DMAC	DMAC0I	198	0318h	Edge	✓	✓	N/A	N/A	IER18.IEN6	IPR198	DTCER198
	DMAC1I	199	031Ch	Edge	✓	✓	N/A	N/A	IER18.IEN7	IPR199	DTCER199
	DMAC2I	200	0320h	Edge	✓	✓	N/A	N/A	IER19.IEN0	IPR200	DTCER200
	DMAC3I	201	0324h	Edge	✓	✓	N/A	N/A	IER19.IEN1	IPR201	DTCER201
RSIP	PROC_BUSY	202	0328h	Edge	✓	N/A	N/A	N/A	IER19.IEN2	IPR202	—
	ROMOK	203	032Ch	Edge	✓	N/A	N/A	N/A	IER19.IEN3	IPR203	—
	LONG_PLG	204	0330h	Edge	✓	N/A	N/A	N/A	IER19.IEN4	IPR204	—
	TEST_BUSY	205	0334h	Edge	✓	N/A	N/A	N/A	IER19.IEN5	IPR205	—
	WRRDY0	206	0338h	Edge	✓	✓	✓	N/A	IER19.IEN6	IPR206	DTCER206
	WRRDY2	207	033Ch	Edge	✓	✓	✓	N/A	IER19.IEN7	IPR207	DTCER207
	RDRDY0	208	0340h	Edge	✓	✓	✓	N/A	IER1A.IEN0	IPR208	DTCER208
	INTEGRATE_RDRDY	209	0344h	Edge	✓	✓	✓	N/A	IER1A.IEN1	IPR209	DTCER209
	INTEGRATE_WRRDY	210	0348h	Edge	✓	✓	✓	N/A	IER1A.IEN2	IPR210	DTCER210
	ECCERR	211	034Ch	Edge	✓	N/A	N/A	N/A	IER1A.IEN3	IPR211	—
RSCI9	BFD	212	0350h	Edge	✓	N/A	N/A	N/A	IER1A.IEN4	IPR212	—
	AED	213	0354h	Edge	✓	✓	✓	N/A	IER1A.IEN5	IPR213	DTCER213
RSCI0	ERI	214	0358h	Level	✓	N/A	N/A	N/A	IER1A.IEN6	IPR214	—
	RXI	215	035Ch	Edge	✓	✓	✓	N/A	IER1A.IEN7		DTCER215
	TXI	216	0360h	Edge	✓	✓	✓	N/A	IER1B.IEN0		DTCER216
	TEI	217	0364h	Level	✓	N/A	N/A	N/A	IER1B.IEN1		—
SCI1	ERI1	218	0368h	Level	✓	N/A	N/A	N/A	IER1B.IEN2	IPR218	—
	RXI1	219	036Ch	Edge	✓	✓	✓	N/A	IER1B.IEN3		DTCER219
	TXI1	220	0370h	Edge	✓	✓	✓	N/A	IER1B.IEN4		DTCER220
	TEI1	221	0374h	Level	✓	N/A	N/A	N/A	IER1B.IEN5		—

Table 14.3 Interrupt Vector Table (6/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	Start DMAC	SSBY Return	IER	IPR	DTCER
SCI5	ERI5	222	0378h	Level	✓	N/A	N/A	N/A	IER1B.IEN6	IPR222	—
	RXI5	223	037Ch	Edge	✓	✓	✓	N/A	IER1B.IEN7		DTCER223
	TXI5	224	0380h	Edge	✓	✓	✓	N/A	IER1C.IEN0		DTCER224
	TEI5	225	0384h	Level	✓	N/A	N/A	N/A	IER1C.IEN1		—
SCI6	ERI6	226	0388h	Level	✓	N/A	N/A	N/A	IER1C.IEN2	IPR226	—
	RXI6	227	038Ch	Edge	✓	✓	✓	N/A	IER1C.IEN3		DTCER227
	TXI6	228	0390h	Edge	✓	✓	✓	N/A	IER1C.IEN4		DTCER228
	TEI6	229	0394h	Level	✓	N/A	N/A	N/A	IER1C.IEN5		—
RSCI8	ERI	230	0398h	Level	✓	N/A	N/A	N/A	IER1C.IEN6	IPR230	—
	RXI	231	039Ch	Edge	✓	✓	✓	N/A	IER1C.IEN7		DTCER231
	TXI	232	03A0h	Edge	✓	✓	✓	N/A	IER1D.IEN0		DTCER232
	TEI	233	03A4h	Level	✓	N/A	N/A	N/A	IER1D.IEN1		—
RSCI9	ERI	234	03A8h	Level	✓	N/A	N/A	N/A	IER1D.IEN2	IPR234	—
	RXI	235	03ACh	Edge	✓	✓	✓	N/A	IER1D.IEN3		DTCER235
	TXI	236	03B0h	Edge	✓	✓	✓	N/A	IER1D.IEN4		DTCER236
	TEI	237	03B4h	Level	✓	N/A	N/A	N/A	IER1D.IEN5		—
SCI12	ERI12	238	03B8h	Level	✓	N/A	N/A	N/A	IER1D.IEN6	IPR238	—
	RXI12	239	03BCh	Edge	✓	✓	✓	N/A	IER1D.IEN7		DTCER239
	TXI12	240	03C0h	Edge	✓	✓	✓	N/A	IER1E.IEN0		DTCER240
	TEI12	241	03C4h	Level	✓	N/A	N/A	N/A	IER1E.IEN1		—
	SCIX0	242	03C8h	Level	✓	N/A	N/A	N/A	IER1E.IEN2	IPR242	—
	SCIX1	243	03CCh	Level	✓	N/A	N/A	N/A	IER1E.IEN3	IPR243	—
	SCIX2	244	03D0h	Level	✓	N/A	N/A	N/A	IER1E.IEN4	IPR244	—
	SCIX3	245	03D4h	Level	✓	N/A	N/A	N/A	IER1E.IEN5	IPR245	—
RIIC0	EEI0	246	03D8h	Level	✓	N/A	N/A	N/A	IER1E.IEN6	IPR246	—
	RXI0	247	03DCh	Edge	✓	✓	✓	N/A	IER1E.IEN7	IPR247	DTCER247
	TXI0	248	03E0h	Edge	✓	✓	✓	N/A	IER1F.IEN0	IPR248	DTCER248
	TEI0	249	03E4h	Level	✓	N/A	N/A	N/A	IER1F.IEN1	IPR249	—
—	Reserved	250	03E8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	251	03ECh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	252	03F0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	253	03F4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	254	03F8h	—	N/A	N/A	N/A	N/A	—	—	—
LPT	LPTCMI1	255	03FCh	Edge	✓	✓	✓	N/A	IER1F.IEN7	IPR255	DTCER255

Note 1. An interrupt source with a smaller vector number takes precedence.

14.3.2 Fast Interrupt Vector Table

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

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 a multiple of 4 in the EXTB register.

14.4 Interrupt Operation

The interrupt controller performs the following processing.

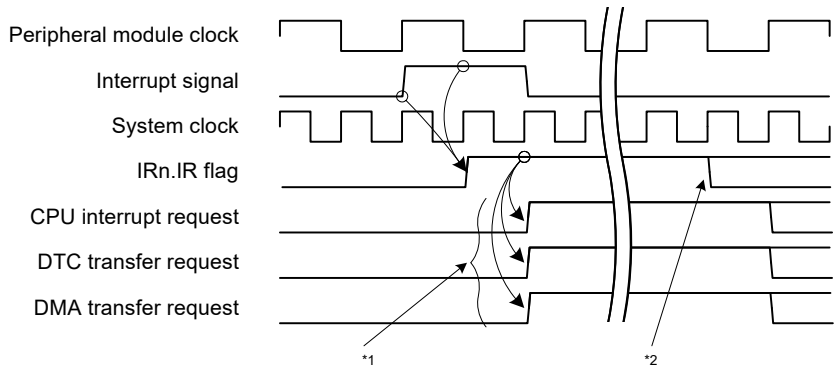
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC trigger, or DMAC trigger)
- Determining priority

14.4.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.
Edge detection or level detection is selected for the IRQi pins (i = 0 to 7) as external interrupt requests by the setting of the IRQMD[1:0] bits in IRQCRi.
For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source.
For the correspondence between interrupt sources and methods of detection, see Table 14.3, Interrupt Vector Table.

14.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 14.2 shows the operation of the IR flag in IRn (n = interrupt vector number) in the case of edge detection of an interrupt from a peripheral module or on an external pin.
The IR flag in IRn is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DMAC or DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DMAC/DTC transfer settings and transfer count. For details, see Table 14.4, Operation When Starting the DMAC/DTC.



Note 1. One of the following requests is issued: CPU interrupt request, DTC transfer request, and DMA transfer request. For details of the setting, see section 14.4.3, Selecting Interrupt Request Destinations.
Note 2. When the CPU interrupt request is specified, this flag is set to 0 on acceptance of a CPU interrupt. For the timing of 0 setting at the DTC transfer request or DMA transfer request, see Table 14.4, Operation When Starting the DMAC/DTC.

Figure 14.2 IRn.IR Flag Operation for Edge Detection Interrupts

Figure 14.3 to Figure 14.6 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK cycles” of delay is added.

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

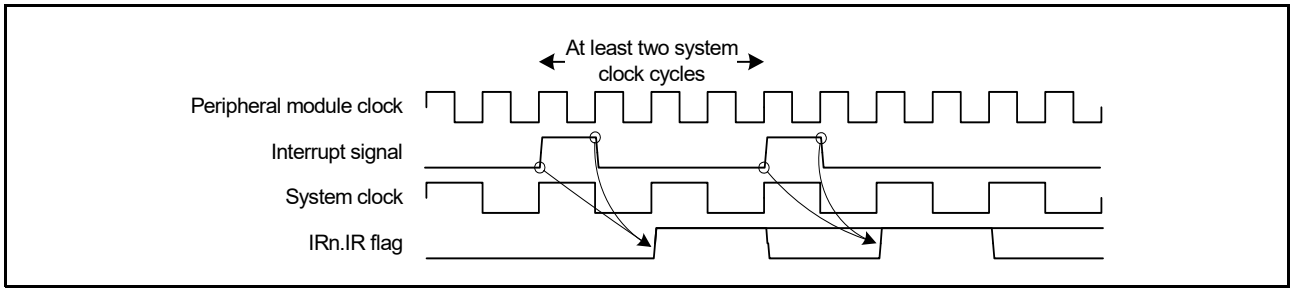


Figure 14.3 Interval Required between Issuance of Continuous Interrupt Requests (when the Frequency of System Clock is Slower than that of the Peripheral Module Clock)

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.*1
Figure 14.4 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the USB, SCI, RSCI, RIIC, or RSPI is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is cleared to 0, the IRn.IR flag is set to 1 again by the retained request. For details, see descriptions of the interrupts in section 31, Serial Communications Interface (SCIk, SCIh), section 33, I2C-bus Interface (RIICa), and section 35, Serial Peripheral Interface (RSPIc).

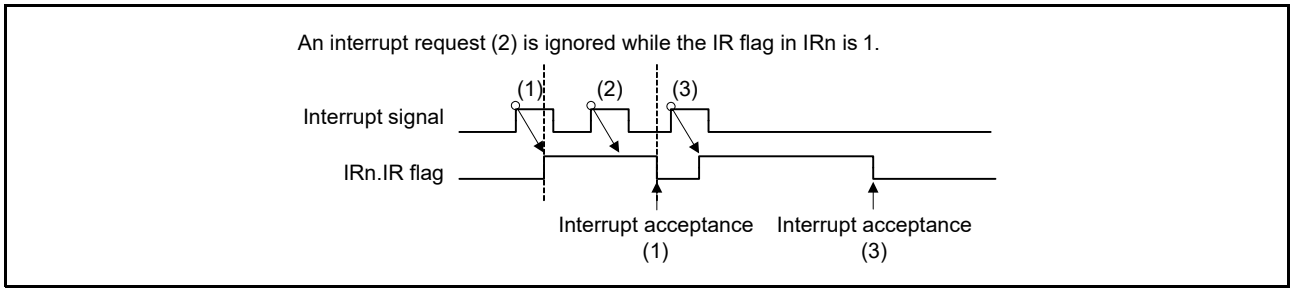


Figure 14.4 Timing for IRn.IR Flag Re-Setting

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 14.5 shows operation when the interrupt is disabled.

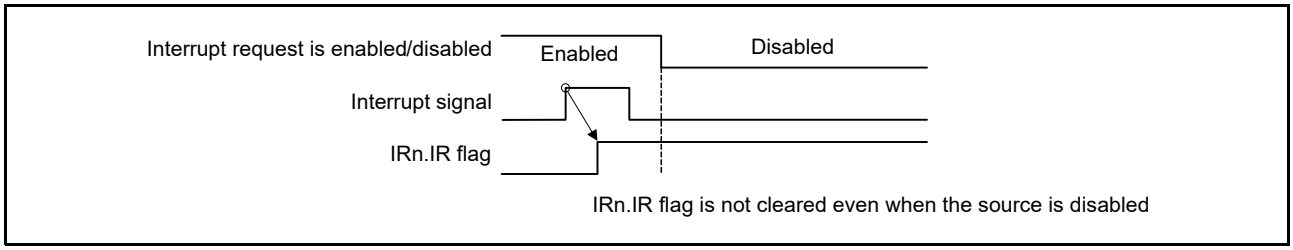


Figure 14.5 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

14.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 14.6 shows the operation of the interrupt status flag (IR flag) in IRn (n = interrupt vector number) in the case of level detection of an interrupt from a peripheral module or an external pin.
The IR flag in IRn remains set to 1 as long as the interrupt signal is asserted. To clear the IRn.IR flag to 0, clear the interrupt request in the source generating the interrupt. Confirm that the interrupt request flag in the source generating the interrupt has been cleared to 0 and that the IRn.IR flag has been cleared to 0, and then complete the interrupt handling.

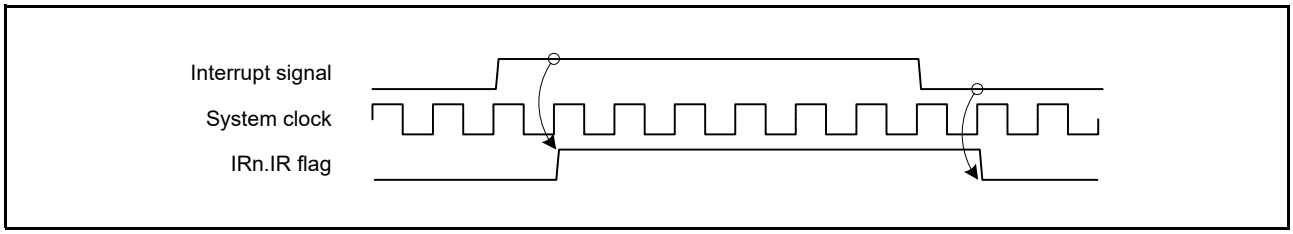


Figure 14.6 IRn.IR Flag Operation for Level Detection Interrupts

Figure 14.7 shows the procedure for handling level detection interrupts.

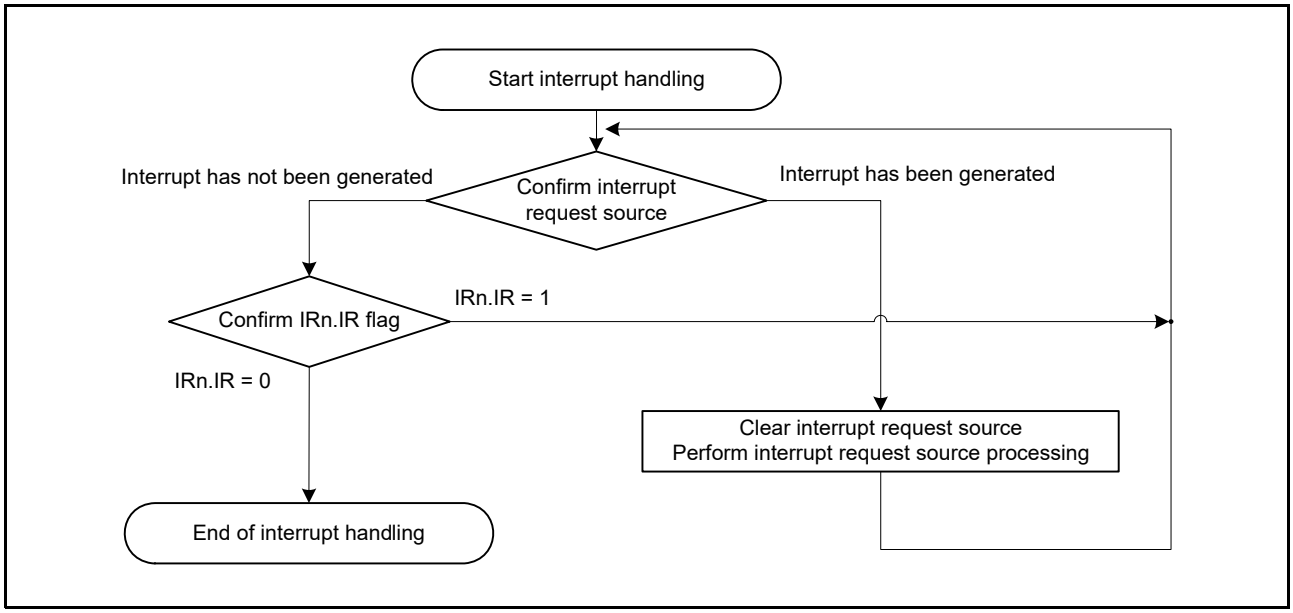


Figure 14.7 Procedure for Handling Level Detection Interrupts

14.4.2 Enabling and Disabling Interrupt Sources

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

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. Enabling of the interrupt by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7)

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag (n = interrupt vector number) is set to 1.
Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.
The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
3. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.*1

Note 1. To disable the transmission or reception interrupt of the SCI, RSCI, RIIC, or RSPI from the enabled state, clear the IRn.IR flag to 0 using the above procedure. For details, see descriptions of the interrupts in section 31, Serial Communications Interface (SCIk, SCIh), section 33, I2C-bus Interface (RIICa), and section 35, Serial Peripheral Interface (RSPIc).

14.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those indicated in Table 14.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a "✓" in Table 14.3.
If the DMAC or DTC is selected as the destination for requests from an IRQi pin (i = 0 to 7), be sure to set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.
The following describes how to specify the destinations of interrupt requests.

(1) DMAC Trigger

Make the following settings for each source while the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) is 0.

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

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

(2) DTC Trigger

Make the following settings for each source while the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) is 0.

- Set the DTC transfer request enable bit in the DTC transfer request enable register (DTCERn.DTCE (n = interrupt vector number)) for the pertinent source to 1.*1

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DTC module start bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DTC module start bit does not matter.
For the DTC setting procedure, refer to section 18.5, DTC Setting Procedure, in section 18, Data Transfer Controller (DTCb).

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

(3) CPU Interrupt Request

If the interrupt request destination is neither the DMAC nor the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) to 1 while neither the DMAC trigger settings nor the DTC trigger settings described above are in place.

Table 14.4 shows operation when the DMAC or DTC is the request destination.

Table 14.4 Operation When Starting the DMAC/DTC

Interrupt Request Destination	DISEL *1	Remaining Number of Transfer Operations	Operation per Request	IR*2	Interrupt Request Destination after Transfer
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DMA transfer	Cleared at the start of DMAC transfer	DMAC
		= 0	DMA transfer*3	Cleared at the start of DMAC transfer*3	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
DTC*4	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer information	DTC
		= 0	DTC transfer → CPU interrupt*3	Cleared on interrupt acceptance by the CPU*3	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.

Note 1. DISEL for the DMAC is set by the DMACm.DMCSL.DISEL bit; DISEL for the DTC is set by the DTC.MRB.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC or DMA transfer request) that is generated again will be ignored.
Note 3. When the DISEL bit is 0, operation with the remaining number of transfer operations being 0 differs according to whether the source is for the DTC or DMAC.
Note 4. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 18.4, Chain Transfer Conditions in section 18, Data Transfer Controller (DTCb).

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

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

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

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

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

14.4.4 Determining Priority

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

(1) Determining Priority when the CPU is the Request Destination of the Interrupt

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPR[3:0]) in IPRn takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

(2) Determining Priority when the DTC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn (n = interrupt vector number) have no effect. An interrupt source with a smaller vector number takes precedence.

(3) Determining Priority when the DMAC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn have no effect. Regarding the order of priority of DMAC channels, see section 17, DMA Controller (DMACA).

14.4.5 Multiple Interrupts

To enable multiple interrupts of the CPU, set the PSW.I bit to 1 (interrupt enabled) in the handling routine of accepted interrupts.
The PSW.IPL[3:0] bits immediately after processing branches to the interrupt handling routine are set to the same value as the interrupt priority level of the accepted interrupt request. If an interrupt request which has an interrupt level higher than that of the PSW.IPL[3:0] bits is generated at this time, this interrupt request (for multiple interrupts) is accepted.
If the interrupt priority level of the accepted interrupt request is 15 (fast interrupt or interrupt when IPR[3:0] are set to 1111b), multiple interrupts are not generated.

14.4.6 Fast Interrupt

The fast interrupt is an interrupt for executing a faster interrupt response by the CPU, so only one of the interrupt sources can be assigned.
The interrupt priority level of the fast interrupt is 15 (highest) regardless of the setting of the IPR[3:0] bits in IPRn (n = interrupt vector number). In addition, the fast interrupt is accepted with precedence over other interrupt sources with level 15. However, when the value of the PSW.IPL[3:0] bits are 1111b (priority level 15), even the fast interrupt cannot be accepted.
To assign an interrupt source to the fast interrupt, specify the vector number of the source in the FIR.FVCT[7:0] bits, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
For details on the fast interrupt, see section 2, CPU and section 13, Exception Handling.

14.4.7 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0 to 7) and NMI pin interrupt.
The digital filter samples input signals at the filter sampling clock (PCLK) and removes the pulses of which length is less than three sampling cycles.
To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTC0.FCLKSELi[1:0] bits and set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).
To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
Figure 14.8 shows an example of digital filter operation.

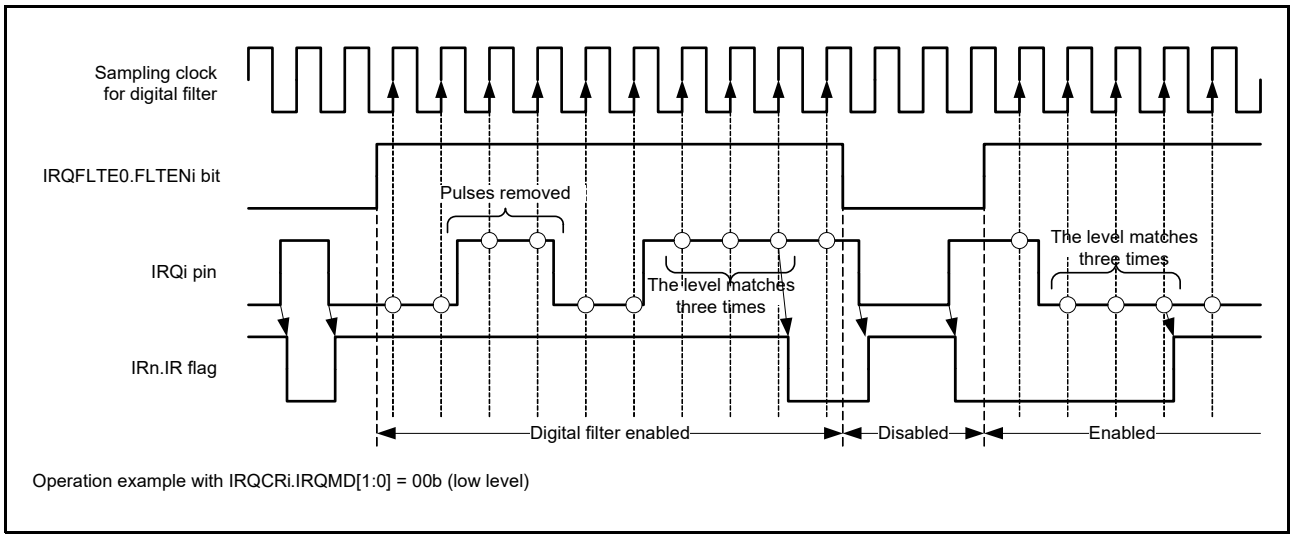


Figure 14.8 Digital Filter Operation Example

Before software standby mode is entered, set the IRQFLTE0.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE0.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

14.4.8 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Clear the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7) to 0 (interrupt request disabled).
2. Clear the IRQFLTE0.FLTENi bit (i = 0 to 7) to 0 (digital filter disabled).*1
3. Set the digital filter sampling clock with the IRQFLTC0.FCLKSELi[1:0] bits.*1
4. Make or confirm the I/O port settings.
5. Set the method of detection for the interrupt in the IRQCRi.IRQMD[1:0] bits.
6. Clear the corresponding IRn.IR flag (n = interrupt vector number) to 0 (if edge detection is in use).
7. Set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).*1
8. If the interrupt is to be used for DMAC trigger, set the DMRSRm.DMRS[7:0] bits. If the interrupt is to be used for DTC trigger, set the DTCERn.DTCE bit. The interrupt will be a CPU interrupt if neither of these settings is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

Note 1. To use the digital filter function, settings must be made beforehand.

14.5 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error, IWDT underflow/refresh error, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and RAM error interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC or DMAC trigger. Non-maskable interrupts take precedence over all interrupts, including the fast interrupt.
Non-maskable interrupt requests are accepted regardless of the states of the I (interrupt enable) bit and IPL[3:0] (processor interrupt priority level) bits in the PSW of the CPU. The current states of the non-maskable interrupts can be checked in the non-maskable interrupt status register (NMISR).
Confirm that all bits in the NMISR have returned to 0 from within the handler for the non-maskable interrupt, before ending the handler.
Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, clear the NMIFLTE.NFLTEN bit to 0 (digital filter disabled).*1
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.*1
4. To use the NMI pin, set the NMI pin detection sense with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
6. To use the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).*1
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

Note 1. To use the digital filter function, settings must be made beforehand.

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. It can be disabled only by a reset.
For the flow of non-maskable interrupt processing, see section 13, Exception Handling.
Writing 1 to the NMICLR.NMICLR bit sets the NMI status flag (NMISR.NMIST) to 0.
Writing 1 to the NMICLR.OSTCLR bit sets the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.
Writing 1 to the NMICLR.WDTCLR bit sets the WDT underflow/refresh error status flag (NMISR.WDTST) to 0.
Writing 1 to the NMICLR.IWDTCLR bit sets the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.
Writing 1 to the NMICLR.LVD1CLR bit sets the voltage monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.
Writing 1 to the NMICLR.LVD2CLR bit sets the voltage monitoring 2 interrupt status flag (NMISR.LVD2ST) to 0.
Writing 0 to the RAM.RAMSTS.RAMERR flag sets the RAM error interrupt status flag (NMISR.RAMST) to 0.

14.6 Return from Low Power Consumption States

Interrupts can be used to return from sleep mode, deep sleep mode, software standby mode, or snooze mode. Refer to section 11, Low Power Consumption for details. This section describes the procedure to set an interrupt source for returning from each low power consumption mode.

14.6.1 Return from Sleep Mode or Deep Sleep Mode

All non-maskable interrupts and all interrupts can be used to return from sleep mode and deep sleep mode. The following conditions must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for return 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.

14.6.2 Return from Software Standby Mode

Non-maskable interrupts and interrupts that have a "✓" in the SSBY return column in Table 14.3, Interrupt Vector Table can be used to return from software standby mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for return 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 to return from 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 IPRn.IPR[3:0] bits (n = interrupt vector number). Set the IPRn.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 14.4.7, Digital Filter for details on the procedure to set the digital filter.

14.6.3 Return from Snooze Mode

Non-maskable interrupts, external pin interrupts (IRQ0 to IRQ7), peripheral function interrupts (voltage monitoring 1, voltage monitoring 2, RTC alarm/periodic, REMC, USB0 resume), and SNZI interrupt (snooze release interrupt) can be used to return from snooze mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for return 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 to return from snooze 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 IPRn.IPR[3:0] bits (n = interrupt vector number). Set the IPRn.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 14.4.7, Digital Filter for details on the procedure to set the digital filter.

14.7 Usage Note

14.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Before executing the WAIT instruction, check to see that all the status flags in NMISR are 0.

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 to 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, ROM) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to ROM
Internal main buses	Internal main bus 1	• Connected to the CPU • Operates in synchronization with the system clock (ICLK)
	Internal main bus 2	• Connected to the DTC and DMAC • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Internal peripheral buses	Internal peripheral bus 1	• Connected to peripheral modules (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 (USB0, CANFD, CTSU, REMC, and RSCI) • Operates in synchronization with the peripheral module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral module (GPTW) • Operates in synchronization with the peripheral module clock (PCLKA)
	Internal peripheral bus 5	• Connected to peripheral module (CANFD (message buffer RAM)) • Operates in synchronization with the peripheral module clock (PCLKA)
	Internal peripheral bus 6	• Connected to ROM (P/E) and E2 DataFlash • Operates in synchronization with the FlashIF clock (FCLK)

P/E: Programming/Erasure

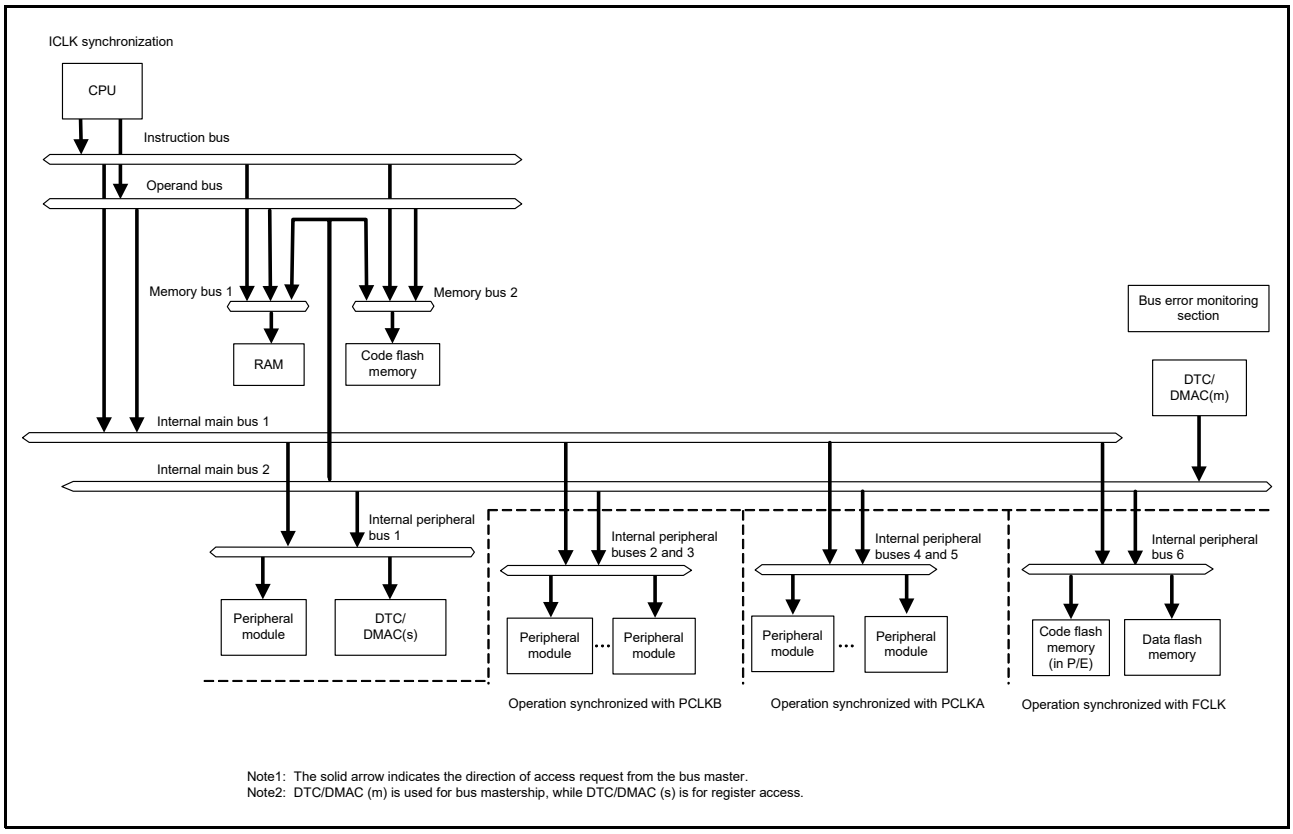


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 FFFFh	Memory bus 1	RAM
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
000C 0000h to 000D FFFFh	Internal peripheral bus 4	
000E 0000h to 000F FFFFh	Internal peripheral bus 5	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	E2 DataFlash memory and ROM (for programming/erasure)
8000 0000h to FEFF FFFFh	Memory bus 2	ROM (for reading only)
FF00 0000h to FFFF FFFFh		

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 ROM provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to ROM by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to ROM and RAM is possible.

15.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. RAM is connected to memory bus 1 and ROM 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 CPU bus and internal main bus 2 can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (ROM) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, the bus for which a request has been accepted has lower priority.

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 DMC, and then DTC as listed 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, and then DTC, regardless of the BUSPRI setting.
If the CPU and another bus master are requesting access to different buses (on-chip memory and internal peripheral buses 1 to 6), 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	Internal Main Bus	Bus Master
High ↑	2	DMAC
		DTC
Low	1	CPU

Note: The above applies when the priority order of the buses is fixed.
The priority order of the internal main bus 1 and another bus (internal main bus 2) can be toggled by using the bus priority control register (BUSPRI) (round-robin method).

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	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	USB0, CANFD, CTSU, REMC, RSCI
Internal peripheral bus 4	GPTW
Internal peripheral bus 5	CANFD (message buff RAM)
Internal peripheral bus 6	ROM (P/E)/E2 DataFlash

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 the 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 buses 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), internal peripheral buses 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 (see Figure 15.2). As shown in the Figure 15.2, the priority order does not change if the priority of the accepted request is low.

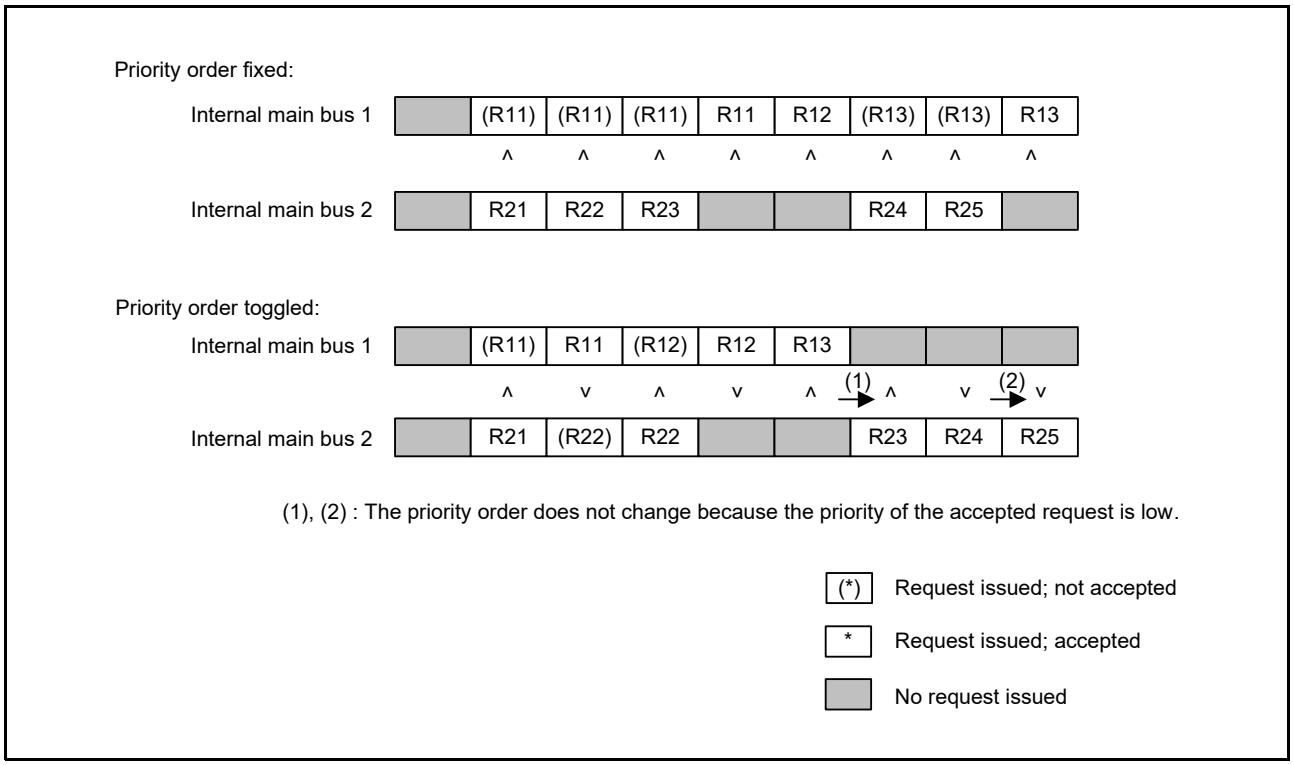


Figure 15.2 Priority Order Between Internal Peripheral Bus Accesses

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 read access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (see Figure 15.3).

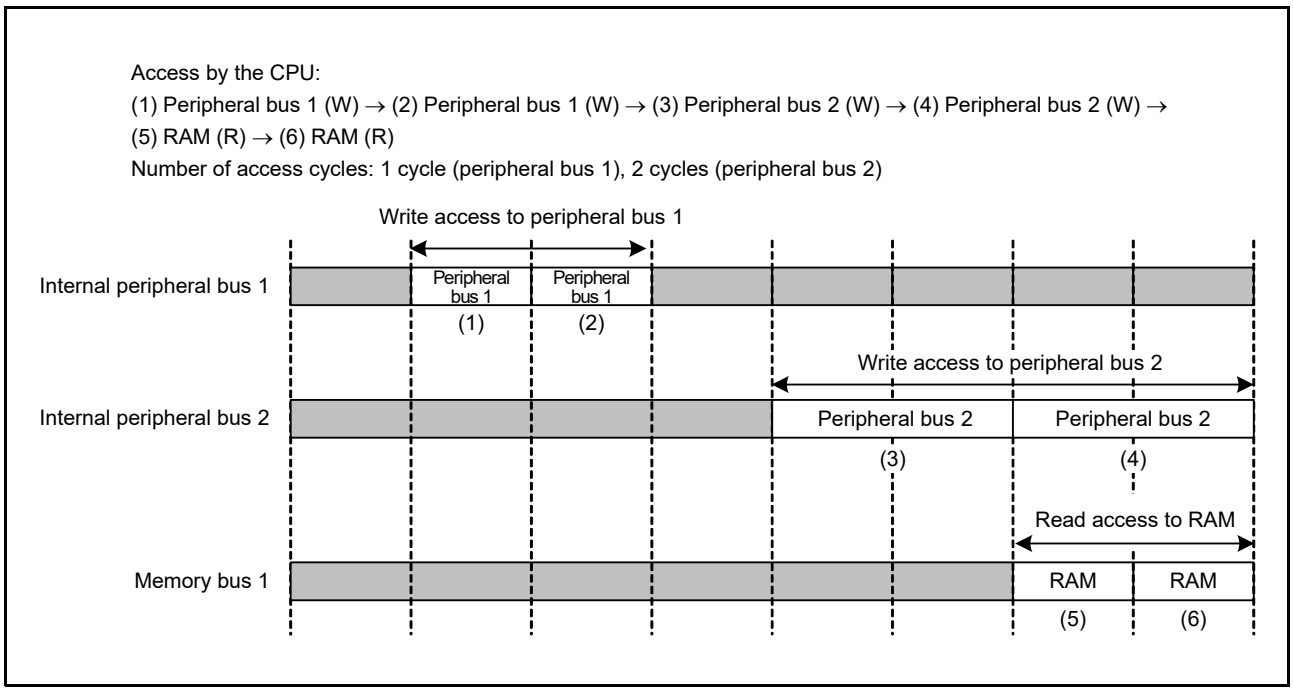


Figure 15.3 Write Buffer Function

15.2.6 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from ROM and an operand from RAM, the DMAC is able to handle transfer between a peripheral bus and peripheral bus at the same time.
An example of parallel operations is shown in Figure 15.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to a peripheral bus during access to RAM and ROM by the CPU.

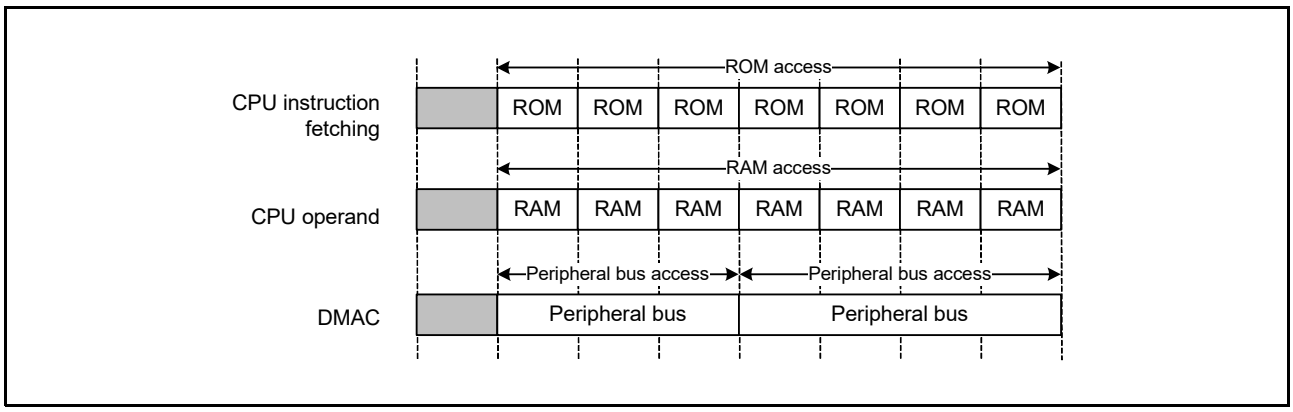


Figure 15.4 Example of Parallel Operations

15.2.7 Restrictions

(1) Prohibition of Access that Spans Multiple 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. Ensure that a single word or longword access does not span across two areas by crossing address space area boundaries.

(2) Restrictions on RMPA and String-Manipulation Instructions

(a) 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.

15.3 Register Descriptions

15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): BSC.BERCLR 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.

15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): BSC.BEREN 0008 1304h

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

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is cleared to 0), bus access can cause the bus to freeze.
Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): BSC.BERSR1 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.

15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): BSC.BERSR2 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

15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): BSC.BUSPRI 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	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 (ROM) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Buses 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 Buses 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
b15 to b12	—	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 stopped. Operation is not guaranteed if it is written more than once.

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 the 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 (ROM) Priority Control)

These bits specify the priority order for memory bus 2 (ROM).

When the priority order is fixed, internal main bus 2 has priority over the 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 Buses 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 Buses 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.

15.4 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.

15.4.1 Type 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.

15.4.1.1 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 to an illegal address area leads to illegal address access errors.
The address ranges where access will lead to illegal address access errors are listed in Table 15.5.

15.4.1.2 Timeout

When the timeout detection enable bit in the bus error monitoring enable register is enabled (BEREN.TOEN = 1), bus access that is not completed within 768 cycles leads to a timeout error.

- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access.
  If a timeout error occurs, accesses from the bus master are not accepted for 256 PCLKB cycles.
- Internal peripheral buses (4 and 5): Bus access is not completed within 768 peripheral module clock (PCLKA) cycles from the start of the access.
  If a timeout error occurs, accesses from the bus master are not accepted 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.
  If a timeout error occurs, accesses from the bus master are not accepted for 256 FCLK cycles. In this MCU, a timeout error does not occur.

15.4.2 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 ICU.IERn register can specify whether to generate an interrupt in the case of a bus error.

15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the type of bus errors for each area in the respective address space.
If an illegal address access error is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1, 2) is cleared), the detected error is reflected in the BERSRn register. Once a bus error occurs, no subsequent bus errors are reflected in 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 the BERSRn register is cleared.

Table 15.5 Type of Bus Errors

Address	Type of Area	Type of Error	
		Illegal Address Access	Timeout
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 00FF FFFFh	Internal peripheral bus 6	Δ	—
0100 0000h to 07FF FFFFh	Reserved area	✓	—
0800 0000h to 0FFF FFFFh	Reserved area	✓	—
1000 0000h to 7FFF FFFFh	Reserved area	✓	—
8000 0000h to FFFF FFFFh	Memory bus 2	—	—

—: A bus error does not result.
Δ: A bus error may or may not result.
✓: A bus error results.
Note: The capacity of the ROM differs depending on the product. For details, refer to section 46, Flash Memory (FLASH).

15.5 Interrupt

15.5.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 15.6 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.

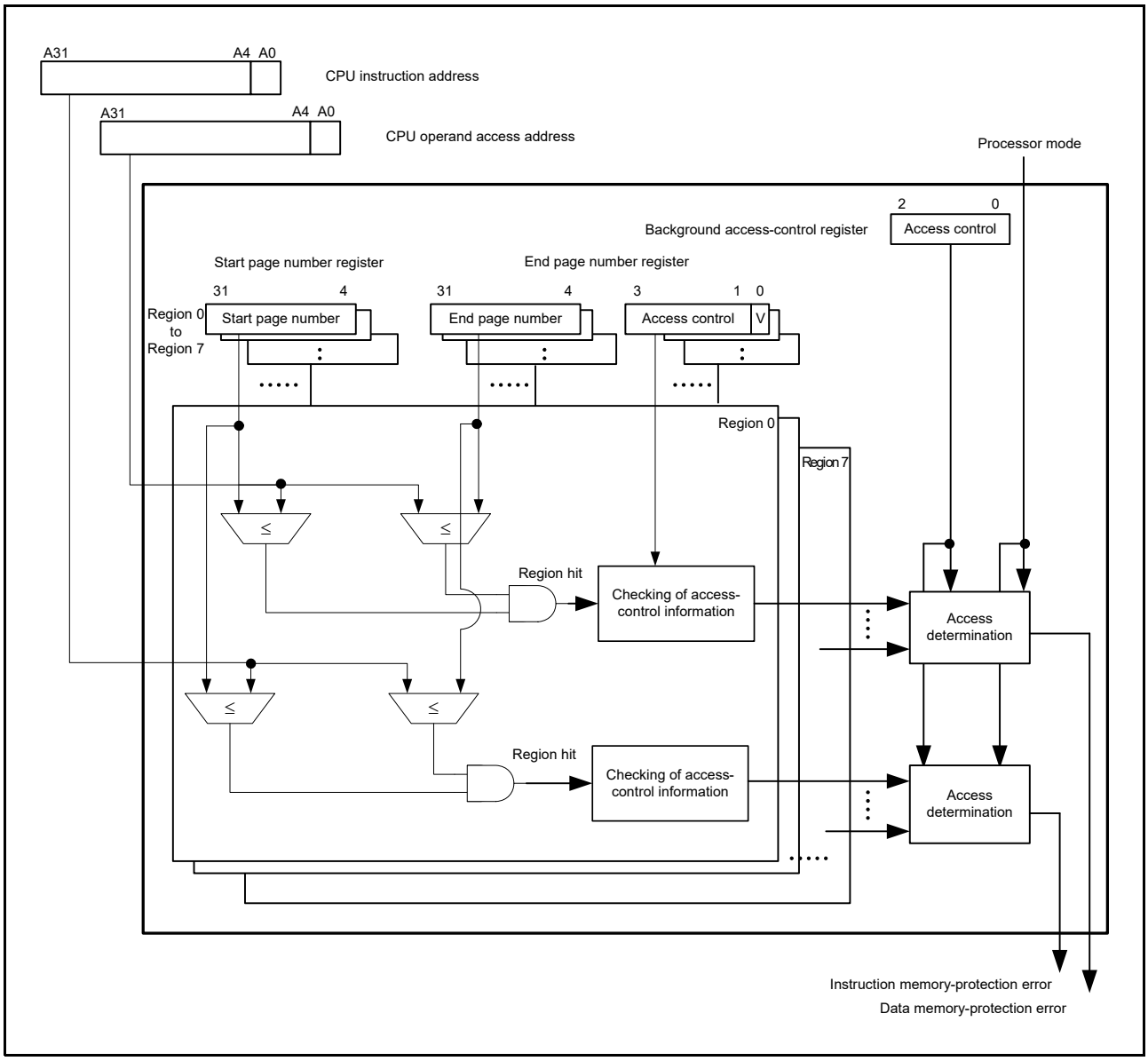


Figure 16.1 Block Diagram of the Memory-Protection Unit

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

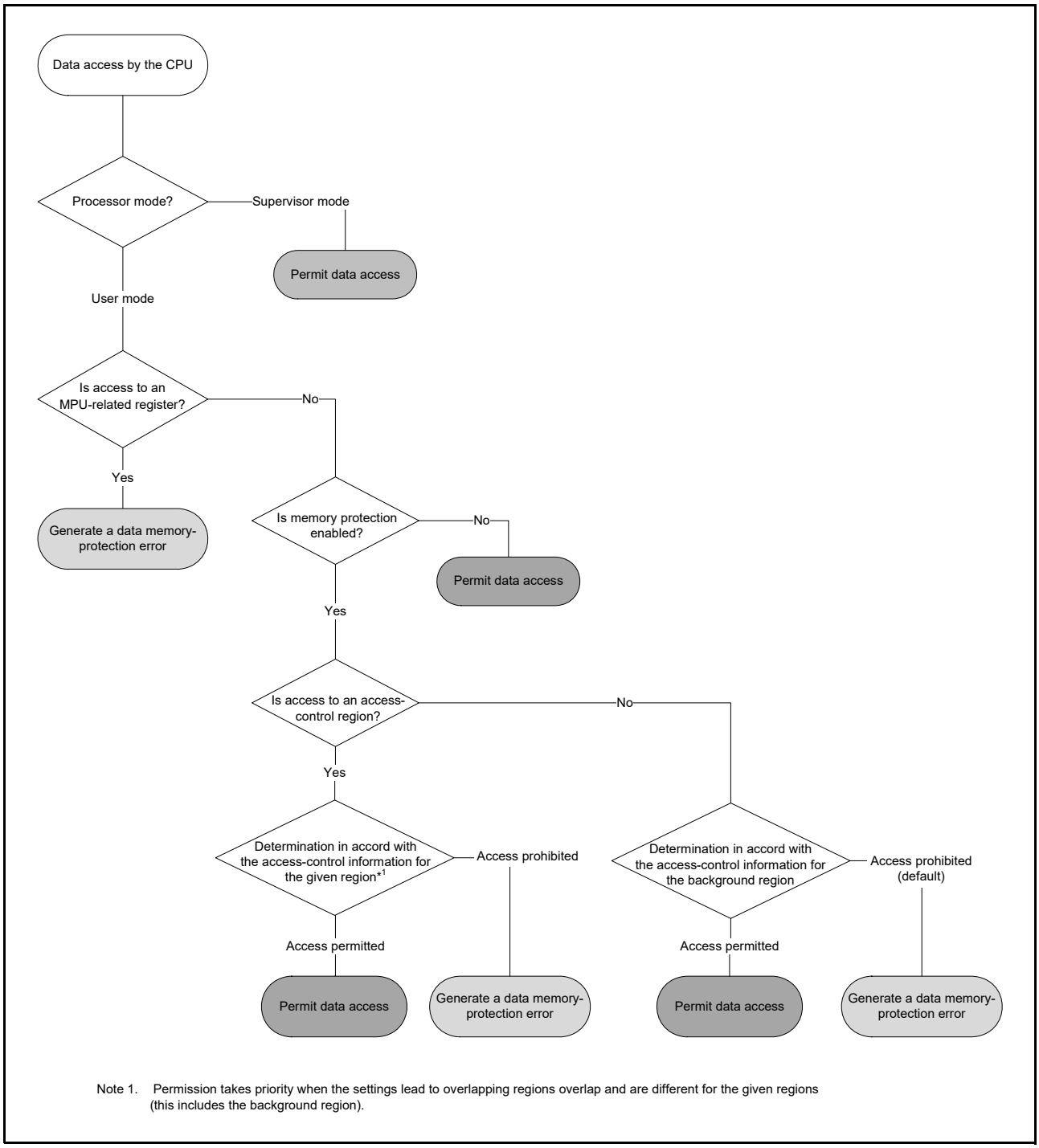
Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOPS) register is set to 1, the address specified 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.



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are different for the given regions (this includes the background region).

Figure 16.2 Flow of Determination for Data Access

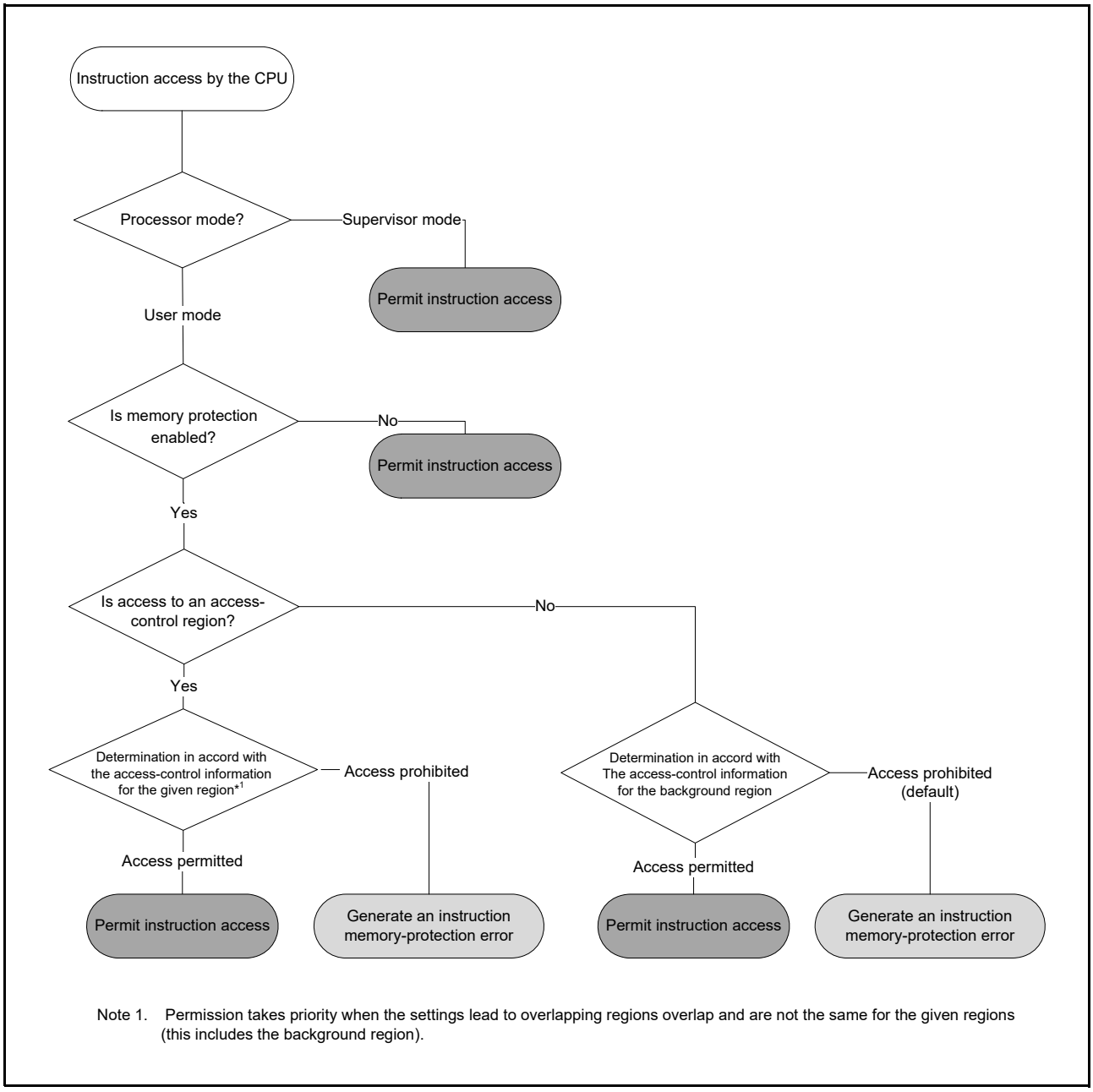


Figure 16.3 Flow of Determination for Instruction Access

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 (MHITI.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 (DMACA)

This MCU incorporates a 4-channel direct memory access controller (DMAC).
The DMAC module performs data transfers without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

17.1 Overview

Table 17.1 lists the specifications of the DMAC, and Figure 17.1 shows a block diagram of the DMAC.

Table 17.1 Specifications of DMAC

Item		Description
Number of channels		4 (DMACm (m = 0 to 3))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		1M data (Maximum number of transfers in block transfer mode: 1,024 data × 1,024 blocks)
DMA request source		• Activation source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Power consumption reduction function		Module stop state can be set.
Event link function		Event link request is generated after one data transfer (for block, after one block transfer).

Note 1. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

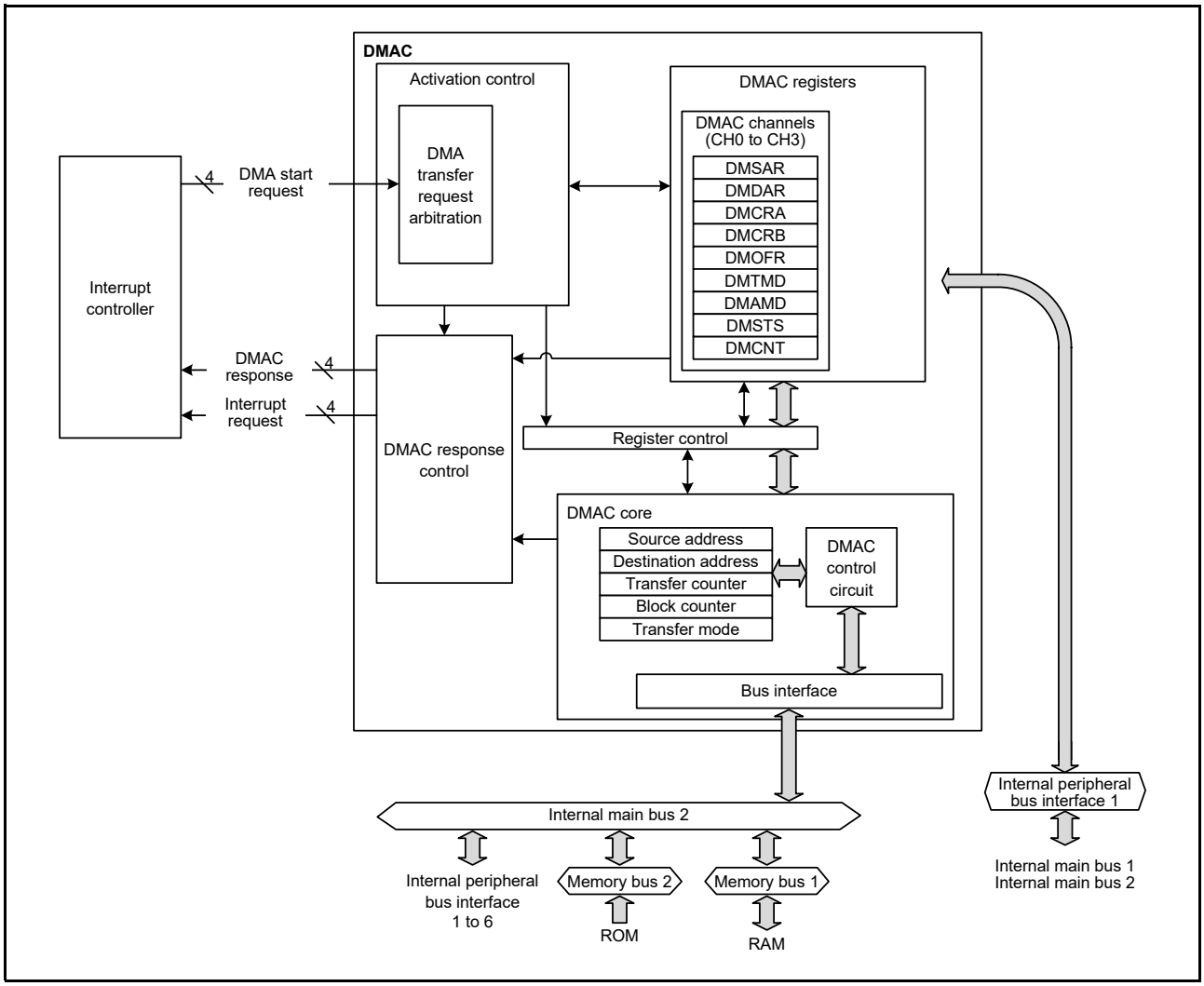


Figure 17.1 Block Diagram of DMAC

17.2 Register Descriptions

17.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h, DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h

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

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMSAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

17.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h, DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h

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

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

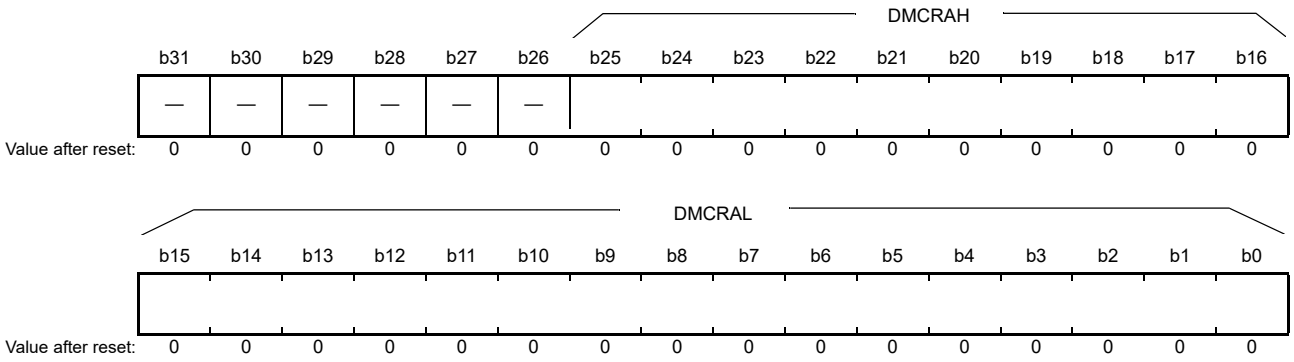
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

DMDAR specifies the start address of the transfer destination. Set DMDAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMDAR returns the extended value.

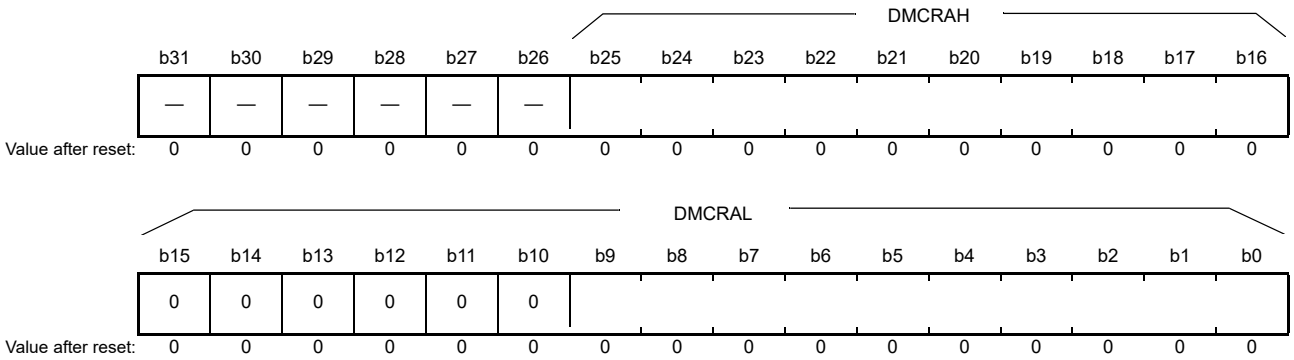
17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for DMCRAH and DMCRAL in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

DMCRAH is not used in normal transfer mode. Write 0000h to DMCRAH.

(2) Repeat Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.
The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.
The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

17.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh

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

Bit	Description	Setting Range	R/W
b9 to b0	Specifies the number of block transfer operations or repeat transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

17.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h, DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	DMA Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. DMAC activation source is selected using the DMRSRm registers of the ICU. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

17.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h, DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF flag in DMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the ESIF flag in DMSTS to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in DMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DTIF bit in DMSTS to 0.

17.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h, DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 17.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 17.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed. 0 1: Offset addition*1 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note 1. Offset addition can be specified only for DMAC0.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, or when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in DMINT set to 1. Table 17.2 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 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, or when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in DMINT set to 1. Table 17.2 lists the settings and the corresponding extended repeat areas.

SM[1:0] Bit (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address. Offset addition can be specified only for DMAC0.

Table 17.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA4 to SARA0 or DARA4 to DARA0	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

17.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

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

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

17.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch, DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE Bit (DMA Transfer Enable)

When the DMST bit in DMAST is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

17.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh

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

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b (DMA activation source is software).

Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

17.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh, DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

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

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in DMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in DMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in DMINT is set to 1 and the SARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in DMINT is set to 1 and the DARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in DMCNT.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer))
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT

ACT Flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

17.2.12 DMA Request Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh, DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt flag of the activation source to 0. 1: At the end of transfer, the interrupt flag of the activation source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DISEL Bit (Interrupt Select)

This bit selects whether the interrupt flag of the activation source of the DMAC is cleared to 0 or issues an interrupt to the CPU, at the beginning of transfer.

When DMTMD.DCTG[1:0] = 00b (activation by software), the setting of the DISEL bit does not affect the operation.

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

DMST Bit (DMAC Operation Enable)

When this bit is set to 1, DMAC activation is enabled for all channels.

When 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC activation is enabled), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is cleared to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

17.3 Operation

17.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL of DMACm. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting DMCRB of DMACm is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 17.3 summarizes the register update operation in normal transfer mode, and Figure 17.2 shows the operation in normal transfer mode.

Table 17.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
DMACm.DMCRAL	Transfer count	Decremented by one/not updated (in free running mode)
DMACm.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACm.DMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for DMAC0.

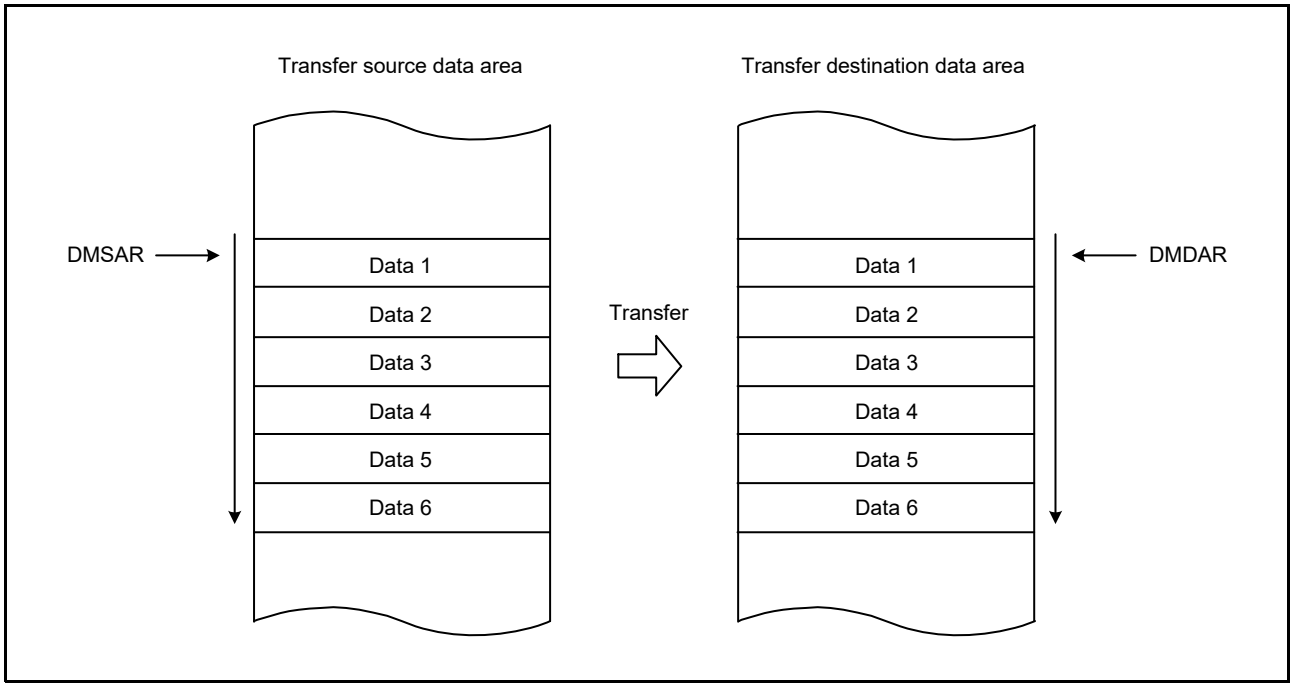


Figure 17.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using DMCRA of the DMACm.
A maximum of 1K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count of repeat transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 17.4 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.4 Register Update Operation in Repeat Transfer Mode

		Update Operation after Completion of a Transfer by One Transfer Request	
Register	**Function**	**When DMACm.DMCRAL is not 1**	**When DMACm.DMCRAL is 1 (Transfer of the Last Data in Repeat Size)**
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer count	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

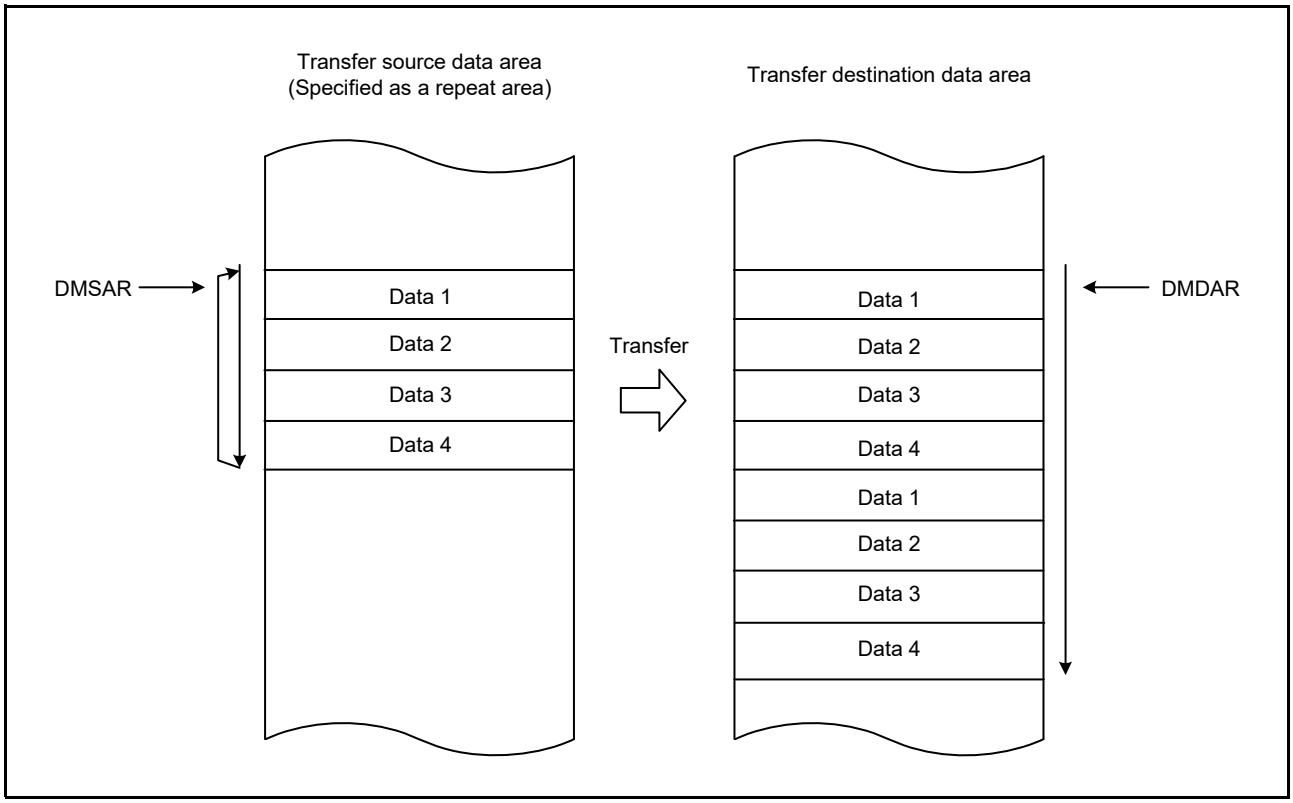


Figure 17.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.
A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACm.
A maximum of 1K can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count of block transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.
Table 17.5 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

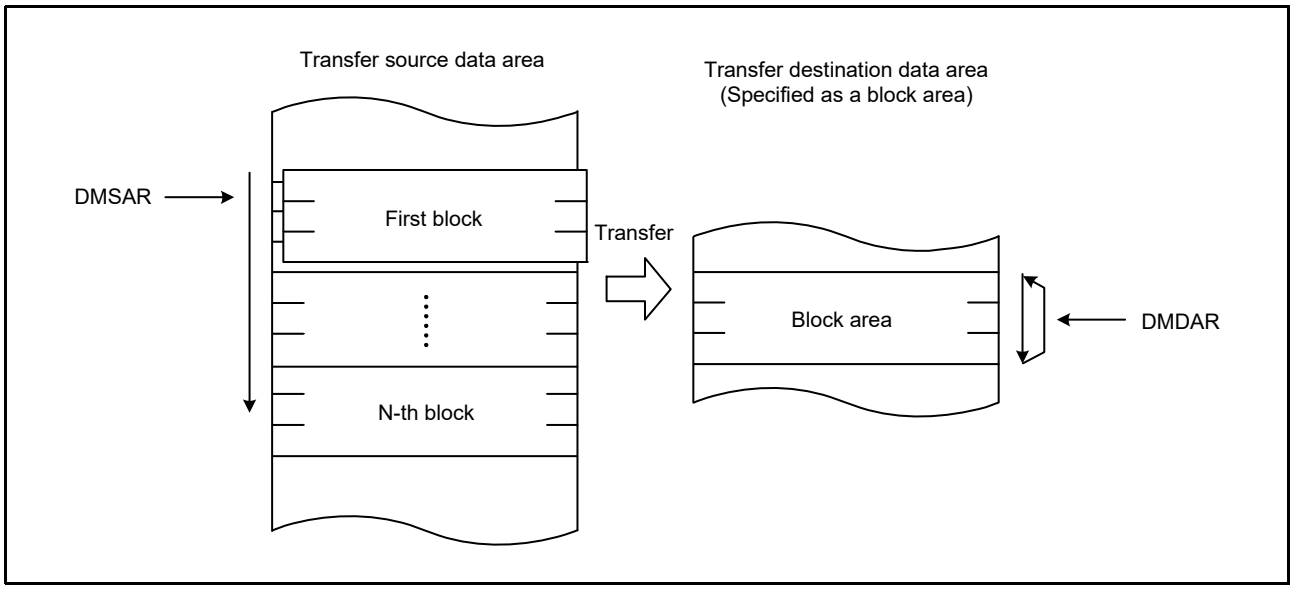


Figure 17.4 Operation in Block Transfer Mode

17.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the SARA[4:0] bits in DMAMD of DMACm. The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMAMD of DMACm. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMINT of DMACm is set to 1, the ESIF flag in DMSTS of DMACm is set to 1 and the DTE bit in DMCNT of DMACm is cleared to 0 to stop DMA transfer. At this time, if the ESIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the DARIE bit in DMINT of DMACm is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the interrupt handling.

Figure 17.5 shows an example of the extended repeat area operation.

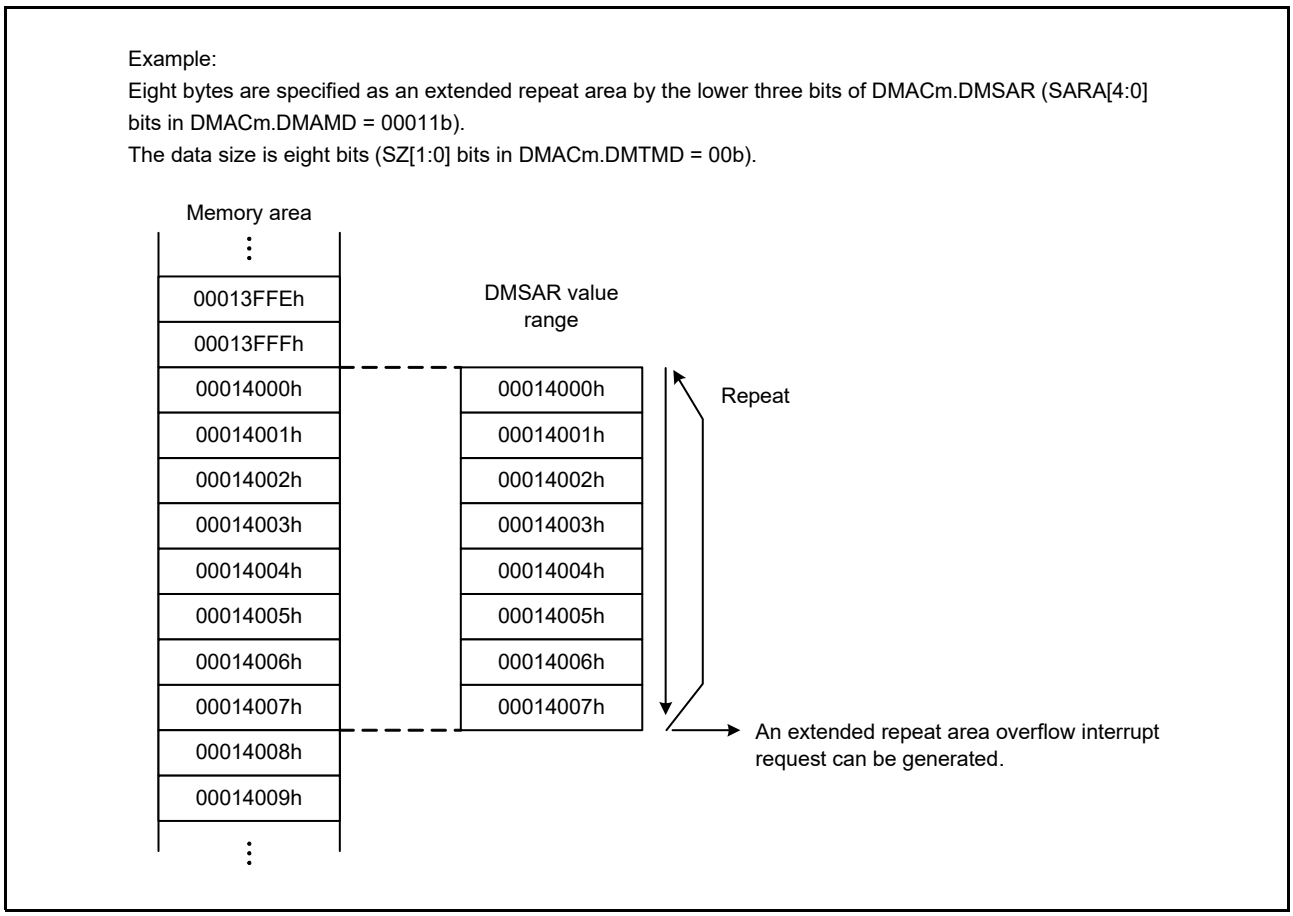


Figure 17.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 17.6 shows an example when the extended repeat area function is used in block transfer mode.

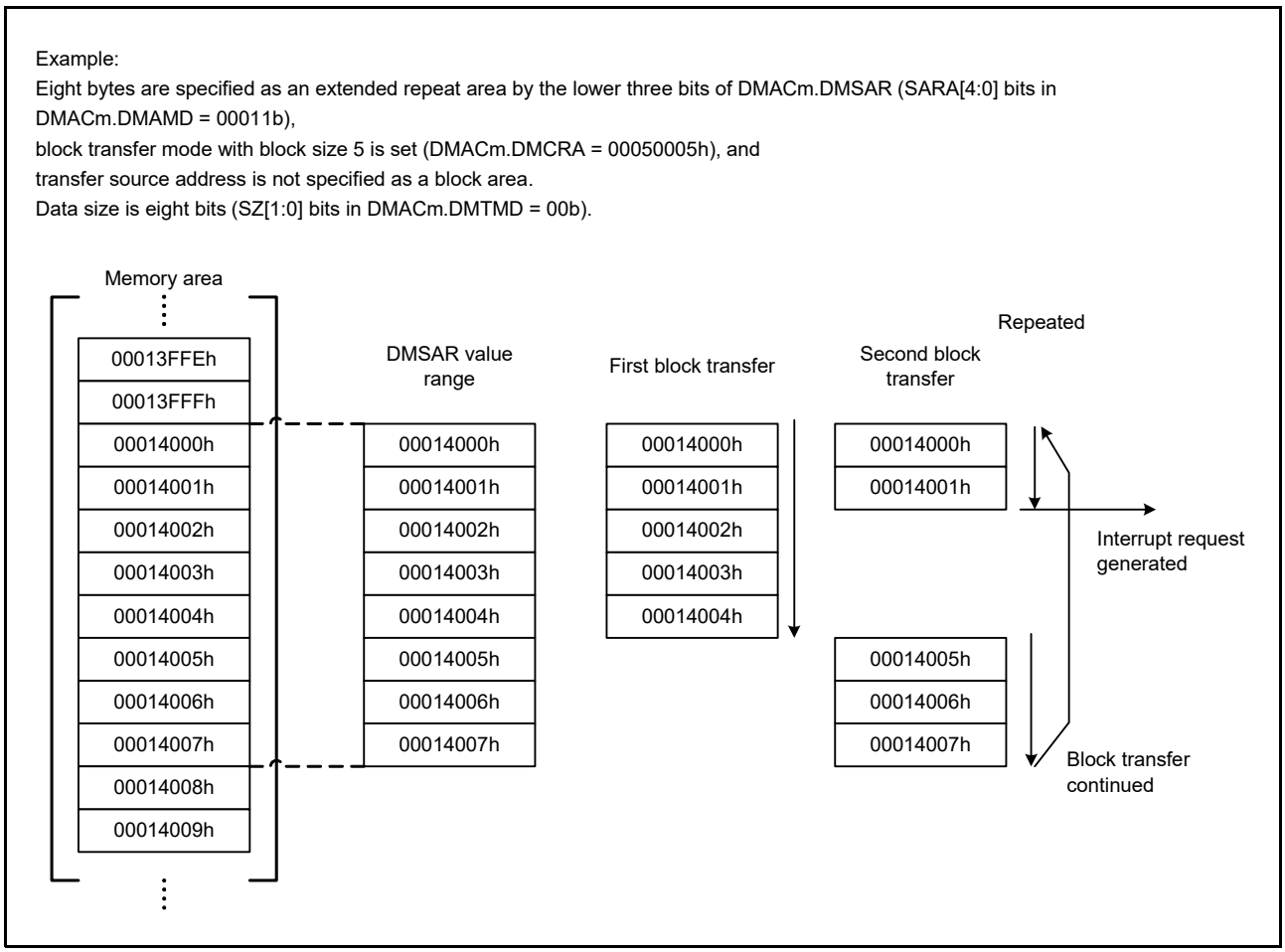


Figure 17.6 Example of Extended Repeat Area Function in Block Transfer Mode

17.3.3 Address Update Function Using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR of DMAC0) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR of DMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 17.6 shows the address update method in each address update mode.

Table 17.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD of DMACm)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 17.7 shows an example of address updating using offset addition.

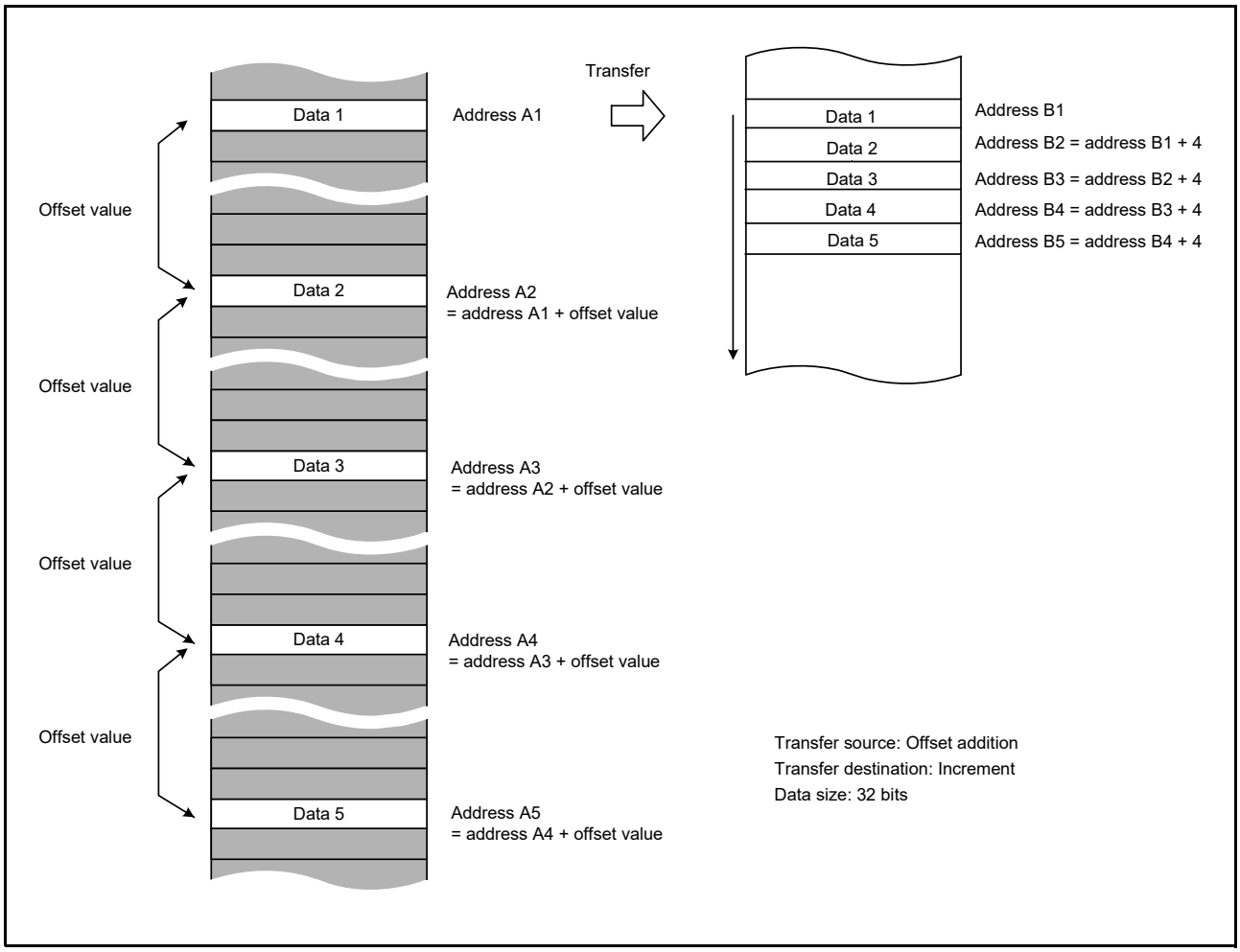


Figure 17.7 Example of Address Updating by Offset Addition

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: Transfer source address update mode: Offset addition
- DMAC0.DMAMD: Transfer destination address update mode: Destination address is incremented.
- DMAC0.DMTMD: Transfer data size select: 32 bits
- DMAC0.DMTMD: Transfer mode select: Repeat transfer
- DMAC0.DMTMD: Repeat area select: The source is specified as the repeat area.
- DMAC0.DMOFR: Offset address: 10h
- DMAC0.DMCRA: Repeat size: 4h
- DMAC0.DMINT: The repeat size end interrupt is enabled.

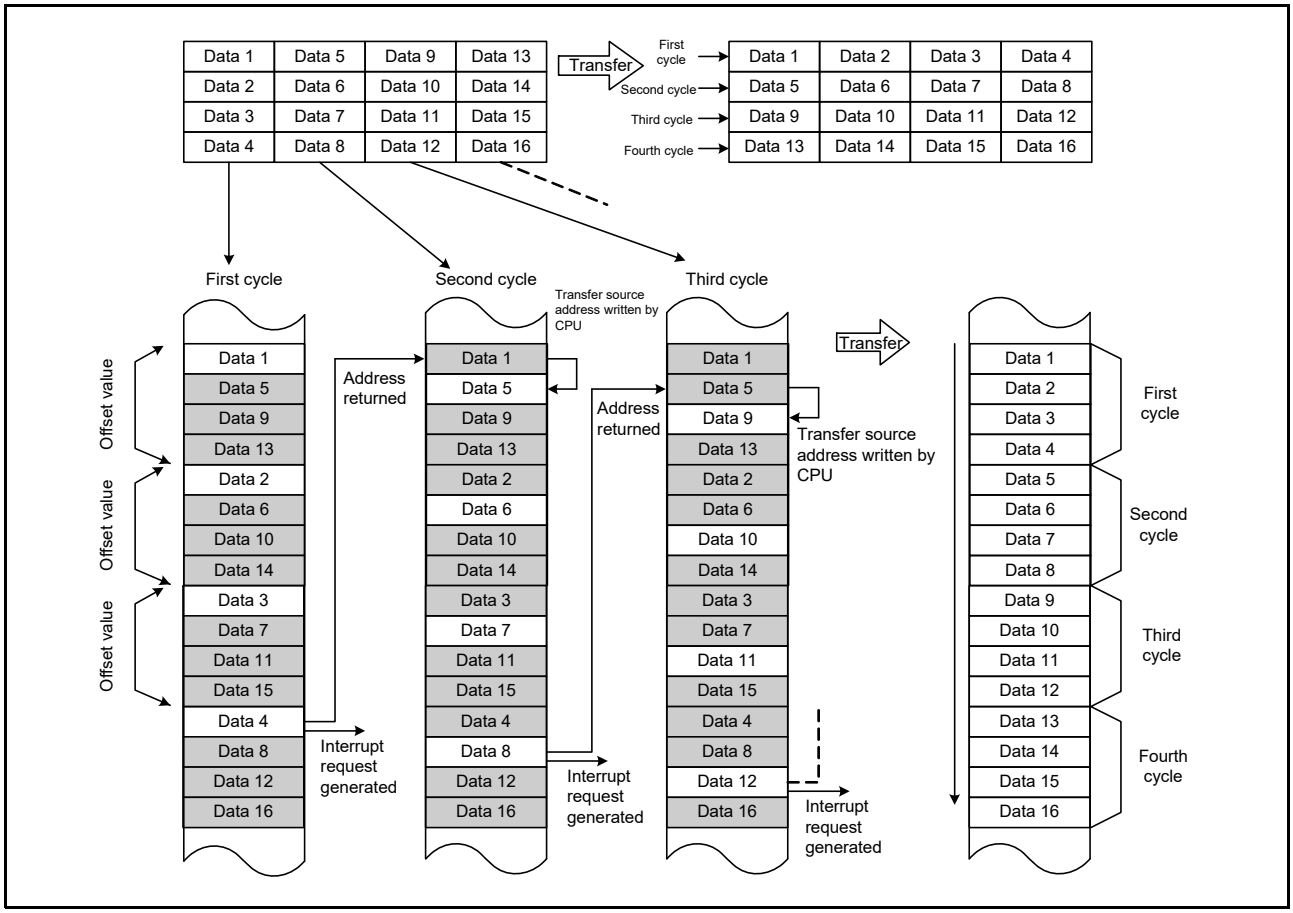


Figure 17.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMAC0.DMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMAC0.DMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 17.9 shows a flowchart of the XY conversion.

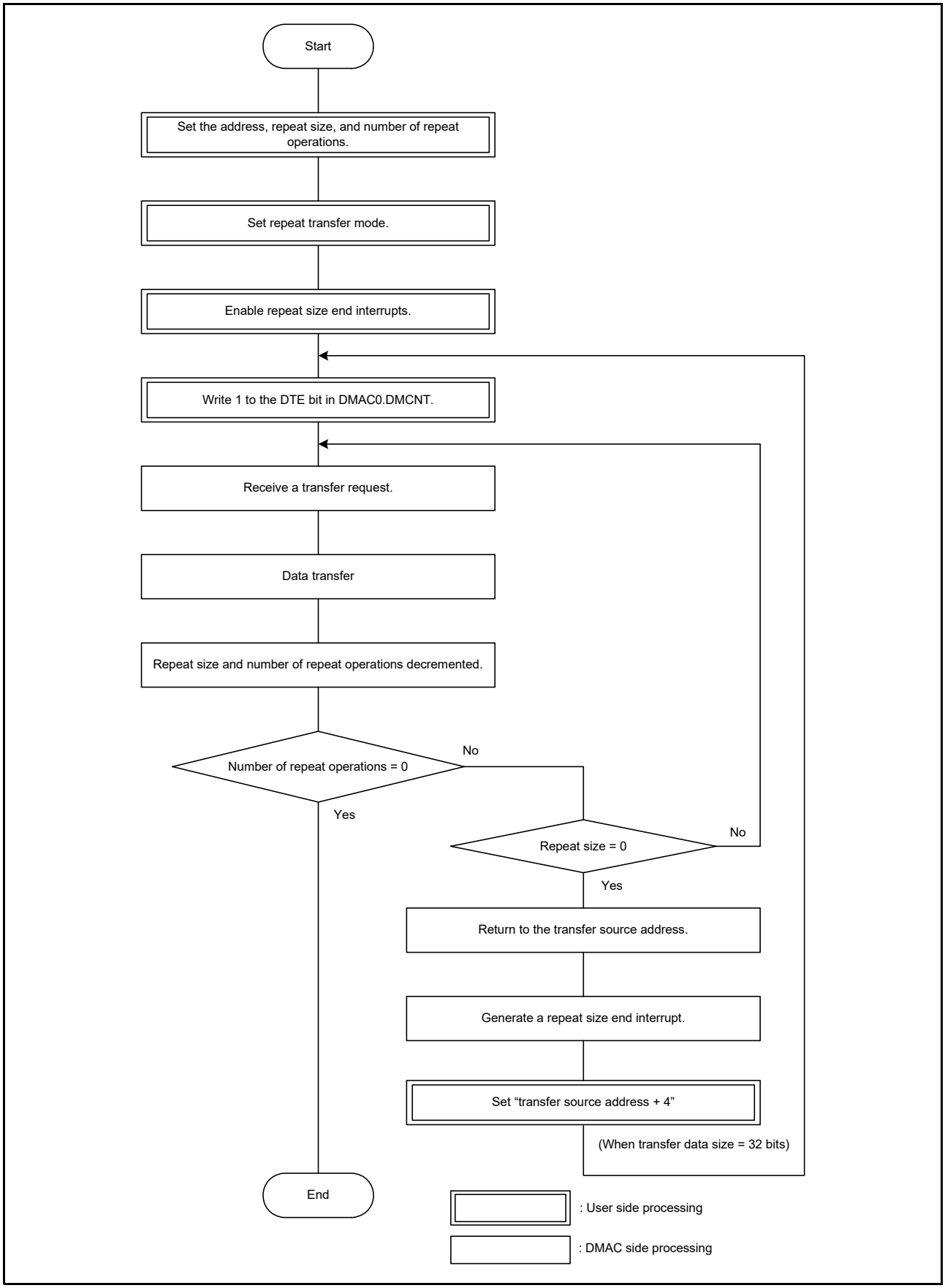


Figure 17.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

17.3.4 Activation Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMAC activation sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the activation source.

(1) DMAC Activation by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the DMAC activation by software.
To start DMA transfer by software, set the DCTG[1:0] bits in DMTMD of DMACm to 00b, and then set the DTE bit in DMCNT of DMACm to 1 (DMA transfer is enabled) and the SWREQ bit in DMREQ of DMACm to 1 (DMA transfer is requested) with the DMST bit in DMAST set to 1 (DMAC activation enabled).
When the DMAC is activated by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is cleared to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) DMAC Activation by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMAC activation sources. The activation source can be selected separately for each channel using the DMRSRm registers (m = 0 to 3) of the ICU.
The DMAC is activated when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DCTG[1:0] bits in DMTMD of DMACm is set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DTE bit in DMCNT of DMACm is set to 1 (DMA transfer is enabled), and the DMST bit in DMAST is set to 1 (DMAC activation is enabled).

For interrupt requests specified as DMAC activation sources, see Table 14.3, Interrupt Vector Table, in section 14, Interrupt Controller (ICUb).

17.3.5 Operation Timing

Figure 17.10 and Figure 17.11 show DMAC operation timing examples.

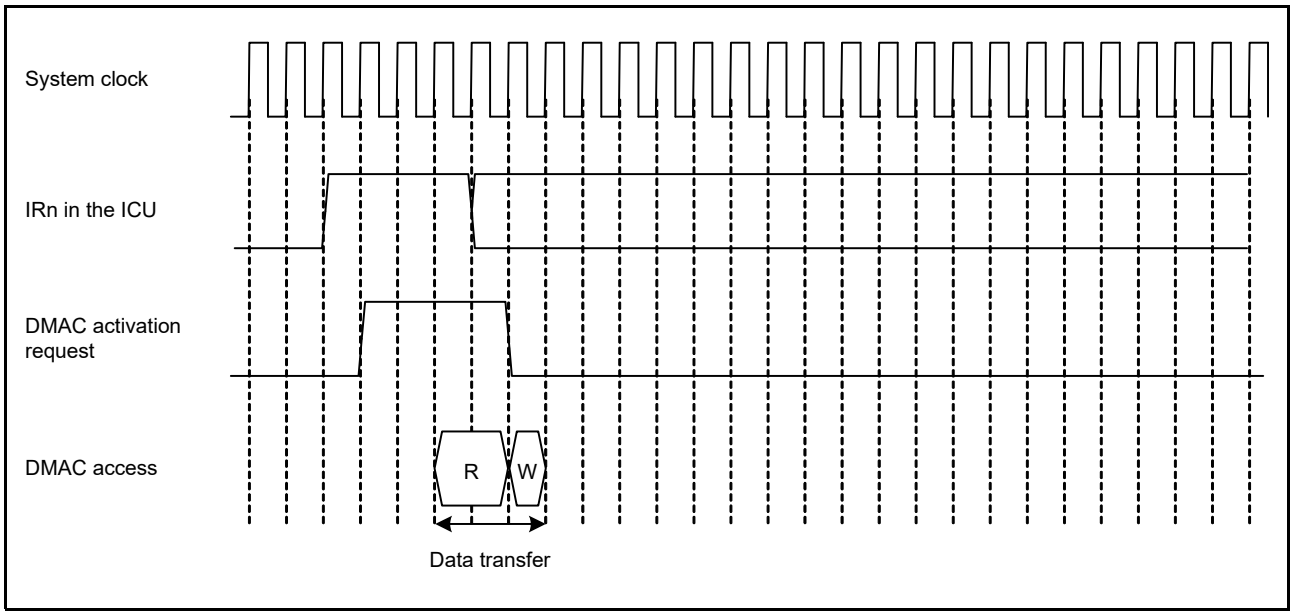


Figure 17.10 DMAC Operation Timing Example (1) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

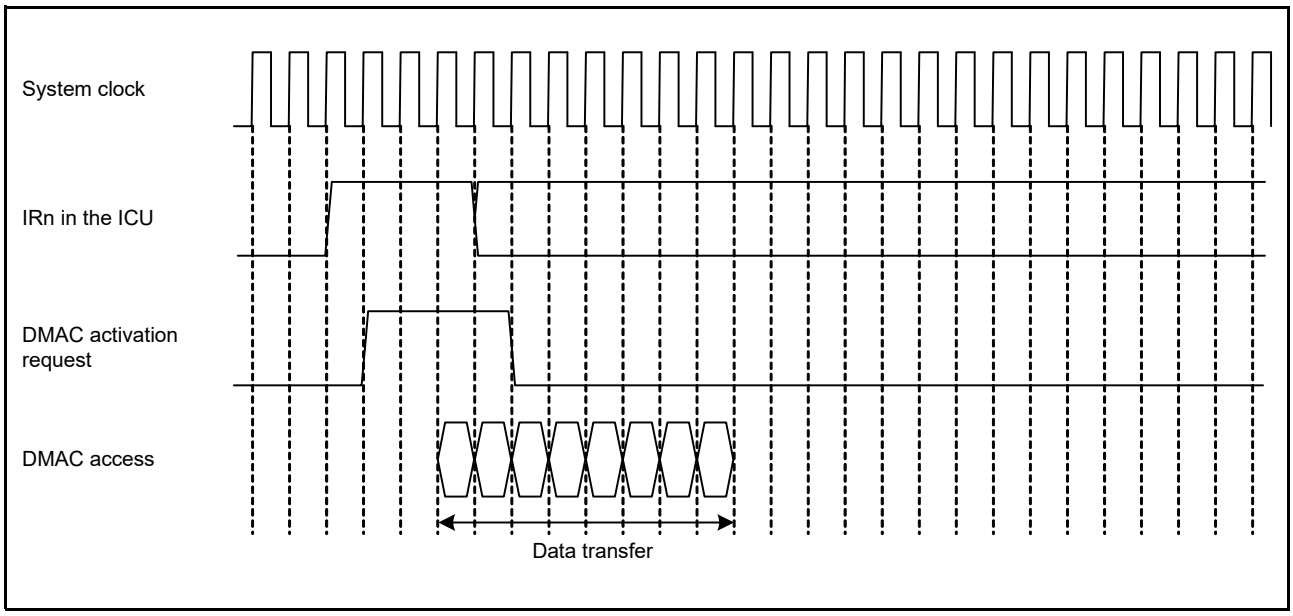


Figure 17.11 DMAC Operation Timing Example (2) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

17.3.6 DMAC Execution Cycles

Table 17.7 lists execution cycles in one DMAC data transfer operation.

Table 17.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 45, RAM, section 46, Flash Memory (FLASH), and section 5, I/O Registers.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, see section 17.3.5, Operation Timing.

17.3.7 Activating the DMAC

Figure 17.12 shows the register setting procedure.

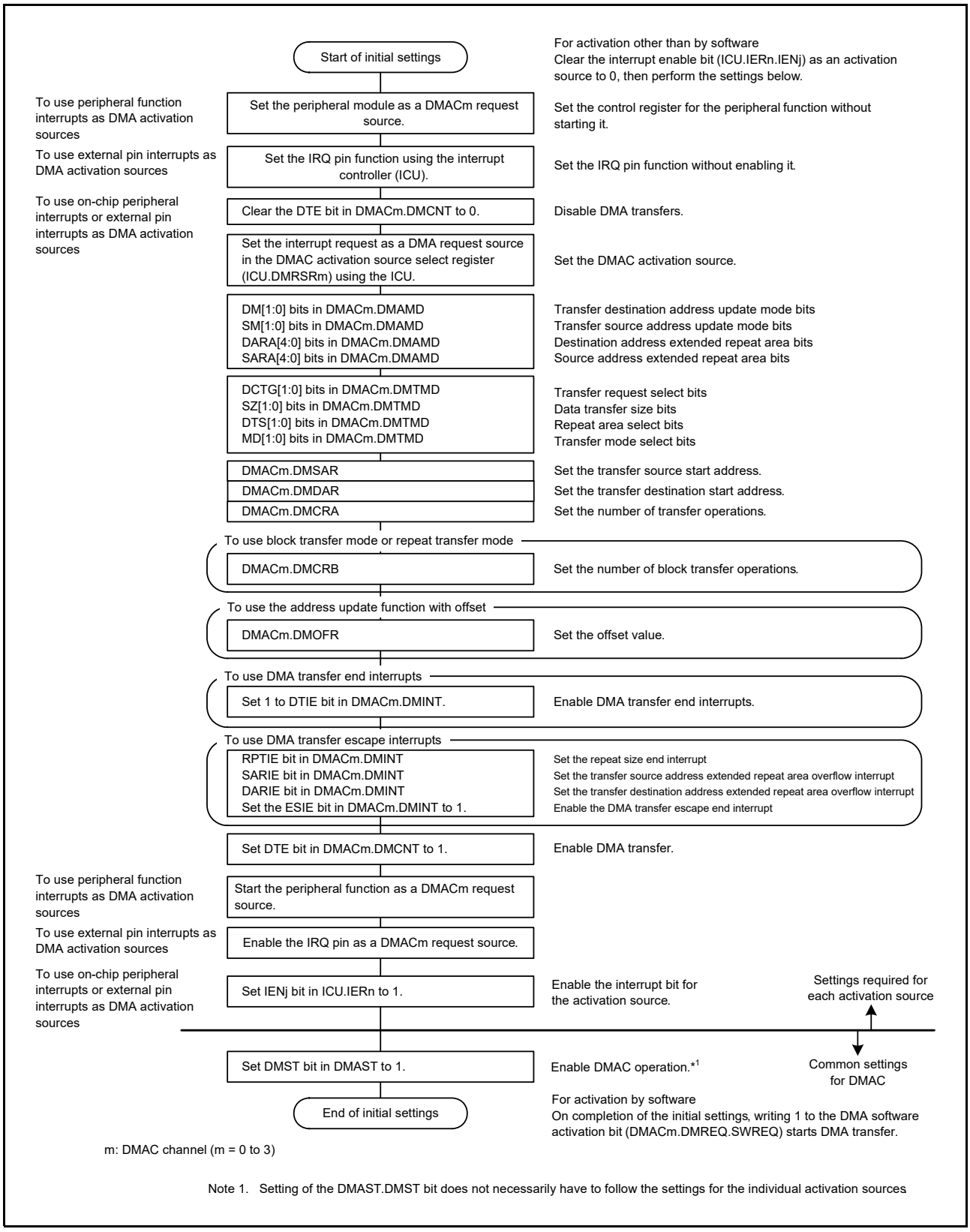


Figure 17.12 Register Setting Procedure

17.3.8 Starting DMA Transfer

Setting the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled) and setting the DMST bit in DMAST to 1 (DMAC start enabled) enable DMA transfer of channel m (m = 0 to 3).
Another activation request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the ACT bit in DMSTS of DMACm is set to 1 (the DMAC is in the active state).

17.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

(1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.
The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMACm.DMCNT.DTE bit is set to 1 is prohibited (except for DMACm.DMCNT). In this case, writing must be performed after the bit is cleared to 0.

(6) DMA Active Flag (DMACm.DMSTS.ACT)

The ACT bit in DMSTS of DMACm indicates whether the DMACm is in the idle or active state.
This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DTE bit in DMCNT of DMACm during DMA transfer, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMSTS of DMACm is set to 1 after DMA transfer of the total transfer size of data is completed.
When both this flag and the DTIE bit in DMINT of DMACm are set to 1, a transfer end interrupt is requested.
This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in DMSTS of DMACm is cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMACm is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMSTS of DMACm is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMINT of DMACm are set to 1, a transfer escape end interrupt is requested.
This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in DMSTS of DMACm is cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMACm is set to 1 during an interrupt handling.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUb).

17.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.
The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 (channel 0: highest).
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

17.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMSTS of DMACm are changed from 1 to 0, indicating that DMA transfer has ended.

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMCRAL of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUb).

17.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in DMINT of DMACm is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in DMCNT of DMACm is cleared to 0 and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm.
A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUb).

17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in DMCNT of DMACm is cleared to 0, and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUb).

17.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 17.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 17.13 shows the schematic logic diagram of interrupt outputs. Figure 17.14 shows the DMAC interrupt handling routine to resume or terminate DMA transfer.

Table 17.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

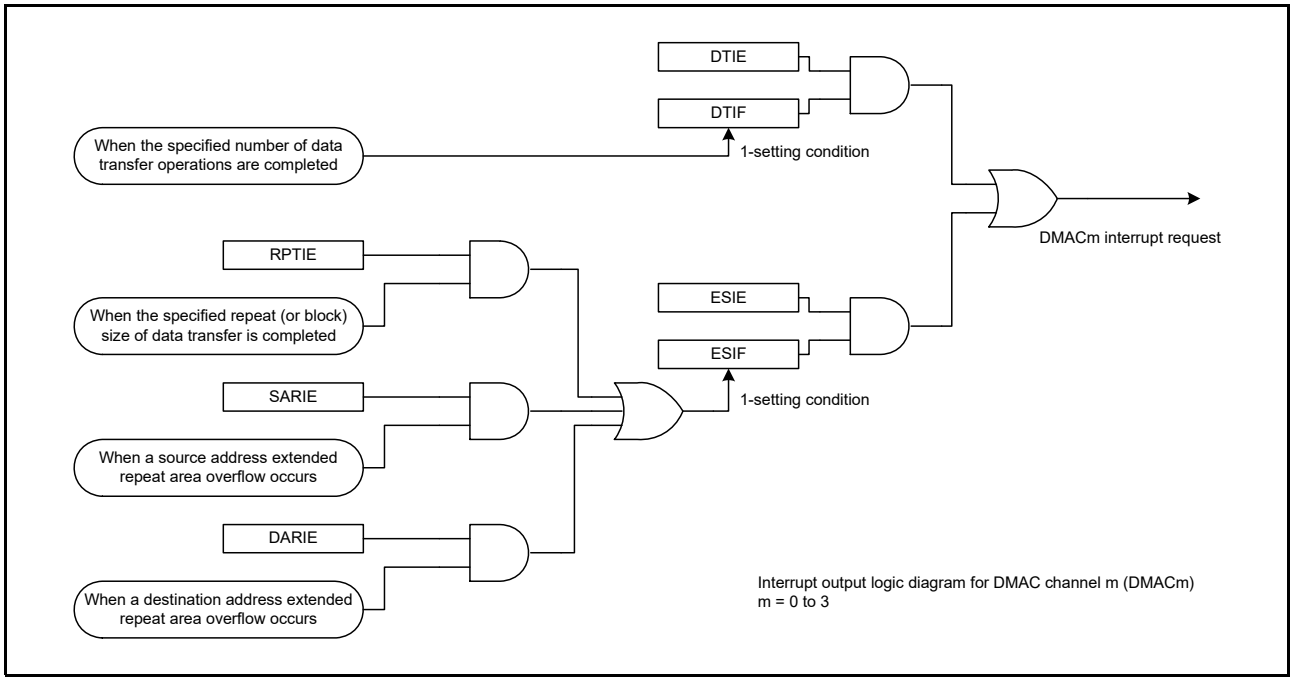


Figure 17.13 Schematic Logic Diagram of Interrupt Outputs

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF bit in DMSTS of DMACm to clear a transfer end interrupt, and to the ESIF bit in DMSTS of DMACm to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACm remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in DMCNT of DMACm. The ESIF bit in DMSTS of DMACm is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

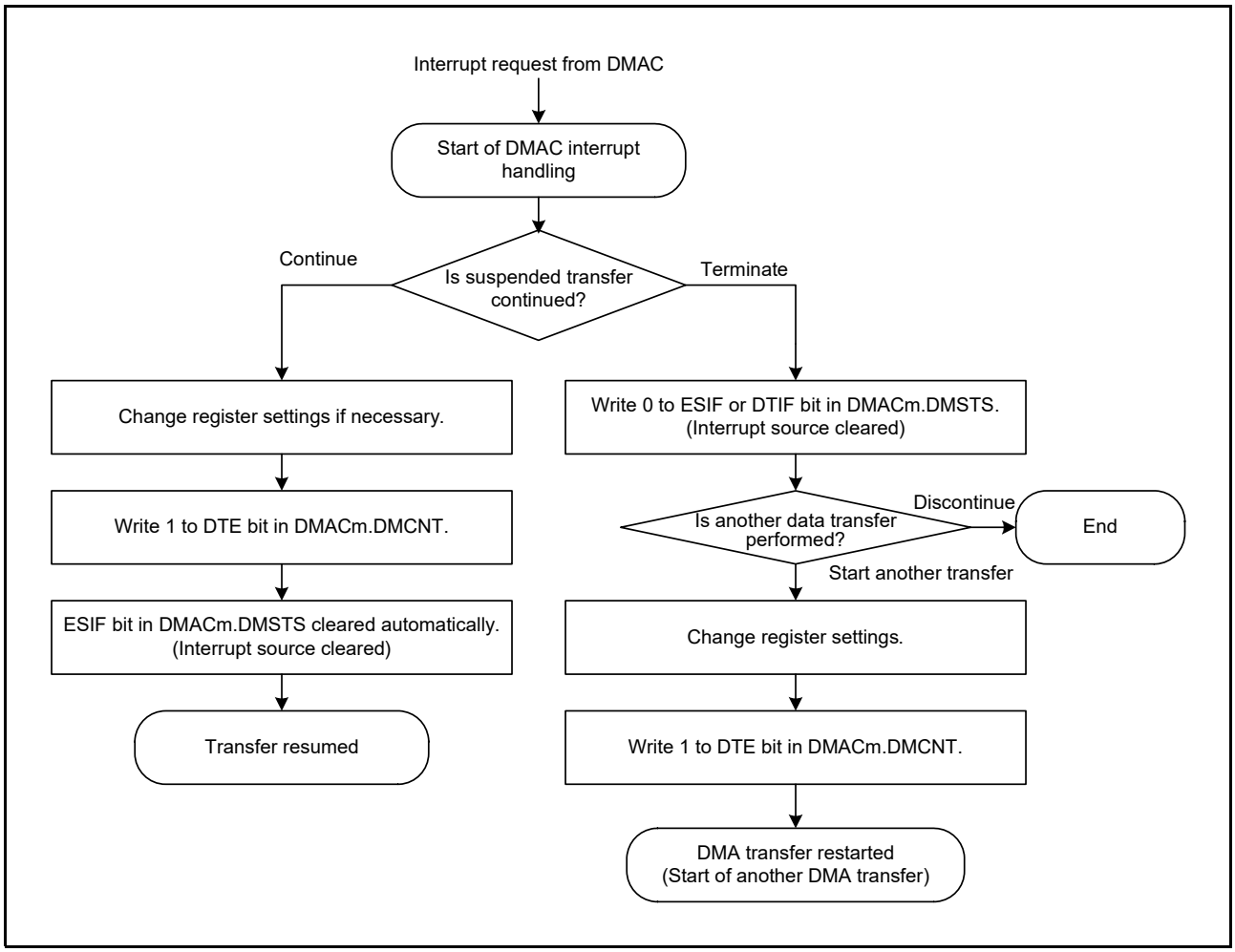


Figure 17.14 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

17.6 Event Link Function

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 internal peripheral bus, an event link request signal is generated when the write to the write buffer is accepted.

17.7 Low Power Consumption Function

Before transition to the module stop state or software standby mode, clear the DMAST.DMST bit to 0 (DMAC activation is disabled), and then perform the following.

(1) Module Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMAC. If DMA transfer is in progress at the time 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) Software Standby

Make settings in accord with the procedure under section 11.6.3.1, Entry to 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 follows the completion of DMA transfer.

(3) Note on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.7.6, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform DMA transfer after returning from low power consumption mode, set the DMST bit in DMAST to 1 again.
To use a request that is generated in 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 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

17.8 Usage Notes

17.8.1 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

17.8.2 Access to the Registers during DMA Transfer

The DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm must not be accessed while the ACT bit in DMSTS of the same channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the same channel is set to 1 (DMA transfer enabled).

17.8.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

17.8.4 Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt is issued to the CPU at the end of each transfer that has been activated by one DMA request. Unlike the transfer end interrupt that the DMAC outputs or the escape end interrupt, the interrupt of this type is issued to the CPU at the end of DMA transfer without clearing the interrupt flag of the DMAC activation source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt flag is not cleared to 0 at the end of DMAC transfer, it should be cleared to 0 by the CPU interrupt routine.
The interrupt flag is cleared when the CPU interrupt is accepted.
For the change of the settings on the interrupt flag or the interrupt request destination, see section 14, Interrupt Controller (ICUb). For the DMACm.DMCSL.DISEL bit setting, see section 17.2.12, DMA Request Source Flag Control Register (DMCSL).

17.8.5 Setting of DMAC Activation Source Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC activation source select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.DTCERm) that corresponds to the same vector number that has been set by the ICU.DMRSRm register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRm, see section 14, Interrupt Controller (ICUb).

17.8.6 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the interrupt enable bit for the activation source (ICU.IERn.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 17.3.7, Activating the DMAC.

18. 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.

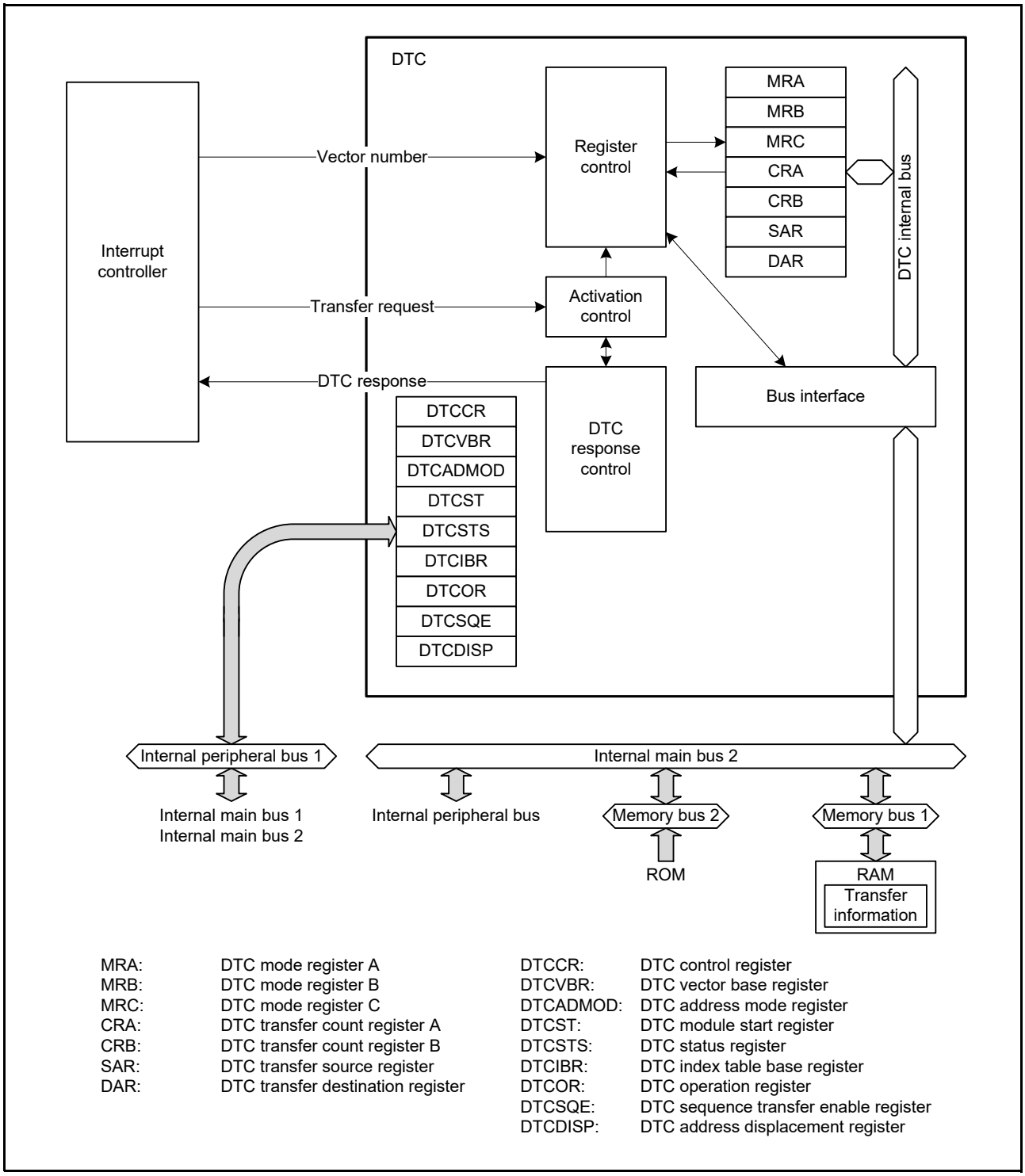


Figure 18.1 DTC Block Diagram

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

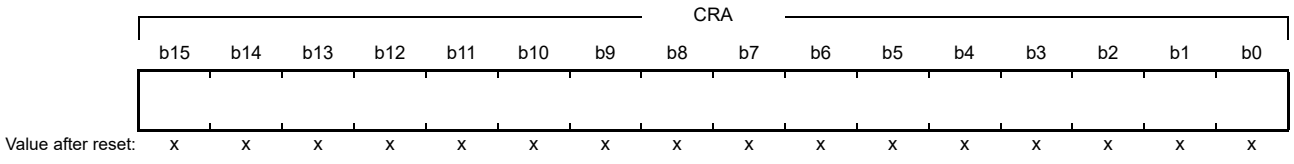
x: Undefined

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

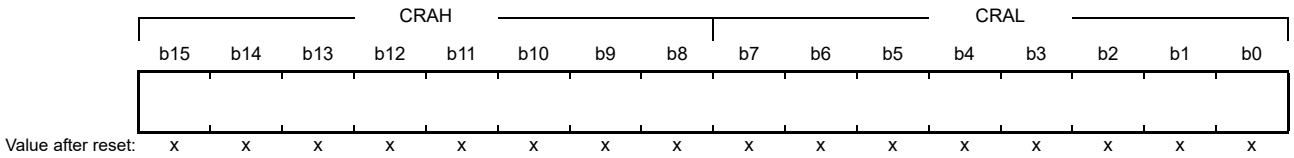
Address(es): (inaccessible directly from the CPU)



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, deep sleep mode, or software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state, deep sleep mode, or software standby mode.

For details on transitions to the module stop state, deep sleep mode, and 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 section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

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.

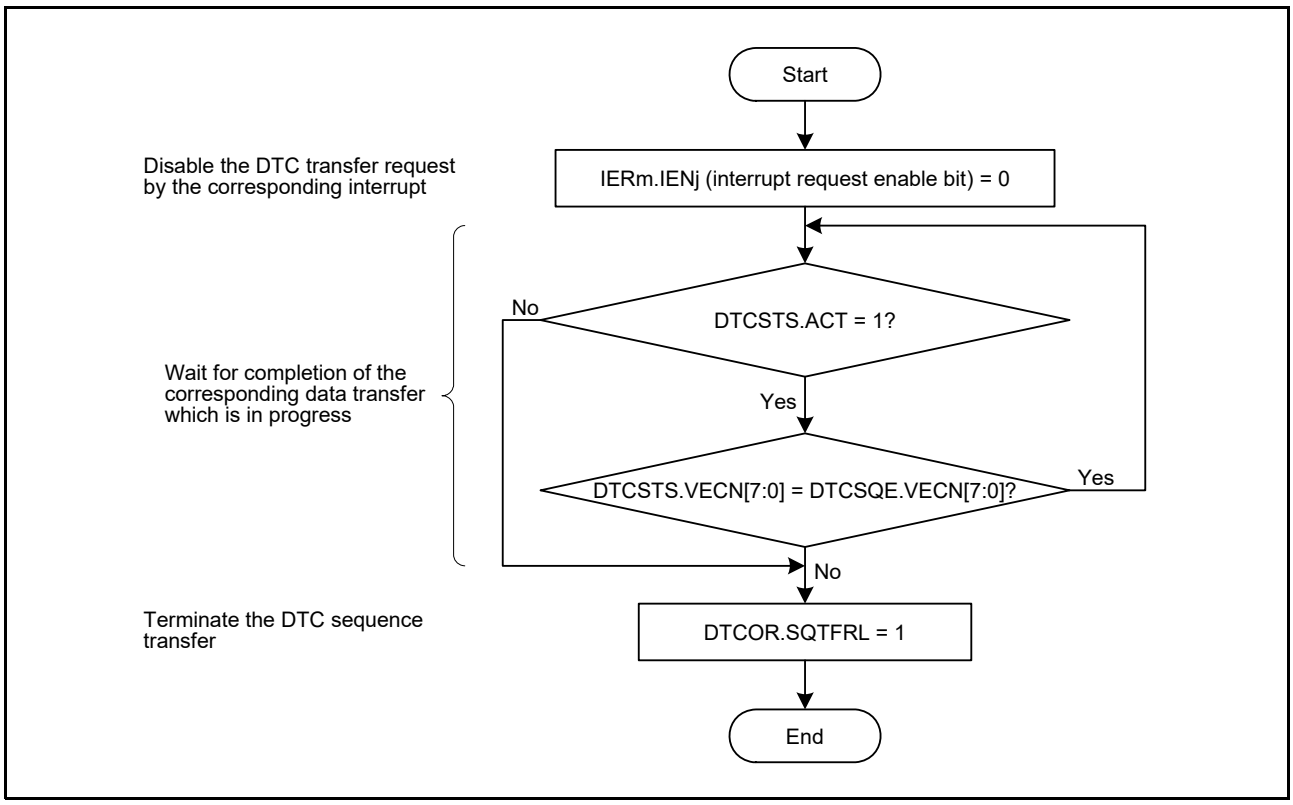


Figure 18.2 Procedure to Terminate Sequence Transfer

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.

section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb) 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 section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb). For request by software, refer to section 14.2.5, Software Interrupt Generation Register (SWINTR) in section 14, Interrupt Controller (ICUb).
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.

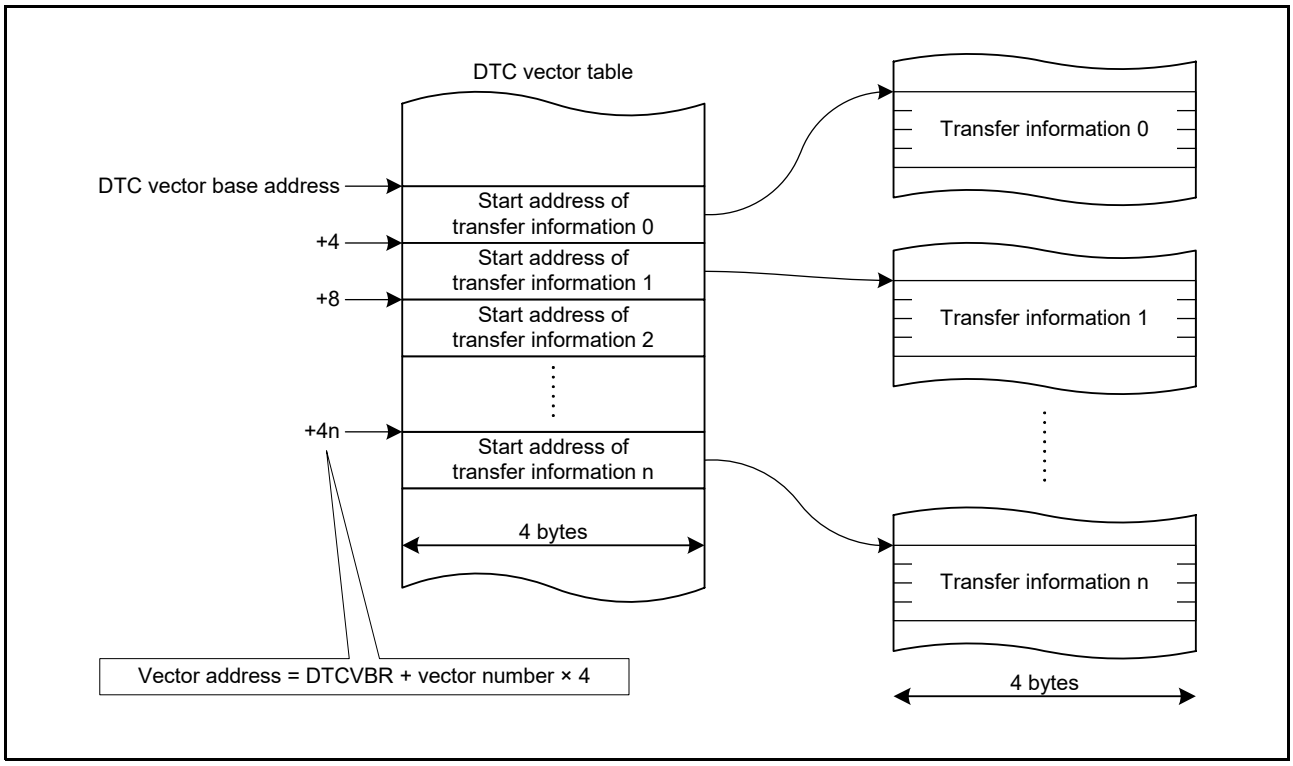


Figure 18.3 DTC Vector Table and Transfer Information

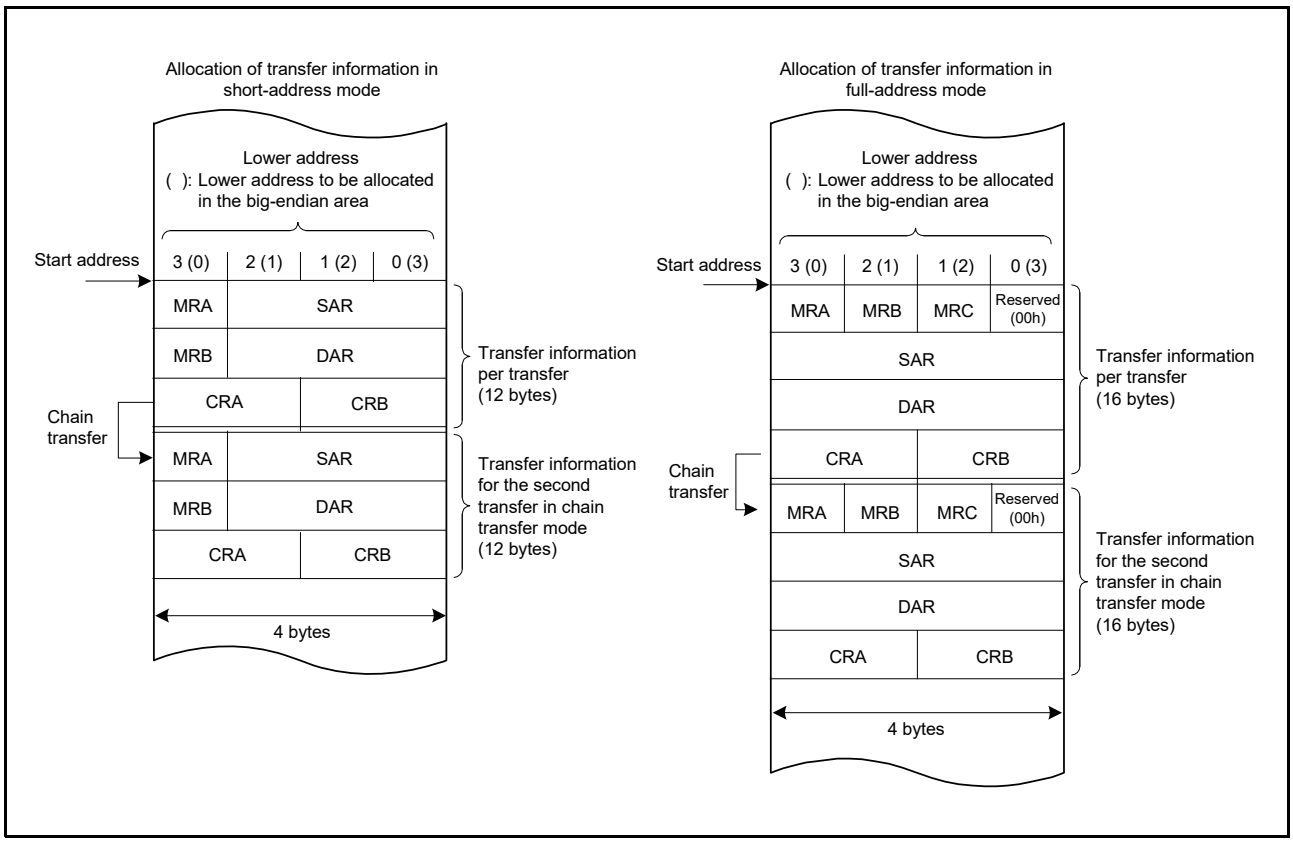


Figure 18.4 Allocation of Transfer Information in the RAM Area

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.

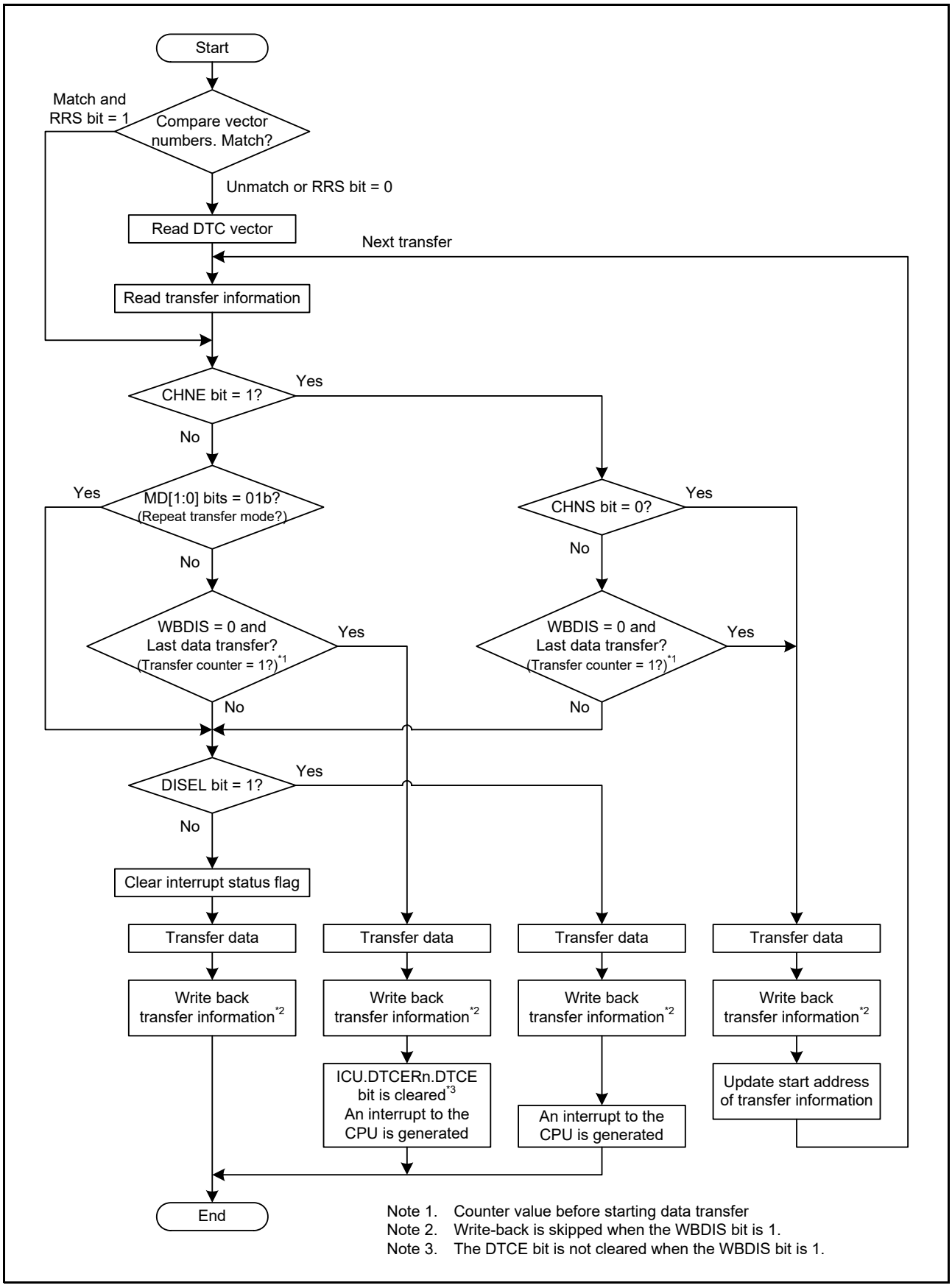


Figure 18.5 Operation Flowchart of the DTC

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.

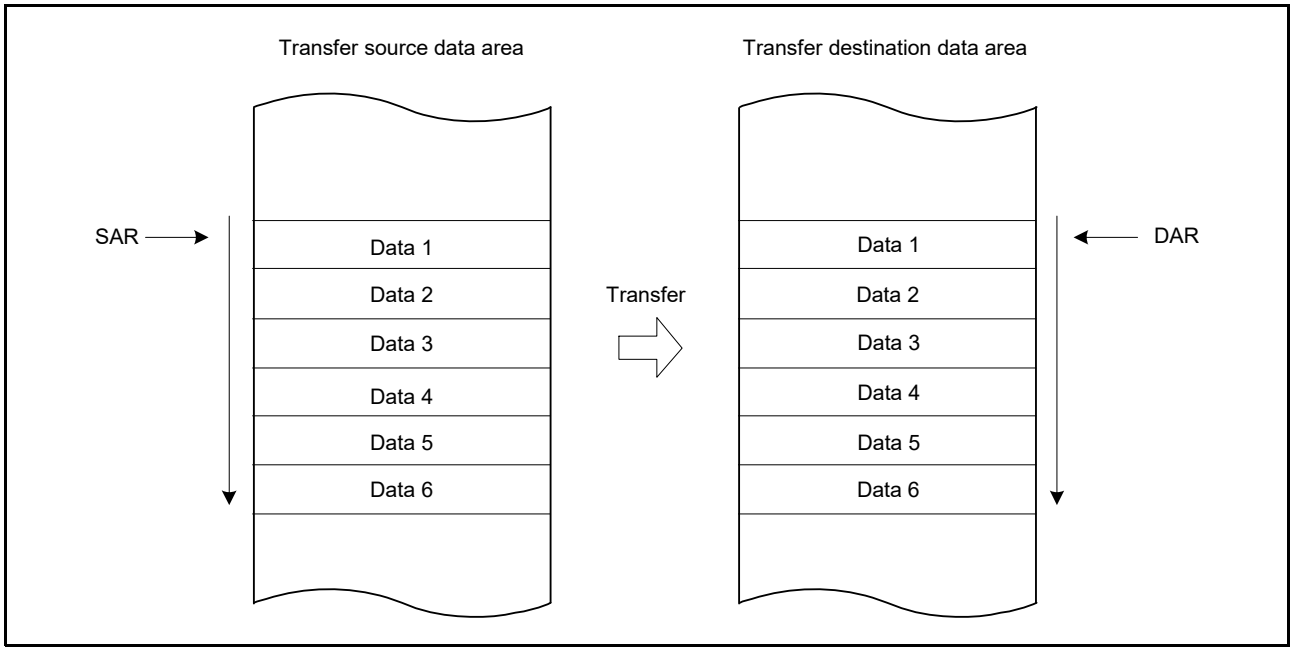


Figure 18.6 Memory Map of Normal Transfer Mode

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.

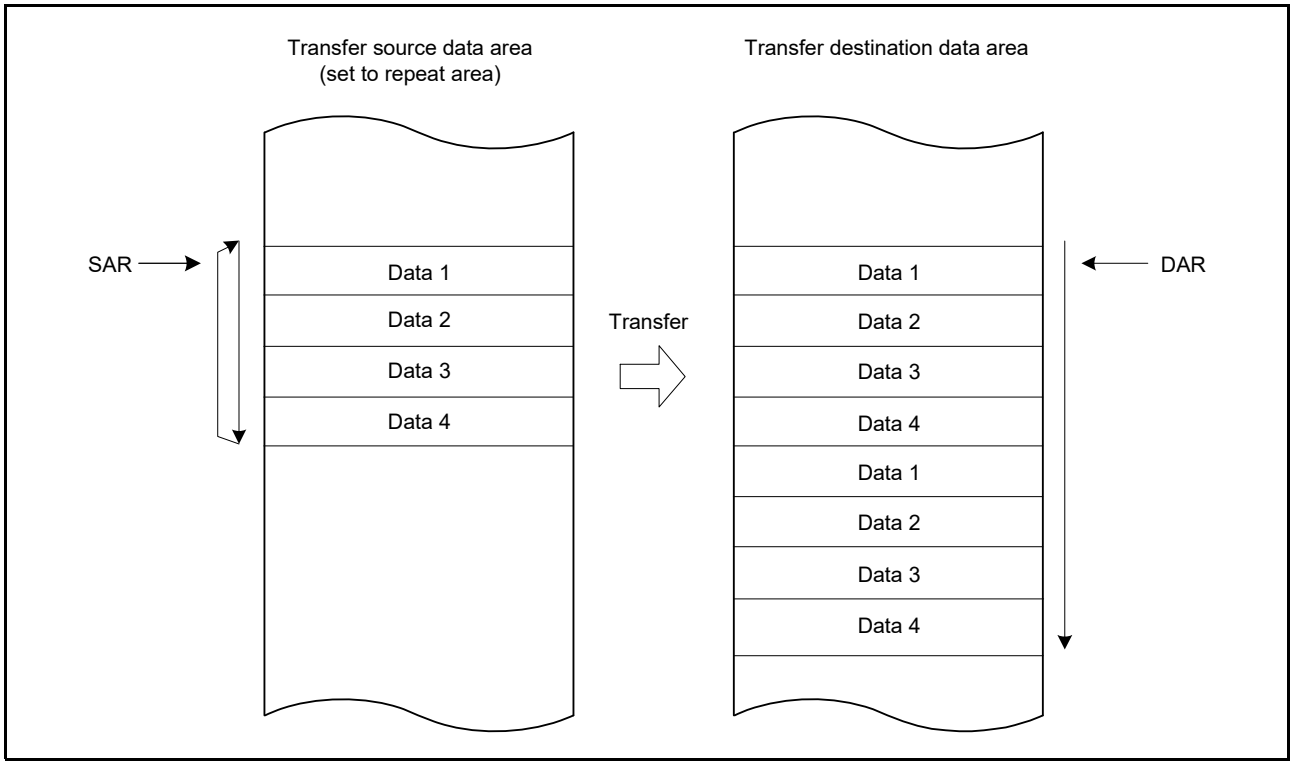


Figure 18.7 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

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.

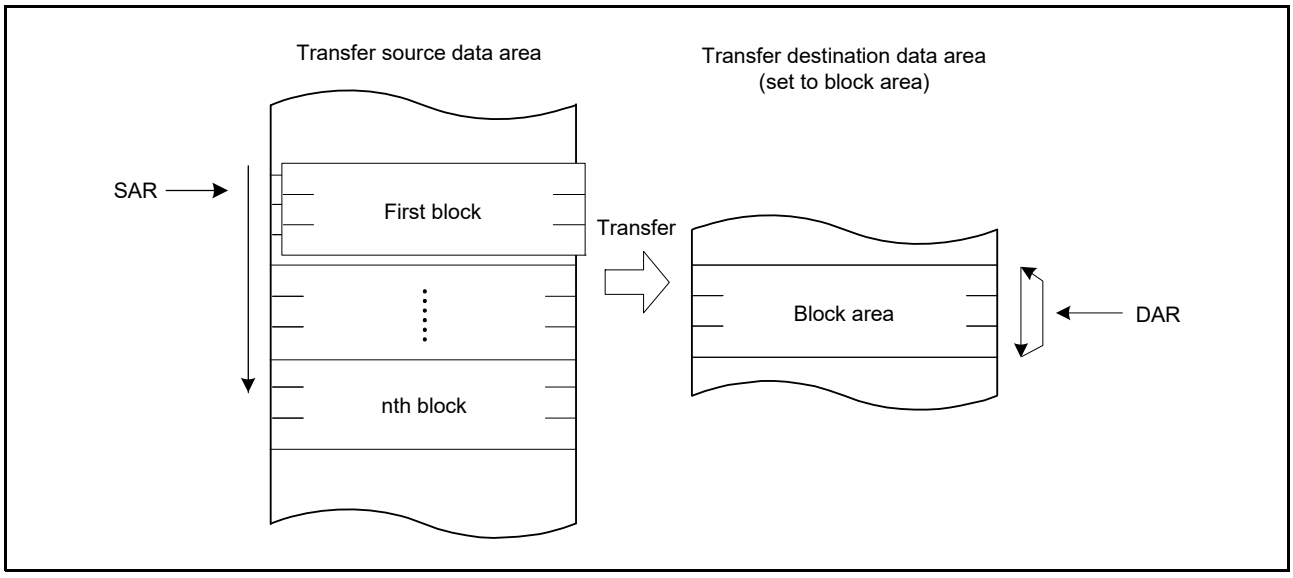


Figure 18.8 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

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.

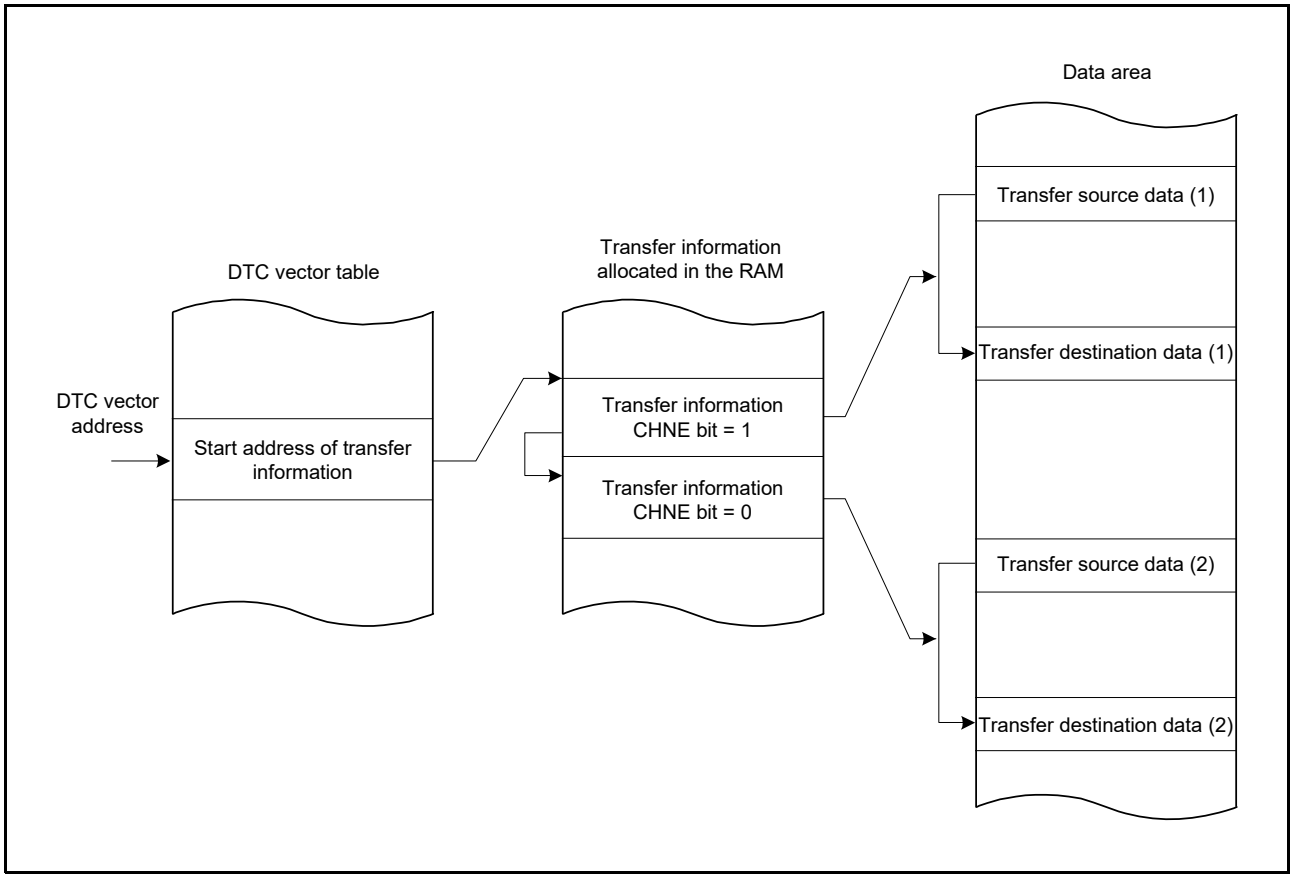


Figure 18.9 Chain Transfer Operation

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.

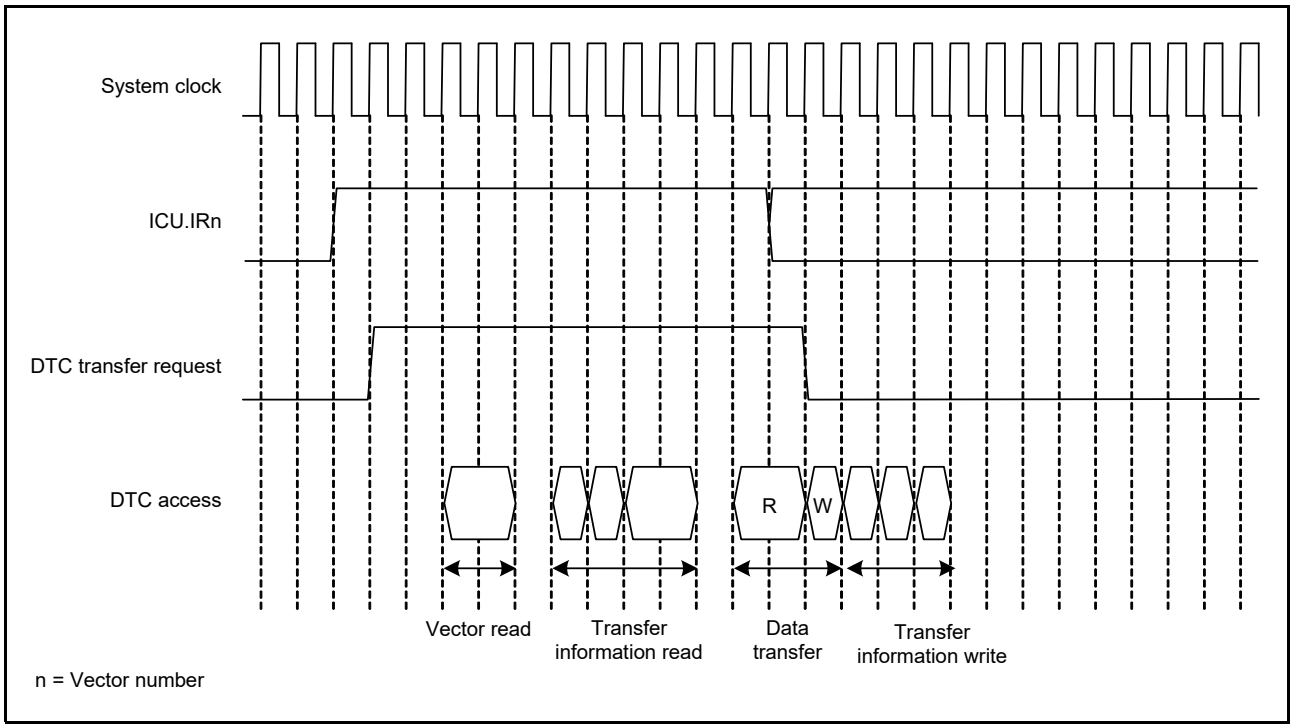


Figure 18.10 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

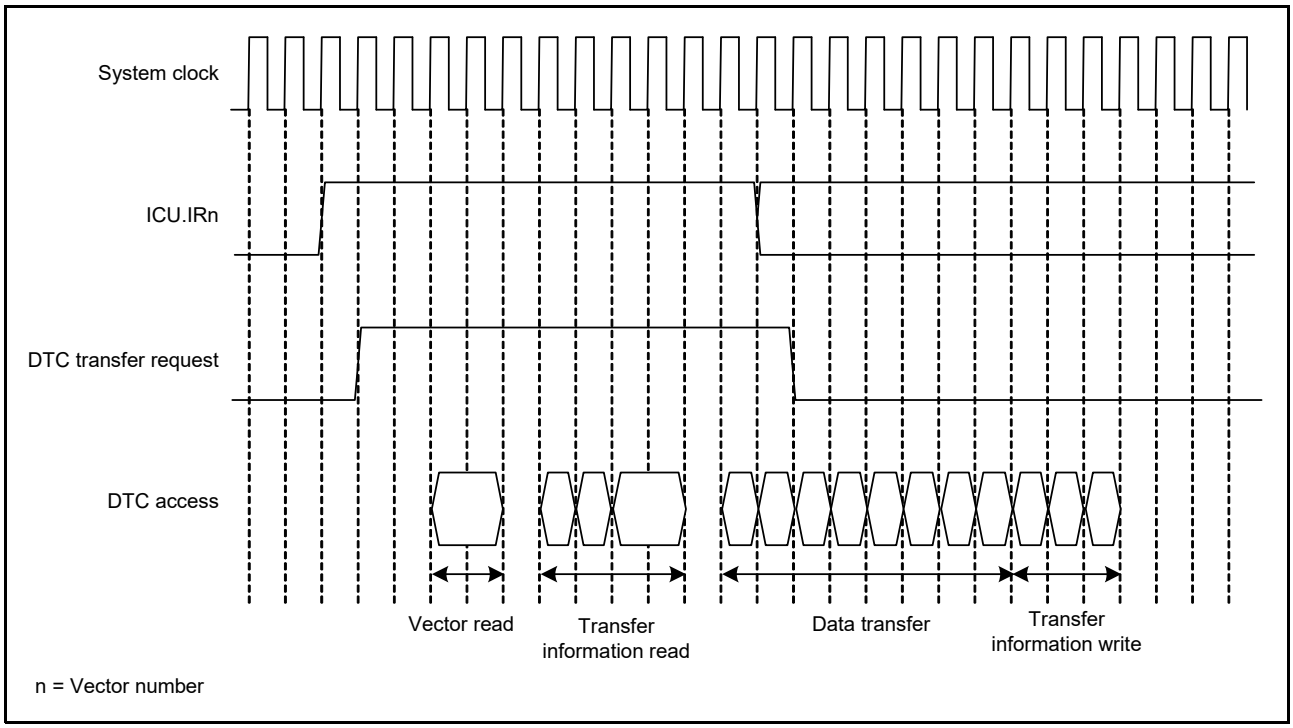


Figure 18.11 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

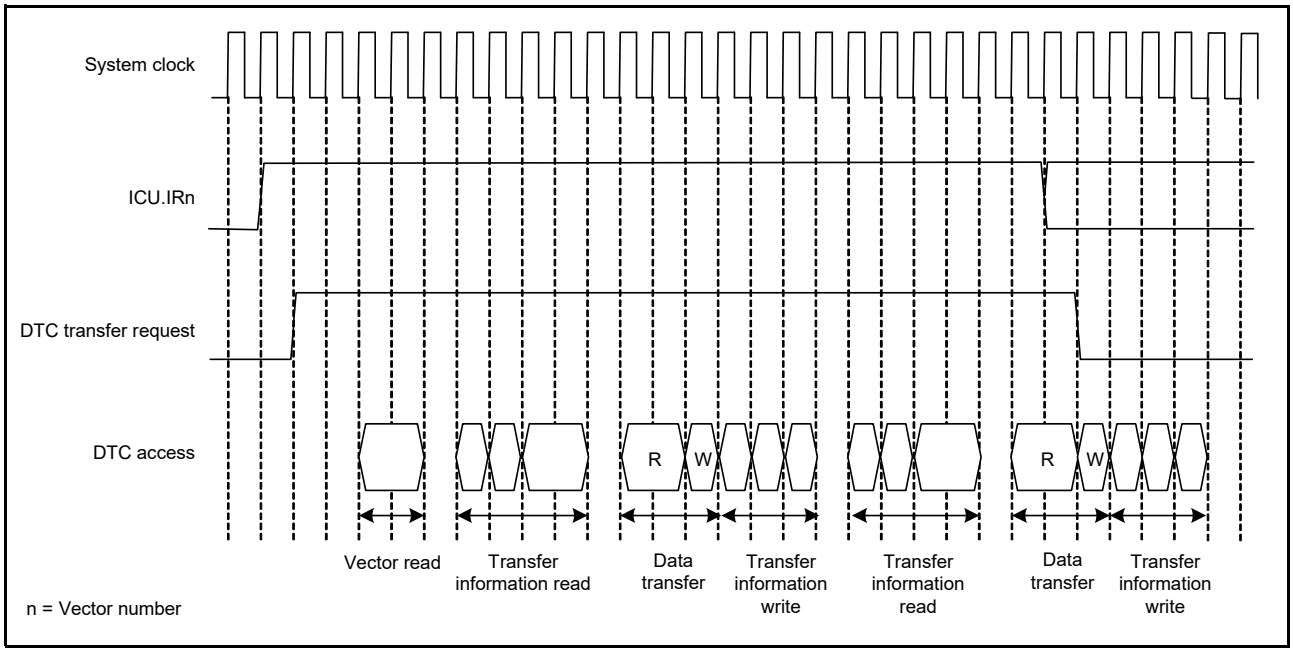


Figure 18.12 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

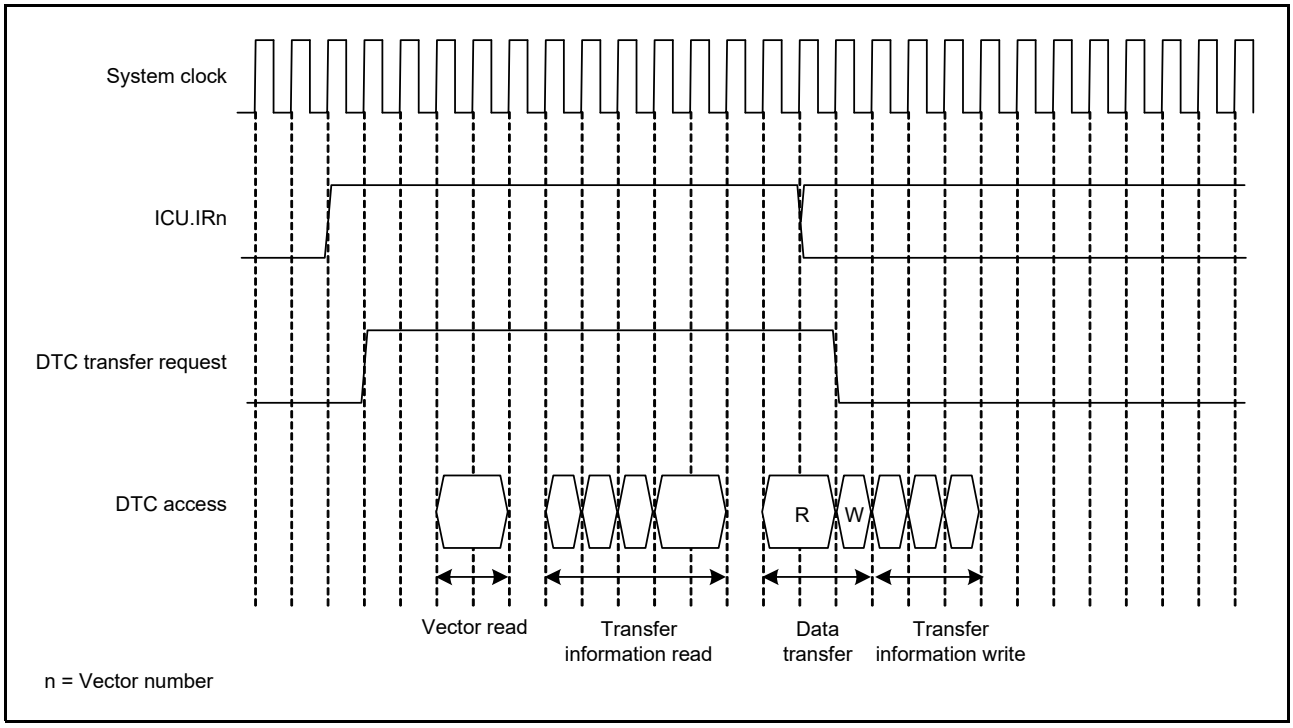
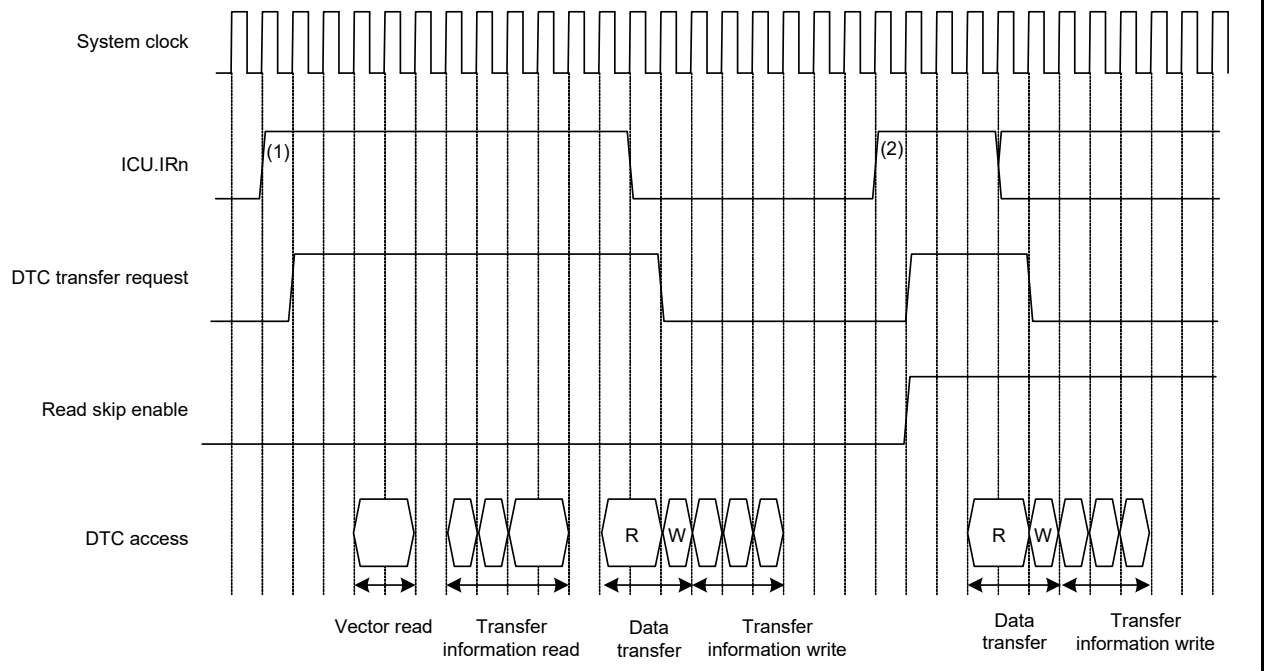


Figure 18.13 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



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 45, RAM, section 46, Flash Memory (FLASH), and section 5, I/O Registers.)

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 (ICUb) for the conditions where the DTCE bit becomes 1.

Figure 18.15 and Figure 18.16 shows a basic sequence transfer operation.

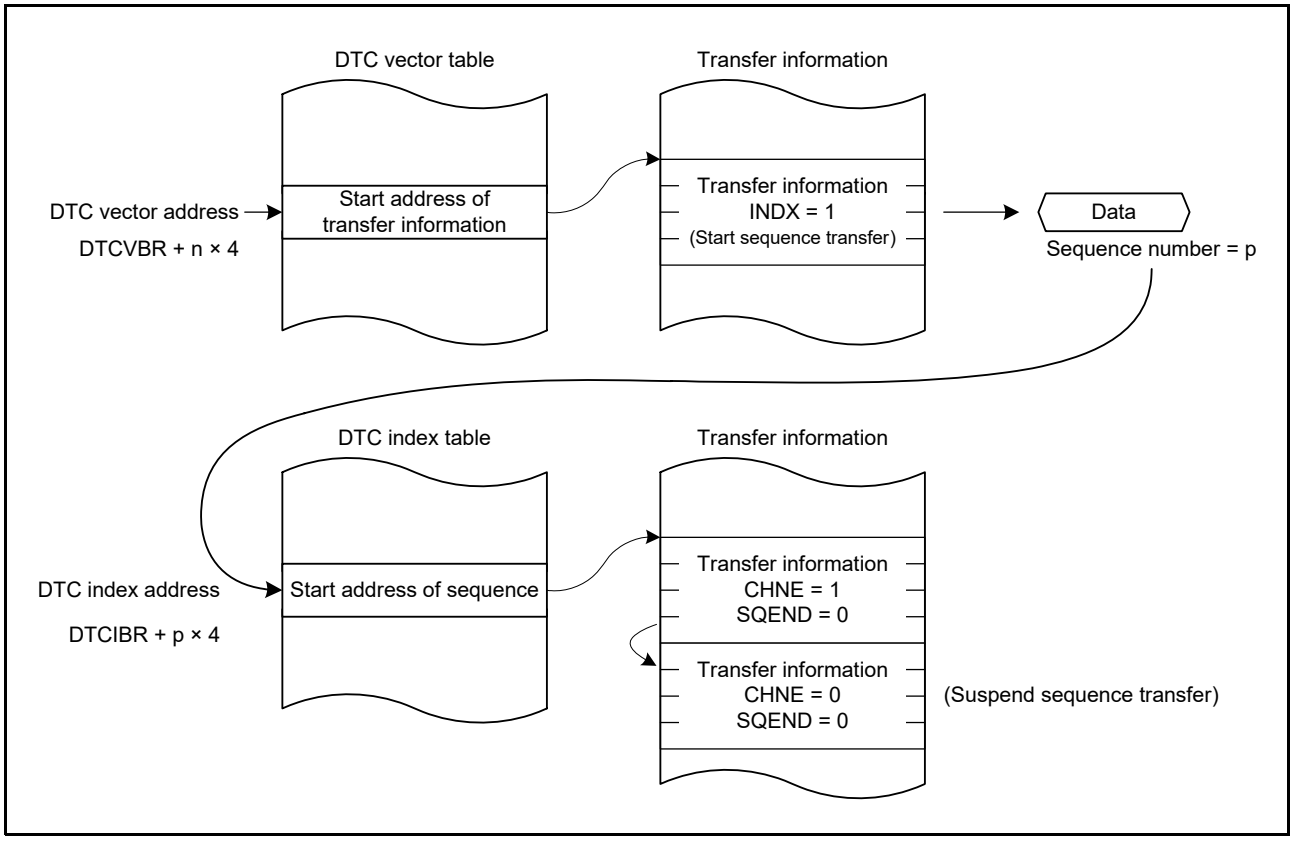


Figure 18.15 Start and Suspension of Sequence Transfer

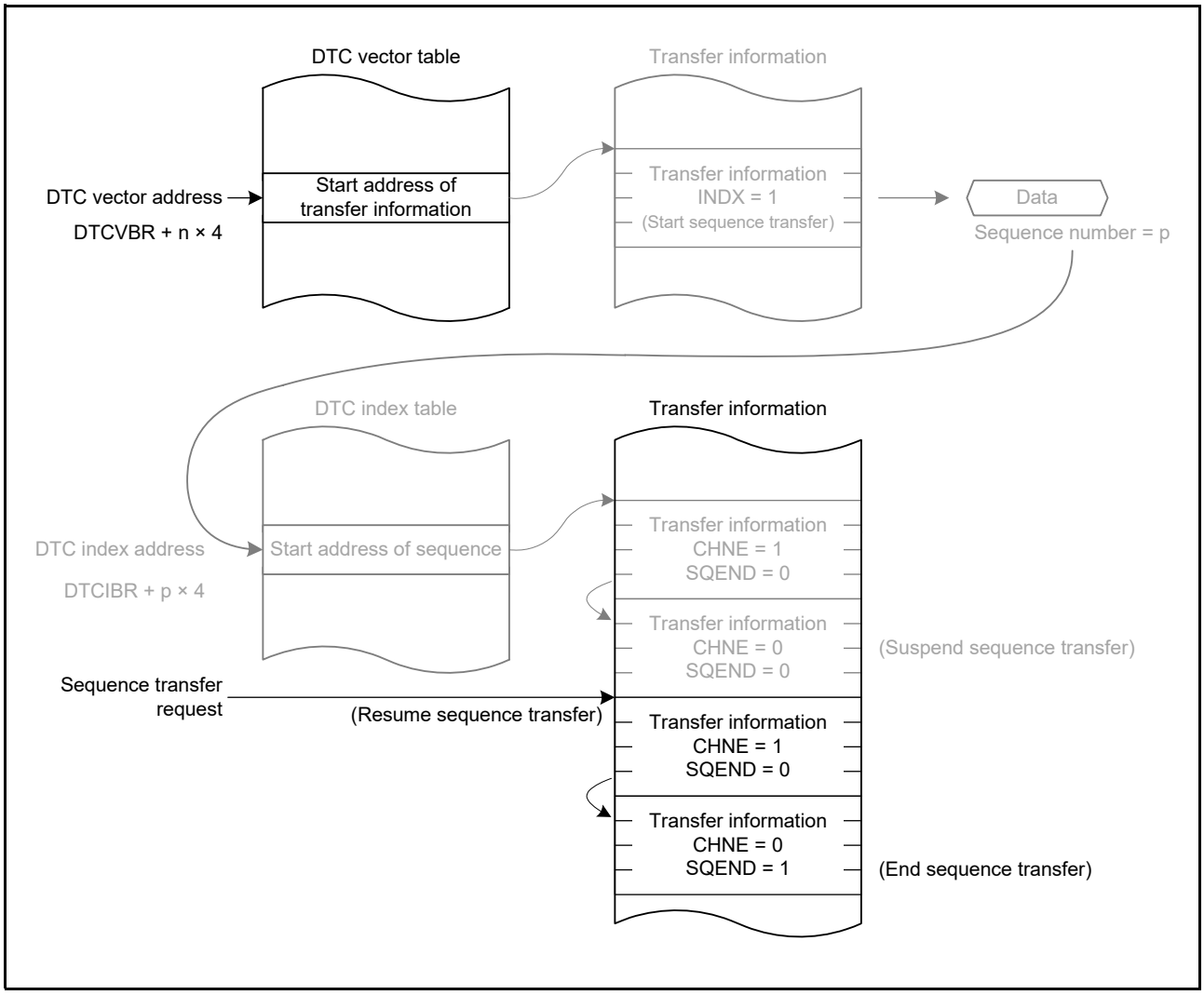


Figure 18.16 Resumption and End of Sequence Transfer

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.

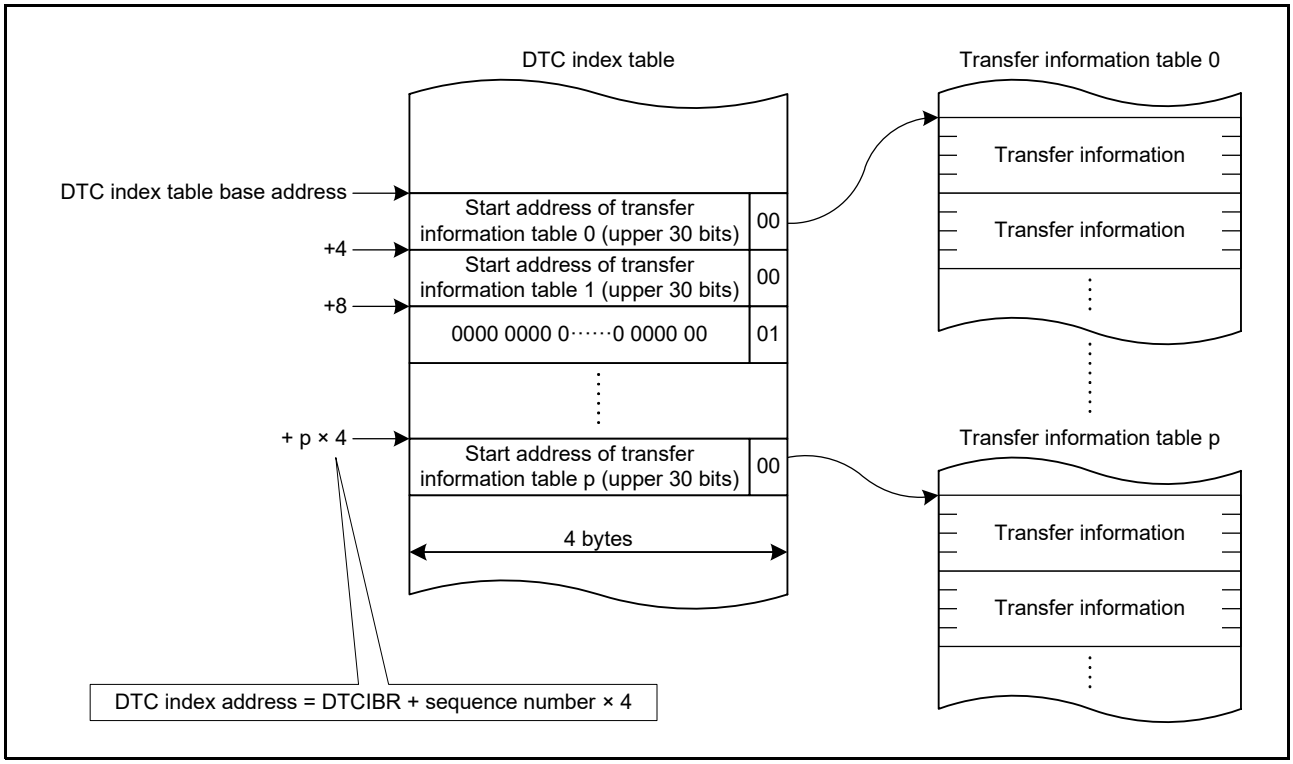
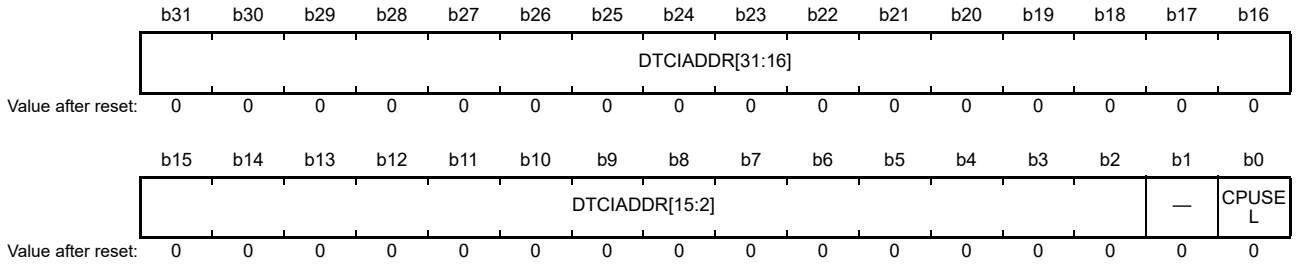


Figure 18.17 DTC Index Table

- 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.

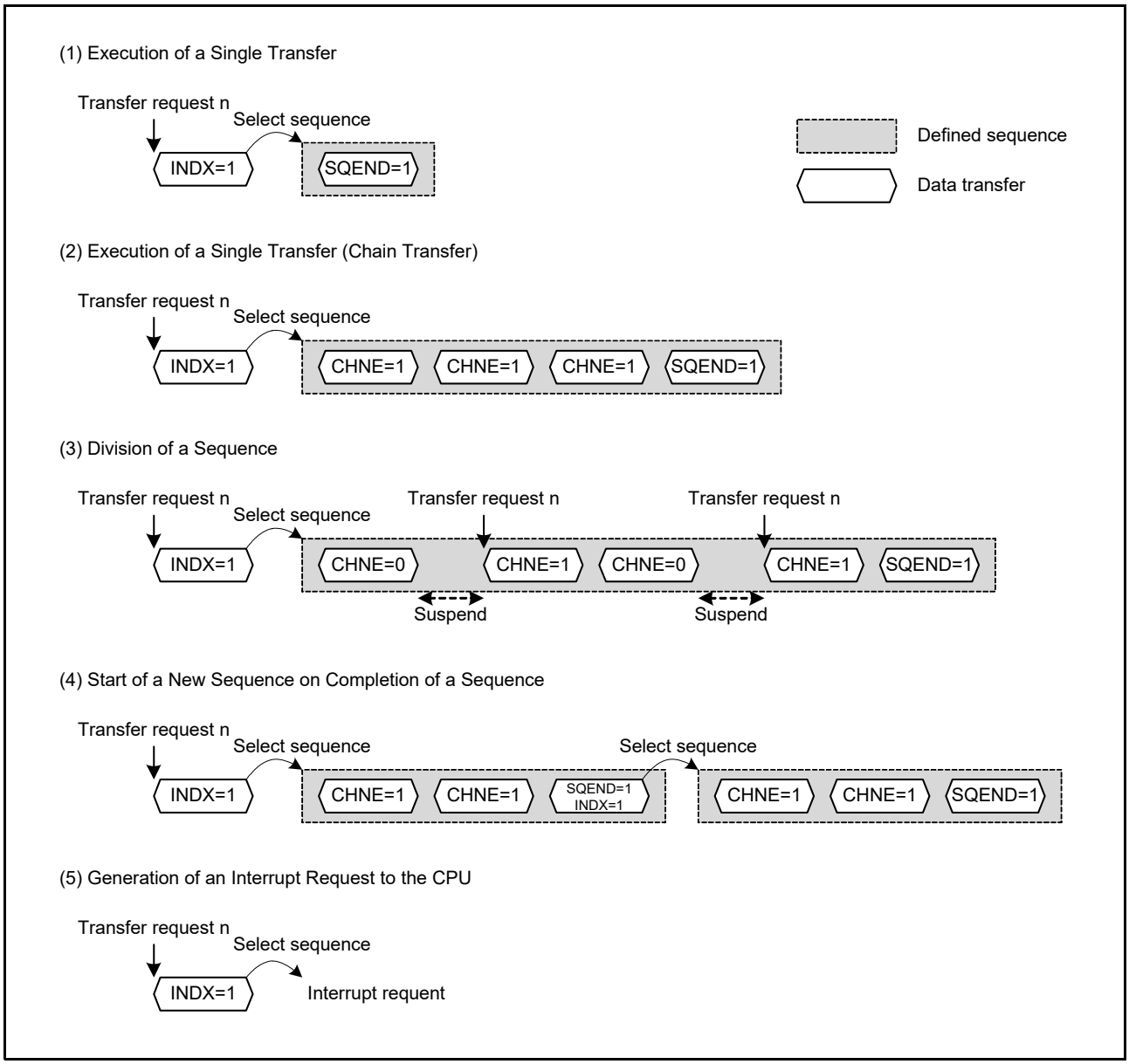


Figure 18.18 Examples of Sequence Transfers

(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.

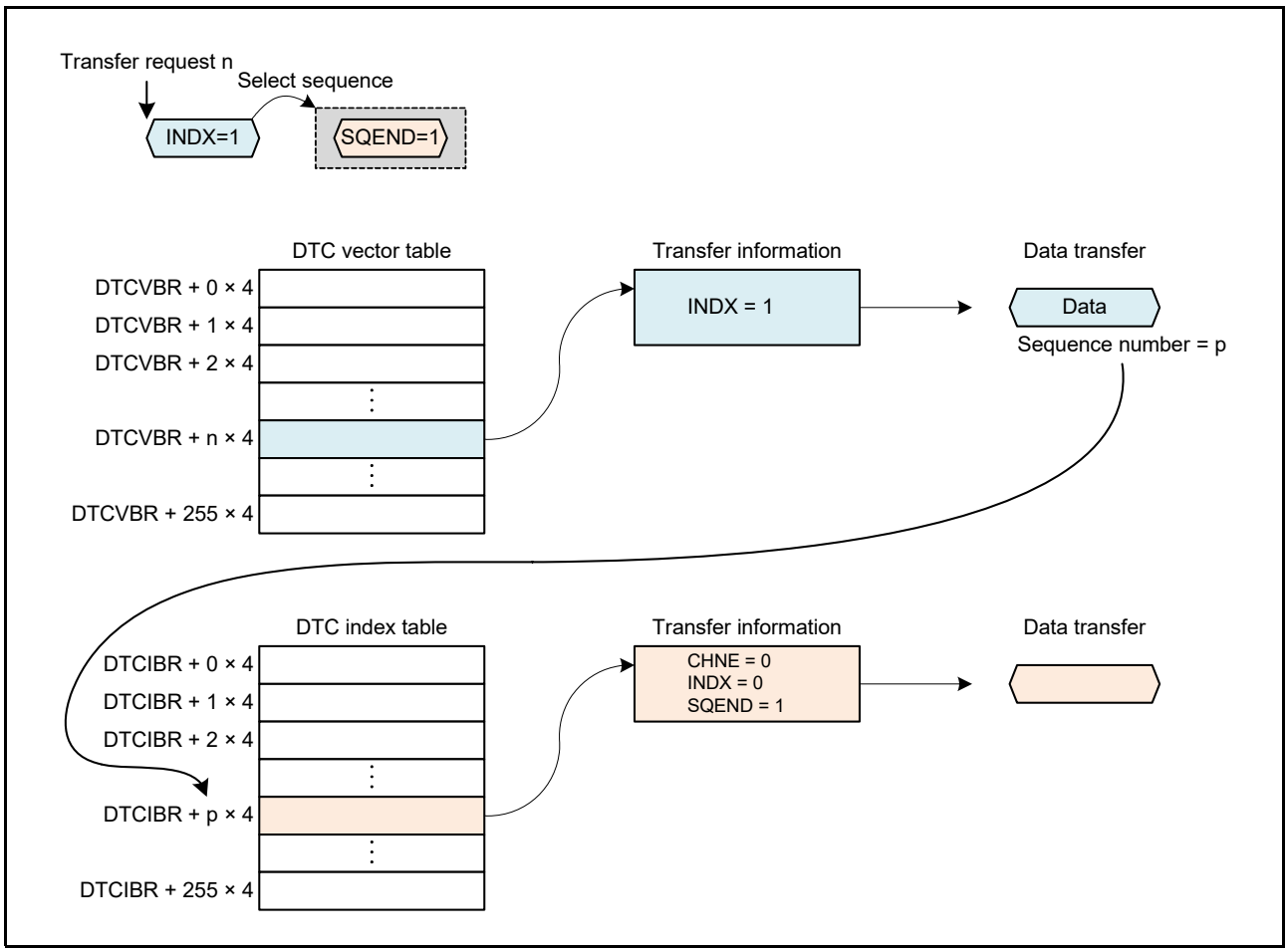


Figure 18.19 Example of a Sequence of Single Transfer

(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.

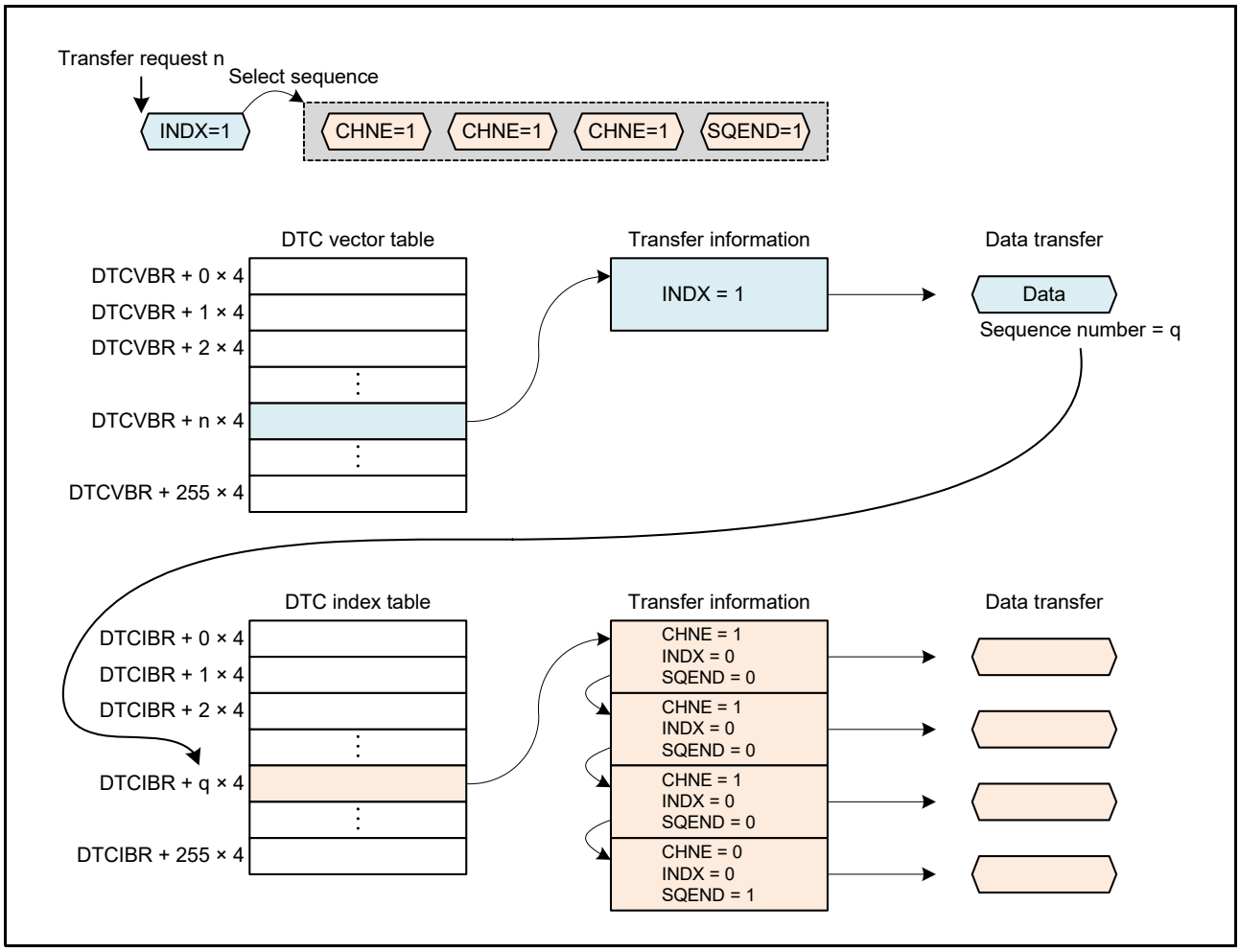


Figure 18.20 Example of Sequence of a Single Chain Transfer

(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.

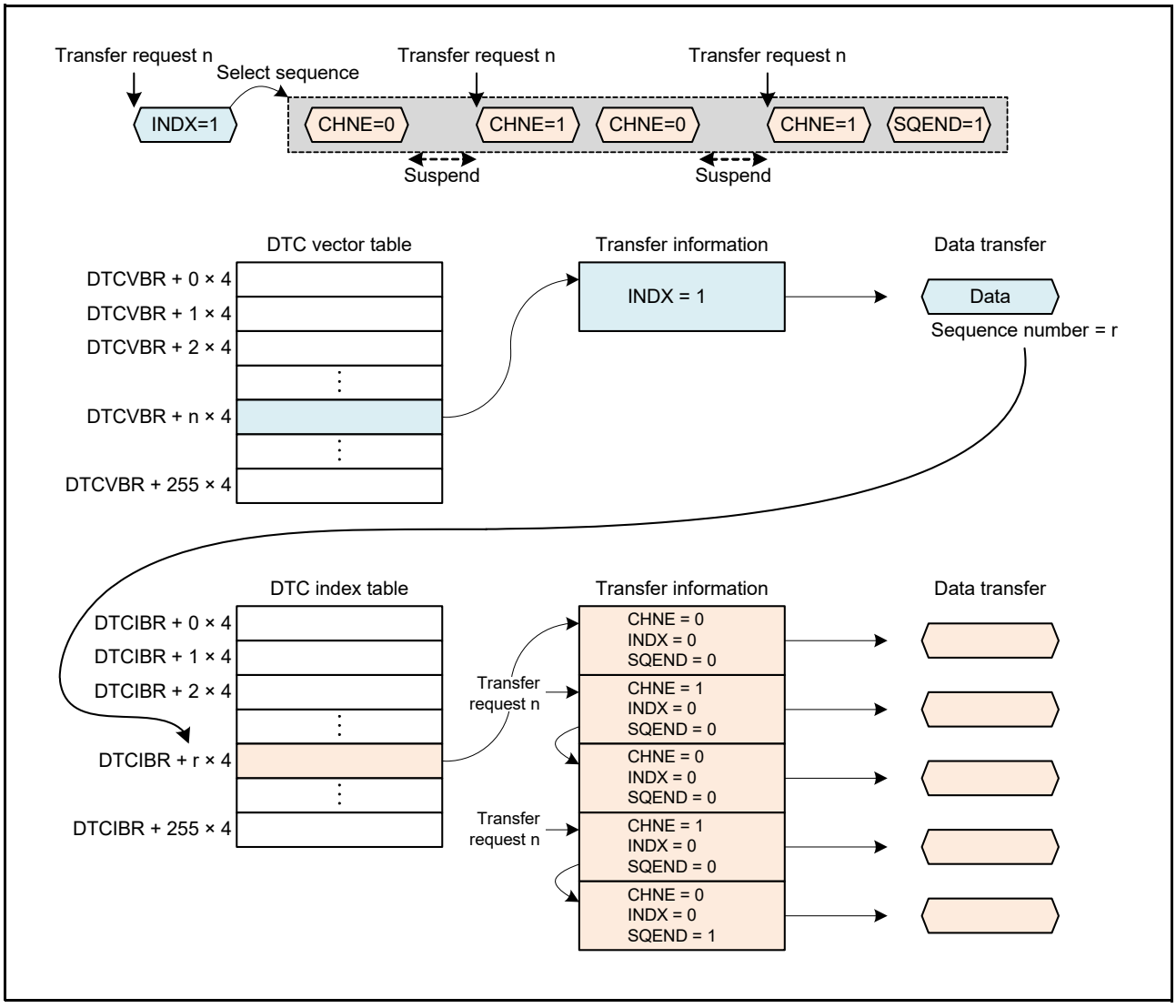


Figure 18.21 Example of Divided Sequence

(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.

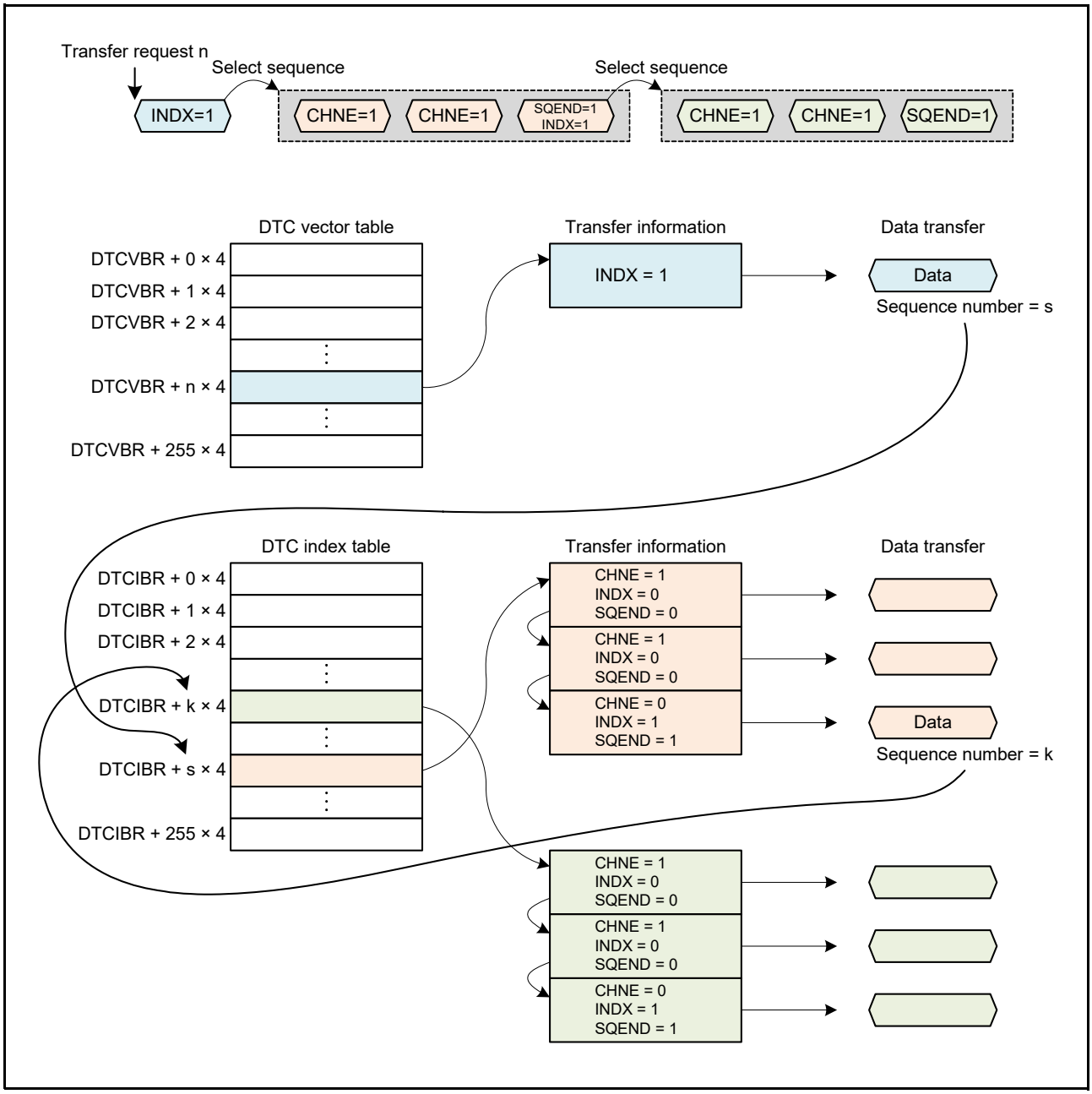


Figure 18.22 Example When Starting a New Sequence on Completion of a Sequence

(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.

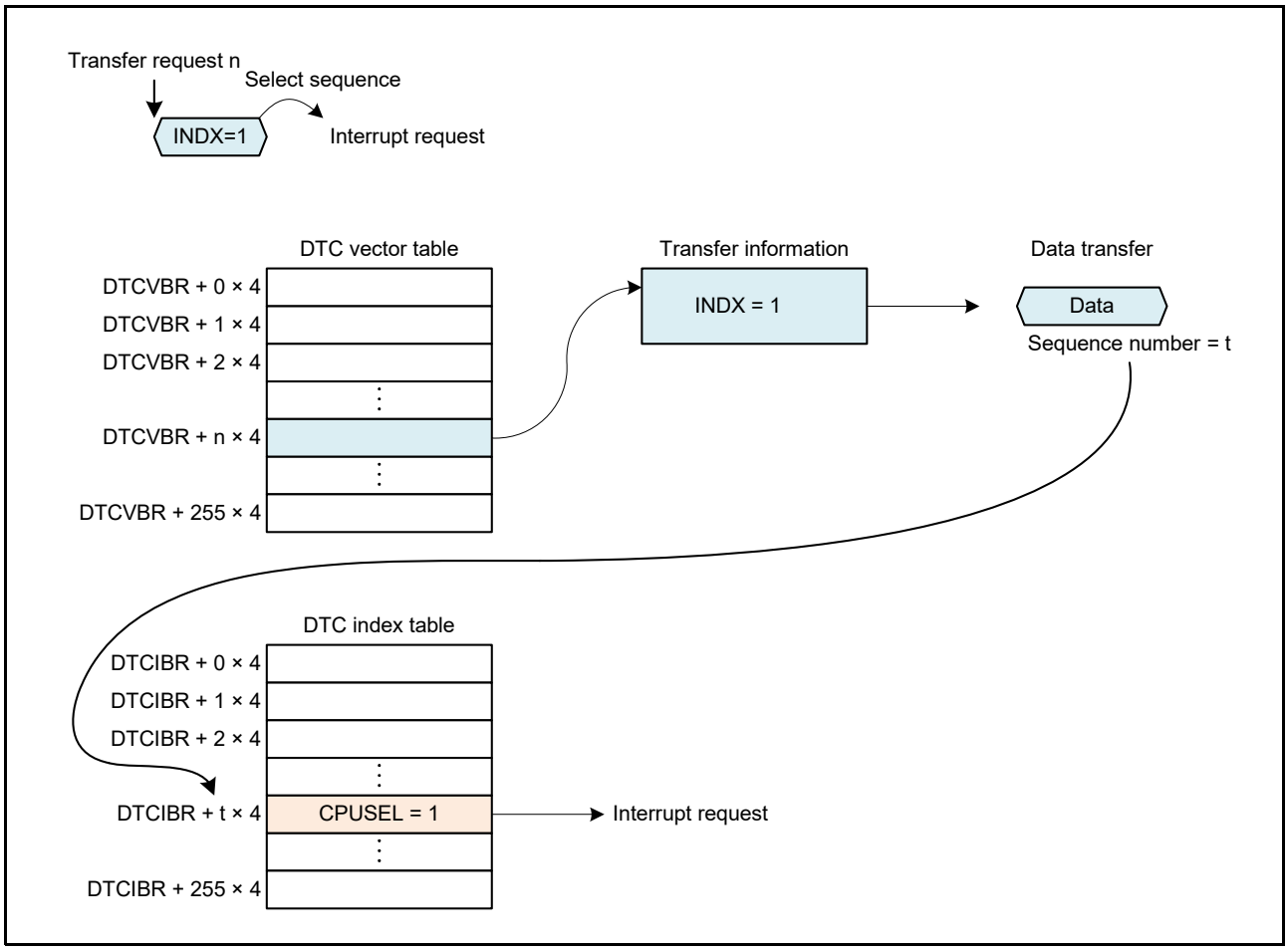


Figure 18.23 Example of Output of an Interrupt Request to the CPU

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.

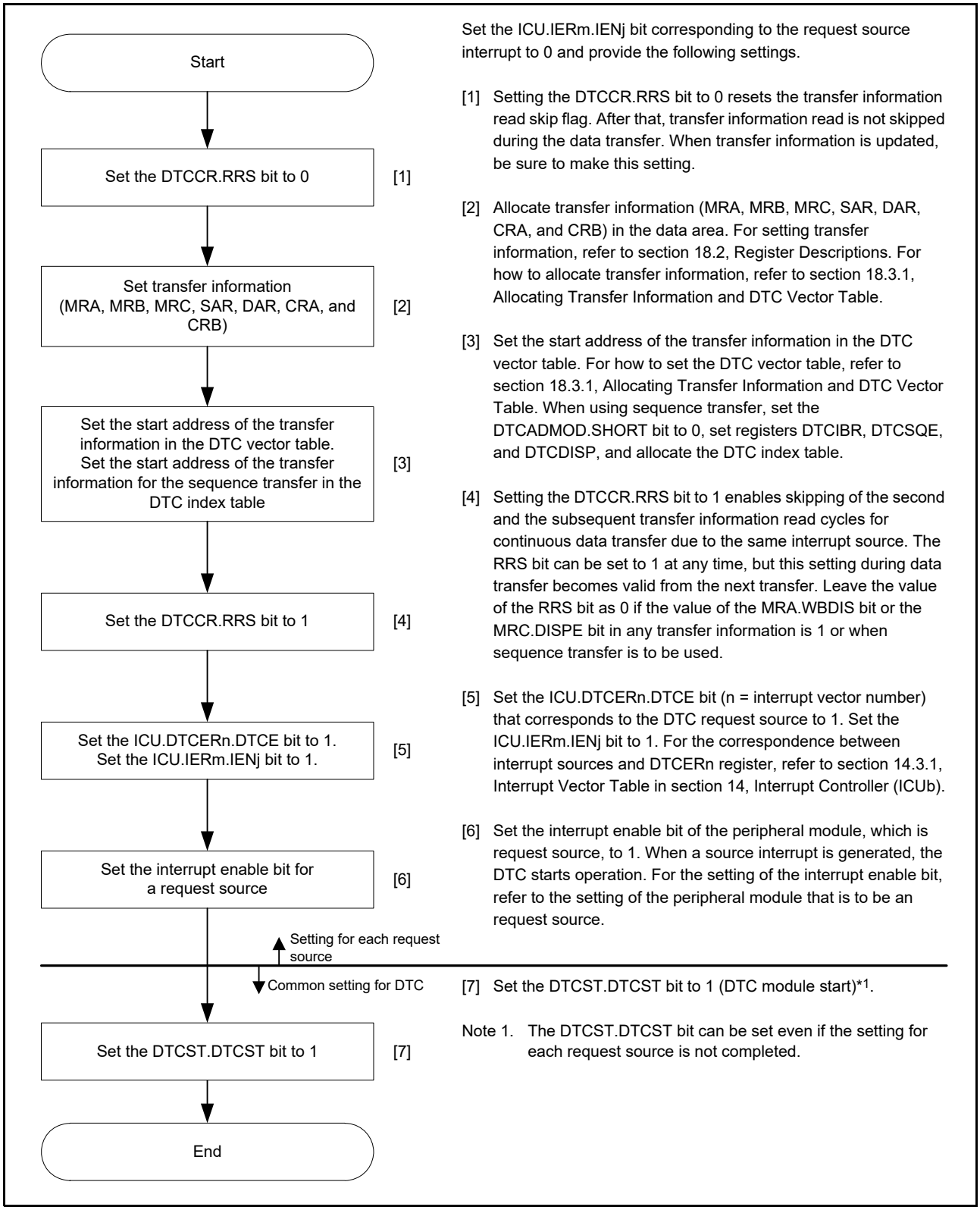


Figure 18.24 Procedure to Set the DTC

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.

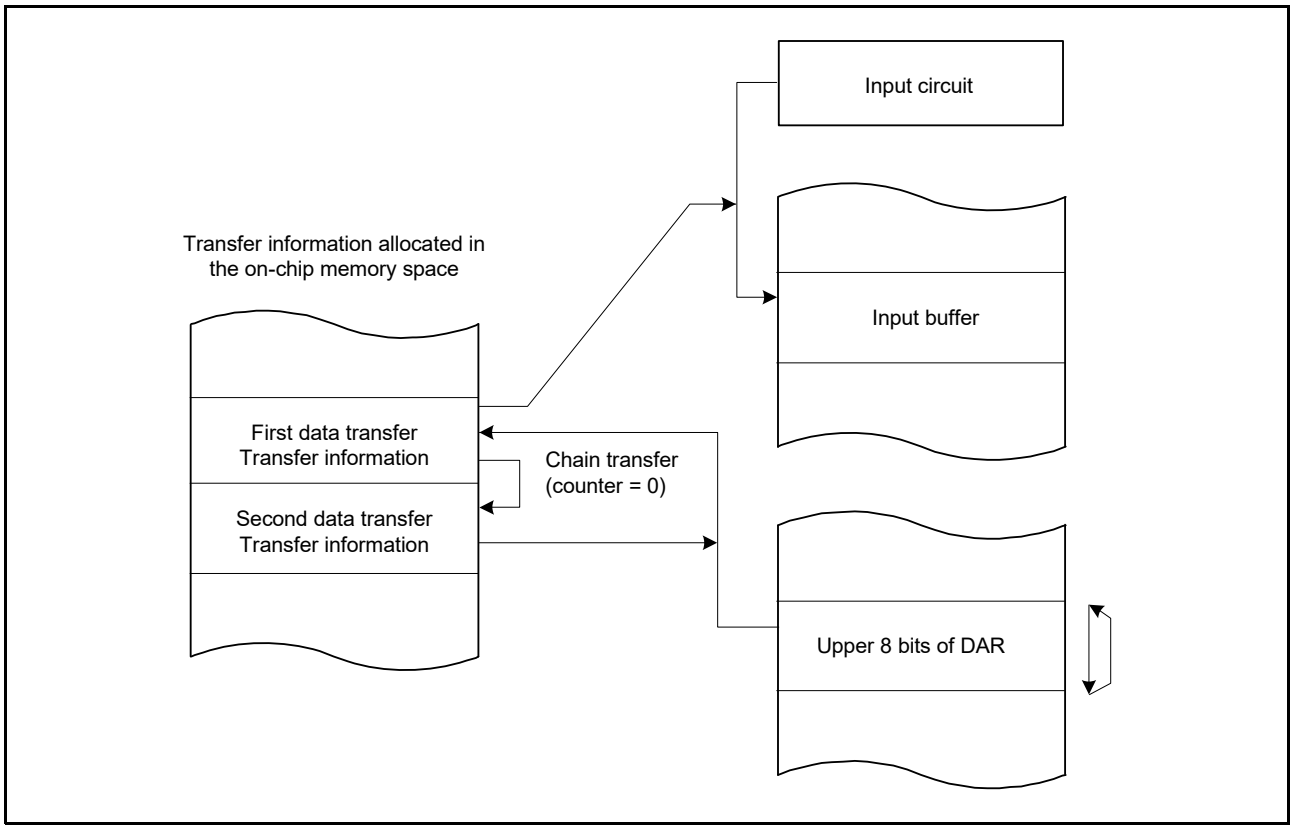


Figure 18.25 Chain Transfer When the Counter is 0

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.

18.9 Low Power Consumption Function

Before making a transition to the module stop state, deep sleep mode, or 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) Deep Sleep Mode

Make settings according to the procedure under section 11.6.2.1, Entry to Deep Sleep 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 deep sleep 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 deep sleep mode.

(3) Software Standby Mode

Make settings according to the procedure under section 11.6.3.1, Entry to 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 follows the completion of the data transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.6, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in deep sleep 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 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), 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.

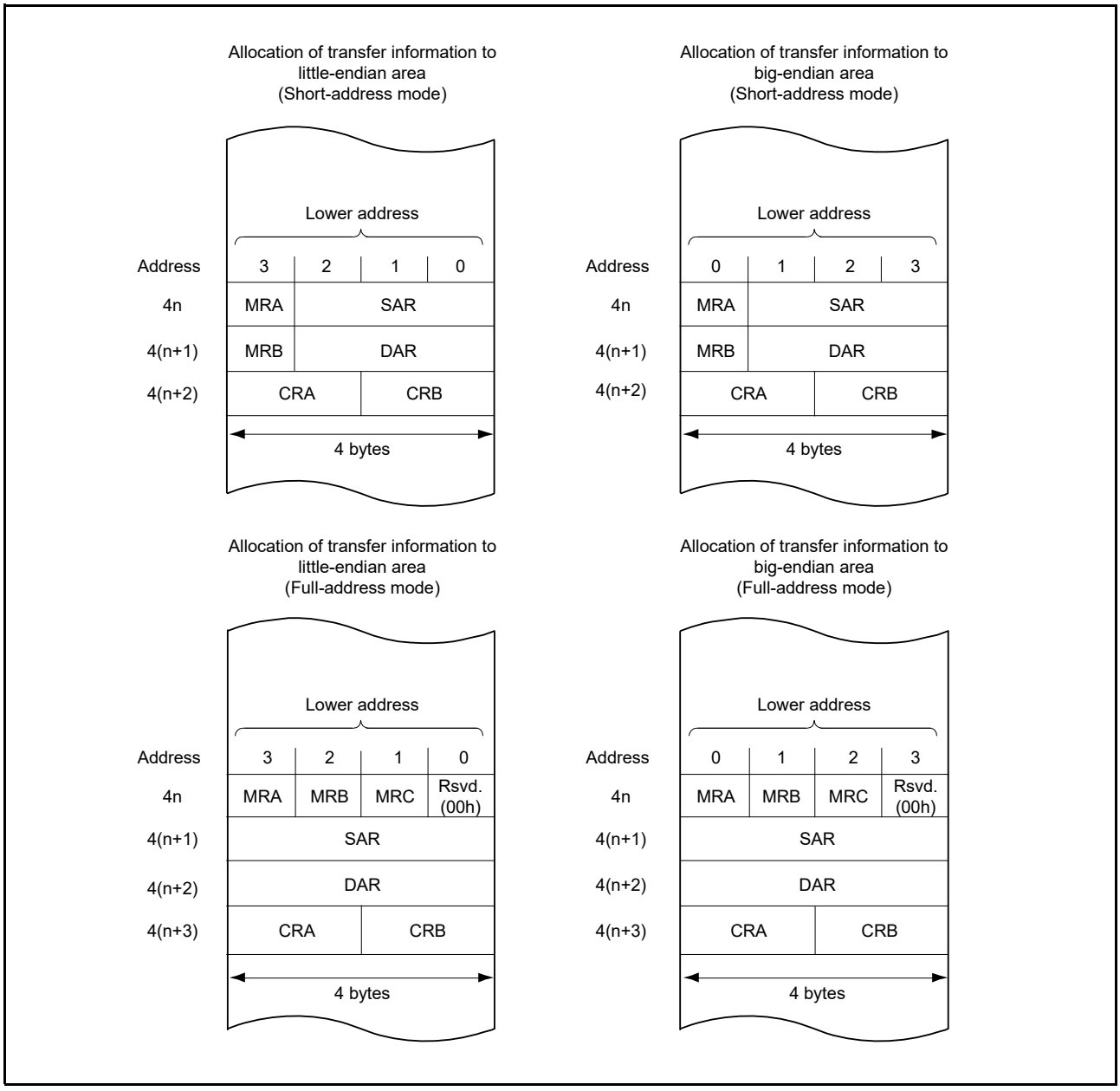


Figure 18.26 Allocation of Transfer Information

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 (ICUb).

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	• 116 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. In products with 64-pin packages, an event linkage operation for PB6 and PB7 becomes impossible when PC0 and PC1 are selected by the port switching register A (PSRA). In products with 48-pin packages, an event linkage operation for PB0, PB1, PB3, and PB5 becomes impossible when PC0 to PC3 are selected by the port switching register B (PSRB).

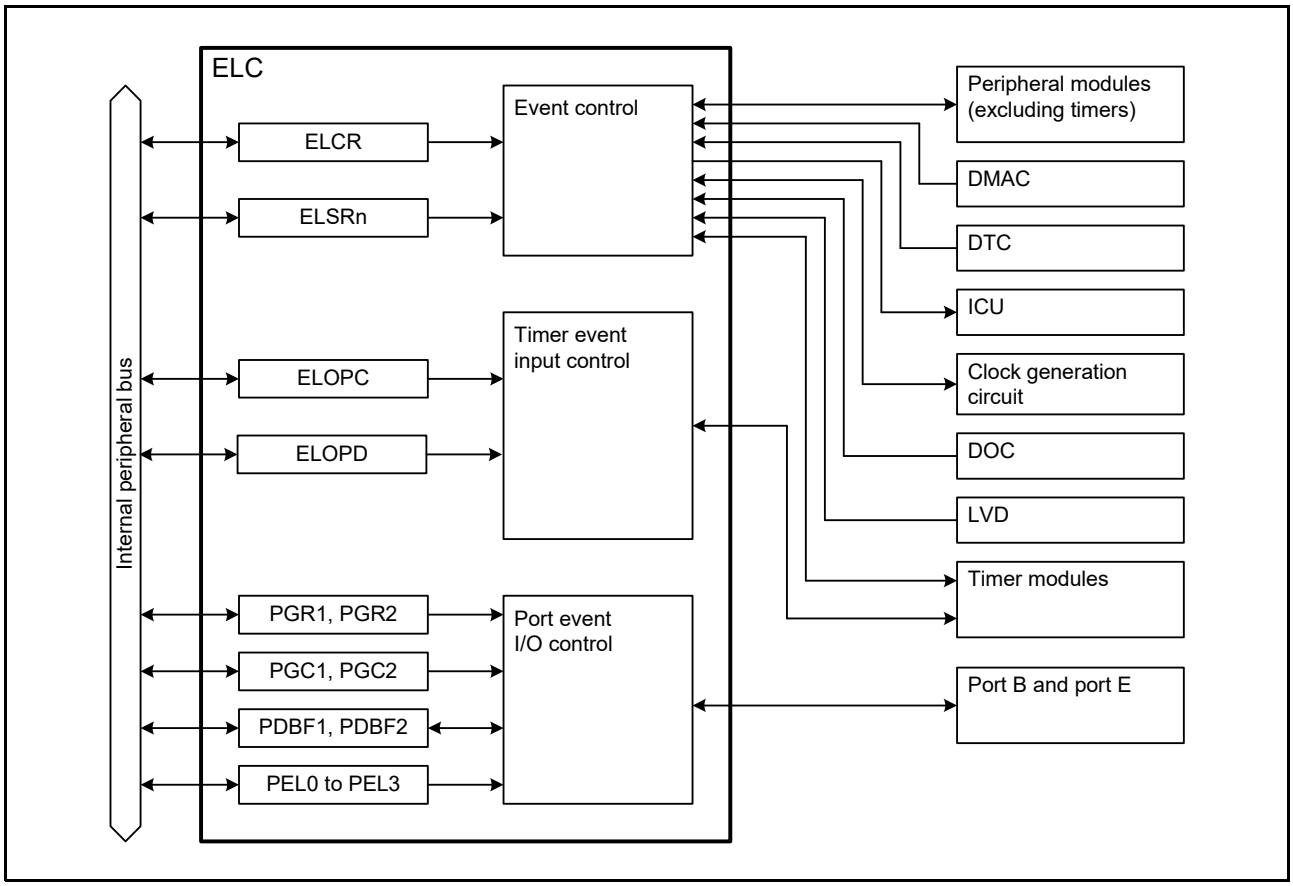


Figure 19.1 ELC Block Diagram (n = 7, 8, 10, 12, 14 to 16, 18 to 28, 48 to 56)

19.2 Register Descriptions

19.2.1 Event Link Control Register (ELCR)

Address(es): ELC.ELCR 0008 B100h

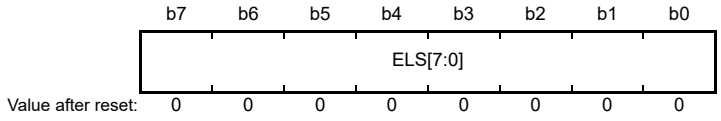
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

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 = 7, 8, 10, 12, 14 to 16, 18 to 28, 48 to 56)

Address(es): ELC.ELSR7 0008 B108h, ELC.ELSR8 0008 B109h, ELC.ELSR10 0008 B10Bh, ELC.ELSR12 0008 B10Dh, ELC.ELSR14 0008 B10Fh, 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.ELSR48 0008 B146h, ELC.ELSR49 0008 B147h, ELC.ELSR50 0008 B148h, ELC.ELSR51 0008 B149h, ELC.ELSR52 0008 B14Ah, ELC.ELSR53 0008 B14Bh, ELC.ELSR54 0008 B14Ch, ELC.ELSR55 0008 B14Dh, 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. 1Fh to C6h: 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 (1/2)

Register Name	Peripheral Module
ELSR7	CMT1
ELSR8	ICU (LPT dedicated interrupt)*1
ELSR10	TMR0
ELSR12	TMR2
ELSR14	CTSU
ELSR15	S12AD (ELCTRG00N)
ELSR16	DA0
ELSR18	ICU (Interrupt 1)*2
ELSR19	ICU (Interrupt 2)*2
ELSR20	Output port group 1
ELSR21	Output port group 2
ELSR22	Input port group 1
ELSR23	Input port group 2
ELSR24	Single port 0
ELSR25	Single port 1
ELSR26	Single port 2
ELSR27	Single port 3

Table 19.2 Correspondence between the ELSRn Register and the Peripheral Modules (2/2)

Register Name	Peripheral Module
ELSR28	Clock source switching to LOCO
ELSR48	GPTW event source A (common to all channels)
ELSR49	GPTW event source B (common to all channels)
ELSR50	GPTW event source C (common to all channels)
ELSR51	GPTW event source D (common to all channels)
ELSR52	GPTW event source E (common to all channels)
ELSR53	GPTW event source F (common to all channels)
ELSR54	GPTW event source G (common to all channels)
ELSR55	GPTW event source H (common to all channels)
ELSR56	S12AD (ELCTRG01N)

Note 1. Specify an event number to 32h (LPT compare match 0).
Note 2. Specify an event number from among 63h to 6Ah. Do not set other values.

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
1Fh	Compare match timer	CMT1 compare match 1
22h	8-bit timers	TMR0 compare match A0
23h		TMR0 compare match B0
24h		TMR0 overflow
28h		TMR2 compare match A2
29h		TMR2 compare match B2
2Ah		TMR2 overflow
2Eh	Realtime clock	RTC periodic event (select 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, or 2 seconds)
31h	Independent watchdog timer	IWDT underflow or refresh error
32h	Low power timer	LPT compare match 0
33h	Low power timer	LPT compare match 1
34h	12-bit A/D converter	S12AD comparison conditions are met
35h		S12AD comparison conditions are not met
3Ah	Serial communications interfaces	SCI5 error (receive error or error signal detection)
3Bh		SCI5 receive data full
3Ch		SCI5 transmit data empty
3Dh		SCI5 transmit end
4Eh	I^2C-bus interface	RIIC0 communication error or event generation
4Fh		RIIC0 receive data full
50h		RIIC0 transmit data empty
51h		RIIC0 transmit end
52h	Serial peripheral interface	RSPI0 error (mode fault, overrun, underrun, or parity error)
53h		RSPI0 idle
54h		RSPI0 receive buffer full
55h		RSPI0 transmit buffer empty
56h		RSPI0 transmit end
58h	12-bit A/D converter	S12AD A/D conversion end
59h	Comparator B0	Comparison result change of comparator B0
5Ah	Comparator B0/B1	Comparison result change of comparator B0/B1
5Bh	Voltage detection circuit	LVD1 voltage detection
5Ch		LVD2 voltage detection
5Dh	DMA controller	DMAC0 transfer end
5Eh		DMAC1 transfer end
5Fh		DMAC2 transfer end
60h		DMAC3 transfer end
61h	Data transfer controller	DTC transfer end
62h	Clock generation circuit	Oscillation stop detection of clock generation circuit
63h	I/O ports	Input edge detection of input port group 1
64h		Input edge detection of input port group 2
65h		Input edge detection of single input port 0
66h		Input edge detection of single input port 1
67h		Input edge detection of single input port 2
68h		Input edge detection of single input port 3
69h	Event link controller	Software event

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
6Ah	Data operation circuit	DOC data operation condition met
80h	General PWM timer	GPTW0 compare match A
81h		GPTW0 compare match B
82h		GPTW0 compare match C
83h		GPTW0 compare match D
84h		GPTW0 compare match E
85h		GPTW0 compare match F
86h		GPTW0 overflow
87h		GPTW0 underflow
88h		GPTW0 A/D converter start request A
89h		GPTW0 A/D converter start request B
8Ah		GPTW1 compare match A
8Bh		GPTW1 compare match B
8Ch		GPTW1 compare match C
8Dh		GPTW1 compare match D
8Eh		GPTW1 compare match E
8Fh		GPTW1 compare match F
90h		GPTW1 overflow
91h		GPTW1 underflow
92h		GPTW1 A/D converter start request A
93h		GPTW1 A/D converter start request B
94h		GPTW2 compare match A
95h		GPTW2 compare match B
96h		GPTW2 compare match C
97h		GPTW2 compare match D
98h		GPTW2 compare match E
99h		GPTW2 compare match F
9Ah		GPTW2 overflow
9Bh		GPTW2 underflow
9Ch		GPTW2 A/D converter start request A
9Dh		GPTW2 A/D converter start request B
9Eh		GPTW3 compare match A
9Fh		GPTW3 compare match B
A0h		GPTW3 compare match C
A1h		GPTW3 compare match D
A2h		GPTW3 compare match E
A3h		GPTW3 compare match F
A4h		GPTW3 overflow
A5h		GPTW3 underflow

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
A6h	General PWM timer	GPTW4 compare match A
A7h		GPTW4 compare match B
A8h		GPTW4 compare match C
A9h		GPTW4 compare match D
AAh		GPTW4 compare match E
ABh		GPTW4 compare match F
ACh		GPTW4 overflow
ADh		GPTW4 underflow
AEh		GPTW5 compare match A
AFh		GPTW5 compare match B
B0h		GPTW5 compare match C
B1h		GPTW5 compare match D
B2h		GPTW5 compare match E
B3h		GPTW5 compare match F
B4h		GPTW5 overflow
B5h		GPTW5 underflow
B6h		GPTW6 compare match A
B7h		GPTW6 compare match B
B8h		GPTW6 compare match C
B9h		GPTW6 compare match D
BAh		GPTW6 compare match E
BBh		GPTW6 compare match F
BCh		GPTW6 overflow
BDh		GPTW6 underflow
BEh		GPTW7 compare match A
BFh		GPTW7 compare match B
C0h		GPTW7 compare match C
C1h		GPTW7 compare match D
C2h		GPTW7 compare match E
C3h		GPTW7 compare match F
C4h		GPTW7 overflow
C5h		GPTW7 underflow
C6h	General PWM timer	GPTW (OPS) U-/V-/W-phase input edge detected
Settings other than above are prohibited.		

19.2.3 Event Link Option Setting Register C (ELOPC)

Address(es): ELC.ELOPC 0008 B121h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LPTMD[1:0]		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
b5, b4	LPTMD[1:0]	LPT Operation Select	b5 b4 0 0: Output the LPT compare match 0 event to ICU as an interrupt request 1 1: Event output is disabled. Settings other than above are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPC register specifies the operations of CMT1 and LPT 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 D (ELOPD)

Address(es): ELC.ELOPD 0008 B122h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TMR2MD[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	—	Reserved	These bits are read as 1. The write value should be 1.	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	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPD register specifies the operations of TMR0 and TMR2 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.5 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 (PGRn)	Port Group Control Register (PGCn)	Port Buffer Register (PDBFn)
Port B	PGR1 register	PGC1 register	PDBF1 register
Port E	PGR2 register	PGC2 register	PDBF2 register

19.2.6 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.

19.2.7 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 19.3.7, 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.

19.2.8 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.

19.2.9 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.

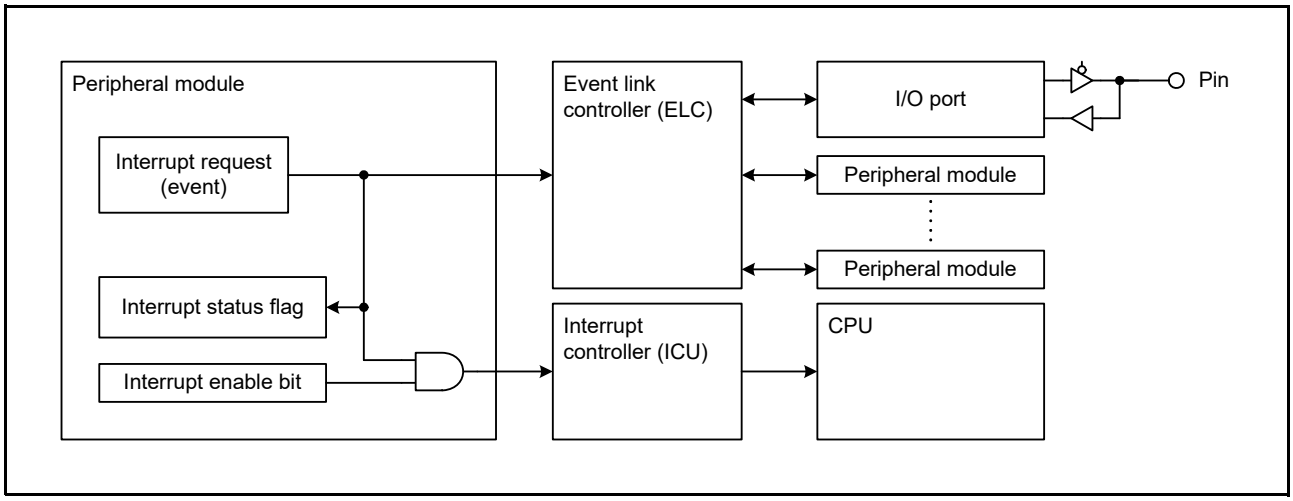


Figure 19.2 Relation between Interrupt Handling and ELC

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		
CMT TMR	The following operations can be selected by setting the ELOPC and ELOPD registers: • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events (CMT, TMR).		
GPTW	The following operations can be selected by setting registers in GPTW: • Starts counting when an event signal is input. • Stops counting when an event signal is input. • Clears counter when an event signal is input. • Increments counter when an event signal is input. • Decrements counter when an event signal is input. • Performs input-capture operation when an event signal is input.		
CTSU	Starts measurement of electrostatic capacitance when an event signal is input.		
A/D converter	Starts A/D conversion when an event signal is input.		
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 except GPTW, set the ELOPC or ELOPD 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.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

19.3.4 Operation of GPTW When Event Signal is Input

Eight event signals specified by the ELSR48 to ELSR55 registers are connected to every channel of GPTW as GPTW event sources A to H. To set the operation of GPTW when inputting an event signal, enable any of the event sources A to H by a corresponding bit in registers listed in Table 19.6.

Table 19.6 Operations When Event Signal is Input and Corresponding Source Select Registers

Operations on Event	Register Symbol	Register Name
Count start	GTSSR	General PWM Timer Start Source Select Register
Count stop	GTPSR	General PWM Timer Stop Source Select Register
Counter clear	GTCSR	General PWM Timer Clear Source Select Register
Count-up	GTUPSR	General PWM Timer Count-Up Source Select Register
Count-down	GTDNSR	General PWM Timer Count-Down Source Select Register
Input capture A	GTICASR	General PWM Timer Input Capture Source Select Register A
Input capture B	GTICBSR	General PWM Timer Input Capture Source Select Register B

19.3.5 Operation of CTSU When Event Signal is Input

When an event signal is input while the CAP and STRT bits in the CTSUCRA register are set to 1, a measurement is started. Refer to the description of the CTSUCRA.STRT bit for details.

19.3.6 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.

19.3.7 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[2:0] 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).

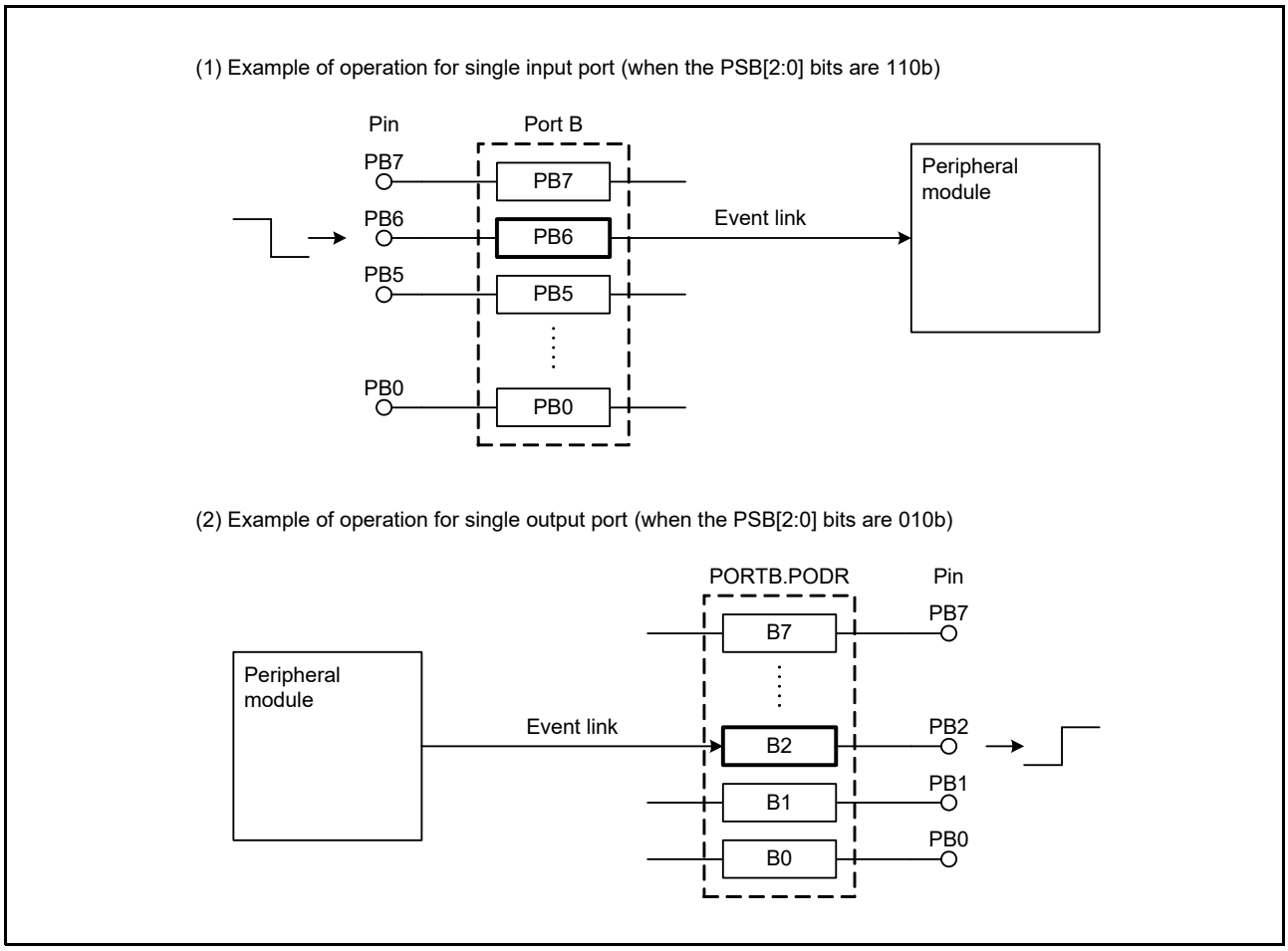


Figure 19.3 Event Linkage Related to Single Ports (Port B)

(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.

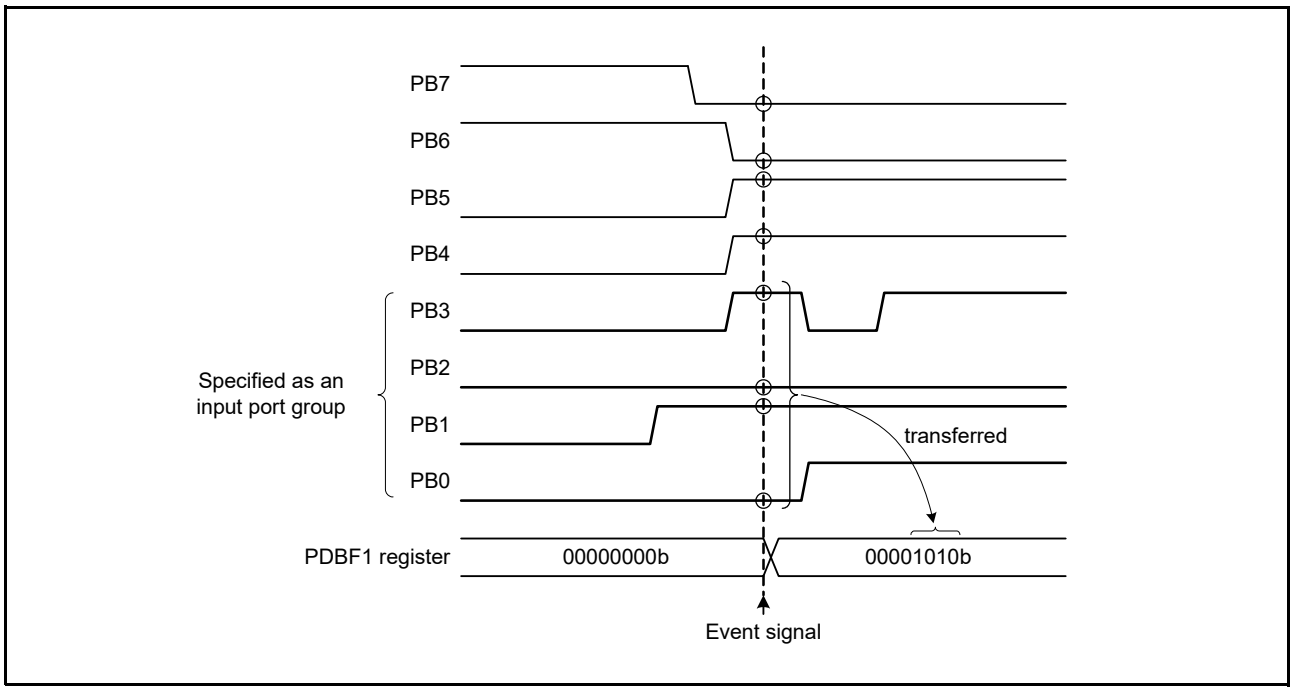


Figure 19.4 Event Linkage Related to Input Port Groups (Port B)

(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.

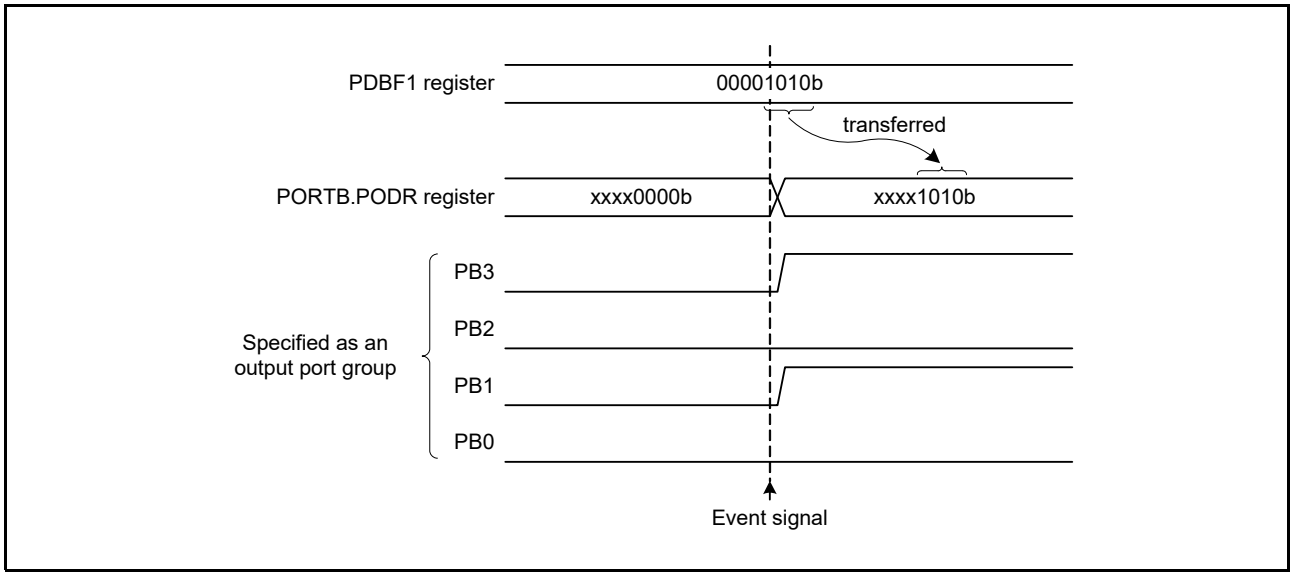


Figure 19.5 Event Linkage Related to Output Port Groups (Port B)

(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.

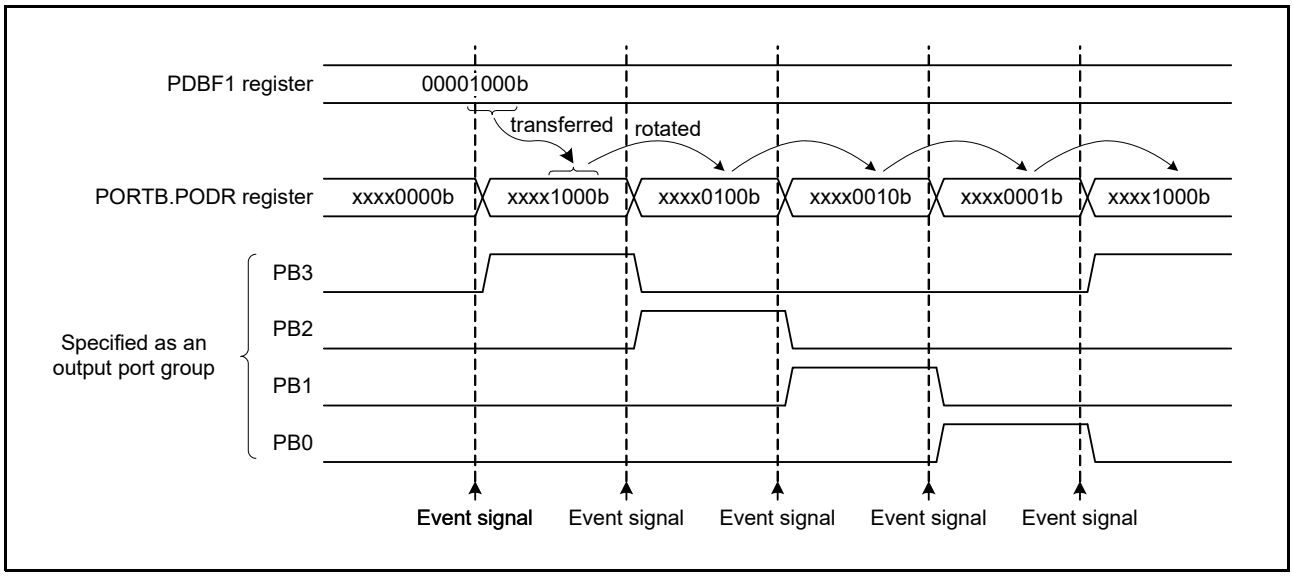


Figure 19.6 Bit-Rotating Operation of Output Port Groups (Port B)

(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.

19.3.8 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 ELOPC and ELOPD registers corresponding to the timer as required.
(5) Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
(6) Set the operation of the peripheral module (source) from which an event signal is output, and activate the module. The preset operation of the destination peripheral module is started by the event signal that is output from the source peripheral module.
(7) To stop event linkage of independent peripheral module, set 00h to the ELSRn register corresponding to the peripheral module. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

Note: If event signal output from the RTC is to be used, make the ELC settings after the RTC settings (initialization, time setting, etc.). Unintended events may be generated if RTC settings are made after the ELC settings.
Note: When using event signal output from the LVD, set the LVD and then the ELC. Set the corresponding ELSRn register to 00h and then disable the LVD.

19.4 Usage Notes

19.4.1 Setting ELSRn Register

(1) Setting ELSR8 Register

Set this register to 32h (LPT compare match 0).

(2) Setting ELSR18 and ELSR19 Registers

Specify an event number from among 63h to 6Ah. Do not set the value other than preceding numbers.

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 (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 I/O ports function as a general I/O port, an I/O pin of a peripheral module, an input pin for an interrupt.
Some of the pins are 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 on-chip 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 data 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 control register (PCR) that controls on/off of the input pull-up MOS, and the port mode register (PMR) that specifies the pin function of each port.
For details on the PMR register, see section 21, Multi-Function Pin Controller (MPC).
In 80-pin, 64-pin and 48-pin packages, port switching register A (PSRA) and port switching register B (PSRB) respectively are individually provided to use PORTC as an 8-bit port by switching the general I/O function of some pins.
The configuration of the I/O ports differs depending on the package. Table 20.1 lists the specifications of I/O ports, and Table 20.2 list the port functions.

Table 20.1 Specifications of I/O Ports

Port	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	P03 to P07	5	P03 to P07	5	P03, P05	2	Not provided	0
PORT1	P12 to P17	6	P12 to P17	6	P14 to P17	4	P14 to P17	4
PORT2	P20 to P27	8	P20, P21, P26, P27	4	P26, P27	2	P26, P27	2
PORT3	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 P42, P45 to P47	6
PORT5	P50 to P55	6	P54, P55	2	P54, P55	2	Not provided	0
PORTA	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, PB1, PB3, PB5 to PB7	6	PB0, PB1, PB3, PB5	4
PORTC	PC0 to PC7	8	PC0 to PC7*1	6*3	PC0 to PC7*1	6*3	PC0 to PC7*2	4*3
PORTD	PD0 to PD7	8	PD0 to PD2	3	Not provided	0	Not provided	0
PORTE	PE0 to PE7	8	PE0 to PE5	6	PE0 to PE5	6	PE1 to PE4	4
PORTG	PG7	1	PG7	1	PG7	1	PG7	1
PORTH	PH0 to PH3, PH6, PH7	6	PH0 to PH3, PH6, PH7	6	PH0 to PH3, PH6, PH7	6	PH0 to PH3	4
PORTJ	PJ1, PJ3, PJ6, PJ7	4	PJ1, PJ6, PJ7	3	PJ6, PJ7	2	PJ6, PJ7	2
	Total of Pins	92	Total of Pins	72	Total of Pins	56	Total of Pins	40

Note 1. PC0 and PC1 are valid only when switching by the port switching register A.
Note 2. PC0 to PC3 are valid only when switching by the port switching register B.
Note 3. The number of the multiplexed pin functions with PB is not included.

Table 20.2 Port Functions

Port	Pin	I/O	Input Pull-up	Open Drain Output	5-V Tolerant	IO Level
PORT0	P03 to P07	I/O	✓	—	—	AVCC0
PORT1	P12, P13, P16, P17	I/O	✓	✓	✓	VCC
	P14, P15	I/O	✓	✓	—	
PORT2	P20 to P27	I/O	✓	✓	—	
PORT3	P30 to P34	I/O	✓	✓	—	
	P35	Input	—	—	—	
	P36, P37	I/O	✓	✓	—	
PORT4	P40 to P47	I/O	✓	—	—	AVCC0
PORT5	P50 to P52, P54	I/O	✓	✓	—	VCC
	P53, P55	I/O	✓	—	—	
PORTA	PA0 to PA7	I/O	✓	✓	—	
PORTB	PB0 to PB7	I/O	✓	✓	—	
PORTC	PC0 to PC7	I/O	✓	✓	—	
PORTD	PD0 to PD2	I/O	✓	✓	—	
	PD3 to PD7	I/O	✓	—	—	
PORTE	PE0 to PE7	I/O	✓	✓	—	
PORTG	PG7	I/O	✓	✓	—	
PORTH	PH0 to PH3	I/O	✓	—	—	
	PH6, PH7	Input	—	—	—	
PORTJ	PJ1, PJ3	I/O	✓	—	—	
	PJ6, PJ7	I/O	✓	—	—	AVCC0

✓: Supported
—: Unsupported

20.2 I/O Port Configuration

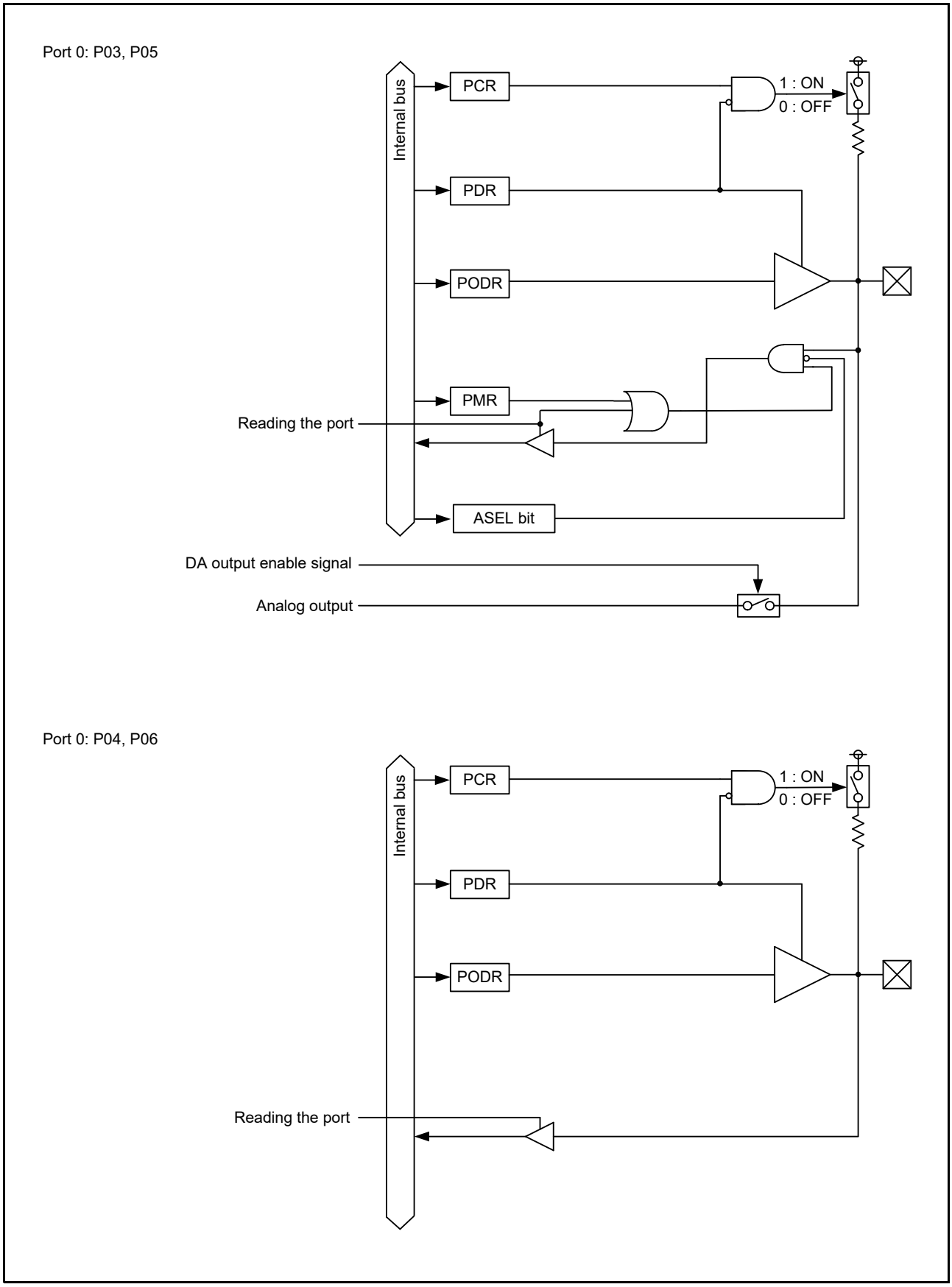


Figure 20.1 I/O Port Configuration (1)

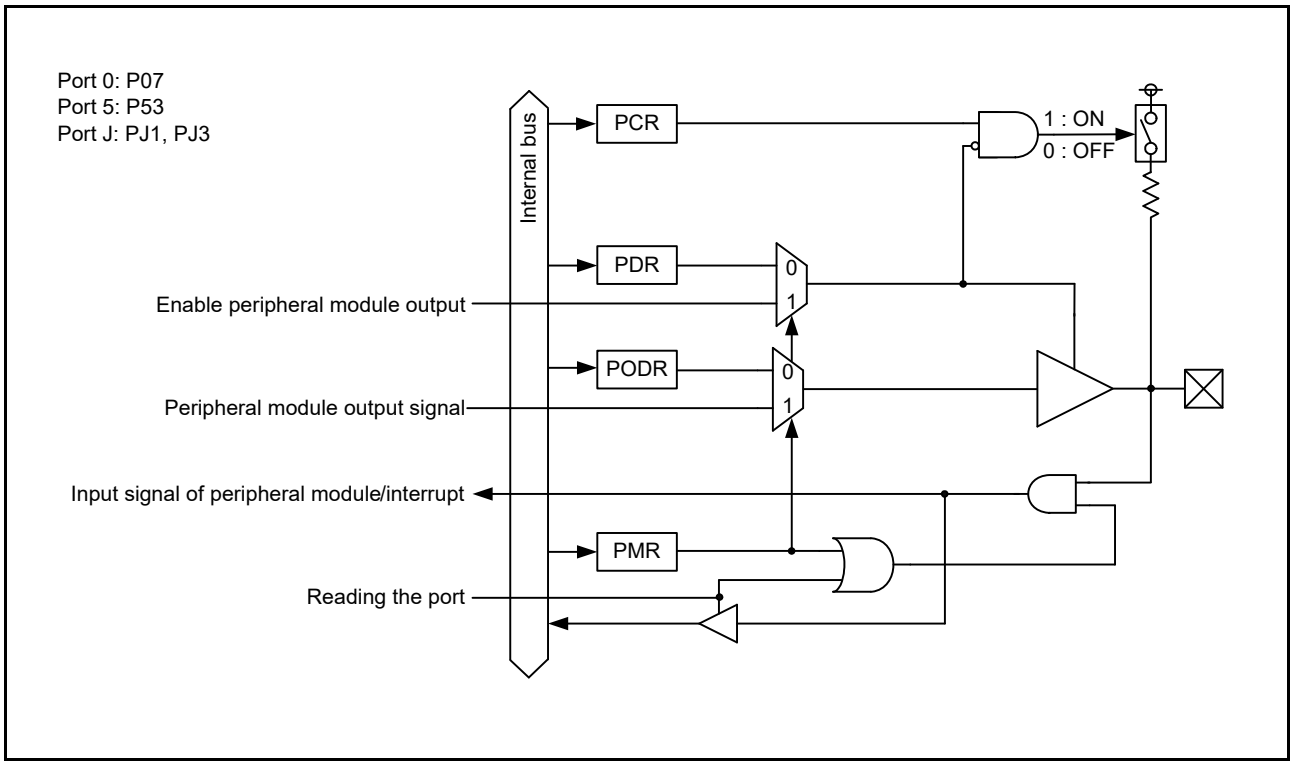
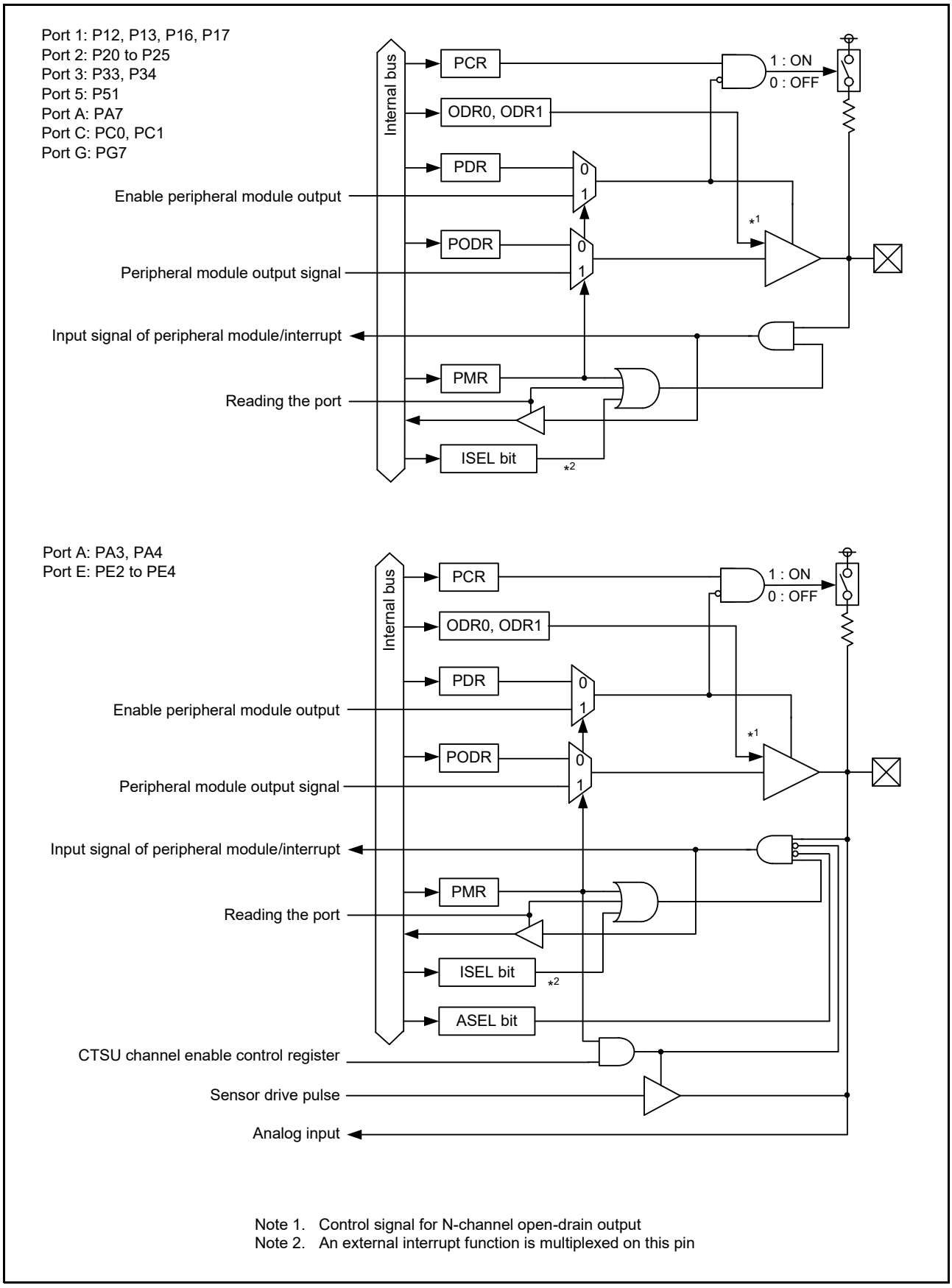


Figure 20.2 I/O Port Configuration (2)



Note 1. Control signal for N-channel open-drain output
Note 2. An external interrupt function is multiplexed on this pin

Figure 20.3 I/O Port Configuration (3)

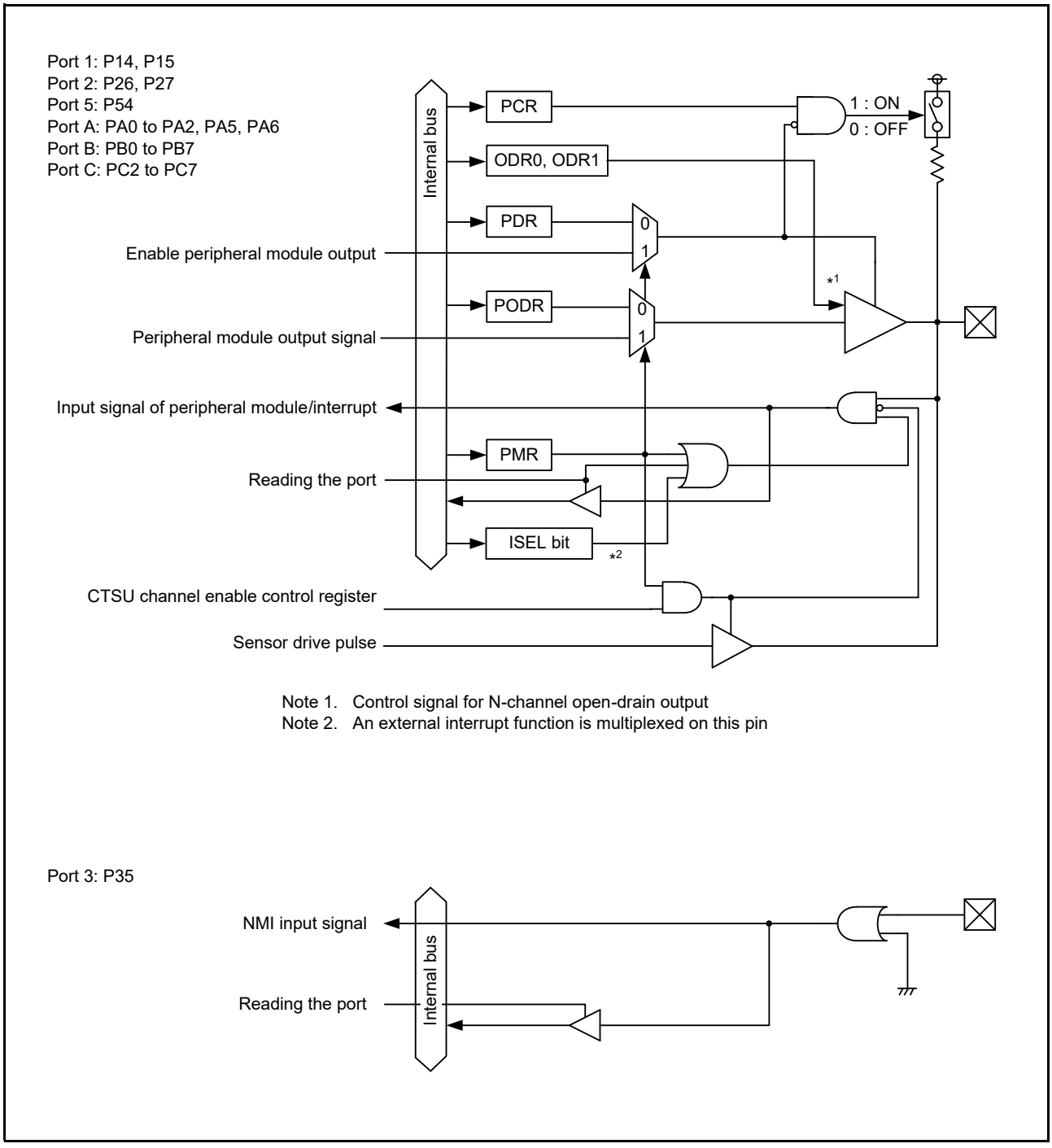
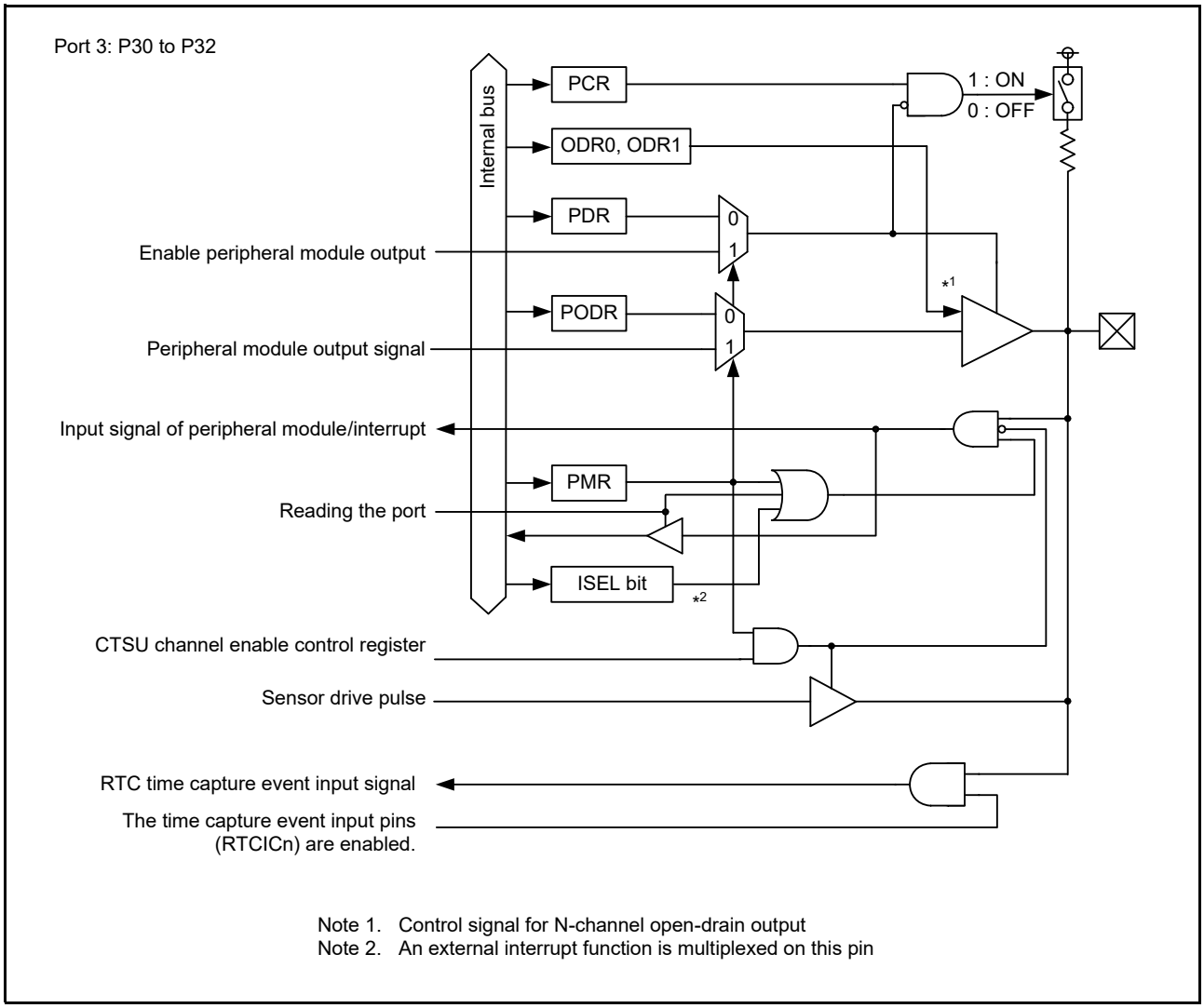
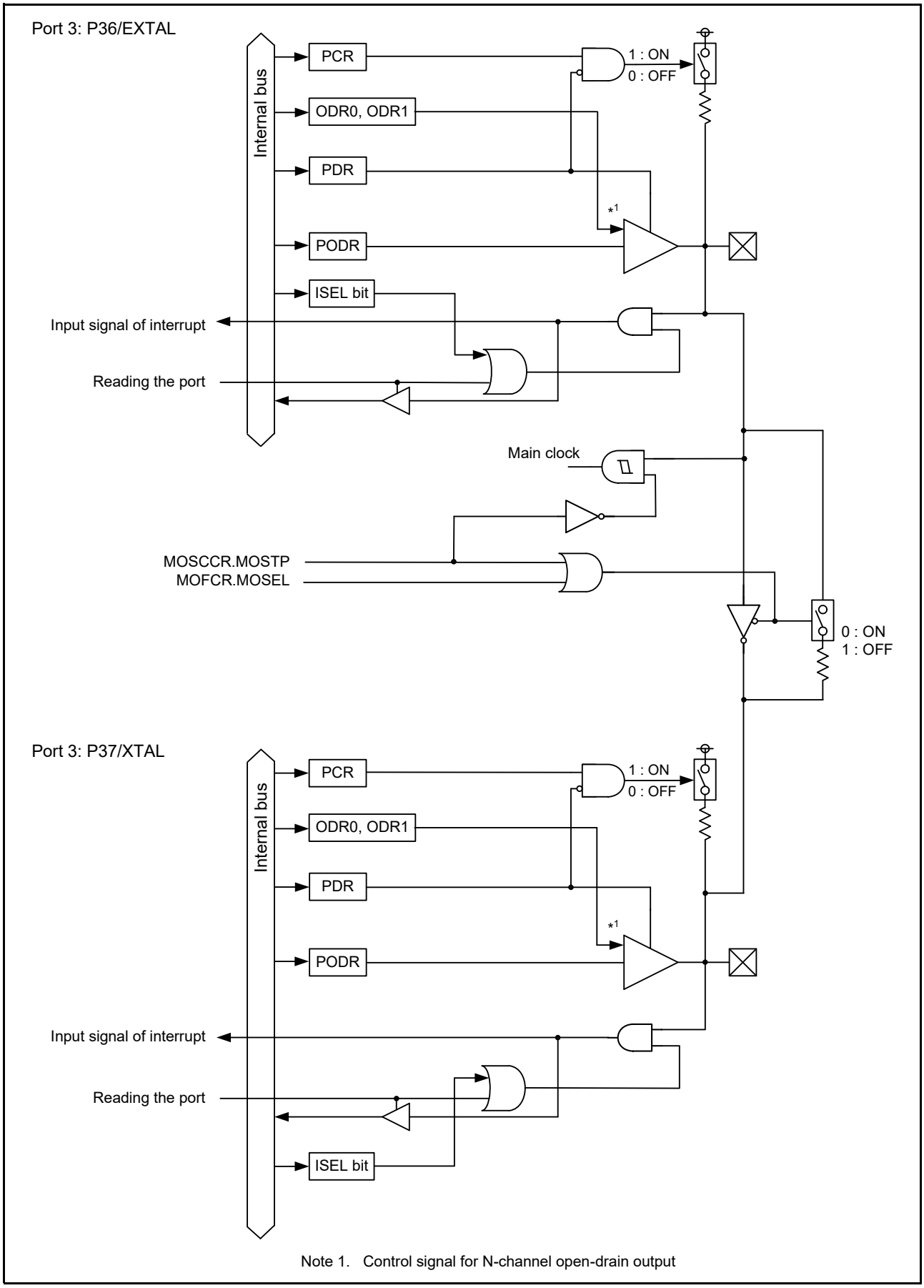


Figure 20.4 I/O Port Configuration (4)



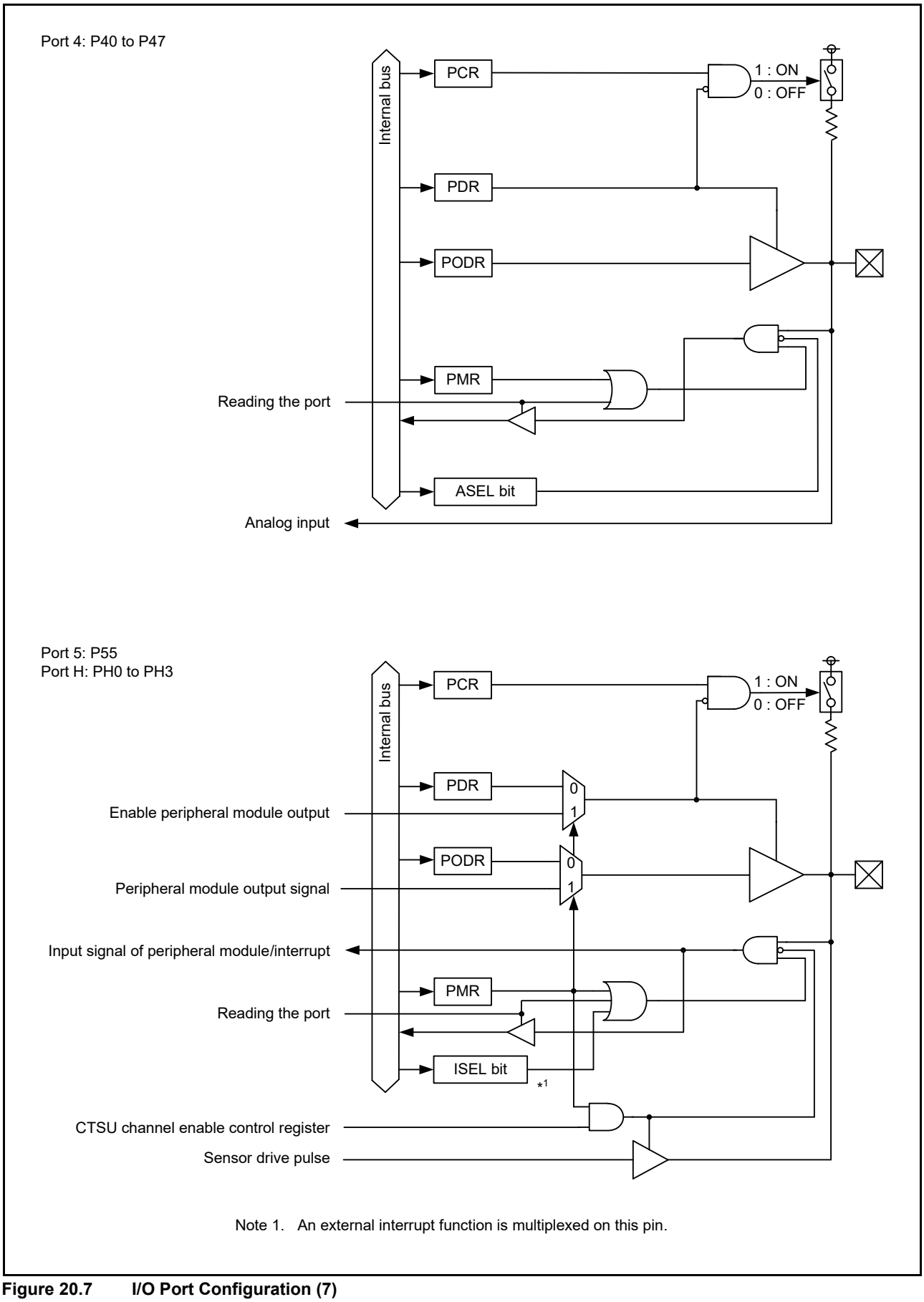
Note 1. Control signal for N-channel open-drain output
Note 2. An external interrupt function is multiplexed on this pin

Figure 20.5 I/O Port Configuration (5)



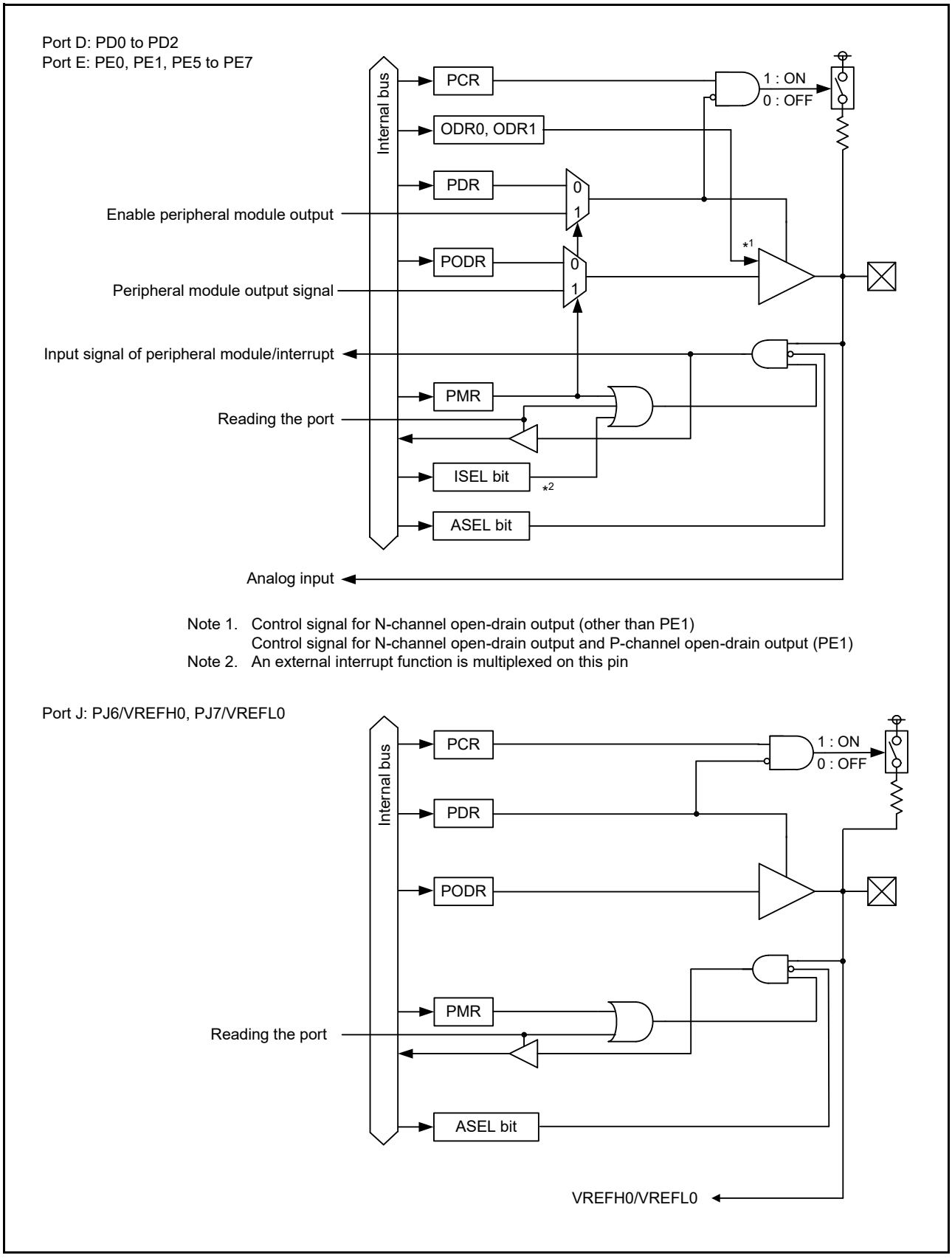
Note 1. Control signal for N-channel open-drain output

Figure 20.6 I/O Port Configuration (6)



Note 1. An external interrupt function is multiplexed on this pin.

Figure 20.7 I/O Port Configuration (7)



Note 1. Control signal for N-channel open-drain output (other than PE1)
Control signal for N-channel open-drain output and P-channel open-drain output (PE1)
Note 2. An external interrupt function is multiplexed on this pin

Figure 20.8 I/O Port Configuration (8)

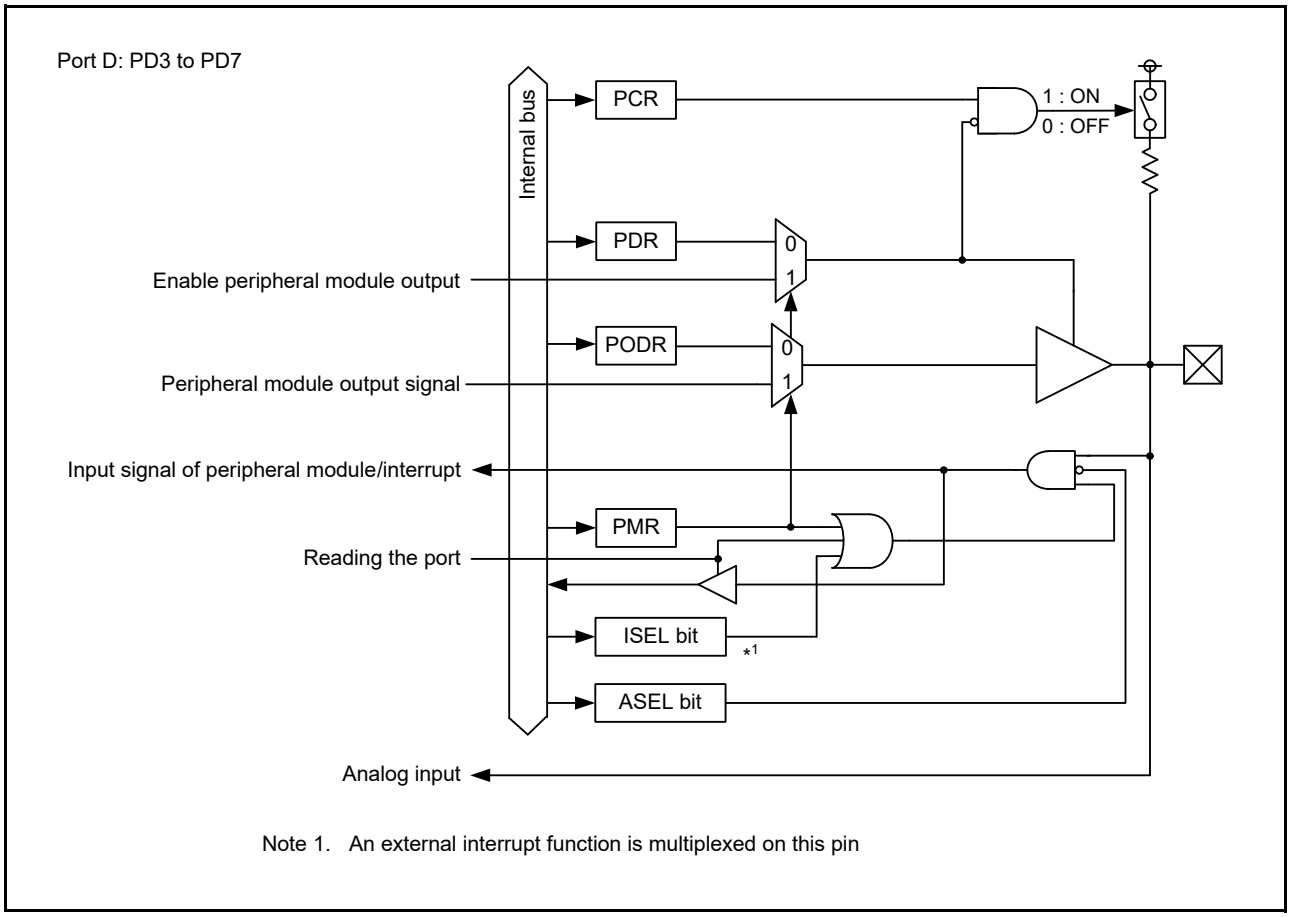


Figure 20.9 I/O Port Configuration (8)

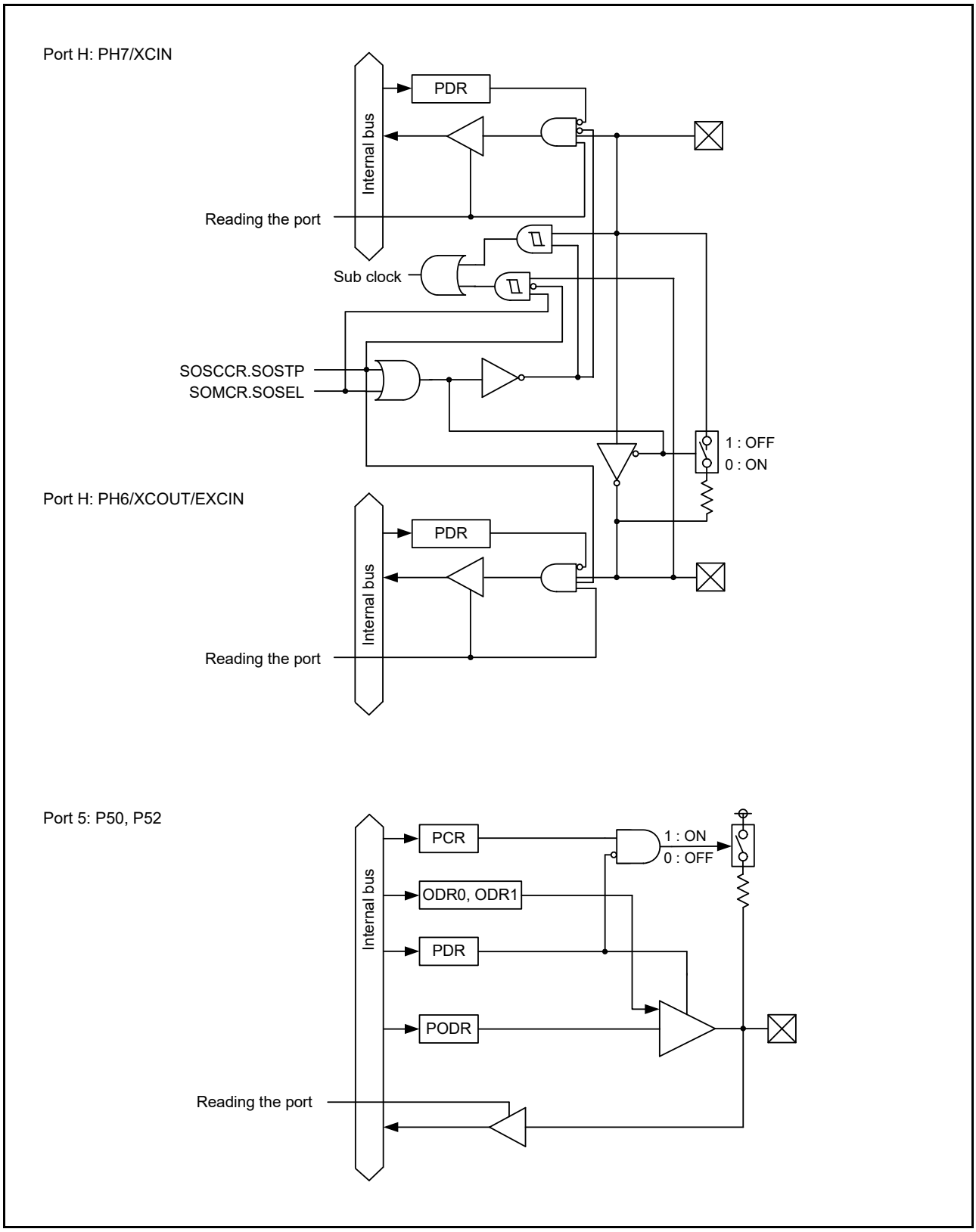


Figure 20.10 I/O Port Configuration (10)

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, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTG.PDR 0008 C010h, PORTH.PDR 0008 C011h, PORTJ.PDR 0008 C012h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 I/O Select	0: Input (Functions as an input pin.) 1: Output (Functions as an output pin.)	R/W
b1	B1	Pm1 I/O Select		R/W
b2	B2	Pm2 I/O Select		R/W
b3	B3	Pm3 I/O Select		R/W
b4	B4	Pm4 I/O Select		R/W
b5	B5	Pm5 I/O Select		R/W
b6	B6	Pm6 I/O Select		R/W
b7	B7	Pm7 I/O Select		R/W

m = 0 to 5, A to E, G, H, J

PDR is used to select the input or output direction for individual pins of the corresponding port m 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.
Write 1 (output) to each bit of PDR corresponding to port m that does not exist.
The PORT3.PDR.B5 bit is reserved, because the P35 pin is input only. A reserved bit is read as 0. The write value should be 0.

20.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTG.PODR 0008 C030h, PORTH.PODR 0008 C031h, PORTJ.PODR 0008 C032h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Data Store	Holds output data.	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 5, A to E, G, H, J

PODR holds the data to be output from the pins used for general output ports.
Bits corresponding to port m on the 100 pin-product but which do not exist on a product with fewer than 100 pins are reserved. Write 0 to these bits.
The PORT3.PODR.B5 bit, the PORTH.PODR.B6 bit, and the PORTH.PODR.B7 bit are reserved, because the P35, PH6, and PH7 pins are input only. The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

20.3.3 Port Input Data 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, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTG.PIDR 0008 C050h, PORTH.PIDR 0008 C051h, PORTJ.PIDR 0008 C052h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	Indicates individual pin states of the corresponding port.	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 5, A to E, G, H, J

PIDR indicates individual pin states of port m.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The NMI pin state is reflected in the P35 bit.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

Note: When using P36 and P37 as general I/O ports, set the MOSCCR.MOSTP bit to 1 (main clock oscillator is stopped).

Note: When using PH6 and PH7 as general input ports, set the SOSCCR.SOSTP bit to 1 (stopping the sub-clock oscillator).

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, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTG.PMR 0008 C070h, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0*1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Use pin as general I/O port. 1: Use pin as I/O port for peripheral functions.	R/W
b1	B1	Pm1 Pin Mode Control		R/W
b2	B2	Pm2 Pin Mode Control		R/W
b3	B3	Pm3 Pin Mode Control		R/W
b4	B4	Pm4 Pin Mode Control		R/W
b5	B5	Pm5 Pin Mode Control		R/W
b6	B6	Pm6 Pin Mode Control		R/W
b7	B7	Pm7 Pin Mode Control	• PG7 0: Use pin as general I/O port. 1: Use the pin as the MD function (initial value). • Other pins 0: Use pin as general I/O port (initial value). 1: Use pin as I/O port for peripheral functions.	R/W

m = 0 to 5, A to E, G, H, J

Note 1. The initial value of the PORTG.PMR.B7 bit is 1.

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.

Bits corresponding to port m on the 100 pin-product but which do not exist on a product with fewer than 100 pins are reserved. Write 0 to these bits.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

20.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT5.ODR0 0008 C08Ah, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTD.ODR0 0008 C09Ah, PORTE.ODR0 0008 C09Ch

	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 Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm1 Output Type Select	• P21, P31, P51, PA1, PB1, PC1, PD1 b2 0: CMOS output 1: N-channel open-drain b3 This bit is read as 0. The write value should be 0. • PE1 b3 b2 0 0: CMOS output 0 1: N-channel open-drain 1 0: P-channel open-drain 1 1: Setting prohibited	R/W
b3	B3			R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm3 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1 to 3, 5, A to E, J

Bits corresponding to port m on the 100 pin-product but which do not exist on a product with fewer than 100 pins are reserved. Write 0 to these bits.

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

20.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT5.ODR1 0008 C08Bh, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTE.ODR1 0008 C09Dh, PORTG.ODR1 0008 C0A1h

	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	Pm4 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm5 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	B3	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm6 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm7 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1 to 3, 5, A to C, E, G

Bits corresponding to port m on the 100 pin-product but which do not exist on a product with fewer than 100 pins are reserved. Write 0 to these bits.

The PORT3.ODR1.B2 bit is reserved, because the P35 pin is input only.

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

20.3.7 Pull-Up 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, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTG.PCR 0008 C0D0h, PORTH.PCR 0008 C0D1h, PORTJ.PCR 0008 C0D2h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0*1	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 5, A to E, G, H, J

Note 1. The initial value of the PORTG.PCR.B7 bit is 1.

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 used as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the settings of the PCR register.

The pull-up resistor is also disabled in the reset state.

The B5 bit in PORT3.PCR is reserved. The bit corresponding to a pin that does not exist is also reserved. A reserved bit is read as 0. The write value should be 0.

20.3.8 Port Switching Register A (PSRA)

Address(es): PORT.PSRA 0008 C121h

	b7	b6	b5	b4	b3	b2	b1	b0
	PSEL7	PSEL6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	PSEL6	PB6/PC0 Switching	0: PB6 general I/O port function is selected 1: PC0 general I/O port function is selected	R/W
b7	PSEL7	PB7/PC1 Switching	0: PB7 general I/O port function is selected 1: PC1 general I/O port function is selected	R/W

Note: The PSRA register is for 80-pin and 64-pin packages.

The PSRA register is used to select either the general I/O functions of PB6 and PB7 or those of PC0 and PC1. When 1 is written to the PSEL6 and PSEL7 bits, port C can be used as an 8-bit port. Figure 20.11 shows the switching general-purpose I/O port by the PSRA register.

As for the I/O functions of the peripheral functions, functions multiplexed with PB6 and PB7 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register.

Rewriting this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

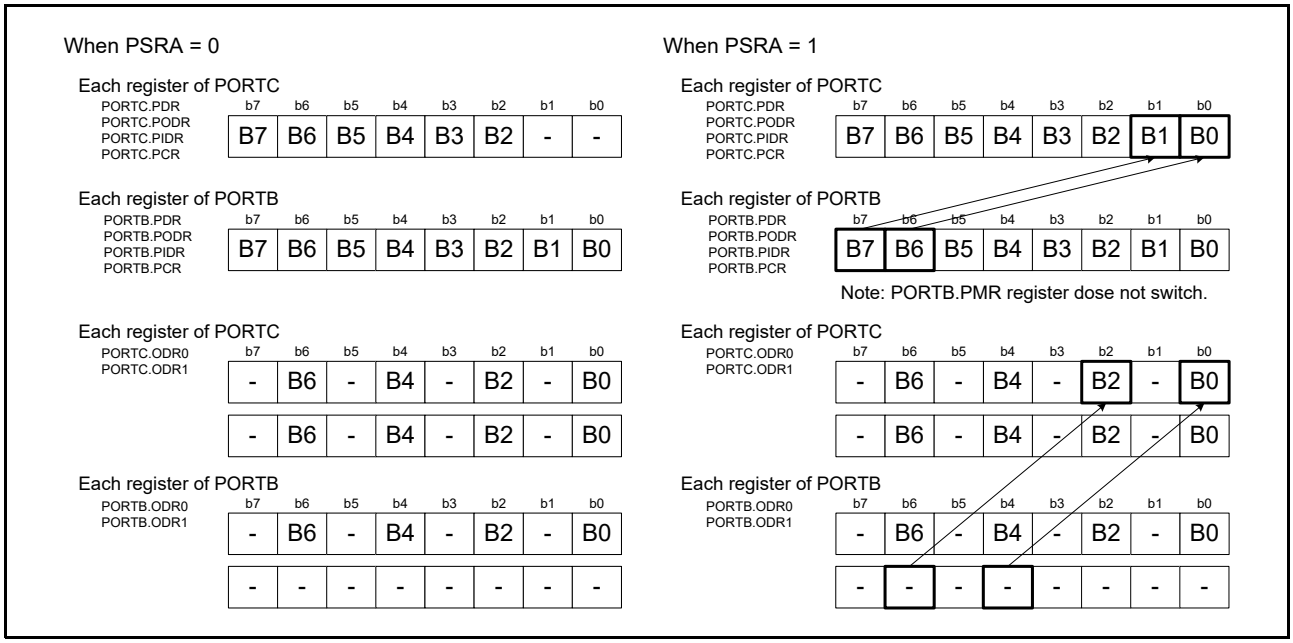


Figure 20.11 Switching General-Purpose I/O Port by the PSRA Register

20.3.9 Port Switching Register B (PSRB)

Address(es): PORT.PSRB 0008 C120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSEL5	—	PSEL3	—	PSEL1	PSEL0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSEL0	PB0/PC0 Switching	0: PB0 general I/O port function is selected 1: PC0 general I/O port function is selected	R/W
b1	PSEL1	PB1/PC1 Switching	0: PB1 general I/O port function is selected 1: PC1 general I/O port function is selected	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PSEL3	PB3/PC2 Switching	0: PB3 general I/O port function is selected 1: PC2 general I/O port function is selected	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	PSEL5	PB5/PC3 Switching	0: PB5 general I/O port function is selected 1: PC3 general I/O port function is selected	R/W
b6, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The PSRB register is for 48-pin packages.

The PSRB register is used to select either the general I/O functions of PB0, PB1, PB3, and PB5 and those of PC0 to PC3. When 1 is written to the PSEL0, PSEL1, PSEL3, and PSEL5 bits, port C can be used as an 8-bit port. Figure 20.12 show the switching general-purpose I/O port by the PSRA register.

As for the I/O functions of the peripheral functions, functions multiplexed with PB0, PB1, PB3, and PB5 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register.

Rewriting this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

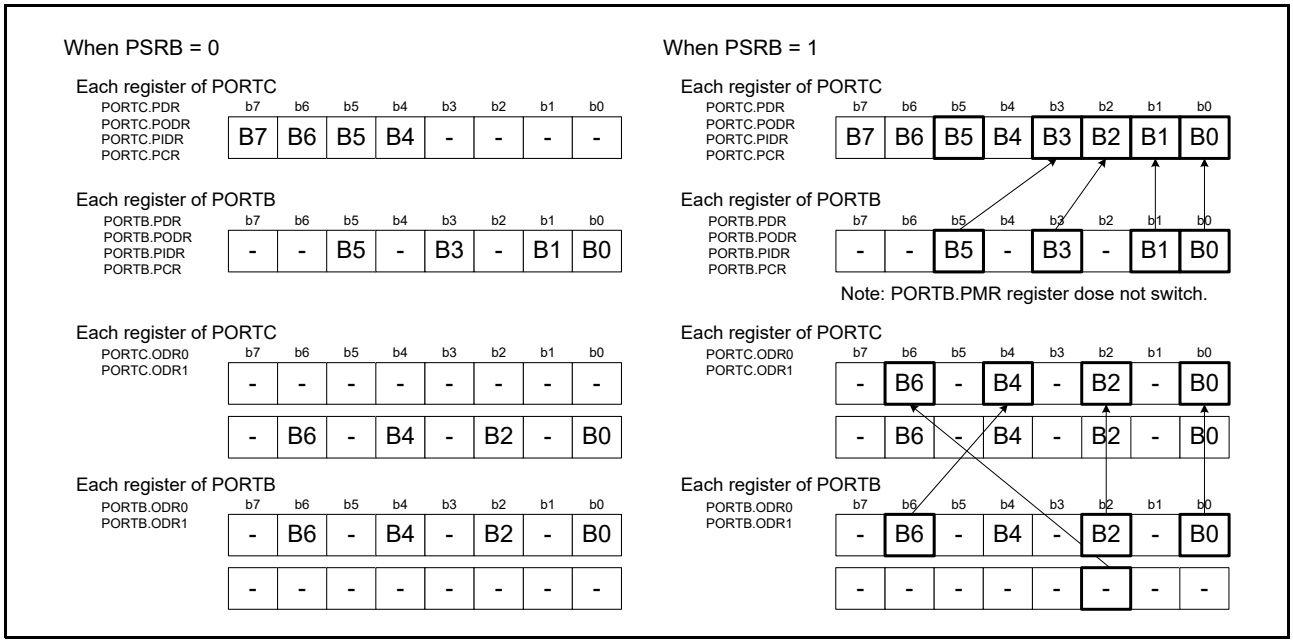


Figure 20.12 Switching General-Purpose I/O Port by the PSRB Register

20.3.10 Port Reading Wait Control Register (PRWCNTR)

Address(es): PORT.PRWCNTR 0008 C122h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAIT[1:0]*1	Port Reading Wait Control	b1 b0 0 0: Setting prohibited 0 1: Insert of one wait cycle 1 0: Insert of two wait cycles (initial value) 1 1: Setting prohibited	R/W
b2 to b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The table below lists the results of the settings of bits 1 and 0 of the PRWCNTR register.

Operating Voltage Range	Operating Frequency Range of PCLKB	WAIT[1:0]	Number of Wait Cycles
2.4 to 5.5 V	to 32 MHz	01	1
		10	2
1.8 to 2.4 V	to 24 MHz	01	1
		10	2
	24 to 32 MHz	01	Setting prohibited
		10	2
1.6 to 1.8 V	to 4 MHz	01	1
		10	2

The PRWCNTR register specifies the number of wait cycles to read the PIDR register.

20.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 20.3 to Table 20.6.

- The blank columns in Table 20.3 to Table 20.6 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 to Table 20.6 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 20.3 to Table 20.6.
  When setting a value to a reserved bit, access in byte units.

Table 20.3 PDR Register Settings in 100-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0						1	1	1
PORT1							1	1
PORT2								
PORT3			0					
PORT4								
PORT5	1	1						
PORTA								
PORTB								
PORTC								
PORTD								
PORTE								
PORTG		1	1	1	1	1	1	1
PORTH*1			1	1				
PORTJ			1	1		1		1

Note 1. RX261 Group products do not have the PH1 and PH2 pins, so set b1 and b2 to 1.

Table 20.4 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
PORTA	1							
PORTB								
PORTC							1	1
PORTD	1	1	1	1	1			
PORTE	1	1						
PORTG		1	1	1	1	1	1	1
PORTH*1			1	1				
PORTJ			1	1	1	1		1

Note 1. RX261 Group products do not have the PH1 and PH2 pins, so set b1 and b2 to 1.

Table 20.5 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
PORTA	1		1			1		
PORTB				1		1		
PORTC							1	1
PORTD	1	1	1	1	1	1	1	1
PORTE	1	1						
PORTG		1	1	1	1	1	1	1
PORTH*1			1	1				
PORTJ			1	1	1	1	1	1

Note 1. RX261 Group products do not have the PH1 and PH2 pins, so set b1 and b2 to 1.

Table 20.6 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
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
PORTG		1	1	1	1	1	1	1
PORTH*1	1	1	1	1				
PORTJ			1	1	1	1	1	1

Note 1. RX261 Group products do not have the PH1 and PH2 pins, so set b1 and b2 to 1.

20.5 Handling of Unused Pins

The configuration of unused pins is listed in Table 20.7.

Table 20.7 Unused Pin Configuration

Pin Name	Description
PG7/MD	(Always used as mode pins)
RES#	Connect this pin to VCC via a pull-up resistor.
USB_DP	Leave this pin open-circuit.
USB_DM	
P35/NMI	Connect this pin to VCC via a pull-up resistor.
P36/EXTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P36). When this pin is not used as port P36, handle as port 1 to 3, 5, A to E, and H.
P37/XTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P37). When this pin is not used as port P37, handle as port 1 to 3, 5, A to E, and H. When the external clock is to be input to the EXTAL pin, pull this pin up to VCC via a resistor, or pull it down to VSS via a resistor.
PH7/XCIN	When not using the sub clock, set the SOSCCR.SOSTP bit to 1 (general purpose port PH7). If not used as port PH7, set the PORTH.PDR.B7 bit to 0 and connect to VSS via a resistor (pull-down) or set it to 1 to leave this pin open-circuit.
PH6/XCOUT	When not using the sub clock, set the SOSCCR.SOSTP bit to 1 (general purpose port PH6). If not used as port PH6, set the PORTH.PDR.B6 bit to 0 and connect to VSS via a resistor (pull-down) or set it to 1 to leave this pin open-circuit.
Ports 1 to 3, 5 Ports A to E, H, J (except for PJ6, PJ7)	• 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), leave these pins open-circuit.*1, *2
Ports 0, 4, J (PJ6, PJ7)	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to AVCC0 (pulled up) via a resistor or to AVSS0 (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), leave these pins open-circuit.*1, *2
PJ6/VREFH0	When this pin is not used as VREFH0, set the PJ6PFS.ASEL bit to 0 (general port PJ6). When this pin is not also used as port PJ6, handle it in the same way as the handling of Ports 0, 4.
PJ7/VREFL0	When this pin is not used as VREFL0, set the PJ7PFS.ASEL bit to 0 (general port PJ7). When this pin is not also used as port PJ7, handle it in the same way as the handling of Ports 0, 4.

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) is used to allocate input and output signals for peripheral modules and input interrupt signals to pins from among multiple ports.

Table 21.1 shows the allocation of pin functions to multiple pins. The symbols ✓ and × in the table indicate whether the pins are or are not present on the given package. Allocating the same function to more than one pin is prohibited.

Table 21.1 Allocation of Pin Functions to Multiple Pins (1 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
Interrupt		NMI (input)	P35	✓	✓	✓	✓
Interrupt	IRQ0	IRQ0 (input)	P30	✓	✓	✓	✓
			PD0	✓	✓	×	×
			PH1	✓	✓	✓	✓
	IRQ1	IRQ1 (input)	P31	✓	✓	✓	✓
			PD1	✓	✓	×	×
			PH2	✓	✓	✓	✓
	IRQ2	IRQ2 (input)	P12	✓	✓	×	×
			P32	✓	✓	✓	×
			P36	✓	✓	✓	✓
			PD2	✓	✓	×	×
	IRQ3	IRQ3 (input)	P13	✓	✓	×	×
			P33	✓	×	×	×
			PD3	✓	×	×	×
	IRQ4	IRQ4 (input)	P14	✓	✓	✓	✓
			P34	✓	✓	×	×
			P37	✓	✓	✓	✓
			PB1	✓	✓	✓	✓
			PD4	✓	×	×	×
	IRQ5	IRQ5 (input)	P15	✓	✓	✓	✓
			PA4	✓	✓	✓	✓
			PD5	✓	×	×	×
			PE5	✓	✓	✓	×
	IRQ6	IRQ6 (input)	P16	✓	✓	✓	✓
			PA3	✓	✓	✓	✓
			PD6	✓	×	×	×
			PE6	✓	×	×	×
	IRQ7	IRQ7 (input)	P17	✓	✓	✓	✓
			PD7	✓	×	×	×
			PE2	✓	✓	✓	✓
			PE7	✓	×	×	×
Clock generation circuit		CLKOUT (output)	PE3	✓	✓	✓	✓
			PE4	✓	✓	✓	✓

Table 21.1 Allocation of Pin Functions to Multiple Pins (2 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
General PWM timer	GPTW0	GTIOC0A (input/output)/ GTIOC0A# (input/output)	P17	✓	✓	✓	✓
			P22	✓	×	×	×
			PA0	✓	✓	✓	×
			PA1	✓	✓	✓	✓
			PB7	✓	✓	✓	×
			PC5	✓	✓	✓	✓
			PH0	✓	✓	✓	✓
		GTIOC0B (input/output)/ GTIOC0B# (input/output)	P16	✓	✓	✓	✓
			P17	✓	✓	✓	✓
			P23	✓	×	×	×
			PA1	✓	✓	✓	✓
			PA6	✓	✓	✓	✓
			PB0	✓	✓	✓	✓
			PB6	✓	✓	✓	×
			PC4	✓	✓	✓	✓
			PH1	✓	✓	✓	✓
	GPTW1	GTIOC1A (input/output)/ GTIOC1A# (input/output)	P24	✓	×	×	×
			P32	✓	✓	✓	×
			P55	✓	✓	✓	×
			PA0	✓	✓	✓	×
			PB3	✓	✓	✓	✓
			PE2	✓	✓	✓	✓
			PE4	✓	✓	✓	✓
		GTIOC1B (input/output)/ GTIOC1B# (input/output)	P25	✓	×	×	×
			P33	✓	×	×	×
			PA3	✓	✓	✓	✓
			PA4	✓	✓	✓	✓
			PB1	✓	✓	✓	✓
			PE1	✓	✓	✓	✓
			PE5	✓	✓	✓	×
			PH2	✓	✓	✓	✓
	GPTW2	GTIOC2A (input/output)/ GTIOC2A# (input/output)	P21	✓	✓	×	×
			P30	✓	✓	✓	✓
			P54	✓	✓	✓	×
			PB0	✓	✓	✓	✓
			PC2	✓	✓	✓	×
			PD1	✓	✓	×	×
			PE3	✓	✓	✓	✓

Table 21.1 Allocation of Pin Functions to Multiple Pins (3 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
General PWM timer	GPTW2	GTIOC2B (input/output)/ GTIOC2B# (input/output)	P20	✓	✓	×	×
			P31	✓	✓	✓	✓
			P55	✓	✓	✓	×
			PA3	✓	✓	✓	✓
			PB1	✓	✓	✓	✓
			PC3	✓	✓	✓	×
			PD2	✓	✓	×	×
			PE4	✓	✓	✓	✓
			PH3	✓	✓	✓	✓
	GPTW3	GTIOC3A (input/output)/ GTIOC3A# (input/output)	P22	✓	×	×	×
			P34	✓	✓	×	×
			PB2	✓	✓	×	×
			PB3	✓	✓	✓	✓
			PC4	✓	✓	✓	✓
		GTIOC3B (input/output)/ GTIOC3B# (input/output)	P13	✓	✓	×	×
			P15	✓	✓	✓	✓
			P23	✓	×	×	×
			PA1	✓	✓	✓	✓
			PB3	✓	✓	✓	✓
	GPTW4	GTIOC4A (input/output)/ GTIOC4A# (input/output)	P20	✓	✓	×	×
			PA4	✓	✓	✓	✓
			PE4	✓	✓	✓	✓
		GTIOC4B (input/output)/ GTIOC4B# (input/output)	P16	✓	✓	✓	✓
			P21	✓	✓	×	×
			PA5	✓	✓	×	×
			PB5	✓	✓	✓	✓
			PE3	✓	✓	✓	✓
	GPTW5	GTIOC5A (input/output)/ GTIOC5A# (input/output)	P26	✓	✓	✓	✓
			PA6	✓	✓	✓	✓
			PB5	✓	✓	✓	✓
		GTIOC5B (input/output)/ GTIOC5B# (input/output)	P15	✓	✓	✓	✓
			P27	✓	✓	✓	✓
			PA7	✓	×	×	×
			PE5	✓	✓	✓	×
	GPTW6	GTIOC6A (input/output)/ GTIOC6A# (input/output)	P14	✓	✓	✓	✓
			P17	✓	✓	✓	✓
			P25	✓	×	×	×
			PB4	✓	✓	×	×
			PC1	✓	×	×	×
			PC7	✓	✓	✓	✓
			PJ1	✓	✓	×	×

Table 21.1 Allocation of Pin Functions to Multiple Pins (4 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
General PWM timer	GPTW6	GTIOC6B (input/output)/ GTIOC6B# (input/output)	P16	✓	✓	✓	✓
			P24	✓	×	×	×
			PB5	✓	✓	✓	✓
			PC0	✓	×	×	×
			PC6	✓	✓	✓	✓
			PJ3	✓	×	×	×
	GPTW7	GTIOC7A (input/output)/ GTIOC7A# (input/output)	P13	✓	✓	×	×
			P32	✓	✓	✓	×
			PB1	✓	✓	✓	✓
			PB6	✓	✓	✓	×
			PC5	✓	✓	✓	✓
		GTIOC7B (input/output)/ GTIOC7B# (input/output)	P14	✓	✓	✓	✓
			P33	✓	×	×	×
			PA3	✓	✓	✓	✓
			PB7	✓	✓	✓	×
	GPTW	GTETRGA (input)	P14	✓	✓	✓	✓
			P24	✓	×	×	×
			PA4	✓	✓	✓	✓
			PC2	✓	✓	✓	×
			PC6	✓	✓	✓	✓
		GTETRGB (input)	P15	✓	✓	✓	✓
			P25	✓	×	×	×
			PA3	✓	✓	✓	✓
			PA6	✓	✓	✓	✓
			PC3	✓	✓	✓	×
			PC7	✓	✓	✓	✓
		GTETRGC (input)	P16	✓	✓	✓	✓
			P22	✓	×	×	×
			PA1	✓	✓	✓	✓
			PB2	✓	✓	×	×
			PC0	✓	×	×	×
			PC4	✓	✓	✓	✓
		GTETRGD (input)	P17	✓	✓	✓	✓
			P23	✓	×	×	×
			PA3	✓	✓	✓	✓
			PB3	✓	✓	✓	✓
			PC1	✓	×	×	×
			PC5	✓	✓	✓	✓
		GTCPPO0 (output)	P14	✓	✓	✓	✓
			P17	✓	✓	✓	✓
			PC1	✓	×	×	×
			PC7	✓	✓	✓	✓
			PJ1	✓	✓	×	×

Table 21.1 Allocation of Pin Functions to Multiple Pins (5 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
General PWM timer	GPTW	GTIU (input)	P34	✓	✓	×	×
			PB3	✓	✓	✓	✓
			PC4	✓	✓	✓	✓
		GTIV (input)	P13	✓	✓	×	×
			P15	✓	✓	✓	✓
			PA1	✓	✓	✓	✓
		GTIW (input)	P32	✓	✓	✓	×
			PB1	✓	✓	✓	✓
			PC5	✓	✓	✓	✓
		GTOULO (output)	P16	✓	✓	✓	✓
			PA6	✓	✓	✓	✓
			PC4	✓	✓	✓	✓
			PH1	✓	✓	✓	✓
		GTOUUP (output)	P17	✓	✓	✓	✓
			PA1	✓	✓	✓	✓
			PC5	✓	✓	✓	✓
			PH0	✓	✓	✓	✓
		GTOVLO (output)	PA3	✓	✓	✓	✓
			PA4	✓	✓	✓	✓
			PB1	✓	✓	✓	✓
			PE1	✓	✓	✓	✓
		GTOVUP (output)	PA0	✓	✓	✓	×
			PB3	✓	✓	✓	✓
			PE2	✓	✓	✓	✓
			PE4	✓	✓	✓	✓
		GTOWLO (output)	P31	✓	✓	✓	✓
			PA3	✓	✓	✓	✓
			PB1	✓	✓	✓	✓
			PE4	✓	✓	✓	✓
		GTOWUP (output)	P30	✓	✓	✓	✓
			PB0	✓	✓	✓	✓
			PC2	✓	✓	✓	×
			PE3	✓	✓	✓	✓
8-bit timer	TMR0	TMO0 (output)	P22	✓	×	×	×
			PB3	✓	✓	✓	✓
			PH1	✓	✓	✓	✓
		TMCI0 (input)	P21	✓	✓	×	×
			PB1	✓	✓	✓	✓
			PH3	✓	✓	✓	✓
		TMRI0 (input)	P20	✓	✓	×	×
			PA4	✓	✓	✓	✓
			PH2	✓	✓	✓	✓

Table 21.1 Allocation of Pin Functions to Multiple Pins (6 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
8-bit timer	TMR1	TMO1 (output)	P17	✓	✓	✓	✓
			P26	✓	✓	✓	✓
		TMCI1 (input)	P12	✓	✓	×	×
			P54	✓	✓	✓	×
			PC4	✓	✓	✓	✓
		TMRI1 (input)	P24	✓	×	×	×
			PB5	✓	✓	✓	✓
	TMR2	TMO2 (output)	P16	✓	✓	✓	✓
			PC7	✓	✓	✓	✓
		TMCI2 (input)	P15	✓	✓	✓	✓
			P31	✓	✓	✓	✓
			PC6	✓	✓	✓	✓
		TMRI2 (input)	P14	✓	✓	✓	✓
			PC5	✓	✓	✓	✓
	TMR3	TMO3 (output)	P13	✓	✓	×	×
			P32	✓	✓	✓	×
			P55	✓	✓	✓	×
		TMCI3 (input)	P27	✓	✓	✓	✓
			P34	✓	✓	×	×
			PA6	✓	✓	✓	✓
		TMRI3 (input)	P30	✓	✓	✓	✓
			P33	✓	×	×	×
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	✓	✓	✓	✓
	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 Allocation of Pin Functions to Multiple Pins (7 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
Serial communications interface	SCI6	RXD6 (input)/ SMISO6 (input/output)/ SSCL6 (input/output)	P33	✓	×	×	×
			PB0	✓	✓	✓	✓
			PD1	✓	✓	×	×
		TXD6 (output)/ SMOSI6 (input/output)/ SSDA6 (input/output)	P32	✓	✓	✓	×
			PB1	✓	✓	✓	✓
			PD0	✓	✓	×	×
		SCK6 (input/output)	P34	✓	✓	×	×
			PB3	✓	✓	✓	✓
			PD2	✓	✓	×	×
		CTS6# (input)/ RTS6# (output)/ SS6# (input)	PB2	✓	✓	×	×
			PJ3	✓	×	×	×
	SCI12	RXD12 (input)/ SMISO12 (input/output)/ SSCL12 (input/output)/ RXDX12 (input)	PE2	✓	✓	✓	✓ (SMISO12 function is not available)
		TXD12 (output)/ SMOSI12 (input/output)/ SSDA12 (input/output)/ TXDX12 (output)/ SIOX12 (input/output)	PE1	✓	✓	✓	✓ (SMOSI12 function is not available)
		SCK12 (input/output)	PE0	✓	✓	✓	×
		CTS12# (input)/RTS12# (output)/SS12# (input)	PE3	✓	✓	✓	✓ (SS12# function is not available)
Serial communications interface	RSCI0	RXD000 (input)/ SMISO000 (input/output)/ SSCL000 (input/output)	P21	✓	✓	×	×
		TXD000 (output)/ TXDA000 (output)/ SMOSI000 (input/output)/ SSDA000 (input/output)	P20	✓	✓ (TXDA000 function is not available)	×	×
		SCK000 (input/output)	P22	✓	×	×	×
		TXDB000 (output)	P22	✓	×	×	×
		CTS000# (input)/ RTS000# (output)/ SS000# (input)	P23	✓	×	×	×
		DE000 (output)	P23	✓	×	×	×
	RSCI8	RXD008 (input)/ SMISO008 (input/output)/ SSCL008 (input/output)	PC6	✓	✓	✓	✓
		TXD008 (output)/ TXDA008 (output)/ SMOSI008 (input/output)/ SSDA008 (input/output)	PC7	✓	✓	✓	✓
		SCK008 (input/output)	PC5	✓	✓	✓	✓
		TXDB008 (output)	PC5	✓	✓	✓	✓
		CTS008# (input)/ RTS008# (output)/ SS008# (input)	PC4	✓	✓	✓	✓
		DE008 (output)	PC4	✓	✓	✓	✓

Table 21.1 Allocation of Pin Functions to Multiple Pins (8 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
Serial communications interface	RSCI9	RXD009 (input)/ SMISO009 (input/output)/ SSCL009 (input/output)	PB6	✓	✓	✓	×
		TXD009 (output)/ TXDA009 (output)/ SMOSI009 (input/output)/ SSDA009 (input/output)	PB7	✓	✓	✓	×
		SCK009 (input/output)	PB5	✓	✓	✓	×
		TXDB009 (output)	PB5	✓	✓	✓	×
		CTS009# (input)/ RTS009# (output)/ SS009# (input)	PB4	✓	✓	×	×
		DE009 (output)	PB4	✓	✓	×	×
I^2C bus interface	RIIC0	SCL0 (input/output)	P12	✓	✓	×	×
			P16	✓	✓	✓	✓
		SDA0 (input/output)	P13	✓	✓	×	×
			P17	✓	✓	✓	✓
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	✓	✓	✓	×
Realtime clock		RTCOUT (output)	P16	✓	✓	✓	×
			P32	✓	✓	✓	×
		RTCIC0 (input)*1	P30	✓	✓	✓	×
		RTCIC1 (input)*1	P31	✓	✓	✓	×
		RTCIC2 (input)*1	P32	✓	✓	✓	×
Low-Power Timer	LPT	LPTO (output)	P26	✓	✓	✓	✓
			PB3	✓	✓	✓	✓
			PC7	✓	✓	✓	✓

Table 21.1 Allocation of Pin Functions to Multiple Pins (9 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
CANFD module	CANFD0	CTX0 (output)	P14	✓	✓	✓	✓
			P32	✓	✓	✓	×
			P54	✓	✓	✓	×
			PD1	✓	✓	×	×
		CRX0 (input)	P15	✓	✓	✓	✓
			P33	✓	×	×	×
			P55	✓	✓	✓	×
			PD2	✓	✓	×	×
USB 2.0 FS host/function module	USB0	USB0_DP (input/output)	PH1*2	✓	✓	✓	✓
		USB0_DM (input/output)	PH2*2	✓	✓	✓	✓
		USB0_VBUS (input)	P16	✓	✓	✓	✓
			PB5	✓	✓	✓	✓
		USB0_EXICEN (output)	P21	✓	✓	×	×
			PC6	✓	✓	✓	✓
		USB0_VBUSEN (output)	P16	✓	✓	✓	✓
			P24	✓	×	×	×
			P26	✓	✓	✓	✓
			P32	✓	✓	✓	×
		USB0_OVRCURA (input)	P14	✓	✓	✓	✓
		USB0_OVRCURB (input)	P16	✓	✓	✓	✓
			P22	✓	×	×	×
		USB0_ID (input)	P20	✓	✓	×	×
			PC5	✓	✓	✓	✓
12-bit A/D converter		AN000 (input)*1	P40	✓	✓	✓	✓
		AN001 (input)*1	P41	✓	✓	✓	✓
		AN002 (input)*1	P42	✓	✓	✓	✓
		AN003 (input)*1	P43	✓	✓	✓	×
		AN004 (input)*1	P44	✓	✓	✓	×
		AN005 (input)*1	P45	✓	✓	✓	✓
		AN006 (input)*1	P46	✓	✓	✓	✓
		AN007 (input)*1	P47	✓	✓	✓	✓
		AN016 (input)*1	PE0	✓	✓	✓	×
		AN017 (input)*1	PE1	✓	✓	✓	✓
		AN018 (input)*1	PE2	✓	✓	✓	✓
		AN019 (input)*1	PE3	✓	✓	✓	✓
		AN020 (input)*1	PE4	✓	✓	✓	✓
		AN021 (input)*1	PE5	✓	✓	✓	×
		AN022 (input)*1	PE6	✓	×	×	×
		AN023 (input)*1	PE7	✓	×	×	×
		AN024 (input)*1	PD0	✓	✓	×	×
		AN025 (input)*1	PD1	✓	✓	×	×
		AN026 (input)*1	PD2	✓	✓	×	×
		AN027 (input)*1	PD3	✓	×	×	×
		AN028 (input)*1	PD4	✓	×	×	×

Table 21.1 Allocation of Pin Functions to Multiple Pins (10 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
12-bit A/D converter		AN029 (input)*1	PD5	✓	×	×	×
		AN030 (input)*1	PD6	✓	×	×	×
		AN031 (input)*1	PD7	✓	×	×	×
		ADTRG0# (input)	P07	✓	✓	×	×
			P16	✓	✓	✓	✓
			P25	✓	×	×	×
D/A converter		DA0 (output)*1	P03	✓	✓	✓	×
		DA1 (output)*1	P05	✓	✓	✓	×
Clock frequency accuracy measurement circuit		CACREF (input)	PA0	✓	✓	✓	×
			PC7	✓	✓	✓	✓
			PH0	✓	✓	✓	✓
LVD voltage detection input		CMPA2 (input)*1	PE4	✓	✓	✓	✓
Comparator B		CMPB0 (input)*1	PE1	✓	✓	✓	✓
		CVREFB0 (input)*1	PE2	✓	✓	✓	✓
		CMPOB0 (output)	PE5	✓	✓	✓	×
		CMPB1 (input)*1	PA3	✓	✓	✓	✓
		CVREFB1 (input)*1	PA4	✓	✓	✓	✓
		CMPOB1 (output)	PB1	✓	✓	✓	✓
Capacitive touch sensing unit (CTSU)		TSCAP (—)	PC4	✓	✓	✓	✓
		TS0 (input/output)	P32	✓	✓	✓	×
		TS1 (input/output)	P31	✓	✓	✓	✓
		TS2 (input/output)	P30	✓	✓	✓	✓
		TS3 (input/output)	P27	✓	✓	✓	✓
		TS4 (input/output)	P26	✓	✓	✓	✓
		TS5 (input/output)	P15	✓	✓	✓	✓
		TS6 (input/output)	P14	✓	✓	✓	✓
		TS7 (input/output)	PH3	✓	✓	✓	✓
		TS8 (input/output)	PH2	✓	✓	✓	✓
		TS9 (input/output)	PH1	✓	✓	✓	✓
		TS10 (input/output)	PH0	✓	✓	✓	✓
		TS11 (input/output)	P55	✓	✓	✓	×
		TS12 (input/output)	P54	✓	✓	✓	×
		TS13 (input/output)	PC7	✓	✓	✓	✓
		TS14 (input/output)	PC6	✓	✓	✓	✓
		TS15 (input/output)	PC5	✓	✓	✓	✓
		TS16 (input/output)	PC3	✓	✓	✓	×
		TS17 (input/output)	PC2	✓	✓	✓	×
		TS18 (input/output)	PB7	✓	✓	✓	×
		TS19 (input/output)	PB6	✓	✓	✓	×
		TS20 (input/output)	PB5	✓	✓	✓	✓
		TS21 (input/output)	PB4	✓	✓	×	×
		TS22 (input/output)	PB3	✓	✓	✓	✓
		TS23 (input/output)	PB2	✓	✓	×	×
		TS24 (input/output)	PB1	✓	✓	✓	✓

Table 21.1 Allocation of Pin Functions to Multiple Pins (11 / 11)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				100-pin	80-pin	64-pin	48-pin
Capacitive touch sensing unit (CTSU)		TS25 (input/output)	PB0	✓	✓	✓	✓
		TS26 (input/output)	PA6	✓	✓	✓	✓
		TS27 (input/output)	PA5	✓	✓	×	×
		TS28 (input/output)	PA4	✓	✓	✓	✓
		TS29 (input/output)	PA3	✓	✓	✓	✓
		TS30 (input/output)	PA2	✓	✓	×	×
		TS31 (input/output)	PA1	✓	✓	✓	✓
		TS32 (input/output)	PA0	✓	✓	✓	×
		TS33 (input/output)	PE4	✓	✓	✓	✓
		TS34 (input/output)	PE3	✓	✓	✓	✓
		TS35 (input/output)	PE2	✓	✓	✓	✓
Remote control signal receiver	REMC0	PMC0 (input)	P51	✓	×	×	×
			P53	✓	×	×	×
			PB3	✓	✓	✓	✓
			PC3	✓	✓	✓	×
			PC4	✓	✓	✓	✓
			PC5	✓	✓	✓	✓

Note 1. Select general input (by setting the Bm bits for the given pin in the PDR and PMR for the given port to 0) for the pin if this pin function is to be used.

Note 2. These pins function as PH1 and PH2 in the RX260, or as USB0_DP and USB0_DM in the RX261.

21.2 Register Descriptions

Registers and bits for pins that are not present due to differences according to the package are reserved. Write the value after a reset when writing to such bits.

21.2.1 Write-Protect Register (PWPR)

Address(es): MPC.PWPR 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 write 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 = 3, 5, 7)

Address(es): MPC.P03PFS 0008 C143h, MPC.P05PFS 0008 C145h, MPC.P07PFS 0008 C147h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.2.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin 1: Used as analog pin P03: DA0 (100/80/64 pins) P05: DA1 (100/80/64 pins)	R/W

The Pmn pin function control register (PmnPFS) selects the pin function.
Bits PSEL[4:0] select the peripheral function assigned to each port pin.
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 100-Pin and 80-Pin

	Pin
PSEL[4:0] Settings	**P07**
00000b (initial value)	Hi-Z
01001b	ADTRG0#

—: Do not specify this value.

21.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 2 to 7)

Address(es): MPC.P12PFS 0008 C14Ah, MPC.P13PFS 0008 C14Bh, MPC.P14PFS 0008 C14Ch, MPC.P15PFS 0008 C14Dh, MPC.P16PFS 0008 C14Eh, MPC.P17PFS 0008 C14Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.3 and Table 21.4.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P12: IRQ2 input switch (100/80 pins) P13: IRQ3 input switch (100/80 pins) P14: IRQ4 input switch (100/80/64/48 pins) P15: IRQ5 input switch (100/80/64/48 pins) P16: IRQ6 input switch (100/80/64/48 pins) P17: IRQ7 input switch (100/80/64/48 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.3 Register Settings for Input/Output Pin Function in 100-Pin and 80-Pin

	Pin					
PSEL[4:0] Settings	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**
00000b (initial value)	Hi-Z					
00001b	—	—	GTCPPO0	—	GTIOC6B#	GTIOC6A#
00010b	—	—	—	—	GTETRGC	GTETRGD
00011b	—	GTIV	—	GTIV	GTOULO	GTCPPO0
00100b	—	—	—	—	—	GTOUUP
00101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
00111b	—	—	—	—	RTCOUT	—
01001b	—	—	—	—	ADTRG0#	—
01010b	—	—	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	CTS1# RTS1# SS1#	—	—	—
01101b	—	—	—	—	MOSIA	MISOA
01111b	SCL0	SDA0	—	—	SCL0	SDA0
10000b	—	—	CTX0	CRX0	—	—
10001b	—	—	USB0_OVRCURA	—	USB0_VBUS	—
10010b	—	—	—	—	USB0_VBUSEN	—
10011b	—	—	—	—	USB0_OVRCURB	—
10100b	—	GTIOC3B	GTIOC6A	GTIOC3B	GTIOC0B	GTIOC0A
10101b	—	GTIOC7A	GTIOC7B	GTIOC5B	GTIOC4B	GTIOC0B
10110b	—	GTIOC3B#	GTIOC6A#	GTIOC3B#	GTIOC0B#	GTIOC0A#
10111b	—	GTIOC7A#	GTIOC7B#	GTIOC5B#	GTIOC4B#	GTIOC0B#
11000b	—	—	GTETRGA	GTETRGB	GTIOC6B	GTIOC6A
11001b	—	—	TS6	TS5	—	—

—: Do not specify this value.

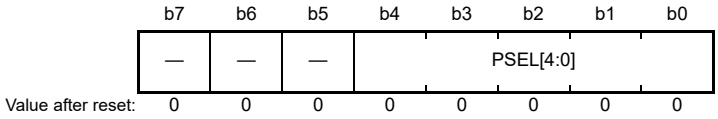
Table 21.4 Register Settings for Input/Output Pin Function in 64-Pin and 48-Pin

	Pin			
PSEL[4:0] Settings	**P14**	**P15**	**P16**	**P17**
00000b (initial value)	Hi-Z			
00001b	GTCPPO0	—	GTIOC6B#	GTIOC6A#
00010b	—	—	GTETRGC	GTETRGD
00011b	—	GTIV	GTOULO	GTCPPO0
00100b	—	—	—	GTOUUP
00101b	TMRI2	TMCI2	TMO2	TMO1
00111b	—	—	RTCOUT*1	—
01001b	—	—	ADTRG0#	—
01010b	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	CTS1# RTS1# SS1#	—	—	—
01101b	—	—	MOSIA	MISOA
01111b	—	—	SCL0	SDA0
10000b	CTX0	CRX0	—	—
10001b	USB0_OVRCURA	—	USB0_VBUS	—
10010b	—	—	USB0_VBUSEN	—
10011b	—	—	USB0_OVRCURB	—
10100b	GTIOC6A	GTIOC3B	GTIOC0B	GTIOC0A
10101b	GTIOC7B	GTIOC5B	GTIOC4B	GTIOC0B
10110b	GTIOC6A#	GTIOC3B#	GTIOC0B#	GTIOC0A#
10111b	GTIOC7B#	GTIOC5B#	GTIOC4B#	GTIOC0B#
11000b	GTETRGA	GTETRGB	GTIOC6B	GTIOC6A
11001b	TS6	TS5	—	—

—: Do not specify this value.
Note 1. This pin is not present in the 48-pin products.

21.2.4 P2n Pin Function Control Register (P2nPFS) (n = 0 to 7)

Address(es): MPC.P20PFS 0008 C150h, MPC.P21PFS 0008 C151h, MPC.P22PFS 0008 C152h, MPC.P23PFS 0008 C153h, MPC.P24PFS 0008 C154h, MPC.P25PFS 0008 C155h, MPC.P26PFS 0008 C156h, MPC.P27PFS 0008 C157h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.5 to Table 21.7.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.5 Register Settings for Input/Output Pin Function in 100-Pin

	Pin							
PSEL[4:0] Settings	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**
00000b (initial value)	Hi-Z							
00101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
01001b	—	—	—	—	—	ADTRG0#	—	—
01010b	TXD000 TXDA000 SMOSI000 SSDA000	RXD000 SMISO000 SSCL000	SCK000	—	—	—	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	CTS000# RTS000# SS000#	—	—	—	—
01100b	—	—	TXDB000	DE000	—	—	—	—
10001b	USB0_ID	USB0_EXICEN	USB0_OVRCURB	—	USB0_VBUSEN	—	USB0_VBUSEN	—
10100b	GTIOC2B	GTIOC2A	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC5A	GTIOC5B
10101b	GTIOC4A	GTIOC4B	GTIOC3A	GTIOC3B	GTIOC6B	GTIOC6A	—	—
10110b	GTIOC2B#	GTIOC2A#	GTIOC0A#	GTIOC0B#	GTIOC1A#	GTIOC1B#	GTIOC5A#	GTIOC5B#
10111b	GTIOC4A#	GTIOC4B#	GTIOC3A#	GTIOC3B#	GTIOC6B#	GTIOC6A#	—	—
11000b	—	—	GTETRGC	GTETRGD	GTETRGA	GTETRGB	—	—
11001b	—	—	—	—	—	—	TS4	TS3
11011b	—	—	—	—	—	—	LPTO	—

—: Do not specify this value.

Table 21.6 Register Settings for Input/Output Pin Function in 80-Pin

	Pin			
PSEL[4:0] Settings	**P20**	**P21**	**P26**	**P27**
00000b (initial value)	Hi-Z			
00101b	TMRI0	TMCI0	TMO1	TMCI3
01010b	TXD000 SMOSI000 SSDA000	RXD000 SMISO000 SSCL000	TXD1 SMOSI1 SSDA1	SCK1
10001b	USB0_ID	USB0_EXICEN	USB0_VBUSEN	—
10100b	GTIOC2B	GTIOC2A	GTIOC5A	GTIOC5B
10101b	GTIOC4A	GTIOC4B	—	—
10110b	GTIOC2B#	GTIOC2A#	GTIOC5A#	GTIOC5B#
10111b	GTIOC4A#	GTIOC4B#	—	—
11001b	—	—	TS4	TS3
11011b	—	—	LPTO	—

—: Do not specify this value.

Table 21.7 Register Settings for Input/Output Pin Function in 64-Pin and 48-Pin

	Pin	
PSEL[4:0] Settings	**P26**	**P27**
00000b (initial value)	Hi-Z	
00101b	TMO1	TMCI3
01010b	TXD1 SMOSI1 SSDA1	SCK1
10001b	USB0_VBUSEN	—
10100b	GTIOC5A	GTIOC5B
10110b	GTIOC5A#	GTIOC5B#
11001b	TS4	TS3
11011b	LPTO	—

—: Do not specify this value.

21.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 4, 6, 7)

Address(es): MPC.P30PFS 0008 C158h, MPC.P31PFS 0008 C159h, MPC.P32PFS 0008 C15Ah, MPC.P33PFS 0008 C15Bh, MPC.P34PFS 0008 C15Ch, MPC.P36PFS 0008 C15Eh, MPC.P37PFS 0008 C15Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.8 to Table 21.11.	R/W
b5	—	Reserved	This bit is 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 P30: IRQ0 input switch (100/80/64/48 pins) P31: IRQ1 input switch (100/80/64/48 pins) P32: IRQ2 input switch (100/80/64 pins) P33: IRQ3 input switch (100 pins) P34: IRQ4 input switch (100/80 pins) P36: IRQ2 input switch (80/64/48 pins) P37: IRQ4 input switch (80/64/48 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.8 Register Settings for Input/Output Pin Function in 100-Pin

	Pin				
PSEL[4:0] Settings	**P30**	**P31**	**P32**	**P33**	**P34**
00000b (initial value)	Hi-Z				
00001b	—	—	—	—	GTIU
00011b	GTOWUP	GTOWLO	GTIW	—	—
00101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
00111b	—	—	RTCOUT	—	—
01010b	RXD1 SMISO1 SSCL1	—	—	—	—
01011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
10000b	—	—	CTX0	CRX0	—
10001b	—	—	USB0_VBUSEN	—	—
10100b	GTIOC2A	GTIOC2B	GTIOC1A	GTIOC1B	GTIOC3A
10101b	—	—	GTIOC7A	GTIOC7B	—
10110b	GTIOC2A#	GTIOC2B#	GTIOC1A#	GTIOC1B#	GTIOC3A#
10111b	—	—	GTIOC7A#	GTIOC7B#	—
11001b	TS2	TS1	TS0	—	—

—: Do not specify this value.

Table 21.9 Register Settings for Input/Output Pin Function in 80-Pin

	Pin			
PSEL[4:0] Settings	**P30**	**P31**	**P32**	**P34**
00000b (initial value)	Hi-Z			
00001b	—	—	—	GTIU
00011b	GTOWUP	GTOWLO	GTIW	—
00101b	TMRI3	TMCI2	TMO3	TMCI3
00111b	—	—	RTCOUT	—
01010b	RXD1 SMISO1 SSCL1	—	—	—
01011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6	SCK6
10000b	—	—	CTX0	—
10001b	—	—	USB0_VBUSEN	—
10100b	GTIOC2A	GTIOC2B	GTIOC1A	GTIOC3A
10101b	—	—	GTIOC7A	—
10110b	GTIOC2A#	GTIOC2B#	GTIOC1A#	GTIOC3A#
10111b	—	—	GTIOC7A#	—
11001b	TS2	TS1	TS0	—

—: Do not specify this value.

Table 21.10 Register Settings for Input/Output Pin Function in 64-Pin

	Pin		
PSEL[4:0] Settings	**P30**	**P31**	**P32**
00000b (initial value)	Hi-Z		
00011b	GTOWUP	GTOWLO	GTIW
00101b	TMRI3	TMCI2	TMO3
00111b	—	—	RTCOUT
01010b	RXD1 SMISO1 SSCL1	—	—
01011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6
10000b	—	—	CTX0
10001b	—	—	USB0_VBUSEN
10100b	GTIOC2A	GTIOC2B	GTIOC1A
10101b	—	—	GTIOC7A
10110b	GTIOC2A#	GTIOC2B#	GTIOC1A#
10111b	—	—	GTIOC7A#
11001b	TS2	TS1	TS0

—: Do not specify this value.

Table 21.11 Register Settings for Input/Output Pin Function in 48-Pin

	Pin	
PSEL[4:0] Settings	**P30**	**P31**
00000b (initial value)	Hi-Z	
00011b	GTOWUP	GTOWLO
00101b	TMRI3	TMCI2
01010b	RXD1 SMISO1 SSCL1	—
01011b	—	CTS1# RTS1# SS1#
10100b	GTIOC2A	GTIOC2B
10110b	GTIOC2A#	GTIOC2B#
11001b	TS2	TS1

—: Do not specify this value.

21.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7)

Address(es): MPC.P40PFS 0008 C160h, MPC.P41PFS 0008 C161h, MPC.P42PFS 0008 C162h, MPC.P43PFS 0008 C163h, MPC.P44PFS 0008 C164h, MPC.P45PFS 0008 C165h, MPC.P46PFS 0008 C166h, MPC.P47PFS 0008 C167h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	—	—	—	—	—
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	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin P40: AN000 (100/80/64/48 pins) P41: AN001 (100/80/64/48 pins) P42: AN002 (100/80/64/48 pins) P43: AN003 (100/80/64 pins) P44: AN004 (100/80/64 pins) P45: AN005 (100/80/64/48 pins) P46: AN006 (100/80/64/48 pins) P47: AN007 (100/80/64/48 pins)	R/W

21.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 1, 3 to 5)

Address(es): MPC.P51PFS 0008 C169h, MPC.P53PFS 0008 C16Bh, MPC.P54PFS 0008 C16Ch, MPC.P55PFS 0008 C16Dh

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.12 and Table 21.13.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.12 Register Settings for Input/Output Pin Function in 100-Pin

	Pin			
PSEL[4:0] Settings	**P51**	**P53**	**P54**	**P55**
00000b (initial value)	Hi-Z			
00101b	—	—	TMCI1	TMO3
10000b	—	—	CTX0	CRX0
10100b	—	—	GTIOC2A	GTIOC1A
10101b	—	—	—	GTIOC2B
10110b	—	—	GTIOC2A#	GTIOC1A#
10111b	—	—	—	GTIOC2B#
11001b	—	—	TS12	TS11
11100b	PMC0	PMC0	—	—

—: Do not specify this value.

Table 21.13 Register Settings for Input/Output Pin Function in 80-Pin and 64-Pin

	Pin	
PSEL[4:0] Settings	**P54**	**P55**
00000b (initial value)	Hi-Z	
00101b	TMCI1	TMO3
10000b	CTX0	CRX0
10100b	GTIOC2A	GTIOC1A
10101b	—	GTIOC2B
10110b	GTIOC2A#	GTIOC1A#
10111b	—	GTIOC2B#
11001b	TS12	TS11

—: Do not specify this value.

21.2.8 PAn Pin Function Control Registers (PAnPFS) (n = 0 to 7)

Address(es): MPC.PA0PFS 0008 C190h, MPC.PA1PFS 0008 C191h, MPC.PA2PFS 0008 C192h, MPC.PA3PFS 0008 C193h, MPC.PA4PFS 0008 C194h, MPC.PA5PFS 0008 C195h, MPC.PA6PFS 0008 C196h, MPC.PA7PFS 0008 C197h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.14 to Table 21.17.	R/W
b5	—	Reserved	This bit is 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 PA3: IRQ6 input switch (100/80/64/48 pins) PA4: IRQ5 input switch (100/80/64/48 pins)	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin PA3: CMPB1 (100/80/64/48 pins) PA4: CVREFB1 (100/80/64/48 pins)	R/W

Table 21.14 Register Settings for Input/Output Pin Function in 100-Pin

	Pin							
PSEL[4:0] Settings	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
00000b (initial value)	Hi-Z							
00001b	—	GTIOC3B#	—	GTIOC7B#	—	—	—	—
00010b	GTOVUP	GTETRGC	—	GTETRGB	GTOVLO	—	—	—
00011b	—	GTIV	—	GTETRGD	—	—	GTOULO	—
00100b	—	GTOUUP	—	GTOVLO	—	—	—	—
00101b	—	—	—	—	TMRI0	—	TMCI3	—
00111b	CACREF	—	—	—	—	—	—	—
01000b	—	—	—	GTOWLO	—	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—	—
01011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10100b	GTIOC0A	GTIOC0A	—	GTIOC1B	GTIOC1B	GTIOC4B	GTIOC0B	GTIOC5B
10101b	GTIOC1A	GTIOC0B	—	GTIOC2B	GTIOC4A	—	GTIOC5A	—
10110b	GTIOC0A#	GTIOC0A#	—	GTIOC1B#	GTIOC1B#	GTIOC4B#	GTIOC0B#	GTIOC5B#
10111b	GTIOC1A#	GTIOC0B#	—	GTIOC2B#	GTIOC4A#	—	GTIOC5A#	—
11000b	—	GTIOC3B	—	GTIOC7B	GTETRGA	—	GTETRGB	—
11001b	TS32	TS31	TS30	TS29	TS28	TS27	TS26	—

—: Do not specify this value.

Table 21.15 Register Settings for Input/Output Pin Function in 80-Pin

	Pin						
PSEL[4:0] Settings	**PA0**	**PA1**	**PA2**	**PA3**	**PA4**	**PA5**	**PA6**
00000b (initial value)	Hi-Z						
00001b	—	GTIOC3B#	—	GTIOC7B#	—	—	—
00010b	GTOVUP	GTETRGC	—	GTETRGB	GTOVLO	—	—
00011b	—	GTIV	—	GTETRGD	—	—	GTOULO
00100b	—	GTOUUP	—	GTOVLO	—	—	—
00101b	—	—	—	—	TMRI0	—	TMCI3
00111b	CACREF	—	—	—	—	—	—
01000b	—	—	—	GTOWLO	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—
01011b	—	—	—	—	—	—	CTS5# RTS5# SS5#
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA
10100b	GTIOC0A	GTIOC0A	—	GTIOC1B	GTIOC1B	GTIOC4B	GTIOC0B
10101b	GTIOC1A	GTIOC0B	—	GTIOC2B	GTIOC4A	—	GTIOC5A
10110b	GTIOC0A#	GTIOC0A#	—	GTIOC1B#	GTIOC1B#	GTIOC4B#	GTIOC0B#
10111b	GTIOC1A#	GTIOC0B#	—	GTIOC2B#	GTIOC4A#	—	GTIOC5A#
11000b	—	GTIOC3B	—	GTIOC7B	GTETRGA	—	GTETRGB
11001b	TS32	TS31	TS30	TS29	TS28	TS27	TS26

—: Do not specify this value.

Table 21.16 Register Settings for Input/Output Pin Function in 64-Pin

	Pin				
PSEL[4:0] Settings	**PA0**	**PA1**	**PA3**	**PA4**	**PA6**
00000b (initial value)	Hi-Z				
00001b	—	GTIOC3B#	GTIOC7B#	—	—
00010b	GTOVUP	GTETRGC	GTETRGB	GTOVLO	—
00011b	—	GTIV	GTETRGD	—	GTOULO
00100b	—	GTOUUP	GTOVLO	—	—
00101b	—	—	—	TMRI0	TMCI3
00111b	CACREF	—	—	—	—
01000b	—	—	GTOWLO	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—
01011b	—	—	—	—	CTS5# RTS5# SS5#
01101b	SSLA1	SSLA2	—	SSLA0	MOSIA
10100b	GTIOC0A	GTIOC0A	GTIOC1B	GTIOC1B	GTIOC0B
10101b	GTIOC1A	GTIOC0B	GTIOC2B	GTIOC4A	GTIOC5A
10110b	GTIOC0A#	GTIOC0A#	GTIOC1B#	GTIOC1B#	GTIOC0B#
10111b	GTIOC1A#	GTIOC0B#	GTIOC2B#	GTIOC4A#	GTIOC5A#
11000b	—	GTIOC3B	GTIOC7B	GTETRGA	GTETRGB
11001b	TS32	TS31	TS29	TS28	TS26

—: Do not specify this value.

Table 21.17 Register Settings for Input/Output Pin Function in 48-Pin

	Pin			
PSEL[4:0] Settings	**PA1**	**PA3**	**PA4**	**PA6**
00000b (initial value)	Hi-Z			
00001b	GTIOC3B#	GTIOC7B#	—	—
00010b	GTETRGC	GTETRGB	GTOVLO	—
00011b	GTIV	GTETRGD	—	GTOULO
00100b	GTOUUP	GTOVLO	—	—
00101b	—	—	TMRI0	TMCI3
01000b	—	GTOWLO	—	—
01010b	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—
01011b	—	—	—	CTS5# RTS5# SS5#
01101b	SSLA2	—	SSLA0	MOSIA
10100b	GTIOC0A	GTIOC1B	GTIOC1B	GTIOC0B
10101b	GTIOC0B	GTIOC2B	GTIOC4A	GTIOC5A
10110b	GTIOC0A#	GTIOC1B#	GTIOC1B#	GTIOC0B#
10111b	GTIOC0B#	GTIOC2B#	GTIOC4A#	GTIOC5A#
11000b	GTIOC3B	GTIOC7B	GTETRGA	GTETRGB
11001b	TS31	TS29	TS28	TS26

—: Do not specify this value.

21.2.9 PBn Pin Function Control Registers (PBnPFS) (n = 0 to 7)

Address(es): MPC.PB0PFS 0008 C198h, MPC.PB1PFS 0008 C199h, MPC.PB2PFS 0008 C19Ah, MPC.PB3PFS 0008 C19Bh, MPC.PB4PFS 0008 C19Ch, MPC.PB5PFS 0008 C19Dh, MPC.PB6PFS 0008 C19Eh, MPC.PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.18 to Table 21.20.	R/W
b5	—	Reserved	This bit is 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 PB1: IRQ4 (100/80/64/48 pins)	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 100-Pin and 80-Pin

	Pin							
PSEL[4:0] Settings	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
00000b (initial value)	Hi-Z							
00001b	—	GTIOC7A#	—	GTIOC3B#	—	GTIOC6B#	—	—
00010b	—	GTOVLO	—	GTETRGD	—	—	—	—
00011b	GTOWUP	GTIW	—	GTIU	—	—	—	—
00100b	—	GTOWLO	—	GTOVUP	—	—	—	—
00101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
01010b	—	—	—	—	—	SCK009	RXD009 SMISO009 SSCL009	TXD009 TXDA009 SMOSI009 SSDA009
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS009# RTS009# SS009#	—	—	—
01100b	—	—	—	—	DE009	TXDB009	—	—
01101b	RSPCKA	—	—	—	—	—	—	—
10000b	—	CMPOB1	—	—	—	—	—	—
10001b	—	—	—	—	—	USB0_VBUS	—	—
10100b	GTIOC0B	GTIOC1B	GTIOC3A	GTIOC1A	GTIOC6A	GTIOC4B	GTIOC0B	GTIOC0A
10101b	GTIOC2A	GTIOC2B	—	GTIOC3A	—	GTIOC5A	GTIOC7A	GTIOC7B
10110b	GTIOC0B#	GTIOC1B#	GTIOC3A#	GTIOC1A#	GTIOC6A#	GTIOC4B#	GTIOC0B#	GTIOC0A#
10111b	GTIOC2A#	GTIOC2B#	—	GTIOC3A#	—	GTIOC5A#	GTIOC7A#	GTIOC7B#
11000b	—	GTIOC7A	GTETRGC	GTIOC3B	—	GTIOC6B	—	—
11001b	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18
11011b	—	—	—	LPTO	—	—	—	—
11100b	—	—	—	PMC0	—	—	—	—

—: Do not specify this value.

Table 21.19 Register Settings for Input/Output Pin Function in 64-Pin

	Pin					
PSEL[4:0] Settings	**PB0**	**PB1**	**PB3**	**PB5**	**PB6**	**PB7**
00000b (initial value)	Hi-Z					
00001b	—	GTIOC7A#	GTIOC3B#	GTIOC6B#	—	—
00010b	—	GTOVLO	GTETRGD	—	—	—
00011b	GTOWUP	GTIW	GTIU	—	—	—
00100b	—	GTOWLO	GTOVUP	—	—	—
00101b	—	TMCI0	TMO0	TMRI1	—	—
01010b	—	—	—	SCK009	RXD009 SMISO009 SSCL009	TXD009 TXDA009 SMOSI009 SSDA009
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—	—	—
01100b	—	—	—	TXDB009	—	—
01101b	RSPCKA	—	—	—	—	—
10000b	—	CMPOB1	—	—	—	—
10001b	—	—	—	USB0_VBUS	—	—
10100b	GTIOC0B	GTIOC1B	GTIOC1A	GTIOC4B	GTIOC0B	GTIOC0A
10101b	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC5A	GTIOC7A	GTIOC7B
10110b	GTIOC0B#	GTIOC1B#	GTIOC1A#	GTIOC4B#	GTIOC0B#	GTIOC0A#
10111b	GTIOC2A#	GTIOC2B#	GTIOC3A#	GTIOC5A#	GTIOC7A#	GTIOC7B#
11000b	—	GTIOC7A	GTIOC3B	GTIOC6B	—	—
11001b	TS25	TS24	TS22	TS20	TS19	TS18
11011b	—	—	LPTO	—	—	—
11100b	—	—	PMC0	—	—	—

—: Do not specify this value.

Table 21.20 Register Settings for Input/Output Pin Function in 48-Pin

	Pin			
PSEL[4:0] Settings	**PB0**	**PB1**	**PB3**	**PB5**
00000b (initial value)	Hi-Z			
00001b	—	GTIOC7A#	GTIOC3B#	GTIOC6B#
00010b	—	GTOVLO	GTETRGD	—
00011b	GTOWUP	GTIW	GTIU	—
00100b	—	GTOWLO	GTOVUP	—
00101b	—	TMCI0	TMO0	TMRI1
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—
01101b	RSPCKA	—	—	—
10000b	—	CMPOB1	—	—
10001b	—	—	—	USB0_VBUS
10100b	GTIOC0B	GTIOC1B	GTIOC1A	GTIOC4B
10101b	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC5A
10110b	GTIOC0B#	GTIOC1B#	GTIOC1A#	GTIOC4B#
10111b	GTIOC2A#	GTIOC2B#	GTIOC3A#	GTIOC5A#
11000b	—	GTIOC7A	GTIOC3B	GTIOC6B
11001b	TS25	TS24	TS22	TS20
11011b	—	—	LPTO	—
11100b	—	—	PMC0	—

—: Do not specify this value.

21.2.10 PCn Pin Function Control Registers (PCnPFS) (n = 0 to 7)

Address(es): MPC.PC0PFS 0008 C1A0h, MPC.PC1PFS 0008 C1A1h, MPC.PC2PFS 0008 C1A2h, MPC.PC3PFS 0008 C1A3h, MPC.PC4PFS 0008 C1A4h, MPC.PC5PFS 0008 C1A5h, MPC.PC6PFS 0008 C1A6h, MPC.PC7PFS 0008 C1A7h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.21 to Table 21.23.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.21 Register Settings for Input/Output Pin Function in 100-Pin

	Pin							
PSEL[4:0] Settings	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z							
00001b	—	GTCPPO0	—	—	GTIU	—	—	GTCPPO0
00010b	—	—	—	—	—	GTOUUP	—	—
00011b	—	—	GTOWUP	—	GTOULO	GTIW	—	—
00101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
00111b	—	—	—	—	—	—	—	CACREF
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK008	RXD008 SMISO008 SSCL008	TXD008 TXDA008 SMOSI008 SSDA008
01011b	CTS5# RTS5# SS5#	—	—	—	CTS008# RTS008# SS008#	—	—	—
01100b	—	—	—	—	DE008	TXDB008	—	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10001b	—	—	—	—	—	USB0_ID	USB0_EXICEN	—
10100b	GTIOC6B	GTIOC6A	GTIOC2A	GTIOC2B	GTIOC0B	GTIOC0A	GTIOC6B	GTIOC6A
10101b	—	—	—	—	GTIOC3A	GTIOC7A	—	—
10110b	GTIOC6B#	GTIOC6A#	GTIOC2A#	GTIOC2B#	GTIOC0B#	GTIOC0A#	GTIOC6B#	GTIOC6A#
10111b	—	—	—	—	GTIOC3A#	GTIOC7A#	—	—
11000b	GTETRGC	GTETRGD	GTETRGA	GTETRGB	GTETRGC	GTETRGD	GTETRGA	GTETRGB
11001b	—	—	TS17	TS16	TSCAP	TS15	TS14	TS13
11011b	—	—	—	—	—	—	—	LPTO
11100b	—	—	—	PMC0	PMC0	PMC0	—	—

—: Do not specify this value.

Table 21.22 Register Settings for Input/Output Pin Function in 80-Pin and 64-Pin

	Pin					
PSEL[4:0] Settings	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
00000b (initial value)	Hi-Z					
00001b	—	—	GTIU	—	—	GTCPPO0
00010b	—	—	—	GTOUUP	—	—
00011b	GTOWUP	—	GTOULO	GTIW	—	—
00101b	—	—	TMCI1	TMRI2	TMCI2	TMO2
00111b	—	—	—	—	—	CACREF
01010b	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK008	RXD008 SMISO008 SSCL008	TXD008 TXDA008 SMOSI008 SSDA008
01011b	—	—	CTS008# RTS008# SS008#	—	—	—
01100b	—	—	DE008	TXDB008	—	—
01101b	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10001b	—	—	—	USB0_ID	USB0_EXICEN	—
10100b	GTIOC2A	GTIOC2B	GTIOC0B	GTIOC0A	GTIOC6B	GTIOC6A
10101b	—	—	GTIOC3A	GTIOC7A	—	—
10110b	GTIOC2A#	GTIOC2B#	GTIOC0B#	GTIOC0A#	GTIOC6B#	GTIOC6A#
10111b	—	—	GTIOC3A#	GTIOC7A#	—	—
11000b	GTETRGA	GTETRGB	GTETRGC	GTETRGD	GTETRGA	GTETRGB
11001b	TS17	TS16	TSCAP	TS15	TS14	TS13
11011b	—	—	—	—	—	LPTO
11100b	—	PMC0	PMC0	PMC0	—	—

—: Do not specify this value.

Table 21.23 Register Settings for Input/Output Pin Function in 48-Pin

	Pin			
PSEL[4:0] Settings	**PC4**	**PC5**	**PC6**	**PC7**
00000b (initial value)	Hi-Z			
00001b	GTIU	—	—	GTCPPO0
00010b	—	GTOUUP	—	—
00011b	GTOULO	GTIW	—	—
00101b	TMCI1	TMRI2	TMCI2	TMO2
00111b	—	—	—	CACREF
01010b	SCK5	SCK008	RXD008 SMISO008 SSCL008	TXD008 TXDA008 SMOSI008 SSDA008
01011b	CTS008# RTS008# SS008#	—	—	—
01100b	DE008	TXDB008	—	—
01101b	SSLA0	RSPCKA	MOSIA	MISOA
10001b	—	USB0_ID	USB0_EXICEN	—
10100b	GTIOC0B	GTIOC0A	GTIOC6B	GTIOC6A
10101b	GTIOC3A	GTIOC7A	—	—
10110b	GTIOC0B#	GTIOC0A#	GTIOC6B#	GTIOC6A#
10111b	GTIOC3A#	GTIOC7A#	—	—
11000b	GTETRGC	GTETRGD	GTETRGA	GTETRGB
11001b	TSCAP	TS15	TS14	TS13
11011b	—	—	—	LPTO
11100b	PMC0	PMC0	—	—

—: Do not specify this value.

21.2.11 PDn Pin Function Control Registers (PDnPFS) (n = 0 to 7)

Address(es): MPC.PD0PFS 0008 C1A8h, MPC.PD1PFS 0008 C1A9h, MPC.PD2PFS 0008 C1AAh, MPC.PD3PFS 0008 C1ABh, MPC.PD4PFS 0008 C1ACh, MPC.PD5PFS 0008 C1ADh, MPC.PD6PFS 0008 C1AEh, MPC.PD7PFS 0008 C1AFh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.24.	R/W
b5	—	Reserved	This bit is 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 PD0: IRQ0 input switch (100/80 pins) PD1: IRQ1 input switch (100/80 pins) PD2: IRQ2 input switch (100/80 pins) PD3: IRQ3 input switch (100 pins) PD4: IRQ4 input switch (100 pins) PD5: IRQ5 input switch (100 pins) PD6: IRQ6 input switch (100 pins) PD7: IRQ7 input switch (100 pins)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin 1: Used as analog pin PD0: AN024 (100/80 pins) PD1: AN025 (100/80 pins) PD2: AN026 (100/80 pins) PD3: AN027 (100 pins) PD4: AN028 (100 pins) PD5: AN029 (100 pins) PD6: AN030 (100 pins) PD7: AN031 (100 pins)	R/W

Table 21.24 Register Settings for Input/Output Pin Function in 100-Pin and 80-Pin

	Pin		
PSEL[4:0] Settings	**PD0**	**PD1**	**PD2**
00000b (initial value)	Hi-Z		
01011b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
10000b	—	CTX0	CRX0
10100b	—	GTIOC2A	GTIOC2B
10110b	—	GTIOC2A#	GTIOC2B#

—: Do not specify this value.

21.2.12 PEn Pin Function Control Registers (PEnPFS) (n = 0 to 7)

Address(es): MPC.PE0PFS 0008 C1B0h, MPC.PE1PFS 0008 C1B1h, MPC.PE2PFS 0008 C1B2h, MPC.PE3PFS 0008 C1B3h, MPC.PE4PFS 0008 C1B4h, MPC.PE5PFS 0008 C1B5h, MPC.PE6PFS 0008 C1B6h, MPC.PE7PFS 0008 C1B7h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.25 and Table 21.26.	R/W
b5	—	Reserved	This bit is 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 PE2: IRQ7 input switch (100/80/64/48 pins) PE5: IRQ5 input switch (100/80/64 pins) PE6: IRQ6 input switch (100 pins) PE7: IRQ7 input switch (100 pins)	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin PE0:AN016 (100/80/64 pins) PE1:AN017 or CMPB0 (100/80/64/48 pins) PE2:AN018 or CVREFB0 (100/80/64/48 pins) PE3:AN019 (100/80/64/48 pins) PE4:AN020 or CMPA2 (100/80/64/48 pins) PE5:AN021 (100/80/64 pins) PE6:AN022 (100 pins) PE7:AN023 (100 pins)	R/W

Table 21.25 Register Settings for Input/Output Pin Function in 100-Pin, 80-Pin, and 64-Pin

	Pin					
PSEL[4:0] Settings	PE0	PE1	PE2	PE3	PE4	PE5
00000b (initial value)	Hi-Z					
00001b	—	—	—	—	GTIOC4A#	—
00010b	—	GTOVLO	GTOVUP	—	GTOVUP	—
00011b	—	—	—	GTOWUP	GTOWLO	—
01001b	—	—	—	CLKOUT	CLKOUT	—
01100b	SCK12	TXD12 TXDX12 SIOX12 SMOSI12 SSDA12	RXD12 RXDX12 SMISO12 SSCL12	CTS12# RTS12# SS12#	—	—
10000b	—	—	—	—	—	CMPOB0
10100b	—	GTIOC1B	GTIOC1A	GTIOC2A	GTIOC1A	GTIOC1B
10101b	—	—	—	GTIOC4B	GTIOC2B	GTIOC5B
10110b	—	GTIOC1B#	GTIOC1A#	GTIOC2A#	GTIOC1A#	GTIOC1B#
10111b	—	—	—	GTIOC4B#	GTIOC2B#	GTIOC5B#
11000b	—	—	—	—	GTIOC4A	—
11001b	—	—	TS35	TS34	TS33	—

—: Do not specify this value.

Table 21.26 Register Settings for Input/Output Pin Function in 48-Pin

	Pin			
PSEL[4:0] Settings	**PE1**	**PE2**	**PE3**	**PE4**
00000b (initial value)	Hi-Z			
00001b	—	—	—	GTIOC4A#
00010b	GTOVLO	GTOVUP	—	GTOVUP
00011b	—	—	GTOWUP	GTOWLO
01001b	—	—	CLKOUT	CLKOUT
01100b	TXD12 TXDX12 SIOX12 SSDA12	RXD12 RXDX12 SSCL12	CTS12# RTS12#	—
10100b	GTIOC1B	GTIOC1A	GTIOC2A	GTIOC1A
10101b	—	—	GTIOC4B	GTIOC2B
10110b	GTIOC1B#	GTIOC1A#	GTIOC2A#	GTIOC1A#
10111b	—	—	GTIOC4B#	GTIOC2B#
11000b	—	—	—	GTIOC4A
11001b	—	TS35	TS34	TS33

—: Do not specify this value.

21.2.13 PHn Pin Function Control Registers (PHnPFS) (n = 0 to 3)

Address(es): MPC.PH0PFS 0008 C1C8h, MPC.PH1PFS 0008 C1C9h, MPC.PH2PFS 0008 C1CAh, MPC.PH3PFS 0008 C1CBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.27.	R/W
b5	—	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 PH1: IRQ0 input switch (100/80/64/48 pins) PH2: IRQ1 input switch (100/80/64/48 pins)	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.27 Register Settings for Input/Output Pin Function in 100-Pin, 80-Pin, 64-Pin, and 48-Pin

	Pin			
PSEL[4:0] Settings	**PH0**	**PH1**	**PH2**	**PH3**
00000b (initial value)	Hi-Z			
00010b	GTOUUP	—	—	—
00011b	—	GTOULO	—	—
00101b	—	TMO0	TMRI0	TMCI0
00111b	CACREF	—	—	—
10100b	GTIOC0A	GTIOC0B	GTIOC1B	GTIOC2B
10110b	GTIOC0A#	GTIOC0B#	GTIOC1B#	GTIOC2B#
11001b	TS10	TS9	TS8	TS7

—: Do not specify this value.

21.2.14 PJn Pin Function Control Registers (PJnPFS) (n = 1, 3, 6, 7)

Address(es): MPC.PJ1PFS 0008 C1D1h, MPC.PJ3PFS 0008 C1D3h, MPC.PJ6PFS 0008 C1D6h, MPC.PJ7PFS 0008 C1D7h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.28 and Table 21.29.	R/W
b6 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used as a pin other than an analog pin 1: Used as an analog pin PJ6: VREFH0 (100/80/64/48 pins) PJ7:VREFL0 (100/80/64/48 pins)	R/W

Table 21.28 Register Settings for Input/Output Pin Function in 100-Pin

	Pin	
PSEL[4:0] Settings	PJ1	PJ3
00000b (initial value)	Hi-Z	
00001b	GTCPPO0	—
01011b	—	CTS6# RTS6# SS6#
10100b	GTIOC6A	GTIOC6B
10110b	GTIOC6A#	GTIOC6B#

—: Do not specify this value.

Table 21.29 Register Settings for Input/Output Pin Function in 80-Pin

	Pin
PSEL[4:0] Settings	PJ1
00000b (initial value)	Hi-Z
00001b	GTCPPO0
10100b	GTIOC6A
10110b	GTIOC6A#

21.3 Usage Notes

21.3.1 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) to 0 to select the general I/O port function.
(2) Specify the assignments of input/output signals for peripheral functions 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 5, A to E, H, J; n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[4:0] 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.

21.3.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is set to 0. If a Pmn pin function control register is set while the PMR register is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
(3) Do not assign a single function to multiple pins through the MPC settings.
(4) Ports 4, D, and E also function as analog input pins for the A/D converter. When using these ports as analog input pins, set them to general input pins by clearing the corresponding bits in the port mode register (PMR) and port direction register (PDR) to 0 and set the PmnPFS.ASEL bit to 1, to avoid degradation of accuracy.
(5) The initial value of the time capture event input pin enable bit (TCEN) of the time capture control register y (RTCCRy, y = 0 to 2) is undefined after a reset. Therefore, set this bit to 0 to avoid unnecessary input.

Table 21.30 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
After a reset	0	0	0	0	00000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the 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 21.29)	Set the 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.
CTSU	1	0	0	0	11001b	PCR.Bn = 0
EXTAL/XTAL	0	0	x	0	x	Set these as general input port pins so that the output buffers are turned off.
XCIN/XCOUT	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (if the IRQ is selected from the multiplexed functions).
Note 1. The pin does not function as the IRQn input pin even if the PmnPFS.ISEL bit is set to 1.

Note: The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[4:0] 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 PCR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.

21.3.3 Note on Using Analog Functions

When an analog function is in use, configure the pin as a general-purpose input by setting the given bits of the port mode register (PMR), of the port direction register (PDR), and the pull-up control register (PCR) to 0, and then set the ASEL bit in the Pmn pin function control register (PmnPFS) to 1.

21.3.4 Notes on Using the CTSU Function of the Capacitive Touch Sensing Unit

When using the CTSU function (TSn (n=0 to 35), TSCAP) of the capacitive touch sensing unit, set the given bits of the port mode register (PMR), the port direction register (PDR), and the pull-up control register (PCR) to 0. Then, use the PmnPFS.PSEL[4:0] bits to select the CTSU function and set the PMR register to 1. When a pin function of the capacitive touch sensing unit, do not use the pin as the IRQ input pin regardless of the ISEL setting of the corresponding bit.

21.3.5 Notes on Inverting Input/Output Function for the GPTW Input/Output Pins

The GPTW input/output pins shown in Table 21.31 can invert and output the signal captured by inverting the input signal when the PmnPFS.PSEL[4:0] bit for the target pins is set. Setting for switching the state between the normal input/output and the inverting input/output must be done while the PMR register for the target pins is set as 0.

Table 21.31 GPTW Input/Output Pins

Module/Function	Channel	Normal Input/Output	Inverting Input/Output
General PWM timer	GPTW0	GTIOC0A	GTIOC0A#
		GTIOC0B	GTIOC0B#
	GPTW1	GTIOC1A	GTIOC1A#
		GTIOC1B	GTIOC1B#
	GPTW2	GTIOC2A	GTIOC2A#
		GTIOC2B	GTIOC2B#
	GPTW3	GTIOC3A	GTIOC3A#
		GTIOC3B	GTIOC3B#
	GPTW4	GTIOC4A	GTIOC4A#
		GTIOC4B	GTIOC4B#
	GPTW5	GTIOC5A	GTIOC5A#
		GTIOC5B	GTIOC5B#
	GPTW6	GTIOC6A	GTIOC6A#
		GTIOC6B	GTIOC6B#
	GPTW7	GTIOC7A	GTIOC7A#
		GTIOC7B	GTIOC7B#

22. General PWM Timer (GPTWa)

This MCU has a general PWM timer (GPTW) consisting of a two-channel 32-bit timer and six-channel 16-bit timer.

22.1 Overview

Table 22.1 lists the specifications for the GPTW, and Table 22.2 shows the functions of the GPTW. Figure 22.1 and Figure 22.2 show block diagrams of the GPTW.

Table 22.1 GPTW Specifications

Item	Description
Functions	• 32 bits × 2 channels and 16 bits × 6 channels • Up-counting or down-counting (sawtooth waves) or up/down-counting (triangle waves) for each counter. • Clock sources independently selectable for each channel • Two I/O pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can take place at crests or troughs, enabling the generation of laterally asymmetrical PWM waveforms. • Registers for setting up frame cycles in each channel (with capability for generating interrupts at overflow or underflow) • Generation of dead times in PWM operation • Generation of high accuracy duty in the vicinity of duty 0% and 100% PWM waveform • In output compare operation, setting compare register is immediately used to generate PWM waveform with dead times • Simultaneous start/stop/clearing of desired channel counters • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by maximum of eight ELC events based on the ELC setting • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by detecting two input signal conditions • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by maximum of four external triggers • Function to control output negation by requests for disabling of output from the POEG • A/D conversion start trigger generation function • Event signals for compare match A to F and for overflow/underflow can be output to the ELC • Input capture input can select noise filter function • External pulse width measuring function • Synchronous counter clearing/counter setting/input capture among channels • Bus clock: PCLKA, GPTW count reference clock: PCLKA

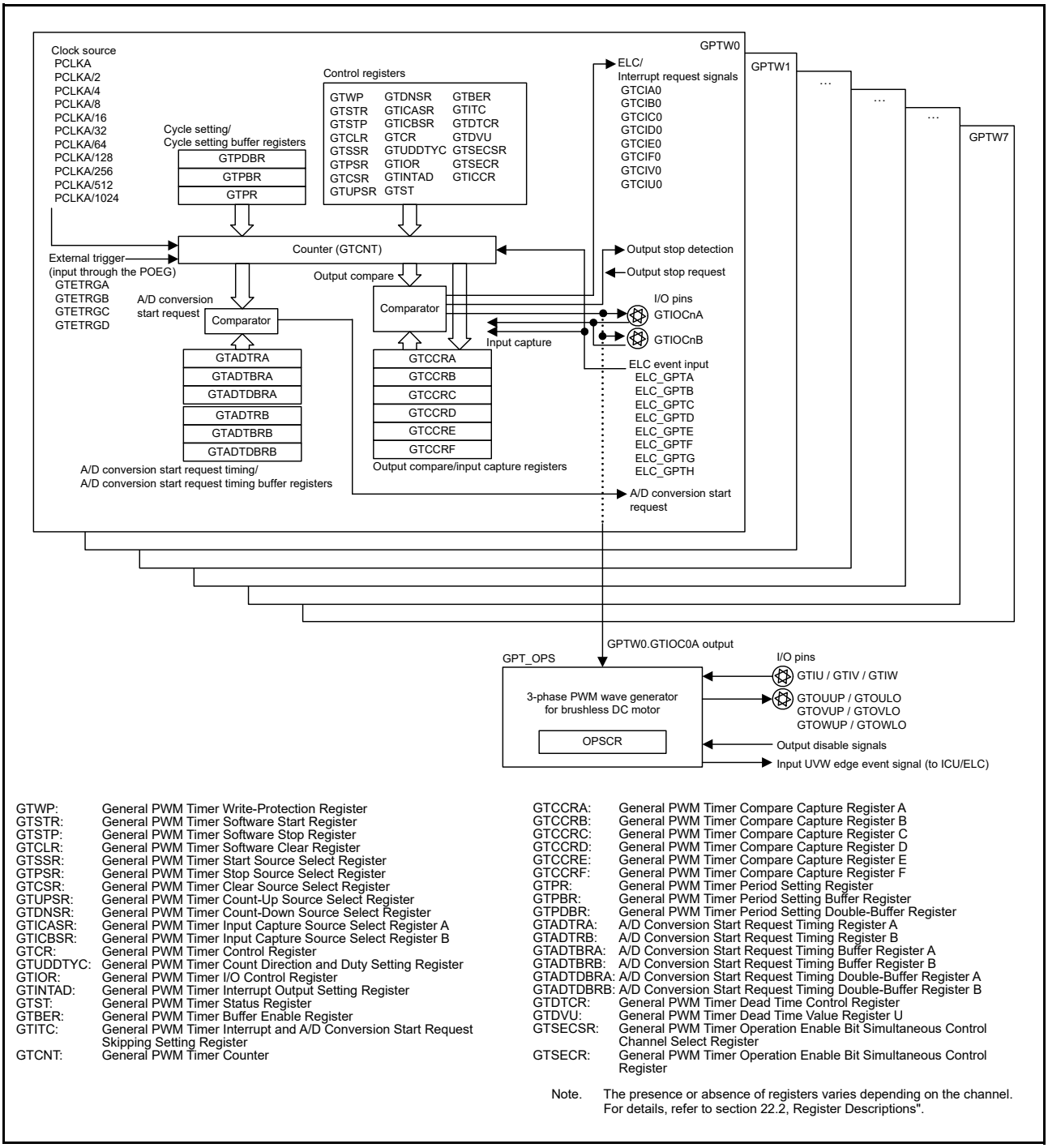
Table 22.2 GPTW Functions (1/2)

Item		GPTW0	GPTW1	GPTW2	GPTW3	GPTW4	GPTW5	GPTW6	GPTW7
Timer counter		16-bit	16-bit	16-bit	16-bit	16-bit	16-bit	32-bit	32-bit
Count clocks		PCLKA, PCLKA/2, PCLKA/4, PCLKA/8, PCLKA/16, PCLKA/32, PCLKA/64, PCLKA/128, PCLKA/256, PCLKA/512, PCLKA/1024, GTETRGA, GTETRGB, GTETRGC, GTETRGD							
Output compare/input capture registers (GTCCR)		GTCCRA, GTCCRB							
Compare/buffer registers		GTCCRC, GTCCRD, GTCCRE, GTCCRF							
PWM period setting register		GTPR							
PWM period setting buffer registers		GTPBR, GTPDBR			GTPBR				
Input capture input/ compare match output/ PWM output pin		GTIOCnA, GTIOCnB							
External trigger input pin (via the POEG)		GTETRGA, GTETRGB, GTETRGC, GTETRGD							
Counter clear sources		Compare match for the GTPR register, input capture, ELC input, the state of input pins, external trigger input, GTCCR register compare match, and Other channel's counter clear sources							
Compare match output	Low output	✓							
	High output	✓							
	Toggle output	✓							
Input capture function		✓							
Automatic addition of dead time		✓			—				
PWM mode	Sawtooth-wave PWM mode 1	✓			✓				
	Sawtooth-wave PWM mode 2	✓			—				
	Sawtooth-wave one-shot pulse mode	✓			✓				
	Triangle-wave PWM mode 1/2/3	✓			✓				
	Complementary PWM mode 1/2/3/4	✓			—				
PWM Synchronous Output		✓	—		—				
Phase counting function		—			✓				
External pulse width measuring function		—			✓			—	
Buffer operation	Double buffer	✓							
	Simultaneous operation disable control for multiple channels	✓							
	Counter clear	✓							
One-shot operation		✓							
DMAC/DTC activation		All interrupt sources							
A/D conversion start trigger		Compare match of GTADTRA or GTADTRB register			✓				
3-phase PWM wave generator for brushless DC motor		✓							

Table 22.2 GPTW Functions (2/2)

Item	GPTW0	GPTW1	GPTW2	GPTW3	GPTW4	GPTW5	GPTW6	GPTW7
Interrupt sources	Eight sources • GTCCRA register compare match/input capture (GTCIAn) • GTCCRB register compare match/input capture (GTCIBn) • GTCCRC register compare match (GTCICn) • GTCCRD register compare match (GTCIDn) • GTCCRE register compare match (GTCIEn) • GTCCRF register compare match (GTCIFn) • GTCNT counter overflow (GTPR register compare match) (GTCIVn) • GTCNT counter underflow (GTCIUn)							
Interrupt skipping function	Skipping of interrupts of GTCNT counter overflow (GTPR register compare match) (GTCIVn) and GTCNT counter underflow (GTCIUn) (interlocked with other interrupts and A/D conversion start requests)			—				
Event operation by the ELC	✓							
Noise filter function	✓							
Synchronous counter clearing/counter setting/input capture	✓							

✓: Possible



GTWP: General PWM Timer Write-Protection Register
GTSTR: General PWM Timer Software Start Register
GTSTP: General PWM Timer Software Stop Register
GTCLR: General PWM Timer Software Clear Register
GTSSR: General PWM Timer Start Source Select Register
GTPSR: General PWM Timer Stop Source Select Register
GTCSR: General PWM Timer Clear Source Select Register
GTUPSR: General PWM Timer Count-Up Source Select Register
GTDNSR: General PWM Timer Count-Down Source Select Register
GTICASR: General PWM Timer Input Capture Source Select Register A
GTICBSR: General PWM Timer Input Capture Source Select Register B
GTCR: General PWM Timer Control Register
GTUDDTYC: General PWM Timer Count Direction and Duty Setting Register
GTIOR: General PWM Timer I/O Control Register
GTINTAD: General PWM Timer Interrupt Output Setting Register
GTST: General PWM Timer Status Register
GTBER: General PWM Timer Buffer Enable Register
GTITC: General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register
GTCNT: General PWM Timer Counter

GTCCRA: General PWM Timer Compare Capture Register A
GTCCRB: General PWM Timer Compare Capture Register B
GTCCRC: General PWM Timer Compare Capture Register C
GTCCRD: General PWM Timer Compare Capture Register D
GTCCRE: General PWM Timer Compare Capture Register E
GTCCRF: General PWM Timer Compare Capture Register F
GTPR: General PWM Timer Period Setting Register
GTPBR: General PWM Timer Period Setting Buffer Register
GTPDBR: General PWM Timer Period Setting Double-Buffer Register
GTADTRA: A/D Conversion Start Request Timing Register A
GTADTRB: A/D Conversion Start Request Timing Register B
GTADTBRA: A/D Conversion Start Request Timing Buffer Register A
GTADTBRB: A/D Conversion Start Request Timing Buffer Register B
GTADTDBRA: A/D Conversion Start Request Timing Double-Buffer Register A
GTADTDBRB: A/D Conversion Start Request Timing Double-Buffer Register B
GTDTCR: General PWM Timer Dead Time Control Register
GTDVU: General PWM Timer Dead Time Value Register U
GTSECSR: General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register
GTSECR: General PWM Timer Operation Enable Bit Simultaneous Control Register

Note. The presence or absence of registers varies depending on the channel. For details, refer to section 22.2, Register Descriptions".

Figure 22.1 GPTW Block Diagram (Sawtooth-wave PWM mode, Sawtooth-wave one-shot pulse mode, and Triangle-wave PWM mode 1/2/3)

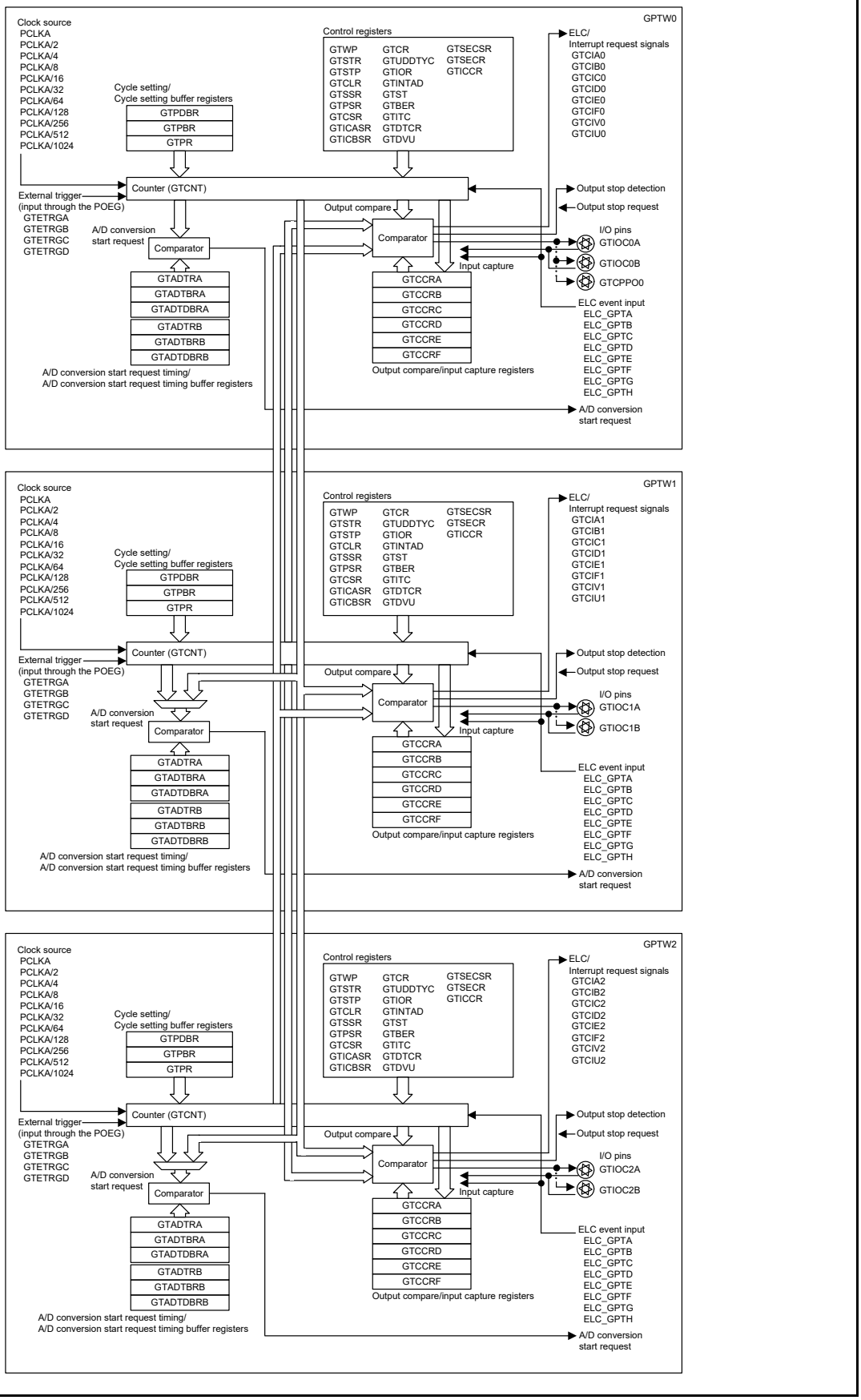


Figure 22.2 GPTW Block Diagram (Sawtooth-wave PWM mode 1/2/3, Sawtooth-wave one-shot pulse mode, Triangle-wave PWM mode 1/2/3, and Complementary PWM 1/2/3/4)

In this specification, three consecutive channels to configure complementary PWM mode is defined as complementary PWM mode channel group. The lowest position channel of complementary PWM mode channel group is defined as master channel. The second channel is defined as slave channel 1. The highest position channel is defined as slave channel 2.

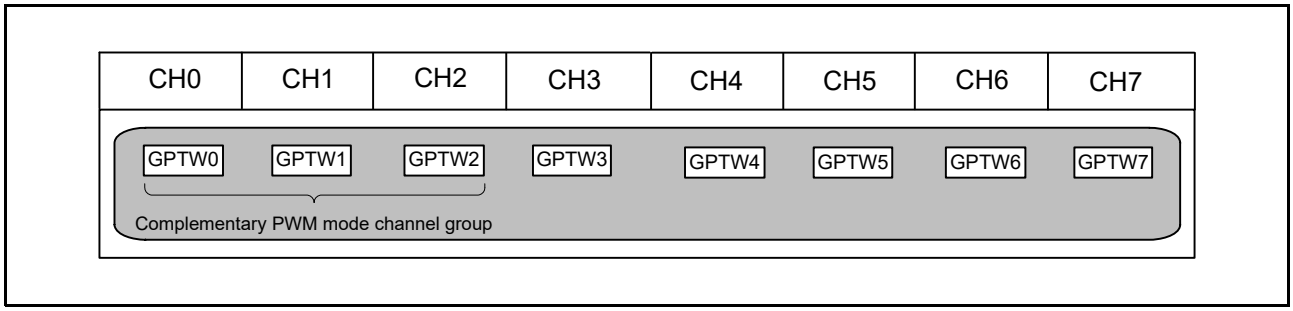


Figure 22.3 Association between GPTW channels and module names

Table 22.3 lists the I/O pins used in the GPTW.

Table 22.3 GPTW I/O Pins (n = 0 to 7)

Channel	Pin Name	I/O	Function
GPTW	GTETRGA	Input	External trigger input pin A (input via the POEG)
	GTETRGB	Input	External trigger input pin B (input via the POEG)
	GTETRGC	Input	External trigger input pin C (input via the POEG)
	GTETRGD	Input	External trigger input pin D (input via the POEG)
GPTWn	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
	GTCPPO0	Output	Toggle output synchronized with PWM period
OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

22.2 Register Descriptions

22.2.1 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPTW0.GTWP 000C 2000h, GPTW1.GTWP 000C 2100h, GPTW2.GTWP 000C 2200h, GPTW3.GTWP 000C 2300h, GPTW4.GTWP 000C 2400h, GPTW5.GTWP 000C 2500h, GPTW6.GTWP 000C 2600h, GPTW7.GTWP 000C 2700h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WP	Register Write Disabled	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b1	STRWP	GTSTR.CSTRT Bit Write Disabled	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b2	STPWP	GTSTP.CSTOP Bit Write Disabled	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b3	CLRWP	GTCLR.CCLR Bit Write Disabled	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b4	CMNWP	Common Register Write Disabled	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	GTWP Key Code	These bits are read as 0. To modify the WP, STRWP, STPWP, CLRWP, and CMNWP bits, write A5h.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification.
Protection by the GTWP register only covers the CPU write operation.
Such protection does not cover a register updating generated in coordination with the CPU write.
For registers that are write enabled or disabled depending on the setting of the GTWP register, see section 22.8.1, Write-Protection for Registers.

WP Bit (Register Write Disabled)

Write enable/disable to the GPTW can be selected with this bit.
Registers which are targets of write enable/disable are as follows:
GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTICCR

STRWP Bit (GTSTR.CSTRT Bit Write Disabled)

The STRWP bits enable or disable starting the updating of counter values by writing to the CSTRTn bit corresponding to a channel number in the GTSTR register.
The bit position of each CSTRTn bit in the GTSTR registers is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for

each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1 (disabling writing), the CSTRT bit for the given channel will not be updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTSTR.CSTRT0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter starts running. When the setting of the GPTW0.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPTW1.GTSTR.CSTRT0 bit when its current setting is 0 will leave the bit with the value 0, and the GPTW0.GTCNT counter will not start running.
If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP Bit (GTSTP.CSTOP Bit Write Disabled)

The STPWP bits enable or disable starting the updating of counter values by writing to the CSTOPn bit corresponding to a channel number in the GTSTP register.
The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1 (disabling writing), the CSTOP bit for the given channel will not be updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTSTP.CSTOP0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter is stopped. When the setting of the GPTW0.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPTW1.GTSTP.CSTOP0 bit when its current setting is 0 will leave the bit with the value 0, and the GPTW0.GTCNT counter will not be stopped.
If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP Bit (GTCLR.CCLR Bit Write Disabled)

The CLRWP bits enable or disable starting the updating of counter values by writing to the CCLRn bit corresponding to a channel number in the GTCLR register.
The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1 (disabling writing), the CCLR bit for the given channel will not be updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTCLR.CCLR0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter is cleared.
If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP Bit (Common Register Write Disabled)

The CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 7) corresponding to a channel number in the GTSECSR register or to the GTSECR register.
The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel in the case of simultaneous writing to all channels.

Thus, in the case of writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1 (disabling writing), the SECSEL bit and the GTSECR register value for the given channel will not be updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) will be updated.
For example, when the setting of the GPTW0.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPTW1.GTSECSR.SECSEL0 bit will lead to updating of the value of the GPTW0.GTSECSR.SECSEL0 bit. In the same way, writing to the GPTW1.GTSECR register will update the value of the GPTW0.GTSECR register. When the setting of the GPTW0.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPTW1.GTSECSR.SECSEL0 bit will not lead to updating of the value of the GPTW0.GTSECSR.SECSEL0 bit. In the same way, writing to the GPTW1.GTSECR register will not lead to updating of the value of the GPTW0.GTSECR register.
If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

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

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

22.2.2 General PWM Timer Software Start Register (GTSTR)

Address(es): GPTW0.GTSTR 000C 2004h, GPTW1.GTSTR 000C 2104h, GPTW2.GTSTR 000C 2204h, GPTW3.GTSTR 000C 2304h, GPTW4.GTSTR 000C 2404h, GPTW5.GTSTR 000C 2504h, GPTW6.GTSTR 000C 2604h, GPTW7.GTSTR 000C 2704h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CSTRT7	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTRT0	Channel 0 Count Start	When reading 0: The counter is stopped. 1: The counter is running. When writing 0: Ignored. 1: The counter starts running.	R/W
b1	CSTRT1	Channel 1 Count Start		R/W
b2	CSTRT2	Channel 2 Count Start		R/W
b3	CSTRT3	Channel 3 Count Start		R/W
b4	CSTRT4	Channel 4 Count Start		R/W
b5	CSTRT5	Channel 5 Count Start		R/W
b6	CSTRT6	Channel 6 Count Start		R/W
b7	CSTRT7	Channel 7 Count Start		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTR register starts the GTCNT counter operation.
A bit position of the GTSTR register indicates a channel number. The GTSTR register of each channel is a common register, and writing 1 to the GTSTR register in any channel and updating it can start operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.
In complementary PWM mode, writing to the CSTRTn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

CSTRTn Bit (Channel n Count Start) (n = 0 to 7)

This bit starts the GTCNT counter for channel n.
A read value indicates the state of the counter operation in a corresponding channel (GTCR.CST bit). The bit with 0 indicates that the counter is stopped, while the bit with 1 indicates that the counter is in operation.

22.2.3 General PWM Timer Software Stop Register (GTSTP)

Address(es): GPTW0.GTSTP 000C 2008h, GPTW1.GTSTP 000C 2108h, GPTW2.GTSTP 000C 2208h, GPTW3.GTSTP 000C 2308h, GPTW4.GTSTP 000C 2408h, GPTW5.GTSTP 000C 2508h, GPTW6.GTSTP 000C 2608h, GPTW7.GTSTP 000C 2708h

	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
	—	—	—	—	—	—	—	—	CSTOP7	CSTOP6	CSTOP5	CSTOP4	CSTOP3	CSTOP2	CSTOP1	CSTOP0
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	CSTOP0	Channel 0 Count Stop	When reading 0: The counter is operating. 1: The counter is stopped. When writing 0: Ignored. 1: The counter is stopped.	R/W
b1	CSTOP1	Channel 1 Count Stop		R/W
b2	CSTOP2	Channel 2 Count Stop		R/W
b3	CSTOP3	Channel 3 Count Stop		R/W
b4	CSTOP4	Channel 4 Count Stop		R/W
b5	CSTOP5	Channel 5 Count Stop		R/W
b6	CSTOP6	Channel 6 Count Stop		R/W
b7	CSTOP7	Channel 7 Count Stop		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTP register stops the GTCNT counter operation.
A bit position of the GTSTP register indicates a channel number. The GTSTP register of each channel is a common register, and writing 1 to the GTSTP register in any channel and updating it can stop operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.
In complementary PWM mode, writing to the CSTOPn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

CSTOPn Bit (Channel n Count Stop) (n = 0 to 7)

This bit stops the GTCNT counter for channel n.
A read value indicates the state of the counter operation in a corresponding channel (inverted GTCR.CST bit). The bit with 0 indicates that the counter is in operation, while the bit with 1 indicates that the counter is stopped.

22.2.4 General PWM Timer Software Clear Register (GTCLR)

Address(es): GPTW0.GTCLR 000C 200Ch, GPTW1.GTCLR 000C 210Ch, GPTW2.GTCLR 000C 220Ch, GPTW3.GTCLR 000C 230Ch, GPTW4.GTCLR 000C 240Ch, GPTW5.GTCLR 000C 250Ch, GPTW6.GTCLR 000C 260Ch, GPTW7.GTCLR 000C 270Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CCLR7	CCLR6	CCLR5	CCLR4	CCLR3	CCLR2	CCLR1	CCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCLR0	Channel 0 Counter Clear	0: Ignored. 1: The counter is to be cleared.	W
b1	CCLR1	Channel 1 Counter Clear		W
b2	CCLR2	Channel 2 Counter Clear		W
b3	CCLR3	Channel 3 Counter Clear		W
b4	CCLR4	Channel 4 Counter Clear		W
b5	CCLR5	Channel 5 Counter Clear		W
b6	CCLR6	Channel 6 Counter Clear		W
b7	CCLR7	Channel 7 Counter Clear		W
b31 to b8	—	Reserved	The write value should be 0.	W

The GTCLR register is a write-only register which sets clearing of the GTCNT counter.
A bit position of the GTCLR register indicates a channel number. The GTCLR register of each channel is a common register, and writing 1 to the GTCLR register in any channel and updating it changes to 0 of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.
In complementary PWM mode, writing to the CCLRn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

CCLRn Bit (Channel n Counter Clear) (n = 0 to 7)

When the counting direction flag is set for decrementation (GTST.TUCF flag = 0) with sawtooth-wave mode selected in the GTCR.MD[2:0] or the GTCR.MD[3:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0000 0000h with other settings.

22.2.5 General PWM Timer Start Source Select Register (GTSSR)

Address(es): GPTW0.GTSSR 000C 2010h, GPTW1.GTSSR 000C 2110h, GPTW2.GTSSR 000C 2210h, GPTW3.GTSSR 000C 2310h, GPTW4.GTSSR 000C 2410h, GPTW5.GTSSR 000C 2510h, GPTW6.GTSSR 000C 2610h, GPTW7.GTSSR 000C 2710h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGA signal. 1 0: The counter starts at a falling edge of the GTETRGA signal. 1 1: The counter starts at both edges of the GTETRGA signal.	R/W *1
b1	SSGTRGAF			R/W *1
b2	SSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGB signal. 1 0: The counter starts at a falling edge of the GTETRGB signal. 1 1: The counter starts at both edges of the GTETRGB signal.	R/W *1
b3	SSGTRGBF			R/W *1
b4	SSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGC signal. 1 0: The counter starts at a falling edge of the GTETRGC signal. 1 1: The counter starts at both edges of the GTETRGC signal.	R/W *1
b5	SSGTRGCF			R/W *1
b6	SSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGD signal. 1 0: The counter starts at a falling edge of the GTETRGD signal. 1 1: The counter starts at both edges of the GTETRGD signal.	R/W *1
b7	SSGTRGDF			R/W *1
b8	SSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter starts at a rising edge of the GTIOCnA signal.	R/W
b9	SSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	SSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to start counting. 0 1: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter starts at a falling edge of the GTIOCnA signal.	R/W
b11	SSCAFBH			R/W
b12	SSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter starts at a rising edge of the GTIOCnB signal.	R/W
b13	SSCBRAH			R/W
b14	SSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to start counting. 0 1: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter starts at a falling edge of the GTIOCnB signal.	R/W
b15	SSCBFAH			R/W
b16	SSELCA	ELCA Event Source Count Start Enable	0: Disables count start by the ELCA event input 1: Enables count start by the ELCA event input	R/W *1
b17	SSELCB	ELCB Event Source Count Start Enable	0: Disables count start by the ELCB event input 1: Enables count start by the ELCB event input	R/W *1
b18	SSELCC	ELCC Event Source Count Start Enable	0: Disables count start by the ELCC event input 1: Enables count start by the ELCC event input	R/W *1
b19	SSELCD	ELCD Event Source Count Start Enable	0: Disables count start by the ELCD event input 1: Enables count start by the ELCD event input	R/W *1
b20	SSELCE	ELCE Event Source Count Start Enable	0: Disables count start by the ELCE event input 1: Enables count start by the ELCE event input	R/W *1
b21	SSELCF	ELCF Event Source Count Start Enable	0: Disables count start by the ELCF event input 1: Enables count start by the ELCF event input	R/W *1
b22	SSELCG	ELCG Event Source Count Start Enable	0: Disables count start by the ELCG event input 1: Enables count start by the ELCG event input	R/W *1
b23	SSELCH	ELCH Event Source Count Start Enable	0: Disables count start by the ELCH event input 1: Enables count start by the ELCH event input	R/W *1
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTRT	Software Source Count Start Enable	0: Disables count start by the GTSTR register 1: Enables count start by the GTSTR register	R/W *1

n = 0 to 7

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW0 to GPTW2.

The GTSSR register sets a count start source for the GTCNT counter.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

22.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPTW0.GTPSR 000C 2014h, GPTW1.GTPSR 000C 2114h, GPTW2.GTPSR 000C 2214h, GPTW3.GTPSR 000C 2314h, GPTW4.GTPSR 000C 2414h, GPTW5.GTPSR 000C 2514h, GPTW6.GTPSR 000C 2614h, GPTW7.GTPSR 000C 2714h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGA signal. 1 0: The counter stops at a falling edge of the GTETRGA signal. 1 1: The counter stops at both edges of the GTETRGA signal.	R/W *1
b1	PSGTRGAF			R/W *1
b2	PSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGB signal. 1 0: The counter stops at a falling edge of the GTETRGB signal. 1 1: The counter stops at both edges of the GTETRGB signal.	R/W *1
b3	PSGTRGBF			R/W *1
b4	PSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGC signal. 1 0: The counter stops at a falling edge of the GTETRGC signal. 1 1: The counter stops at both edges of the GTETRGC signal.	R/W *1
b5	PSGTRGCF			R/W *1
b6	PSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGD signal. 1 0: The counter stops at a falling edge of the GTETRGD signal. 1 1: The counter stops at both edges of the GTETRGD signal.	R/W *1
b7	PSGTRGDF			R/W *1
b8	PSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter stops at a rising edge of the GTIOCnA signal.	R/W
b9	PSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	PSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to stop counting. 0 1: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter stops at a falling edge of the GTIOCnA signal.	R/W
b11	PSCAFBH			R/W
b12	PSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter stops at a rising edge of the GTIOCnB signal.	R/W
b13	PSCBRAH			R/W
b14	PSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to stop counting. 0 1: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter stops at a falling edge of the GTIOCnB signal.	R/W
b15	PSCBFAH			R/W
b16	PSELCA	ELCA Event Source Count Stop Enable	0: Disables count stop by the ELCA event input 1: Enables count stop by the ELCA event input	R/W *1
b17	PSELCB	ELCB Event Source Count Stop Enable	0: Disables count stop by the ELCB event input 1: Enables count stop by the ELCB event input	R/W *1
b18	PSELCC	ELCC Event Source Count Stop Enable	0: Disables count stop by the ELCC event input 1: Enables count stop by the ELCC event input	R/W *1
b19	PSELCD	ELCD Event Source Count Stop Enable	0: Disables count stop by the ELCD event input 1: Enables count stop by the ELCD event input	R/W *1
b20	PSELCE	ELCE Event Source Count Stop Enable	0: Disables count stop by the ELCE event input 1: Enables count stop by the ELCE event input	R/W *1
b21	PSELCF	ELCF Event Source Count Stop Enable	0: Disables count stop by the ELCF event input 1: Enables count stop by the ELCF event input	R/W *1
b22	PSELCG	ELCG Event Source Count Stop Enable	0: Disables count stop by the ELCG event input 1: Enables count stop by the ELCG event input	R/W *1
b23	PSELCH	ELCH Event Source Count Stop Enable	0: Disables count stop by the ELCH event input 1: Enables count stop by the ELCH event input	R/W *1
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTOP	Software Source Count Stop Enable	0: Disables count stop by the GTSTP register 1: Enables count stop by the GTSTP register	R/W *1

n = 0 to 7

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW0 to GPTW2.

The GTPSR register sets a count stop source for the GTCNT counter.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

22.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPTW0.GTCSR 000C 2018h, GPTW1.GTCSR 000C 2118h, GPTW2.GTCSR 000C 2218h, GPTW3.GTCSR 000C 2318h, GPTW4.GTCSR 000C 2418h, GPTW5.GTCSR 000C 2518h, GPTW6.GTCSR 000C 2618h, GPTW7.GTCSR 000C 2718h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCLR	—	—	—	CP1CCE	CSCMSC[2:0]			CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGA signal. 1 0: The counter is cleared at a falling edge of the GTETRGA signal. 1 1: The counter is cleared at both edges of the GTETRGA signal.	R/W *1
b1	CSGTRGAF			R/W *1
b2	CSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGB signal. 1 0: The counter is cleared at a falling edge of the GTETRGB signal. 1 1: The counter is cleared at both edges of the GTETRGB signal.	R/W *1
b3	CSGTRGBF			R/W *1
b4	CSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGC signal. 1 0: The counter is cleared at a falling edge of the GTETRGC signal. 1 1: The counter is cleared at both edges of the GTETRGC signal.	R/W *1
b5	CSGTRGCF			R/W *1
b6	CSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGD signal. 1 0: The counter is cleared at a falling edge of the GTETRGD signal. 1 1: The counter is cleared at both edges of the GTETRGD signal.	R/W *1
b7	CSGTRGDF			R/W *1
b8	CSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is cleared at a rising edge of the GTIOCnA signal.	R/W
b9	CSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	CSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is cleared at a falling edge of the GTIOCnA signal.	R/W
b11	CSCAFBH			R/W
b12	CSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is cleared at a rising edge of the GTIOCnB signal.	R/W
b13	CSCBRAH			R/W
b14	CSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is cleared at a falling edge of the GTIOCnB signal.	R/W
b15	CSCBFAH			R/W
b16	CSELCA	ELCA Event Source Counter Clear Enable	0: Disables counter clear by the ELCA event input 1: Enables counter clear by the ELCA event input	R/W *1
b17	CSELCB	ELCB Event Source Counter Clear Enable	0: Disables counter clear by the ELCB event input 1: Enables counter clear by the ELCB event input	R/W *1
b18	CSELCC	ELCC Event Source Counter Clear Enable	0: Disables counter clear by the ELCC event input 1: Enables counter clear by the ELCC event input	R/W *1
b19	CSELCD	ELCD Event Source Counter Clear Enable	0: Disables counter clear by the ELCD event input 1: Enables counter clear by the ELCD event input	R/W *1
b20	CSELCE	ELCE Event Source Counter Clear Enable	0: Disables counter clear by the ELCE event input 1: Enables counter clear by the ELCE event input	R/W *1
b21	CSELCF	ELCF Event Source Counter Clear Enable	0: Disables counter clear by the ELCF event input 1: Enables counter clear by the ELCF event input	R/W *1
b22	CSELCG	ELCG Event Source Counter Clear Enable	0: Disables counter clear by the ELCG event input 1: Enables counter clear by the ELCG event input	R/W *1
b23	CSELCH	ELCH Event Source Counter Clear Enable	0: Disables counter clear by the ELCH event input 1: Enables counter clear by the ELCH event input	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b26 to b24	CSCMSC[2:0]	Compare Match/Input Capture/ Synchronous Counter Clearing Source Counter Clear Enable	b26 b24 0 0 0: Counter clear disabled by Compare match/Input capture/Synchronous counter clearing group. 0 0 1: Counter clear enabled at the GTCCRA register compare match/Input capture. 0 1 0: Counter clear enabled at the GTCCRB register compare match/Input capture. 0 1 1: Counter clear enabled at the GTCCRC register compare match. 1 0 0: Counter clear enabled at the GTCCRD register compare match. 1 0 1: Counter clear enabled at the GTCCRE register compare match. 1 1 0: Counter clear enabled at the GTCCRF register compare match. 1 1 1: Counter clear enabled at the synchronous counter clearing group.	R/W
b27	CP1CCE	Complementary PWM Mode 1 Crest Source Counter Clear Enable*3	0: Counter clear disabled at the crest of complementary PWM mode 1 1: Counter clear enabled at the crest of complementary PWM mode 1	R/W *1
b30 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CCLR	Software Source Counter Clear Enable	0: Disables counter clear by the GTCLR register 1: Enables counter clear by the GTCLR register	R/W *1

n = 0 to 7

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW0 to GPTW2.

Note 3. This bit in each of GPTW3 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTCSR register sets a counter clear source for the GTCNT counter.

Counter clearing can be executed whether the counter is running (the GTCR.CST bit = 1) or stopped (the GTCR.CST bit = 0).

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

CSCMSC[2:0] Bits (Compare Match/Input Capture/Synchronous Counter Clearing Source Counter Clear Enable)

Select whether to enable or disable for the counter clear of the GTCNT counter by compare match/input capture/ synchronous counter clearing group. When counter clearing in response to a match in comparison or input capture is enabled, either can be used as a source for synchronous clearing by inter channel cooperation as described in section 22.3.8.3, Synchronous Clear Operation by Inter Channel Cooperation. When counter clearing by input capture is enabled while the setting of the CSCMSC[2:0] bits is 001b or 010b, the same sources for input capture as one set in the GTICmSR (m = A, B) register must be set as the counter clearing sources in the GTCSR register. If the timer prescaler is not to be used (GTCR.TPCS[3:0] = 0000b), input capture triggered by a source in another channel can be used as the trigger for clearing the counter (by setting GTICASR.ASOC or GTICBSR.BSOC to 1). Input capture triggered by a source in another channel cannot be used as the trigger for synchronous clearing in interlinked channel operation, but can be used as the trigger for clearing of the counter. Setting of GTCSR is not required in such cases.

Since the compare match by the register that is performing the buffer operation (including the wave mode specific case) does not occur, the counter clear enable setting that makes the target register of the buffer operation the compare match factor is invalid.

In complementary PWM mode, the counter clear enable setting for compare match of the GTCCRB register, GTCCRE register, and GTCCRF register is invalid even when the buffer operation is not performed.

CP1CCE Bit (Complementary PWM Mode 1 Crest Source Counter Clear Enable)

Select enable or disable for the counter clear at the crest of complementary PWM mode 1.
To enable this bit, do not set 1 to the PSYE bit of the GTIOR register.
It is valid only for the master channel in complementary PWM mode. The master channel setting also clears the GTCNT counter of the slave channel in complementary PWM mode.

22.2.8 General PWM Timer Count-Up Source Select Register (GTUPSR)

Address(es): GPTW3.GTUPSR 000C 231Ch, GPTW4.GTUPSR 000C 241Ch, GPTW5.GTUPSR 000C 251Ch, GPTW6.GTUPSR 000C 261Ch, GPTW7.GTUPSR 000C 271Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	USILVL[3:0]				USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGA signal. 1 0: The counter is incremented at a falling edge of the GTETRGA signal. 1 1: The counter is incremented at both edges of the GTETRGA signal.	R/W
b1	USGTRGAF			R/W
b2	USGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGB signal. 1 0: The counter is incremented at a falling edge of the GTETRGB signal. 1 1: The counter is incremented at both edges of the GTETRGB signal.	R/W
b3	USGTRGBF			R/W
b4	USGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGC signal. 1 0: The counter is incremented at a falling edge of the GTETRGC signal. 1 1: The counter is incremented at both edges of the GTETRGC signal.	R/W
b5	USGTRGCF			R/W
b6	USGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGD signal. 1 0: The counter is incremented at a falling edge of the GTETRGD signal. 1 1: The counter is incremented at both edges of the GTETRGD signal.	R/W
b7	USGTRGDF			R/W
b8	USCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is incremented at a rising edge of the GTIOCnA signal.	R/W
b9	USCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	USCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is incremented at a falling edge of the GTIOCnA signal.	R/W
b11	USCAFBH			R/W
b12	USCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is incremented at a rising edge of the GTIOCnB signal.	R/W
b13	USCBRAH			R/W
b14	USCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is incremented at a falling edge of the GTIOCnB signal.	R/W
b15	USCBFAH			R/W
b16	USELCA	ELCA Event Source Count-Up Enable	0: Disables count-up by the ELCA event input 1: Enables count-up by the ELCA event input	R/W
b17	USELCB	ELCB Event Source Count-Up Enable	0: Disables count-up by the ELCB event input 1: Enables count-up by the ELCB event input	R/W
b18	USELCC	ELCC Event Source Count-Up Enable	0: Disables count-up by the ELCC event input 1: Enables count-up by the ELCC event input	R/W
b19	USELCD	ELCD Event Source Count-Up Enable	0: Disables count-up by the ELCD event input 1: Enables count-up by the ELCD event input	R/W
b20	USELCE	ELCE Event Source Count-Up Enable	0: Disables count-up by the ELCE event input 1: Enables count-up by the ELCE event input	R/W
b21	USELCF	ELCF Event Source Count-Up Enable	0: Disables count-up by the ELCF event input 1: Enables count-up by the ELCF event input	R/W
b22	USELCG	ELCG Event Source Count-Up Enable	0: Disables count-up by the ELCG event input 1: Enables count-up by the ELCG event input	R/W
b23	USELCH	ELCH Event Source Count-Up Enable	0: Disables count-up by the ELCH event input 1: Enables count-up by the ELCH event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b27to b24	USILVL[3:0]	External Input Level Source Count-Up Enable*1	b27 b24 0 0 0 0: Disable count-up by external input. 0 0 0 1: Setting prohibited. 0 0 1 0: Enable count-up by GTIOCnA pin input level Low. 0 0 1 1: Enable count-up by GTIOCnA pin input level High. 0 1 0 0: Enable count-up by GTIOCnB pin input level Low. 0 1 0 1: Enable count-up by GTIOCnB pin input level High. 0 1 1 0: Setting prohibited. 0 1 1 1: Setting prohibited. 1 0 0 0: Enable count-up by GTETRGA pin input level Low. 1 0 0 1: Enable count-up by GTETRGA pin input level High. 1 0 1 0: Enable count-up by GTETRGB pin input level Low. 1 0 1 1: Enable count-up by GTETRGB pin input level High. 1 1 0 0: Enable count-up by GTETRGC pin input level Low. 1 1 0 1: Enable count-up by GTETRGC pin input level High. 1 1 1 0: Enable count-up by GTETRGD pin input level Low. 1 1 1 1: Enable count-up by GTETRGD pin input level High.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 3 to 7

Note 1. This bit in each of GPTW6 and GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTUPSR register sets a count-up source for the GTCNT counter.

When at least one bit among bits in the GTUPSR register is set as 1, counting of the GTCNT counter by the count clock set by the GTCR.TPCS[3:0] bits becomes invalid, and the count-up by a source set as 1 by the GTUPSR register is executed.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

22.2.9 General PWM Timer Count-Down Source Select Register (GTDNSR)

Address(es): GPTW3.GTDNSR 000C 2320h, GPTW4.GTDNSR 000C 2420h, GPTW5.GTDNSR 000C 2520h, GPTW6.GTDNSR 000C 2620h, GPTW7.GTDNSR 000C 2720h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	DSILVL[3:0]				DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGA signal. 1 0: The counter is decremented at a falling edge of the GTETRGA signal. 1 1: The counter is decremented at both edges of the GTETRGA signal.	R/W
b1	DSGTRGAF			R/W
b2	DSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGB signal. 1 0: The counter is decremented at a falling edge of the GTETRGB signal. 1 1: The counter is decremented at both edges of the GTETRGB signal.	R/W
b3	DSGTRGBF			R/W
b4	DSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGC signal. 1 0: The counter is decremented at a falling edge of the GTETRGC signal. 1 1: The counter is decremented at both edges of the GTETRGC signal.	R/W
b5	DSGTRGCF			R/W
b6	DSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGD signal. 1 0: The counter is decremented at a falling edge of the GTETRGD signal. 1 1: The counter is decremented at both edges of the GTETRGD signal.	R/W
b7	DSGTRGDF			R/W
b8	DSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is decremented at a rising edge of the GTIOCnA signal.	R/W
b9	DSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	DSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is decremented at a falling edge of the GTIOCnA signal.	R/W
b11	DSCAFBH			R/W
b12	DSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is decremented at a rising edge of the GTIOCnB signal.	R/W
b13	DSCBRAH			R/W
b14	DSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is decremented at a falling edge of the GTIOCnB signal.	R/W
b15	DSCBFAH			R/W
b16	DSELCA	ELCA Event Source Count-Down Enable	0: Disables count-down by the ELCA event input 1: Enables count-down by the ELCA event input	R/W
b17	DSELCB	ELCB Event Source Count-Down Enable	0: Disables count-down by the ELCB event input 1: Enables count-down by the ELCB event input	R/W
b18	DSELCC	ELCC Event Source Count-Down Enable	0: Disables count-down by the ELCC event input 1: Enables count-down by the ELCC event input	R/W
b19	DSELCD	ELCD Event Source Count-Down Enable	0: Disables count-down by the ELCD event input 1: Enables count-down by the ELCD event input	R/W
b20	DSELCE	ELCE Event Source Count-Down Enable	0: Disables count-down by the ELCE event input 1: Enables count-down by the ELCE event input	R/W
b21	DSELCF	ELCF Event Source Count-Down Enable	0: Disables count-down by the ELCF event input 1: Enables count-down by the ELCF event input	R/W
b22	DSELCG	ELCG Event Source Count-Down Enable	0: Disables count-down by the ELCG event input 1: Enables count-down by the ELCG event input	R/W
b23	DSELCH	ELCH Event Source Count-Down Enable	0: Disables count-down by the ELCH event input 1: Enables count-down by the ELCH event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	DSILVL[3:0]	External Input Level Source Count-Down Enable*1	b27 b24 0 0 0 0: Disable count-down by external input. 0 0 0 1: Setting prohibited. 0 0 1 0: Enable count-down by GTIOCnA pin input level Low. 0 0 1 1: Enable count-down by GTIOCnA pin input level High. 0 1 0 0: Enable count-down by GTIOCnB pin input level Low. 0 1 0 1: Enable count-down by GTIOCnB pin input level High. 0 1 1 0: Setting prohibited. 0 1 1 1: Setting prohibited. 1 0 0 0: Enable count-down by GTETRGA pin input level Low. 1 0 0 1: Enable count-down by GTETRGA pin input level High. 1 0 1 0: Enable count-down by GTETRGB pin input level Low. 1 0 1 1: Enable count-down by GTETRGB pin input level High. 1 1 0 0: Enable count-down by GTETRGC pin input level Low. 1 1 0 1: Enable count-down by GTETRGC pin input level High. 1 1 1 0: Enable count-down by GTETRGD pin input level Low. 1 1 1 1: Enable count-down by GTETRGD pin input level High.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 3 to 7

Note 1. This bit in each of GPTW6 and GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTDNSR register sets a count-down source for the GTCNT counter.

When at least one bit among bits in the GTDNSR register is set as 1, counting of the GTCNT counter by the count clock set by the GTCR.TPCS[3:0] bits becomes invalid, and the count-down by a source set as 1 by the GTDNSR register is executed.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

22.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR)

Address(es): GPTW0.GTICASR 000C 2024h, GPTW1.GTICASR 000C 2124h, GPTW2.GTICASR 000C 2224h, GPTW3.GTICASR 000C 2324h, GPTW4.GTICASR 000C 2424h, GPTW5.GTICASR 000C 2524h, GPTW6.GTICASR 000C 2624h, GPTW7.GTICASR 000C 2724h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ASOC	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ASGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGA signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGA signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGA signal.	R/W
b1	ASGTRGAF			R/W
b2	ASGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGB signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGB signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGB signal.	R/W
b3	ASGTRGBF			R/W
b4	ASGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGC signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGC signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGC signal.	R/W
b5	ASGTRGCF			R/W
b6	ASGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGD signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGD signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGD signal.	R/W
b7	ASGTRGDF			R/W
b8	ASCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal.	R/W
b9	ASCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	ASCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal.	R/W
b11	ASCAFBH			R/W
b12	ASCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal.	R/W
b13	ASCBRAH			R/W
b14	ASCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal.	R/W
b15	ASCBFAH			R/W
b16	ASELCA	ELCA Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCA event input 1: Enables GTCCRA input capture by the ELCA event input	R/W
b17	ASELCB	ELCB Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCB event input 1: Enables GTCCRA input capture by the ELCB event input	R/W
b18	ASELCC	ELCC Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCC event input 1: Enables GTCCRA input capture by the ELCC event input	R/W
b19	ASELCD	ELCD Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCD event input 1: Enables GTCCRA input capture by the ELCD event input	R/W
b20	ASELCE	ELCE Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCE event input 1: Enables GTCCRA input capture by the ELCE event input	R/W
b21	ASELCF	ELCF Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCF event input 1: Enables GTCCRA input capture by the ELCF event input	R/W
b22	ASELCG	ELCG Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCG event input 1: Enables GTCCRA input capture by the ELCG event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b23	ASELCH	ELCH Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCH event input 1: Enables GTCCRA input capture by the ELCH event input	R/W
b24	ASOC	Other Channel Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the other channel factor 1: Enables GTCCRA input capture by the other channel factor	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The GTICASR register sets an input capture source for the GTCCRA register.
When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

ASOC Bit (Other Channel Source GTCCRA Input Capture Enable)

This bit enables or disables triggering of input capture of a timer value by the GTCCRA register through interlinked channel operation with the trigger source specified in the GTICCR register in another channel. Input capture triggered by a source in another channel enabled by this bit cannot be used as a source to trigger input capture in another channel through specification by the ICAFA and ICBFA bits in the GTICCR register in its own channel.

22.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR)

Address(es): GPTW0.GTICBSR 000C 2028h, GPTW1.GTICBSR 000C 2128h, GPTW2.GTICBSR 000C 2228h, GPTW3.GTICBSR 000C 2328h, GPTW4.GTICBSR 000C 2428h, GPTW5.GTICBSR 000C 2528h, GPTW6.GTICBSR 000C 2628h, GPTW7.GTICBSR 000C 2728h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	BSOC	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGA signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGA signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGA signal.	R/W *1
b1	BSGTRGAF			R/W *1
b2	BSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGB signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGB signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGB signal.	R/W *1
b3	BSGTRGBF			R/W *1
b4	BSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGC signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGC signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGC signal.	R/W *1
b5	BSGTRGCF			R/W *1
b6	BSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGD signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGD signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGD signal.	R/W *1
b7	BSGTRGDF			R/W *1
b8	BSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal.	R/W
b9	BSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	BSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal.	R/W
b11	BSCAFBH			R/W
b12	BSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal.	R/W
b13	BSCBRAH			R/W
b14	BSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal.	R/W
b15	BSCBFAH			R/W
b16	BSELCA	ELCA Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCA event input 1: Enables GTCCRB input capture by the ELCA event input	R/W *1
b17	BSELCB	ELCB Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCB event input 1: Enables GTCCRB input capture by the ELCB event input	R/W *1
b18	BSELCC	ELCC Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCC event input 1: Enables GTCCRB input capture by the ELCC event input	R/W *1
b19	BSELCD	ELCD Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCD event input 1: Enables GTCCRB input capture by the ELCD event input	R/W *1
b20	BSELCE	ELCE Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCE event input 1: Enables GTCCRB input capture by the ELCE event input	R/W *1
b21	BSELCF	ELCF Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCF event input 1: Enables GTCCRB input capture by the ELCF event input	R/W *1
b22	BSELCG	ELCG Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCG event input 1: Enables GTCCRB input capture by the ELCG event input	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b23	BSELCH	ELCH Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCH event input 1: Enables GTCCRB input capture by the ELCH event input	R/W *1
b24	BSOC	Other Channel Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the other channel factor 1: Enables GTCCRB input capture by the other channel factor	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTICBSR register sets an input capture source for the GTCCRB register.
When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

BSOC Bit (Other Channel Source GTCCRB Input Capture Enable)

This bit enables or disables triggering of input capture of a timer value by the GTCCRB register through interlinked channel operation with the trigger source specified in the GTICCR register in another channel. Input capture triggered by a source in another channel enabled by this bit cannot be used as a source to trigger input capture in another channel through specification by the ICAFB and ICBFB bits in the GTICCR register in its own channel.

22.2.12 General PWM Timer Control Register (GTCR)

- GPTW0.GTCR, GPTW1.GTCR, GPTW2.GTCR

Address(es): GPTW0.GTCR 000C 202Ch, GPTW1.GTCR 000C 212Ch, GPTW2.GTCR 000C 222Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CKEG[1:0]		TPCS[3:0]				—	—	—	MD[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
	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is started	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ICDS	Input Capture Operation Select at Count Stop	0: Input capture is operated at count stop 1: Input capture is not operated at count stop	R/W
b9	SCGTIOC	GTIOC input Source Synchronous Clear Enable	0: Disables to use the counter clear by GTIOC input as the clear factor for other channels 1: Enables to use the counter clear by GTIOC input as the clear factor for other channels	R/W
b11, b10	SSCGRP[1:0]	Synchronous Set/Clear Group Select	b11 b10 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W *1
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSCEN	Synchronous Set/Clear Enable	0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W *1
b19 to b16	MD[3:0]	Mode Select	b19 b16 0 0 0 0: Sawtooth-wave PWM mode 1 (single buffer or double buffer possible) 0 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 0 1 0: Sawtooth-wave PWM mode 2 (single buffer or double buffer possible) 0 0 1 1: Setting prohibited 0 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 0 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 0 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 0 1 1 1: Setting prohibited 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Setting prohibited 1 0 1 1: Setting prohibited 1 1 0 0: Complementary PWM mode 1 (transfer at crest) 1 1 0 1: Complementary PWM mode 2 (transfer at trough) 1 1 1 0: Complementary PWM mode 3 (transfer at crest and trough) 1 1 1 1: Complementary PWM mode 4 (immediate transfer)	R/W *1
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26 to b23	TPCS[3:0]	Timer Prescaler Select	b26 b23 0 0 0 0: PCLKA 0 0 0 1: PCLKA/2 0 0 1 0: PCLKA/4 0 0 1 1: PCLKA/8 0 1 0 0: PCLKA/16 0 1 0 1: PCLKA/32 0 1 1 0: PCLKA/64 0 1 1 1: PCLKA/128 1 0 0 0: PCLKA/256 1 0 0 1: PCLKA/512 1 0 1 0: PCLKA/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (via the POEG) 1 1 0 1: GTETRGB (via the POEG) 1 1 1 0: GTETRGC (via the POEG) 1 1 1 1: GTETRGD (via the POEG)	R/W *1
b28, b27	CKEG[1:0]	Clock Edge Select	b28 b27 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count 1 0: Select both edge of GTETRG for clock count 1 1: Select both edge of GTETRG for clock count	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTCR register controls the GTCNT counter.

CST Bit (Count Start)

This bit controls start or stop for the GTCNT counter.

[Setting conditions]

- When 1 is written to a bit related to a channel number for the GTSTR register while the GTSSR.CSTRT bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input (n = 0 to 2), which is enabled as a count start source by the GTSSR register, is generated
- 1 is written by software

[Clearing conditions]

- When 1 is written to a bit related to a channel number for the GTSTP register while the GTPSR.CSTOP bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input, which is enabled as a count stop source by the GTPSR register, is generated
- 0 is written by software

ICDS Bit (Input Capture Operation Select at Count Stop)

This bit selects Input capture operation at count stop when the input capture function is selected.

SCGTIOC Bit (GTIOC input Source Synchronous Clear Enable)

This bit selects enables or disables the use of counter clearing at the GTIOCnA/GTIOCnB input pin selected by the GTCSR register as a counter clear factor for other channels.

SSCGRP[1:0] Bits (Synchronous Set/Clear Group Select)

This bit selects the synchronization set/clear channel group.

SSCEN Bit (Synchronous Set/Clear Enable)

This bit selects disable or enable of Synchronous set/clear.
In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[3:0] Bits (Mode Select)

These bits select the GPTW operating mode.
In complementary PWM mode, the master channel setting of the MD[3:0] bits is also used to control the slave channels.
Only the MD[3:2] bits are valid at input capture.
Counting in sawtooth-wave mode is performed with 00b for the MD[3:2] bits, counting in triangle-wave mode is performed with 01b for the MD[3:2] bits.
Set the MD[3:0] bits while the GTCNT counter operation is stopped.
During the event count operation (at least one of the bits in the GTUPSR or GTDNSR register is 1), set the MD[3:0] bits to the initial value (0000b).

TPCS[3:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.
Set the TPCS[3:0] bits while the GTCNT counter operation is stopped.

CKEG[1:0] Bits (Clock Edge Select)

When GTETRG input is selected by TPCS[3:0] bits, select the edge of GTETRG input used as the clock of GTCNT counter.
Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

- GPTW3.GTCR, GPTW4.GTCR, GPTW5.GTCR, GPTW6.GTCR, GPTW7.GTCR

Address(es): GPTW3.GTCR 000C 232Ch, GPTW4.GTCR 000C 242Ch, GPTW5.GTCR 000C 252Ch, GPTW6.GTCR 000C 262Ch, GPTW7.GTCR 000C 272Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CKEG[1:0]		TPCS[3:0]				—	—	—	—	MD[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
	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is started	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ICDS	Input Capture Operation Select at Count Stop	0: Input capture is operated at count stop 1: Input capture is not operated at count stop	R/W
b9	SCGTIOC	GTIOC input Source Synchronous Clear Enable	0: Disables to use the counter clear by GTIOC input as the clear factor for other channels 1: Enables to use the counter clear by GTIOC input as the clear factor for other channels	R/W
b11, b10	SSCGRP[1:0]	Synchronous Set/Clear Group Select	b11 b10 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSCEN	Synchronous Set/Clear Enable	0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W
b18 to b16	MD[2:0]	Mode Select	b18 b16 0 0 0: Sawtooth-wave PWM mode (single buffer or double buffer possible) 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
b22 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26 to b23	TPCS[3:0]	Timer Prescaler Select	b26 b23 0 0 0 0: PCLKA 0 0 0 1: PCLKA/2 0 0 1 0: PCLKA/4 0 0 1 1: PCLKA/8 0 1 0 0: PCLKA/16 0 1 0 1: PCLKA/32 0 1 1 0: PCLKA/64 0 1 1 1: PCLKA/128 1 0 0 0: PCLKA/256 1 0 0 1: PCLKA/512 1 0 1 0: PCLKA/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (via the POEG) 1 1 0 1: GTETRGB (via the POEG) 1 1 1 0: GTETRGC (via the POEG) 1 1 1 1: GTETRGD (via the POEG)	R/W
b28, b27	CKEG[1:0]	Clock Edge Select	b28 b27 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count 1 0: Select both edge of GTETRG for clock count 1 1: Select both edge of GTETRG for clock count	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTCR register controls the GTCNT counter.

CST Bit (Count Start)

This bit controls start or stop for the GTCNT counter.

[Setting conditions]

- When 1 is written to a bit related to a channel number for the GTSTR register while the GTSSR.CSTRT bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input (n = 3 to 7), which is enabled as a count start source by the GTSSR register, is generated
- 1 is written by software

[Clearing conditions]

- When 1 is written to a bit related to a channel number for the GTSTP register while the GTPSR.CSTOP bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input, which is enabled as a count stop source by the GTPSR register, is generated
- 0 is written by software

ICDS Bit (Input Capture Operation Select at Count Stop)

This bit selects Input capture operation at count stop when the input capture function is selected.

SCGTIOC Bit (GTIOC input Source Synchronous Clear Enable)

This bit selects enables or disables the use of counter clearing at the GTIOCnA/GTIOCnB input pin selected by the GTCSR register as a counter clear factor for other channels.

SSCGRP[1:0] Bits (Synchronous Set/Clear Group Select)

This bit selects the synchronization set/clear channel group.

SSCEN Bit (Synchronous Set/Clear Enable)

This bit selects disable or enable of Synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[2:0] Bits (Mode Select)

These bits select the GPTW operating mode.
Only the MD[2] bit is valid at input capture.
Counting in sawtooth-wave mode is performed with 0 for the MD[2] bit, counting in triangle-wave mode is performed with 1 for the MD[2] bit.
Set the MD[2:0] bits while the GTCNT counter operation is stopped.
During the event count operation (at least one of the bits in the GTUPSR or GTDNSR register is 1), set the MD[2:0] bits to the initial value (000b).

TPCS[3:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.
Set the TPCS[3:0] bits while the GTCNT counter operation is stopped.

CKEG[1:0] Bits (Clock Edge Select)

When GTETRG input is selected by TPCS[3:0] bits, select the edge of GTETRG input used as the clock of GTCNT counter.
Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

22.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)

Address(es): GPTW0.GTUDDTYC 000C 2030h, GPTW1.GTUDDTYC 000C 2130h, GPTW2.GTUDDTYC 000C 2230h, GPTW3.GTUDDTYC 000C 2330h, GPTW4.GTUDDTYC 000C 2430h, GPTW5.GTUDDTYC 000C 2530h, GPTW6.GTUDDTYC 000C 2630h, GPTW7.GTUDDTYC 000C 2730h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	UD	Count Direction Setting	0: GTCNT counts down. 1: GTCNT counts up.	R/W
b1	UDF	Forcible Count Direction Setting	0: Count direction is not forcibly set. 1: Count direction is forcibly set.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	OADTY[1:0]	GTIOCnA Pin Output Duty Setting	b17b16 0 x: Compare matches determine the duty cycle of the output on the GTIOCnA pin. 1 0: The duty cycle of the output on the GTIOCnA pin is 0%. 1 1: The duty cycle of the output on the GTIOCnA pin is 100%.	R/W
b18	OADTYF	GTIOCnA Pin Output Duty Forced Setting	0: Duty of the GTIOCnA pin output is not forcibly set. 1: Duty of the GTIOCnA pin output is forcibly set.	R/W
b19	OADTYR	Output after Release of GTIOCnA Pin Output 0%/ 100% Duty Cycle Settings	0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OBDTY[1:0]	GTIOCnB Pin Output Duty Setting	b25b24 0 x: Compare matches determine the duty cycle of the output on the GTIOCnB pin 1 0: The duty cycle of the output on the GTIOCnB pin is 0%. 1 1: The duty cycle of the output on the GTIOCnB pin is 100%.	R/W
b26	OBDTYF	GTIOCnB Pin Output Duty Forced Setting	0: Duty of the GTIOCnB pin output is not forcibly set. 1: Duty of the GTIOCnB pin output is forcibly set.	R/W
b27	OBDTYR	Output after Release of GTIOCnB Pin Output 0%/ 100% Duty Cycle Settings	0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The GTUDDTYC register sets the count direction (up-counting/down-counting) for the GTCNT counter and configures duty setting for output from the GTIOCnA and GTIOCnB pins.
The setting is invalid during the event count operation, sawtooth-wave PWM mode 2, Complementary PWM mode.

(1) Setting of Count Direction

- In sawtooth-wave mode

When the UD bit value is set to 0 during up-counting, the count direction is changed at an overflow (count clock when the GTCNT counter value is equal to the GTPR register value). When the UD bit value is set to 1 during down-counting, the count direction is changed at an underflow (count clock when the GTCNT counter value is equal to 0000 0000h).
If the UD bit value is changed from 1 to 0 with the UDF bit being 0 and while counting is stopped, the counter starts up-counting and the count direction is changed at an overflow (count clock when the GTCNT counter value is equal to the GTPR register value).
If the UD bit value is changed from 0 to 1 with the UDF bit being 0 and while counting is stopped, the counter starts down-counting and the count direction is changed at an underflow (count clock when the GTCNT counter value is equal to 0000 0000h).
When the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

- In triangle-wave mode

Even if the UD bit value is changed during counting, the change will not be reflected in the count direction.
If the UD bit value is modified while the UDF bit is 0 and counting is stopped, the change will not be reflected in the count direction when counting starts. If the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

UD Bit (Count Direction Setting)

This bit sets the count direction (up-counting or down-counting) for the GTCNT counter.

UDF Bit (Forcible Count Direction Setting)

This bit forcibly sets the count direction when the GTCNT counter starts operation as the UD bit value.
Only 0 should be written to this bit during count operation.
When 1 has been written to this bit while counting is stopped, this bit should be returned to 0 before counting starts.

(2) Setting of Output Duty

- In sawtooth-wave mode

When the OADTY[1:0]/OBDTY[1:0] bits are changed during up-count operation, the duty setting changed at an overflow is reflected.
When the OADTY[1:0]/OBDTY[1:0] bits are changed during down-count operation, the duty setting changed at an underflow is reflected.
When the OADTYF/OBDTYF bit is 0 while count operation is stopped and the OADTY[1:0]/OBDTY[1:0] bits are changed, the change in the duty-cycle setting is not reflected in the first count operation, but the change is reflected on an overflow in the case of counting up and on an underflow in the case of counting down.
When the OADTYF/OBDTYF bit is 1 while count operation is stopped and the OADTY[1:0]/OBDTY[1:0] bits are changed, the change in the duty-cycle setting is immediately reflected during the first count operation.

- In triangle-wave mode

When the OADTY[1:0]/OBDTY[1:0] bits are changed during count operation, the duty-cycle setting changed at an underflow is reflected.
When the OADTYF/OBDTYF bit is 0 during count operation and the OADTY[1:0]/OBDTY[1:0] bits are changed, the duty-cycle setting changed during the first count operation is not reflected, the duty-cycle setting changed at an underflow is reflected.
When the OADTYF/OBDTYF bit is 1 while count operation is stopped and the OADTY[1:0]/OBDTY[1:0] bits are changed, the change in the duty-cycle setting is immediately reflected during the first count operation.

Whether it’s a sawtooth wave or a triangle wave, When the OADTYF/OBDTYF bit is set to 1 during the count stop and

the duty of the first cycle after the count start is set to the OADTY[1:0]/OBDTY[1:0] bits, Even if the OADTYF/OBDTYF bit is set to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set again, When the count starts, the duty of the first cycle and the duty of the next cycle will be the duty setting of the OADTY[1:0]/OBDTY[1:0] bits set during the count stop.

OmDTY[1:0] Bits (GTIOCnm Pin Output Duty Setting) (n = 0 to 7; m = A, B)

These bits set the output of the duty cycle (0%, 100% or compare match control) on the GTIOCnm pin.

OmDTYF Bit (GTIOCnm Pin Output Duty Forced Setting) (n = 0 to 7; m = A, B)

This bit forcibly specifies the duty cycle at the start of the GTCNT counter in the OmDTY[1:0] bits.
Always write 0 to the bit during count operation.
When writing 1 to the bit while count operation is stopped, set the bit back to 0 by the end of the first cycle after the count operation started and specify the next cycle.

OmDTYR Bit (Output after Release of GTIOCnm Pin Output 0%/100% Duty Cycle Settings) (n = 0 to 7; m = A, B)

When the setting of a GTIOCnm pin for a 0% or 100% duty cycle is forcibly changed due to a compare match, the output level on the pin is maintained after the period has elapsed if the value of the GTIOR.GTIOm[3:2] bits is 00b, but maintenance and a value to control toggling are selected after the period has elapsed if the output has become toggled and the setting is 11b.
The GPTW internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmDTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

22.2.14 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPTW0.GTIOR 000C 2034h, GPTW1.GTIOR 000C 2134h, GPTW2.GTIOR 000C 2234h, GPTW3.GTIOR 000C 2334h, GPTW4.GTIOR 000C 2434h, GPTW5.GTIOR 000C 2534h, GPTW6.GTIOR 000C 2634h, GPTW7.GTIOR 000C 2734h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCSB[1:0]		NFBEN	—	OBEOCD	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[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
	NFCSA[1:0]		NFAEN	PSYE	OAEOCD	OADF[1:0]		OAE	OAHLD	OADFLT	CPSCIR	GTIOA[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	GTIOA[4:0]	GTIOCnA Pin Function Select	See Table 22.4 and Table 22.5.	R/W
b5	CPSCIR	Complementary PWM Mode Initial Output at Synchronous Clear Disable*1	0: Output the initial value set by the GTIOR.GTIOA[4:0] and GTIOB[4:0] bits when synchronous clear occurs in Trough section of complementary PWM mode 1: Disable output the initial value	R/W
b6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop	0: The GTIOCnA pin outputs low when counting is stopped. 1: The GTIOCnA pin outputs high when counting is stopped.	R/W
b7	OAHLD	GTIOCnA Pin Output Retention at the Start/Stop Count	0: The GTIOCnA pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnA pin output level is retained at start/stop of counting.	R/W
b8	OAE	GTIOCnA Pin Output Enable	0: Pin output is disabled 1: Pin output is enabled	R/W
b10, b9	OADF[1:0]	GTIOCnA Pin Negate Value Setting	b10 b9 0 0: None of the following sources is specified 0 1: GTIOCnA pin is placed in the Hi-Z state in response to control for output negation. 1 0: GTIOCnA pin is set to 0 in response to control for output negation. 1 1: GTIOCnA pin is set to 1 in response to control for output negation.	R/W
b11	OAEOCD	GTCCRA Register Compare Match Cycle End Output Invalidate*1	0: Validate GTIOA[3:0] setting 1: Invalidate GTIOA[3:0] setting (GTIOCnA pin output is retained)	R/W
b12	PSYE	PWM Synchronous Output Enable*2	0: Disable GTCPPO0 pin output 1: Enable GTCPPO0 pin output	R/W
b13	NFAEN	GTIOCnA Pin Input Noise Filter Enable	0: The noise filter for GTIOCnA pin input is disabled 1: The noise filter for GTIOCnA pin input is enabled	R/W
b15, b14	NFCSA[1:0]	GTIOCnA Pin Input Noise Filter Sampling Clock Select	b15b14 0 0: PCLKA 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W
b20 to b16	GTIOB[4:0]	GTIOCnB Pin Function Select	See Table 22.4.	R/W
b21	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop	0: The GTIOCnB pin outputs low when counting is stopped. 1: The GTIOCnB pin outputs high when counting is stopped.	R/W
b23	OBHLD	GTIOCnB Pin Output Retention at the Start/Stop Count	0: The GTIOCnB pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnB pin output level is retained at start/stop of counting.	R/W
b24	OBE	GTIOCnB Pin Output Enable	0: Pin output is disabled 1: Pin output is enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
b26, b25	OBDF[1:0]	GTIOCnB Pin Negate Value Setting	b26b25 0 0: None of the following sources is specified 0 1: GTIOCnB pin is placed in the Hi-Z state in response to control for output negation. 1 0: GTIOCnB pin is set to 0 in response to control for output negation. 1 1: GTIOCnB pin is set to 1 in response to control for output negation.	R/W
b27	OBEOCD	GTCCRB Register Compare Match Cycle End Output Invalidate*1	• When sawtooth-wave PWM mode 1 0: Validate GTIOB[3:2] setting 1: Invalidate GTIOB[3:2] setting (GTIOCnB pin output is retained) • When sawtooth-wave PWM mode 2 0: Validate GTIOA[3:2] setting 1: Invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFBEN	GTIOCnB Pin Input Noise Filter Enable	0: The noise filter for GTIOCnB pin input is disabled 1: The noise filter for GTIOCnB pin input is enabled	R/W
b31, b30	NFCSB[1:0]	GTIOCnB Pin Input Noise Filter Sampling Clock Select	b31b30 0 0: PCLKA 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W

n = 0 to 7

Note 1. This bit in each of GPTW3 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

Note 2. This bit in each of GPTW1 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

GTIOR sets the functions of the GTIOCnA, GTIOCnB, and GTCPPO0 pins.

GTIOA[4:0] Bits (GTIOCnA Pin Function Select) (n = 0 to 7)

These bits select the GTIOCnA pin function. For details, see Table 22.4.

CPSCIR Bit (Complementary PWM Mode Initial Output at Synchronous Clear Disable)

Select the output waveform when synchronous clear occurs in complementary PWM mode.

The initial output is disabled by this function only when synchronous clear occurs in the trough in complementary PWM mode. If a synchronous clear occurs at any other time, the initial value set by the GTIOA[4]/GTIOB[4] bits is output regardless of the CPSCIR bit setting. In addition, the initial value set by the GTIOA[4]/GTIOB[4] bits is output even when the synchronous clear occurs in the trough immediately after the count starts.

OADFLT Bit (GTIOCnA Pin Output Value Setting at the Count Stop) (n = 0 to 7)

This bit sets whether the GTIOCnA pin outputs high or low when counting is stopped.

OAHLD Bit (GTIOCnA Pin Output Retention at the Start/Stop Count) (n = 0 to 7)

This bit specifies whether the GTIOCnA pin output level is retained or the level depends on the register setting when counting is started or stopped.

[Setting conditions]

- The output is retained when counting starts or stops.

[Clearing conditions]

- The value specified by the GTIOA[4] bit is output when counting starts.
- The value specified by the OADFLT bit is output when counting stops.
- If the OADFLT bit is modified while counting is stopped, it is immediately reflected in the output.

OAE Bit (GTIOCnA Pin Output Enable) (n = 0 to 7)

This bit selects whether the GTIOCnA pin output is performed or not.
When the GTCCRA register is used as an input capture register (at least one bit among the bits for the GTICASR register is set to 1), the GTIOCnA pin output is not performed regardless of the setting of the OAE bit.

OADF[1:0] Bits (GTIOCnA Pin Negate Value Setting) (n = 0 to 7)

Select the value for output from the GTIOCnA pin in response to a request to disable output from the POEG.

OAEOCD Bit (GTCCRA Register Compare Match Cycle End Output Invalidate)

When the end of the cycle and the timing of the compare match of the GTCCRA register match in sawtooth-wave PWM mode 1 and 2, select whether to disable or enable the GTIOA[3:2] bit setting.
When this bit is 1 (disabled), the GTIOCnA pin holds the output when the cycle end and the GTCCRA compare match timing match.

PSYE Bit (PWM Synchronous Output Enable)

This bit sets enable or disable of output signal from GTCPPO0 pin synchronized with the PWM cycle that toggles at the of complementary PWM mode, crest/trough/GTCNT counter clear at Triangle-wave mode, or the end of the cycle of the Sawtooth-wave mode. The initial output of the GTCPPO0 output pin is at the low level, and the output goes to the high level when counting starts.

NFAEN Bit (GTIOCnA Pin Input Noise Filter Enable) (n = 0 to 7)

This bit sets enable or disable of the noise filter function for the GTIOCnA pin input. Switch these bits while the corresponding pin function for the timer I/O control register is set as the output compare function, because unintended internal edge may be generated when these bits are switched.

NFCSA[1:0] Bits (GTIOCnA Pin Input Noise Filter Sampling Clock Select) (n = 0 to 7)

These bits set a sampling clock for the noise filter to the GTIOCnA pin input. Set to the input capture function after waiting two cycles for the sampling clock cycle following the setting of these bits.

GTIOB[4:0] Bits (GTIOCnB Pin Function Select) (n = 0 to 7)

These bits select the GTIOCnB pin function. For details, see Table 22.4.
In sawtooth-wave PWM mode 2, only the GTIOB[1:0] bits are valid, and the GTIOCnA pin output is selected instead of the GTIOCnB pin by the GTCCRB register compare match.

OBDFLT Bit (GTIOCnB Pin Output Value Setting at the Count Stop) (n = 0 to 7)

This bit sets whether the GTIOCnB pin outputs high or low when counting is stopped.

OBHLD Bit (GTIOCnB Pin Output Retention at the Start/Stop Count) (n = 0 to 7)

This bit specifies whether the GTIOCnB pin output level is retained or the level depends on the register setting when counting is started or stopped.
[Setting conditions]

- The output is retained when counting starts or stops.

[Clearing conditions]

- The value specified by the GTIOB[4] bit is output when counting starts.
- The value specified by the OBDFLT bit is output when counting stops.
- If the OBDFLT bit is modified while counting is stopped, it is immediately reflected in the output.

OBE Bit (GTIOCnB Pin Output Enable) (n = 0 to 7)

This bit selects whether the GTIOCnB pin output is performed or not.
When the GTCCRB register is used as an input capture register (at least one bit among the bits for the GTICBSR register is set to 1), the GTIOCnB pin output is not performed regardless of the setting of the OBE bit.

OBDF[1:0] Bits (GTIOCnB Pin Negate Value Setting) (n = 0 to 7)

Select the value for output from the GTIOCnB pin in response to a request to disable output from the POEG.

OBEOCD Bit (GTCCRB Register Compare Match Cycle End Output Invalidate)

When the end of the cycle and the timing of the compare match of the GTCCRB register match in sawtooth-wave PWM mode 1 and 2, select whether to disable or enable the GTIOB[3:2] bit setting.
When this bit is 1 (disabled), the GTIOCnB pin holds the output when the cycle end and the GTCCRB compare match timing match.

NFBEN Bit (GTIOCnB Pin Input Noise Filter Enable) (n = 0 to 7)

This bit sets enable or disable of the noise filter function for the GTIOCnB pin input. Switch these bits while the corresponding pin function for the timer I/O control register is set as the output compare function, because unintended internal edge may be generated when these bits are switched.

NFCSB[1:0] Bits (GTIOCnB Pin Input Noise Filter Sampling Clock Select) (n = 0 to 7)

These bits set a sampling clock for the noise filter to the GTIOCnB pin input. Set to the input capture function after waiting two cycles for the sampling clock cycle following the setting of these bits.

Table 22.4 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Sawtooth-wave mode, Triangle-wave mode) (1/2)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	0	0	0	Initial output is Low.	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Table 22.4 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Sawtooth-wave mode, Triangle-wave mode) (2/2)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
1	0	0	0	0	Initial output is High.	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note: In sawtooth-wave mode, “end of a cycle” refers to an overflow (the value of the GTCNT counter changing from that of the GTPR register to 0000 0000h in up-counting), an underflow (the value of the GTCNT counter changing from 0000 0000h to that of the GTPR register in down-counting), or counter clearing. It refers to a trough in triangle-wave mode (the value of the GTCNT counter changing from 0000 0000h to 0000 0001h)).

Note: When the timing of an end of a cycle and the timing of a GTCCRA/GTCCRB register compare match are the same in a compare-match operation, if the OAEOCD and OBEOCD bits are set to 0 and the end of cycle output is enabled, the b3 and b2 settings are given priority in sawtooth-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note: During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), the b3 and b2 settings become invalid (GPTW3 to GPTW7).

Note: In sawtooth-wave PWM mode 2, the GTIOB[4:2] bits are invalid. Since only the GTIOCnA pin is the output pin, set the GTIOA[4] bit for the initial output. Set the GTIOA[3:2] bits for output at the end of the cycle.

Table 22.5 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Complementary PWM mode)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
					Initial Output, Active level	Up-count Compare Match input	Down-count Compare Match output
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	1	1	0	Initial output is Low. Active level is High.	Low output	High output
0	1	0	0	1		High output	Low output
1	0	1	1	0	Initial output is High. Active level is Low.	Low output	High output
1	1	0	0	1		High output	Low output

Note: In complementary PWM mode, the only values that can be set in the GTIOA[4:0] bits are 01001b, and 10110b. Setting other values is prohibited.

Note: In complementary PWM mode, the only values that can be set in the GTIOB[4:0] bits are 00110b, and 11001b. Setting other values is prohibited.

Note: In complementary PWM mode, setting the GTIOB[4:0] bits does not use compare match of the GTCCRB register. The combination of counter and register that is the target of compare match depends on the operation period of complementary PWM mode. For details, see (7), Complementary PWM mode 1, 2, 3.

22.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPTW0.GTINTAD 000C 2038h, GPTW1.GTINTAD 000C 2138h, GPTW2.GTINTAD 000C 2238h, GPTW3.GTINTAD 000C 2338h, GPTW4.GTINTAD 000C 2438h, GPTW5.GTINTAD 000C 2538h, GPTW6.GTINTAD 000C 2638h, GPTW7.GTINTAD 000C 2738h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	GRPABL	GRPABH	—	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCFPU	SCFPO	SCFF	SCFE	SCFD	SCFC	SCFB	SCFA	GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GTINTA	GTCCRA Register Compare Match/ Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	GTINTB	GTCCRB Register Compare Match/ Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b2	GTINTC	GTCCRC Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b3	GTINTD	GTCCRD Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b4	GTINTE	GTCCRE Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b5	GTINTF	GTCCRF Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b7, b6	GTINTPR[1:0]	GTPR Register Compare Match Interrupt Enable	b7 b6 0 0: Interrupt request is disabled. 0 1: In sawtooth-wave mode, interrupt requests are enabled at overflows. In triangle-wave mode, interrupt requests are enabled at crests. 1 0: In sawtooth-wave mode, interrupt requests are enabled at underflows. In triangle-wave mode, interrupt requests are enabled at troughs. 1 1: In sawtooth-wave mode, interrupt requests are enabled at both overflows and underflows. In triangle-wave mode, interrupt requests are enabled at both crests and troughs.	R/W
b8	SCFA	GTCCRA Register Compare Match/ Input Capture Source Synchronous Clear Enable	0: Disables the GTCCRA register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRA register comparison match/ input capture as a source for other channel clears.	R/W
b9	SCFB	GTCCRB Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRB register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRB register comparison match/ input capture as a source for other channel clears.	R/W
b10	SCFC	GTCCRC Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRC register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRC register comparison match/ input capture as a source for other channel clears.	R/W
b11	SCFD	GTCCRD Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRD register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRD register comparison match/ input capture as a source for other channel clears.	R/W
b12	SCFE	GTCCRE Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRE register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRE register comparison match/ input capture as a source for other channel clears.	R/W

Bit	Symbol	Bit Name	Description	R/W
b13	SCFF	GTCCRF Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRF register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRF register comparison match/ input capture as a source for other channel clears.	R/W
b14	SCFPO	Overflow Source Synchronous Clear Enable	0: Disables Overflow as a source for other channel clears. 1: Enables Overflow as a source for other channel clears.	R/W
b15	SCFPU	Underflow Source Synchronous Clear Enable	0: Disables Underflow as a source for other channel clears. 1: Enables Underflow as a source for other channel clears.	R/W
b16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable*1	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable*1	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable*1	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable*1	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Stop Group Select	b25b24 0 0: Group A is selected 0 1: Group B is selected 1 0: Group C is selected 1 1: Group D is selected	R/W
b28 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	GRPABH	Simultaneous High Output Stop Detection Enable	0: Simultaneous high output stop detection is disabled 1: Simultaneous high output stop detection is enabled	R/W
b30	GRPABL	Simultaneous Low Output Stop Detection Enable	0: Simultaneous low output stop detection is disabled 1: Simultaneous low output stop detection is enabled	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. This bit in each of GPTW3 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTINTAD register sets enabling or disabling for interrupt request, A/D conversion start request, and output stop detection.

GTINTA Bit (GTCCRA Register Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRA register compare match/input capture (GTCIA).

GTINTB Bit (GTCCRB Register Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRB register compare match/input capture (GTCIB).

GTINTC Bit (GTCCRC Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRC register compare match (GTCIC).

GTINTD Bit (GTCCRD Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRD register compare match (GTCID).

GTINTE Bit (GTCCRE Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRE register compare match (GTCIE).

GTINTF Bit (GTCCRF Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRF register compare match (GTCIF).

GTINTPR[1:0] Bits (GTPR Register Compare Match Interrupt Enable)

These bits enable or disable interrupt requests by a GTPR register compare match (GTCNT counter overflow) and those by a GTCNT counter underflow (GTCIV/GTCIU).

SCFA Bit (GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables the GTCCRA register compare match/input capture as a clearing factor for other channels.

SCFB Bit (GTCCRB Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRB register compare match/input capture as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFC Bit (GTCCRC Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRC register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFD Bit (GTCCRD Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRD register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFE Bit (GTCCRE Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRE register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFF Bit (GTCCRF Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRF register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFPO Bit (Overflow Source Synchronous Clear Enable)

This bit enables or prohibits overflow as a clearing factor for other channels.
In complementary PWM mode, this setting is valid only for the master channel.

SCFPU Bit (Underflow Source Synchronous Clear Enable)

This bit enables or prohibits underflow as a clearing factor for other channels.
In complementary PWM mode, this setting is valid only for the master channel.

ADTRAUEN Bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRA register compare matches during GTCNT counter up-counting.

ADTRADEN Bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRA register compare matches during GTCNT counter down-counting.

ADTRBUEN Bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRB register compare matches during GTCNT counter up-counting.

ADTRBDEN Bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRB register compare matches during GTCNT counter down-counting.

GRP[1:0] Bits (Output Stop Group Select)

Select the group to detect disabling of output from the GPTW to the POEG and to request of disabling of output from the POEG to the GPTW.

Each signal indicating detection of disabling in response to simultaneous driving of outputs to the high or low level is output to the group selected in the GRP[1:0] bits while the respective output disable detection enable bits are 1.

Requests to disabling output from the POEG for the group selected in the GRP[1:0] bits can be monitored with the GTST.ODF flag.

Set the GRP[1:0] bits while both of the OAE and OBE bits in the GTIOR register are set to 0.

GRPABH Bit (Simultaneous High Output Stop Detection Enable)

This bit enables or disables the output stop detection when the GTIOCnA and GTIOCnB pins have become high simultaneously.

GRPABL Bit (Simultaneous Low Output Stop Detection Enable)

This bit enables or disables the output stop detection when the GTIOCnA and GTIOCnB pins have become low simultaneously.

22.2.16 General PWM Timer Status Register (GTST)

Address(es): GPTW0.GTST 000C 203Ch, GPTW1.GTST 000C 213Ch, GPTW2.GTST 000C 223Ch, GPTW3.GTST 000C 233Ch, GPTW4.GTST 000C 243Ch, GPTW5.GTST 000C 253Ch, GPTW6.GTST 000C 263Ch, GPTW7.GTST 000C 273Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	OABLF	OABHF	—	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
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
b0	TCFA	Input Capture/Compare Match Flag A	0: No input capture/compare match of GTCCRA is generated. 1: An input capture/compare match of GTCCRA is generated.	R/(W) *1
b1	TCFB	Input Capture/Compare Match Flag B	0: No input capture/compare match of GTCCRB is generated. 1: An input capture/compare match of GTCCRB is generated.	R/(W) *1
b2	TCFC	Compare Match Flag C	0: No compare match of GTCCRC is generated. 1: An compare match of GTCCRC is generated.	R/(W) *1
b3	TCFD	Compare Match Flag D	0: No compare match of GTCCRD is generated. 1: An compare match of GTCCRD is generated.	R/(W) *1
b4	TCFE	Compare Match Flag E	0: No compare match of GTCCRE is generated. 1: An compare match of GTCCRE is generated.	R/(W) *1
5	TCFF	Compare Match Flag F	0: No compare match of GTCCRF is generated. 1: An compare match of GTCCRF is generated.	R/(W) *1
b6	TCFPO	Overflow Flag	0: No overflow (crest) occurred. 1: An overflow (crest) occurred.	R/(W) *1
b7	TCFPU	Underflow Flag	0: No underflow (trough) occurred. 1: An underflow (trough) occurred.	R/(W) *1
b10 to b8	ITCNT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Counter*2	Counter for counting the number of times a timer interrupt has been skipped.	R
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TUCF	Count Direction Flag	0: The GTCNT counter counts downward. 1: The GTCNT counter counts upward.	R
b16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag*2	0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/(W) *1
b17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag*2	0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/(W) *1
b18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag*2	0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/(W) *1
b19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag*2	0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/(W) *1
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ODF	Output Stop Request Flag	0: No output stop request has occurred. 1: An output stop request has occurred.	R
b28 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b29	OABHF	Simultaneous High Output Flag	0: No simultaneous generation of 1 both for the GTIOCnA and GTIOCnB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCnA and GTIOCnB pins has occurred.	R
b30	OABLF	Simultaneous Low Output Flag	0: No simultaneous generation of 0 both for the GTIOCnA and GTIOCnB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCnA and GTIOCnB pins has occurred.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

n = 0 to 7

Note 1. Only 0 can be written to clear the flag. When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

Note 2. This bit in each of GPTW3 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTST register indicates the status of the GPTW.

TCFA Flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of the GTCCRA register.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- The GTCNT counter value is transferred to the GTCCRA register by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB Flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of the GTCCRB register.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- The GTCNT counter value is transferred to the GTCCRB register by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC Flag (Compare Match Flag C)

The TCFC flag indicates the status for the compare match of the GTCCRC register.

When the GTCCRC register performs buffer operation, the GTCCRC register doesn’t perform compare match.

[Setting conditions]

- GTCNT = GTCCRC

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (the GTCCRC register performs buffer operation).

TCFD Flag (Compare Match Flag D)

The TCFD flag indicates the status for the compare match of the GTCCRD register.

When the GTCCRD register performs buffer operation, the GTCCRD register doesn’t perform compare match.

[Setting conditions]

- GTCNT = GTCCRD

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (the GTCCRD register performs buffer operation).

TCFE Flag (Compare Match Flag E)

The TCFE flag indicates the status for the compare match of the GTCCRE register.
When the GTCCRE register performs buffer operation, the GTCCRE register doesn't perform compare match.

[Setting conditions]

- GTCNT = GTCCRE

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (the GTCCRE register performs buffer operation).

TCFF Flag (Compare Match Flag F)

The TCFF flag indicates the status for the compare match of the GTCCRF register.
When the GTCCRF register performs buffer operation, the GTCCRF register doesn't perform compare match.

[Setting conditions]

- GTCNT = GTCCRF

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode).
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (the GTCCRF register performs buffer operation).

TCFPO Flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.

[Setting conditions]

- In sawtooth-wave mode, an overflow (the GTCNT counter changes from the GTPR register value to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (the GTCNT counter changes from the GTPR register value to GTPR – 1) has occurred
- In counting by hardware sources, an overflow (the GTCNT counter changes from the GTPR register value to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU Flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.
[Setting conditions]

- In sawtooth-wave mode, an underflow (the GTCNT counter changes from 0 to the GTPR register value in down-counting) has occurred
- In triangle-wave mode, a trough (the GTCNT counter changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (the GTCNT counter changes from 0 to the GTPR register value in down-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

ITCNT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Counter)

When the GTCIV/GTCIU interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GTCIV/GTCIU interrupt source is generated.
[Clearing conditions]

- The GTCIV/GTCIU interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the IVTC[1:0] bits are 00b).
- The GTCIV/GTCIU interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the IVTT[2:0] bits).
- When the count operation is stopped.

TUCF Flag (Count Direction Flag)

This flag indicates the count direction of the GTCNT counter.
This flag indicates 1 when in up-counting and indicates 0 when in down-counting during the event count operation.

ADTRAUF Flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting

[Clearing conditions]

- 0 is written to the ADTRAUF flag

ADTRADF Flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in down-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting

[Clearing conditions]

- 0 is written to the ADTRADF flag

ADTRBUF Flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in up-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting

[Clearing conditions]

- 0 is written to the ADTRBUF flag

ADTRBDF Flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in down-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting

[Clearing conditions]

- 0 is written to the ADTRBDF flag

ODF Flag (Output Stop Request Flag)

This flag monitors the output stop request for the group selected by the GTINTAD.GRP[1:0] bits.
Even if a request to disable output is de-asserted after the request is made, releasing the control of negation on the PWM pins has to wait until the end of the PWM cycle.

OABHF Flag (Simultaneous High Output Flag)

This flag indicates that the simultaneous output of 1 both from the GTIOCnA and GTIOCnB pins.
When one of the GTIOCnA and GTIOCnB pins changes to 0, the flag changes to 0.
This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When the output stop detection by the OABHF flag is enabled (the GTINTAD.GRPABH bit is 1), the OABHF flag is output as the output stop detection to the POEG. The GPTW does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.
[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 simultaneously while both the OAE and OBE bits in the GTIOR register are set to 1.

[Clearing conditions]

- The GTIOCnA and GTIOCnB pins output different values while both the OAE and OBE bits are set to 1.
- The GTIOCnA and GTIOCnB pins output 0 simultaneously while both the OAE and OBE bits are set to 1.
- At least one of the OAE and OBE bits is set to 0.

OABLF Flag (Simultaneous Low Output Flag)

This flag indicates that the simultaneous output of 0 both from the GTIOCnA and GTIOCnB pins.
When one of the GTIOCnA and GTIOCnB pins changes to 1, the flag changes to 0.
This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When the output stop detection by the OABLF flag is enabled (the GTINTAD.GRPABL bit is 1), the OABLF flag is output as the output stop detection to the POEG. The GPTW does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.
[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 simultaneously while both the OAE and OBE bits in the GTIOR register are set to 1.

[Clearing conditions]

- The GTIOCnA and GTIOCnB pins output different values while both the OAE and OBE bits are set to 1.
- The GTIOCnA and GTIOCnB pins output 1 simultaneously while both the OAE and OBE bits are set to 1.
- At least one of the OAE and OBE bits is set to 0.

Comparison for the output value to generate the OABHF or OABLF flag uses the value before the compare match output (PWM output) is masked by the output negate function. Even during the output negate condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

22.2.17 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPTW0.GTBER 000C 2040h, GPTW1.GTBER 000C 2140h, GPTW2.GTBER 000C 2240h, GPTW3.GTBER 000C 2340h, GPTW4.GTBER 000C 2440h, GPTW5.GTBER 000C 2540h, GPTW6.GTBER 000C 2640h, GPTW7.GTBER 000C 2740h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BD[2]	BD[1]	BD[0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BD[0]	GTCCRA/GTCCRB Registers Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b1	BD[1]	GTPR Register Buffer Operation Disable		R/W *1
b2	BD[2]	GTADTRA/GTADTRB Registers Buffer Operation Disable*2		R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	CCRA[1:0]	GTCCRA Register Buffer Operation	b17b16 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA register ⇔ GTCCRC register) 1 x: Double buffer operation (GTCCRA register ⇔ GTCCRC register ⇔ GTCCRD register)	R/W
b19, b18	CCRB[1:0]	GTCCRB Register Buffer Operation	b19b18 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB register ⇔ GTCCRE register) 1 x: Double buffer operation (GTCCRB register ⇔ GTCCRE register ⇔ GTCCRF register)	R/W
b21, b20	PR[1:0]	GTPR Register Buffer Operation*3	b21b20 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR register ⇒ GTPR register) 1 x: Double buffer operation (GTPDBR register ⇒ GTPBR register ⇒ GTPR register)	R/W
b22	CCRSWT	GTCCRA and GTCCRB Registers Forcible Buffer Operation	Writing 1 to this bit forcibly performs buffer transfer of GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25, b24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select*2	• Triangle waves or in complementary PWM mode b25b24 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b25b24 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W

Bit	Symbol	Bit Name	Description	R/W
b26	ADTDA	GTADTRA Register Double Buffer Operation*2	0: Single buffer operation (GTADTBRA register ⇒ GTADTRA register) 1: Double buffer operation (GTADTDBRA register ⇒ GTADTBRA register ⇒ GTADTRA register)	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b29, b28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select*2	• Triangle waves or in complementary PWM mode b29b28 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b29b28 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
b30	ADTDB	GTADTRB Register Double Buffer Operation*2	0: Single buffer operation (GTADTBRB register ⇒ GTADTRB register) 1: Double buffer operation (GTADTDBRB register ⇒ GTADTBRB register ⇒ GTADTRB register)	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit in each of GPTW3 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.
Note 3. The PR[1] bit is reserved in GPTW3 to GPTW7, that is, only the PR[0] bit is present in those channels. The PR[1] bit is read as 0. When writing, write 0 to the bit.

The GTBER register makes settings for buffer operation.
Set the GTBER register except the BD[3:0] bits while the GTCNT counter is stopped.

BD[0] Bit (GTCCRA/GTCCRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTCCRA, GTCCRC, and GTCCRD registers together and buffer operation using the GTCCRB, GTCCRE, and GTCCRF registers together.
Even though the BD[0] bit is set to 0, buffer operation in the GTCCRB register is not performed if the GTDTCR.TDE bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode. Instead, the compare match value for negative-phase waveform with dead time is set automatically.
In complementary PWM mode, it is valid only for the buffer operation of GTCCRC register and GTCCRE register. The buffer operation of GTCCRA registers cannot be disabled. No buffer transfer to GTCCRB is performed in complementary PWM mode.
A value for the BD[0] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDCE or SBDCD bit in the GTSECR register.

BD[1] Bit (GTPR Register Buffer Operation Disable)

This bit disables buffer operation using the GTPR, GTPBR, and GTPDBR registers together.
A value for the BD[1] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDPE or SBDPD bit in the GTSECR register.
In complementary PWM mode, the slave channel is also controlled by setting the BD[1] bit of the master channel.

BD[2] Bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

A value for the BD[2] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

CCRA[1:0] Bits (GTCCRA Register Buffer Operation)

These bits set buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR register setting is given priority.*1

CCRB[1:0] Bits (GTCCRB Register Buffer Operation)

These bits set buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR register setting is given priority.*1

PR[1:0] Bits (GTPR Register Buffer Operation)

These bits set buffer operation with the GTPR, GTPBR, and GTPDBR registers combined.
The setting is invalid in complementary PWM mode. Buffer operation unique to comple-mentary PWM mode is performed regardless of the setting value of the PR[1:0] bits.

CCRSWT Bit (GTCCRA and GTCCRB Registers Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forcibly performs buffer transfer of the GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.
This bit is valid only when counting is stopped with compare match operation specified.
The setting is invalid in complementary PWM mode.

ADTTA[1:0] Bits (GTADTRA Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.

ADTDA Bit (GTADTRA Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

ADTTB[1:0] Bits (GTADTRB Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.

ADTDB Bit (GTADTRB Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

Note 1. The buffer operation mode is fixed in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

22.2.18 General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register (GTITC)

Address(es): GPTW0.GTITC 000C 2044h, GPTW1.GTITC 000C 2144h, GPTW2.GTITC 000C 2244h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITLA	GTCCRA Register Compare Match/Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b1	ITLB	GTCCRB Register Compare Match/Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b2	ITLC	GTCCRC Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b3	ITLD	GTCCRD Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b4	ITLE	GTCCRE Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b5	ITLF	GTCCRF Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b7, b6	IVTC[1:0]	GTCIV/GTCIU Interrupt Skipping Function Select	b7 b6 0 0: Skipping is not performed 0 1: Both overflow and underflow for sawtooth waves and crest for triangle waves are counted and skipped. 1 0: Both overflow and underflow for sawtooth waves and trough for triangle waves are counted and skipped. 1 1: Both overflow and underflow for sawtooth waves and both crest and trough for triangle waves are counted and skipped	R/W
b10 to b8	IVTT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Select	b10 b8 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADTAL	GTADTRA Register A/D Conversion Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	ADTBL	GTADTRB Register A/D Conversion Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTITC sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GTCIV)/ underflow interrupt (GTCIU) and also sets whether to link the other interrupts and A/D conversion start requests with the GTCIV/GTCIU interrupt skipping function. Note that detection of disabling of output to the POEG cannot be linked with the GTCIV or GTCIU interrupt skipping function. Additionally, if the interrupt skipping function is performed, the change in the status flag is also skipped.

ITLA Bit (GTCCRA Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRA compare match/input capture interrupt (GTCIA) with the GTCIV/ GTCIU interrupt skipping function.

ITLB Bit (GTCCRB Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRB compare match/input capture interrupt (GTCIB) with the GTCIV/GTCIU interrupt skipping function.

ITLC Bit (GTCCRC Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRC compare match interrupt (GTCIC) with the GTCIV/GTCIU interrupt skipping function.

ITLD Bit (GTCCRD Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRD compare match interrupt (GTCID) with the GTCIV/GTCIU interrupt skipping function.

ITLE Bit (GTCCRE Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRE compare match interrupt (GTCIE) with the GTCIV/GTCIU interrupt skipping function.

ITLF Bit (GTCCRF Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRF compare match interrupt (GTCIF) with the GTCIV/GTCIU interrupt skipping function.

IVTC[1:0] Bits (GTCIV/GTCIU Interrupt Skipping Function Select)

These bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV)/ GTCNT counter underflow interrupt (GTCIU).

IVTT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Select)

These bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV)/GTCNT counter underflow interrupt (GTCIU).
When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL Bit (GTADTRA Register A/D Conversion Start Request Link)

This bit specifies whether to link the A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRA register, with the GTCIV/GTCIU interrupt skipping function.

ADTBL Bit (GTADTRB Register A/D Conversion Start Request Link)

This bit specifies whether to link the A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRB register, with the GTCIV/GTCIU interrupt skipping function.

22.2.19 General PWM Timer Counter (GTCNT)

Address(es): GPTW0.GTCNT 000C 2048h, GPTW1.GTCNT 000C 2148h, GPTW2.GTCNT 000C 2248h, GPTW3.GTCNT 000C 2348h, GPTW4.GTCNT 000C 2448h, GPTW5.GTCNT 000C 2548h, GPTW6.GTCNT 000C 2648h, GPTW7.GTCNT 000C 2748h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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 GTCNT counter is a 32-bit readable/writable counter. There is one GTCNT counter for each channel. Writing is available only when counting is stopped, and thus writing is disabled while counting is in progress (when CST = 1). The upper 16 bits of GPTW0 to GPTW5 are reserved, and these bits are read as 0.

Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units.

When the sawtooth or triangle waves are to be used in GPTW0 to GPTW2, set the GTCNT counter within the range from 0 up to the value of the GTPR register.

22.2.20 General PWM Timer Compare Capture Register m (GTCCRm) (m = A to F)

Address(es): GPTW0.GTCCRA 000C 204Ch, GPTW1.GTCCRA 000C 214Ch, GPTW2.GTCCRA 000C 224Ch, GPTW3.GTCCRA 000C 234Ch, GPTW4.GTCCRA 000C 244Ch, GPTW5.GTCCRA 000C 254Ch, GPTW6.GTCCRA 000C 264Ch, GPTW7.GTCCRA 000C 274Ch, GPTW0.GTCCRB 000C 2050h, GPTW1.GTCCRB 000C 2150h, GPTW2.GTCCRB 000C 2250h, GPTW3.GTCCRB 000C 2350h, GPTW4.GTCCRB 000C 2450h, GPTW5.GTCCRB 000C 2550h, GPTW6.GTCCRB 000C 2650h, GPTW7.GTCCRB 000C 2750h, GPTW0.GTCCRC 000C 2054h, GPTW1.GTCCRC 000C 2154h, GPTW2.GTCCRC 000C 2254h, GPTW3.GTCCRC 000C 2354h, GPTW4.GTCCRC 000C 2454h, GPTW5.GTCCRC 000C 2554h, GPTW6.GTCCRC 000C 2654h, GPTW7.GTCCRC 000C 2754h, GPTW0.GTCCRE 000C 2058h, GPTW1.GTCCRE 000C 2158h, GPTW2.GTCCRE 000C 2258h, GPTW3.GTCCRE 000C 2358h, GPTW4.GTCCRE 000C 2458h, GPTW5.GTCCRE 000C 2558h, GPTW6.GTCCRE 000C 2658h, GPTW7.GTCCRE 000C 2758h, GPTW0.GTCCRD 000C 205Ch, GPTW1.GTCCRD 000C 215Ch, GPTW2.GTCCRD 000C 225Ch, GPTW3.GTCCRD 000C 235Ch, GPTW4.GTCCRD 000C 245Ch, GPTW5.GTCCRD 000C 255Ch, GPTW6.GTCCRD 000C 265Ch, GPTW7.GTCCRD 000C 275Ch, GPTW0.GTCCRF 000C 2060h, GPTW1.GTCCRF 000C 2160h, GPTW2.GTCCRF 000C 2260h, GPTW3.GTCCRF 000C 2360h, GPTW4.GTCCRF 000C 2460h, GPTW5.GTCCRF 000C 2560h, GPTW6.GTCCRF 000C 2660h, GPTW7.GTCCRF 000C 2760h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*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 1. The value after reset is 0 for GPTW0 to GPTW5 and 1 for GPTW6, GPTW7.

The GTCCRm register is a 32-bit readable/writable register. The upper 16 bits of GPTW0 to GPTW5 are reserved, and these bits are read as 0.

Access in 8-bit or 16-bit units to the GTCCRm register is prohibited, and it should be accessed in 32-bit units.

The GTCCRA and GTCCRB registers are registers used for both output compare and input capture.

The GTCCRC and GTCCRE registers are compare match registers, and can also function as buffer registers for the GTCCRA and GTCCRB registers.

The GTCCRD and GTCCRF registers are compare match registers, and can also function as buffer registers for the GTCCRC and GTCCRE registers (double buffer registers for the GTCCRA and GTCCRB registers).

22.2.21 General PWM Timer Period Setting Register (GTPR)

Address(es): GPTW0.GTPR 000C 2064h, GPTW1.GTPR 000C 2164h, GPTW2.GTPR 000C 2264h, GPTW3.GTPR 000C 2364h, GPTW4.GTPR 000C 2464h, GPTW5.GTPR 000C 2564h, GPTW6.GTPR 000C 2664h, GPTW7.GTPR 000C 2764h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*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 1. The value after reset is 0 for GPTW0 to GPTW5 and 1 for GPTW6, GPTW7.

The GTPR register is a 32-bit readable/writable register that sets the maximum count value of the GTCNT counter. The upper 16 bits of GPTW0 to GPTW5 are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. In sawtooth-wave mode except the sawtooth-wave PWM mode 2, the value of (GTPR register + 1) is the count period. In triangle-wave mode or complementary PWM mode, the value of (GTPR register value × 2) is the count period.
When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

22.2.22 General PWM Timer Period Setting Buffer Register (GTPBR)

Address(es): GPTW0.GTPBR 000C 2068h, GPTW1.GTPBR 000C 2168h, GPTW2.GTPBR 000C 2268h, GPTW3.GTPBR 000C 2368h, GPTW4.GTPBR 000C 2468h, GPTW5.GTPBR 000C 2568h, GPTW6.GTPBR 000C 2668h, GPTW7.GTPBR 000C 2768h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*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 1. The value after reset is 0 for GPTW0 to GPTW5 and 1 for GPTW6, GPTW7.

The GTPBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPR register. The upper 16 bits of GPTW0 to GPTW5 are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

22.2.23 General PWM Timer Period Setting Double-Buffer Register (GTPDBR)

Address(es): GPTW0.GTPDBR 000C 206Ch, GPTW1.GTPDBR 000C 216Ch, GPTW2.GTPDBR 000C 226Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPDBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPBR register (a double buffer register for the GTPR register). The upper 16 bits are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

22.2.24 A/D Conversion Start Request Timing Register m (GTADTRm) (m = A, B)

Address(es): GPTW0.GTADTRA 000C 2070h, GPTW1.GTADTRA 000C 2170h, GPTW2.GTADTRA 000C 2270h,
GPTW0.GTADTRB 000C 207Ch, GPTW1.GTADTRB 000C 217Ch, GPTW2.GTADTRB 000C 227Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTRm register is 32-bit readable/writable register that set the timing of A/D conversion start request generation. The upper 16 bits are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTADTRm register is prohibited, and it should be accessed in 32-bit units.
When the GTADTRm register value matches the GTCNT counter value, an A/D conversion start request is generated.
In complementary PWM mode, A/D conversion start request is generated when the GTCNT counter of the master channel matches this register.

22.2.25 A/D Conversion Start Request Timing Buffer Register m (GTADTBRm) (m = A, B)

Address(es): GPTW0.GTADTBRA 000C 2074h, GPTW1.GTADTBRA 000C 2174h, GPTW2.GTADTBRA 000C 2274h, GPTW0.GTADTBRB 000C 2080h, GPTW1.GTADTBRB 000C 2180h, GPTW2.GTADTBRB 000C 2280h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTBRm register is 32-bit readable/writable register that function as buffer register for the GTADTRm register. The upper 16 bits are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTADTBRm register is prohibited, and it should be accessed in 32-bit units.

22.2.26 A/D Conversion Start Request Timing Double-Buffer Register m (GTADTDBRm) (m = A, B)

Address(es): GPTW0.GTADTDBRA 000C 2078h, GPTW1.GTADTDBRA 000C 2178h, GPTW2.GTADTDBRA 000C 2278h, GPTW0.GTADTDBRB 000C 2084h, GPTW1.GTADTDBRB 000C 2184h, GPTW2.GTADTDBRB 000C 2284h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTDBRm register is 32-bit readable/writable register that function as buffer register for the GTADTBRm register (double buffer register for the GTADTRm register). The upper 16 bits are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTADTDBRm register is prohibited, and it should be accessed in 32-bit units.

22.2.27 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPTW0.GTDTCR 000C 2088h, GPTW1.GTDTCR 000C 2188h, GPTW2.GTDTCR 000C 2288h

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Negative-Phase Waveform Setting	0: GTCCRB register is set without using GTDVU register. 1: GTDVU register is used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB register.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time. This register is invalid in sawtooth-wave PWM mode 2 or complementary PWM mode.
The dead time buffer operation is not available. Only GTDVU register is used for setting dead time value.

TDE Bit (Negative-Phase Waveform Setting)

This bit sets whether to use the GTDVU register. When the GTDVU register is used, the compare match value for a negative-phase waveform with dead time that was obtained by the compare match value of a positive-phase waveform (GTCCRA register) and the dead time value (GTDVU register) is automatically set in the GTCCRB register.
The TDE bit setting is ignored in sawtooth-wave PWM mode, and the GTCCRB register is not automatic setting.
The automatically set GTCCRB register value has the following upper and lower limit values.

- Triangle wave PWM mode
  Upper limit value: the value set in the GTPR register – 1
  Lower limit value: 0000 0001h in up-counting, 0000 0000h in down-counting
- Sawtooth-wave one-shot pulse mode
  Upper limit value: the value set in the GTPR register
  Lower limit value: 0000 0000h

If the obtained GTCCRB register value is not within the range between the upper and lower limits, the upper or lower limit is set in the GTCCRB register.

22.2.28 General PWM Timer Dead Time Value Register U (GTDVU)

Address(es): GPTW0.GTDVU 000C 208Ch, GPTW1.GTDVU 000C 218Ch, GPTW2.GTDVU 000C 228Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTDVU register is 32-bit readable/writable register that set the dead time for generating PWM waveforms with dead time. The upper 16 bits are reserved, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTDVU register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in sawtooth-wave PWM mode 2.
In complementary PWM mode, the GTDVU register is used as the dead time value during both up-counting and down-counting. No matter which GTDVU register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
The GTDVU register is used for up-counting in triangle-wave mode.
Do not set this register to a value equal to or larger than that of the GTPR register.
In complementary PWM mode, set the GTDVU register so that all of the following conditions are satisfied.

- GTDVU > 0
- GTDVU < GTPR/2
- GTDVU + GTPR ≤ FFFF FFFFh

In addition, when using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When using the GTDVU register, writing to the GTCCRB register is prohibited. In modes other than complementary PWM, setting this register to 0 leads to the output of waveforms without dead time.

22.2.29 General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register (GTSECSR)

Address(es): GPTW0.GTSECSR 000C 20D0h, GPTW1.GTSECSR 000C 21D0h, GPTW2.GTSECSR 000C 22D0h, GPTW3.GTSECSR 000C 23D0h, GPTW4.GTSECSR 000C 24D0h, GPTW5.GTSECSR 000C 25D0h, GPTW6.GTSECSR 000C 26D0h, GPTW7.GTSECSR 000C 27D0h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SECSEL7	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select	0: Disable simultaneous control. 1: Enable simultaneous control.	R/W
b1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select		R/W
b2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select		R/W
b3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select		R/W
b4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select		R/W
b5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select		R/W
b6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select		R/W
b7	SECSEL7	Channel 7 Operation Enable Bit Simultaneous Control Channel Select		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel for updating an operation enable bit by the GTSECR register.
A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.
Access in 8-bit or 16-bit units to the GTSECSR register is prohibited, and it should be accessed in 32-bit units.

SECSELn Bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 7)

This bit enables or disables the simultaneous control of operation enable in channel n.
When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

22.2.30 General PWM Timer Operation Enable Bit Simultaneous Control Register (GTSECR)

Address(es): GPTW0.GTSECR 000C 20D4h, GPTW1.GTSECR 000C 21D4h, GPTW2.GTSECR 000C 22D4h, GPTW3.GTSECR 000C 23D4h, GPTW4.GTSECR 000C 24D4h, GPTW5.GTSECR 000C 25D4h, GPTW6.GTSECR 000C 26D4h, GPTW7.GTSECR 000C 27D4h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SBDAD	SBDPD	SBDCD	—	—	—	—	—	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
b1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W *1
b2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable *2	0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
b9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W *1
b10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable *2	0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
b16 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17	SSCE	Synchronous Set/Clear Simultaneous Enable	0: Disable simultaneous enabling synchronous set/clear 1: Enable synchronous set/clear simultaneously	R/W *1
b24 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	SSCD	Synchronous Set/Clear Simultaneous Disable	0: Disable simultaneous disabling synchronous set/clear 1: Disable synchronous set/clear simultaneously	R/W *1
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit in each of GPTW3 to GPTW7 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.
Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECR registers. Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.
A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.
Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

SBDCE Bit (GTCCR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB,

GTCCRE, and GTCCRF registers are enabled.
Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE Bit (GTPR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPBR, and GTPDBR registers are enabled.
Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE Bit (GTADTR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.
Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDCD Bit (GTCCR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.
Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD Bit (GTPR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPBR, and GTPDBR registers are disabled.
Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD Bit (GTADTR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.
Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SSCE Bit (Synchronous Set/Clear Simultaneous Enable)

Writing 1 to this bit simultaneously sets any GTCR.SSCEN bits to 1 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and enables the synchronous set/clear function.
Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

SSCD Bit (Synchronous Set/Clear Simultaneous Disable)

Writing 1 to this bit simultaneously sets any GTCR.SSCEN bits to 0 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and disables the synchronous set/clear function.
Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

22.2.31 General PWM Timer Inter Channel Cooperation Input Capture Control Register (GTICCR)

Address(es): GPTW0.GTICCR 000C 20ECh, GPTW1.GTICCR 000C 21ECh, GPTW2.GTICCR 000C 22ECh, GPTW3.GTICCR 000C 23ECh, GPTW4.GTICCR 000C 24ECh, GPTW5.GTICCR 000C 25ECh, GPTW6.GTICCR 000C 26ECh, GPTW7.GTICCR 000C 27ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICBGRP[1:0]		—	—	—	—	—	ICBCLK	ICBFPU	ICBFPO	ICBFF	ICBFE	ICBFD	ICBFC	ICBFB	ICBFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ICAGRP[1:0]		—	—	—	—	—	ICACLK	ICAFPU	ICAFPO	ICAFF	ICAFE	ICAFD	ICAFC	ICAFB	ICAFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICAFA	Enable GTCCRA Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRA register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels 1: Enable GTCCRA register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels	R/W
b1	ICAFB	Enable GTCCRB Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRB register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels 1: Enable GTCCRB register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels	R/W
b2	ICAFC	Enable GTCCRC Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRC register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRC register comparison match, which is the GTCCRA input capture source of other channels	R/W
b3	ICAFD	Enable GTCCRD Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRD register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRD register comparison match, which is the GTCCRA input capture source of other channels	R/W
b4	ICAFE	Enable GTCCRE Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRE register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRE register comparison match, which is the GTCCRA input capture source of other channels	R/W
b5	ICAFF	Enable GTCCRF Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRF register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRF register comparison match, which is the GTCCRA input capture source of other channels	R/W
b6	ICAFPO	Enable Overflow as Source of Other Channel GTCCRA Input Capture	0: Disable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels 1: Enable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels	R/W
b7	ICAFPU	Enable Underflow as Source of Other Channel GTCCRA Input Capture	0: Disable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels 1: Enable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels	R/W
b8	ICACLK	Enable Count Clock as Source of Other Channel GTCCRA Input Capture	0: Disable count clock, which are the GTCCRA input capture source of other channels 1: Enable count clock, which are the GTCCRA input capture source of other channels	R/W

Bit	Symbol	Bit Name	Description	R/W
b13 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ICAGRP[1:0]	GTCCRA Input Capture Group Select	b15b14 0 0: Group A 0 1: Group B 1 0: Group D 1 1: Group D	R/W
b16	ICBFA	Enable GTCCRA Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRA register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels 1: Enable GTCCRA register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels	R/W
b17	ICBFB	Enable GTCCRB Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRB register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels 1: Enable GTCCRB register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels	R/W
b18	ICBFC	Enable GTCCRC Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRC register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRC register comparison match, which is the GTCCRB input capture source of other channels	R/W
b19	ICBFD	Enable GTCCRD Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRD register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRD register comparison match, which is the GTCCRB input capture source of other channels	R/W
b20	ICBFE	Enable GTCCRE Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRE register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRE register comparison match, which is the GTCCRB input capture source of other channels	R/W
b21	ICBFF	Enable GTCCRF Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRF register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRF register comparison match, which is the GTCCRB input capture source of other channels	R/W
b22	ICBFPO	Enable Overflow as Source of Other Channel GTCCRB Input Capture	0: Disable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels 1: Enable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels	R/W
b23	ICBFPU	Enable Underflow as Source of Other Channel GTCCRB Input Capture	0: Disable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels 1: Enable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels	R/W
b24	ICBCLK	Enable Count Clock as Source of Other Channel GTCCRB Input Capture	0: Disable count clock, which are the GTCCRB input capture source of other channels 1: Enable count clock, which are the GTCCRB input capture source of other channels	R/W
b29 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31, b30	ICBGRP[1:0]	GTCCRB Input Capture Group Select	b31b30 0 0: Group A 0 1: Group B 1 0: Group D 1 1: Group D	R/W

The GTICCR register is a register that controls input capture by inter channel link.

For channels that perform input capture in cooperation, the input capture factor set in the GTICASR and GTICASR registers is invalid.

ICAFm Bit (Enable GTCCRm Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRA register.

ICAFn Bit (Enable GTCCRn Register Compare Match as Source of Other Channel GTCCRA Input Capture) (n = C, D, E, F)

Enables/disables the use of compare match of GTCCRn register as the input capture source of other channel's GTCCRA register.

ICAFPO Bit (Enable Overflow as Source of Other Channel GTCCRA Input Capture)

Enable/disable to use the overflow of sawtooth-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICAFPU Bit (Enable Underflow as Source of Other Channel GTCCRA Input Capture)

Enable/disable to use the underflow of sawtooth-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICACLK Bit (Enable Count Clock as Source of Other Channel GTCCRA Input Capture)

Enable/disable to use count clock as the input capture source of other channel's GTCCRA register.

ICAGRP[1:0] Bits (GTCCRA Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRA register.
For channels that accept input capture of the GTCCRA register due to input capture sources from other channels, set the GTICASR.ASOC bit to 1 and select the group of inter channel cooperation with the ICAGRP[1:0] bits.

ICBFm Bit (Enable GTCCRm Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRB register.

ICBFn Bit (Enable GTCCRn Register Compare Match as Source of Other Channel GTCCRB Input Capture) (n = C, D, E, F)

Enables/disables the use of compare match of GTCCRn register as the input capture source of other channel's GTCCRB register.

ICBFPO Bit (Enable Overflow as Source of Other Channel GTCCRB Input Capture)

Enable/disable to use the overflow of sawtooth-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBFPU Bit (Enable Underflow as Source of Other Channel GTCCRB Input Capture)

Enable/disable to use the underflow of sawtooth-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBCLK Bit (Enable Count Clock as Source of Other Channel GTCCRB Input Capture)

Enable/disable to use count clock as the input capture source of other channel's GTCCRB register.

ICBGRP[1:0] Bits (GTCCRB Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRB register.

For channels that accept input capture of the GTCCRB register due to input capture sources from other channels, set the GTICBSR.BSOC bit to 1 and select the group of inter channel cooperation with the ICBGRP[1:0] bits.

22.2.32 Output Phase Switching Control Register (OPSCR)

Address(es): GPTW.OPSCR 000C 2B00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Input Phase Software Setting	These bits set the input phase from software settings. Setting these bits is valid when FB = 1.	R/W
b1	VF			R/W
b2	WF			R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	U	Input U-Phase Monitor	This bit monitors the state of the input phase. • FB = 0 External input that are synchronized by PCLKA • FB = 1 The software settings (UF, VF, WF bit)	R
b5	V	Input V-Phase Monitor		R
b6	W	Input W-Phase Monitor		R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	EN	Output Phase Enable	0: Do not output (Hi-Z external pin) 1: Output*1	R/W
b15-b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from software settings and external input. 0: The external input 1: The software setting (UF, VF, WF bit)	R/W
b17	P	Positive-Phase Output (P) Control	0: Level signal output 1: PWM signal output (PWM of GPTW0)	R/W
b18	N	Negative-Phase Output (N) Control	0: Level signal output 1: PWM signal output (PWM of GPTW0)	R/W
b19	INV	Output Phase Invert Control	0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
b20	RV	Output Phase Rotation Direction Reversal Control	0: Positive rotation 1: Reverse rotation	R/W
b21	ALIGN	Input Phase Alignment	0: Input phase aligned to PCLKA 1: Input phase aligned to the falling edge of PWM	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disabled Source Selection	b25b24 0 0: Group A output disable source 0 1: Group B output disable source 1 0: Group C output disable source 1 1: Group D output disable source	R/W
b26	GODF	Group Output Disable Function	0: This bit function is ignored 1: When the signal value of the source selected by the GRP[1:0] bits becomes High, the EN bit is cleared	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFEN	External Input Noise Filter Enable	0: The noise filter for the external input is disabled. 1: The noise filter for the external input is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31, b30	NFCS[1:0]	External Input Noise Filter Clock Select	b31b30 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W

Note 1. When the GODF bit is 1 and the signal value of the source selected by the GRP[1:0] bits becomes High, the EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF, VF, WF Bits (Input Phase Software Setting)

The UF, VF, and WF bits set the input phase from the software settings. When the FB bit is 1, these bits are valid. The set value of the UF, VF, and WF bit takes the place of the U, V, W external input.

U, V, W Bits (Input m-Phase Monitor) (m = U, V, W)

When the FB bit is 0, external inputs that are synchronized by PCLKA are monitored by these bits. When the FB bit is 1, the U, V, and W bits can read the UF, VF, and WF bits.

EN Bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).
When the EN bit is 1, the signal waveform is output.
When the EN bit is 0, first set FB, VF, and WF (software setting is selected), P, N, INV, RV, ALIGN, GRP[1:0], GODF, NFEN, and NFCS[1:0] bits. Then, set this bit to 1. The EN bit should be set when output disable request doesn't occur from POEG. Also when GODF is 1 and the signal value selected in the GRP[1:0] bits are high, the EN bit is set to 0. And 1 by software to be written, the EN bit remains at 0.
For the return, after clearing the output disable request by software, set the EN bit to 1.

- EN bit priority order (conflict)

When 1 write by software and set to 0 by the output disable request has been conflicting for EN bit, set to 0 by the Output Disable Request is activated.

FB Bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (UF, VF, and WF bits) and external input such as a Hall element.

P Bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N Bit (Negative-Phase Output (N) Control)

The P bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV Bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV Bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN Bit (Input Phase Alignment)

The ALIGN bit selects the PCLKA or PWM for the sampling of the input phase (input phase is specified in the FB bit).

When ALIGN bit is 0, input phase is aligned to PCLKA.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] Bits (Output Disabled Source Selection)

The GRP[1:0] bits select the output disable source.

When the GODF bit is 0, set the GRP[1:0] bits.

GODF Bit (Group Output Disable Function)

When the GODF bit is 1 and the signal value selected by the GRP[1:0] bits are high, the EN bit is set to 0.

When the GODF bit is 0, this bit is ignored.

Set the GODF bit while there is not output disable request from POEG.

NFEN Bit (External Input Noise Filter Enable)

The NFEN bit disables or enables the external input noise filter.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by this bit switching.

NFCS[1:0] Bits (External Input Noise Filter Clock Select)

The NFCS[1:0] bits select the clock for the external input noise filter.

1. Set the NFCS[1:0] bits.
2. Wait for 2 cycles.
3. Set the EN bit to 1.

22.3 Operation

22.3.1 Basic Operation

The timer in each channel performs cycle count operation by count clock or hardware source. The GTCNT counter performs up-counting, or down-counting. The timer period is controlled by the GTPR or GTCCRm (m = A to F) registers. When the GTCNT counter value matches the values for GTCCRA or GTCCRB registers, the GTIOCnA or GTIOCnB pin outputs (n = 0 to 7) can be changed respectively. Also, the GTCCRA or GTCCRB registers can be used as the input capture registers by hardware source. The GTCCRC and GTCCRD registers can be performed as buffer registers for GTCCRA register, and the GTCCRE and GTCCRF registers can be performed as buffer registers for GTCCRB register.

22.3.1.1 Counter Operation

(1) Count Start/Count Stop

The counter in each channel starts counting operation when the corresponding GTCR.CST bit is set to 1, and stops counting when the bit is set to 0.
CST bit value can be changed by the following sources:

- Writing to the GTCR register
- Writing 1 to the bit corresponding to the channel number for the GTSTR register while the GTSSR.CSTRT bit is 1
- Writing 1 to the bit corresponding to the channel number for the GTSTP register while the GTPSR.CSTOP bit is 1
- Hardware source specified by the GTSSR register
- Hardware source specified by the GTPSR register

(2) Cycle Count Operation (In Up-Counting by the Count Clock)

The counter in each channel starts up-counting when the corresponding CST bit is set to 1 while the GTUPSR register and the GTDNSR register are 0000 0000h. When the value of the GTCNT counter changes from that of the GTPR register to 0000 0000h (an overflow) or the GTCCRm (m = A to F) value selected by the GTCSR.CSCMSC[2:0] bits to 0000 0000h in sawtooth-wave PWM mode 2 while the GTINTAD.GTINTPR[0] bit is 1, a GTCIV interrupt request is also issued. After GTCNT counter overflows, up-counting is resumed from 0000 0000h.
Figure 22.4 shows an example of cycle count operation in up-counting by count clock.

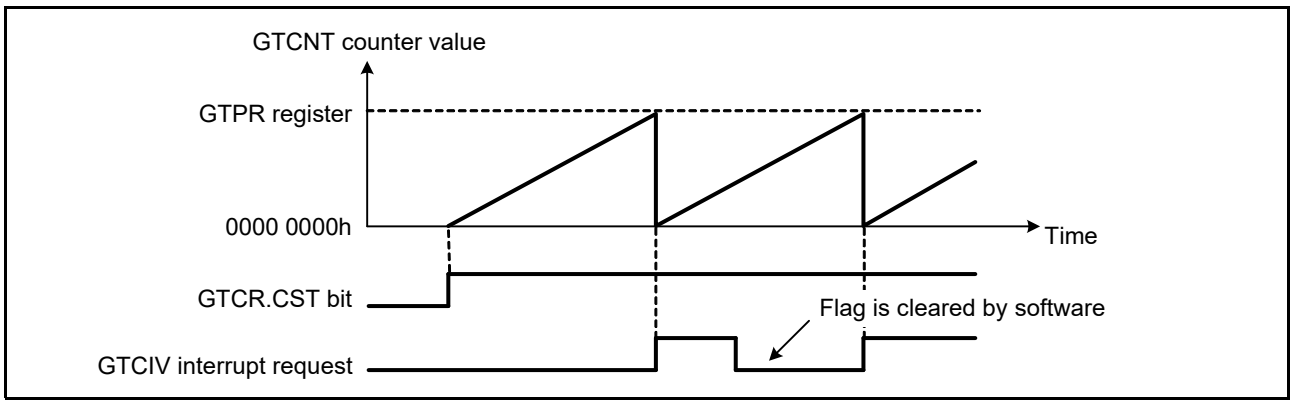


Figure 22.4 Example of Cycle Count Operation (In Up-Counting by the Count Clock)

Figure 22.5 shows an example for setting cycle count operation in up-counting.

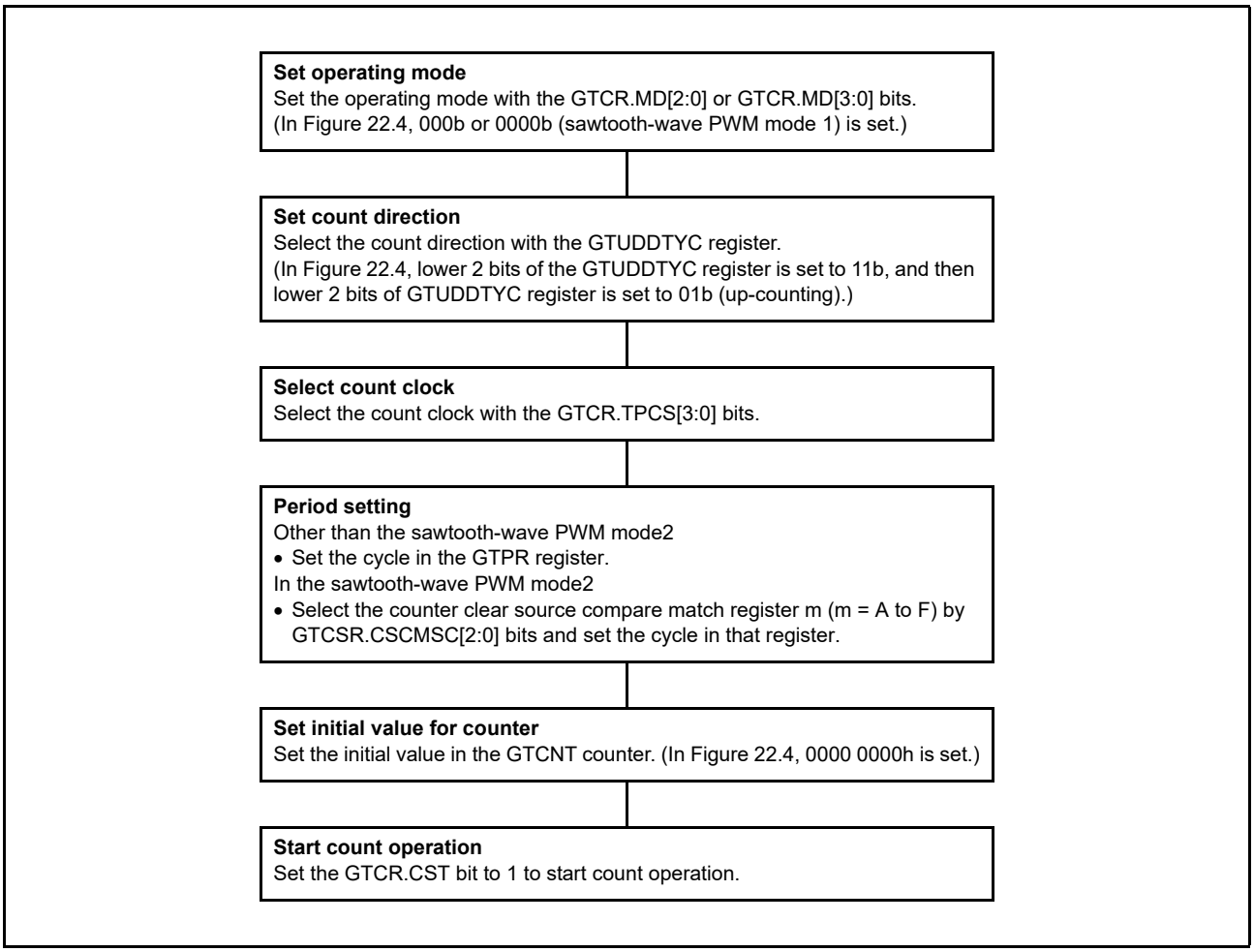


Figure 22.5 Example for Setting Cycle Count Operation (In Up-Counting by the Count Clock)

(3) Cycle Count Operation (In Down-Counting by the Count Clock)

The counter in each channel can start down-counting when the corresponding GTUDDTYC.UD bit is set in a state of the GTUPSR register and the GTDNSR register as 0000 0000h. When the value of the GTCNT counter changes from 0000 0000h to that of the GTPR register (an underflow) while the GTINTAD.GTINTPR[1] bit is 1, a GTCIU interrupt request is also issued. After the GTCNT counter underflows, down-counting is resumed from the GTPR register value.
Figure 22.6 shows an example of cycle count operation in down-counting by count clock.

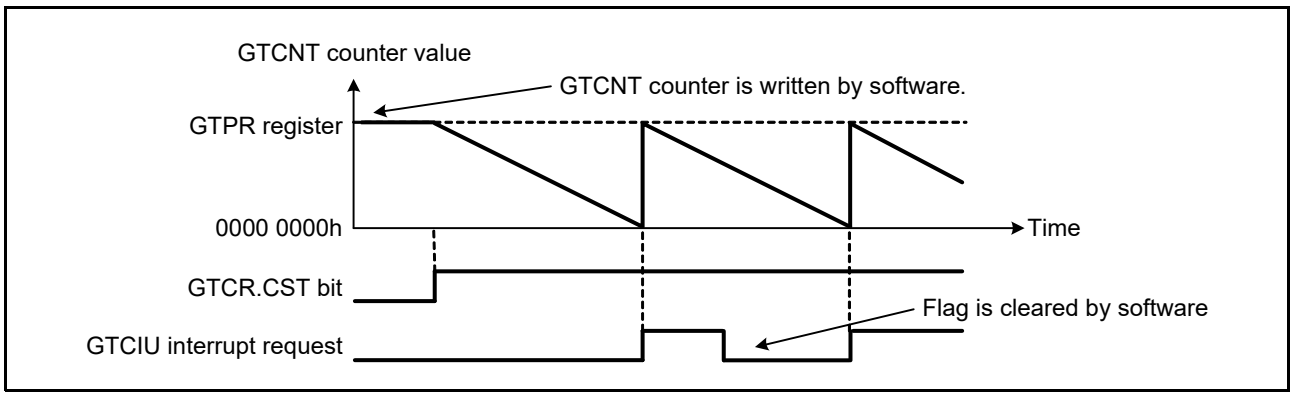


Figure 22.6 Example of Cycle Count Operation (In Down-Counting by the Count Clock)

Figure 22.7 shows an example for setting cycle count operation in down-counting by count clock.

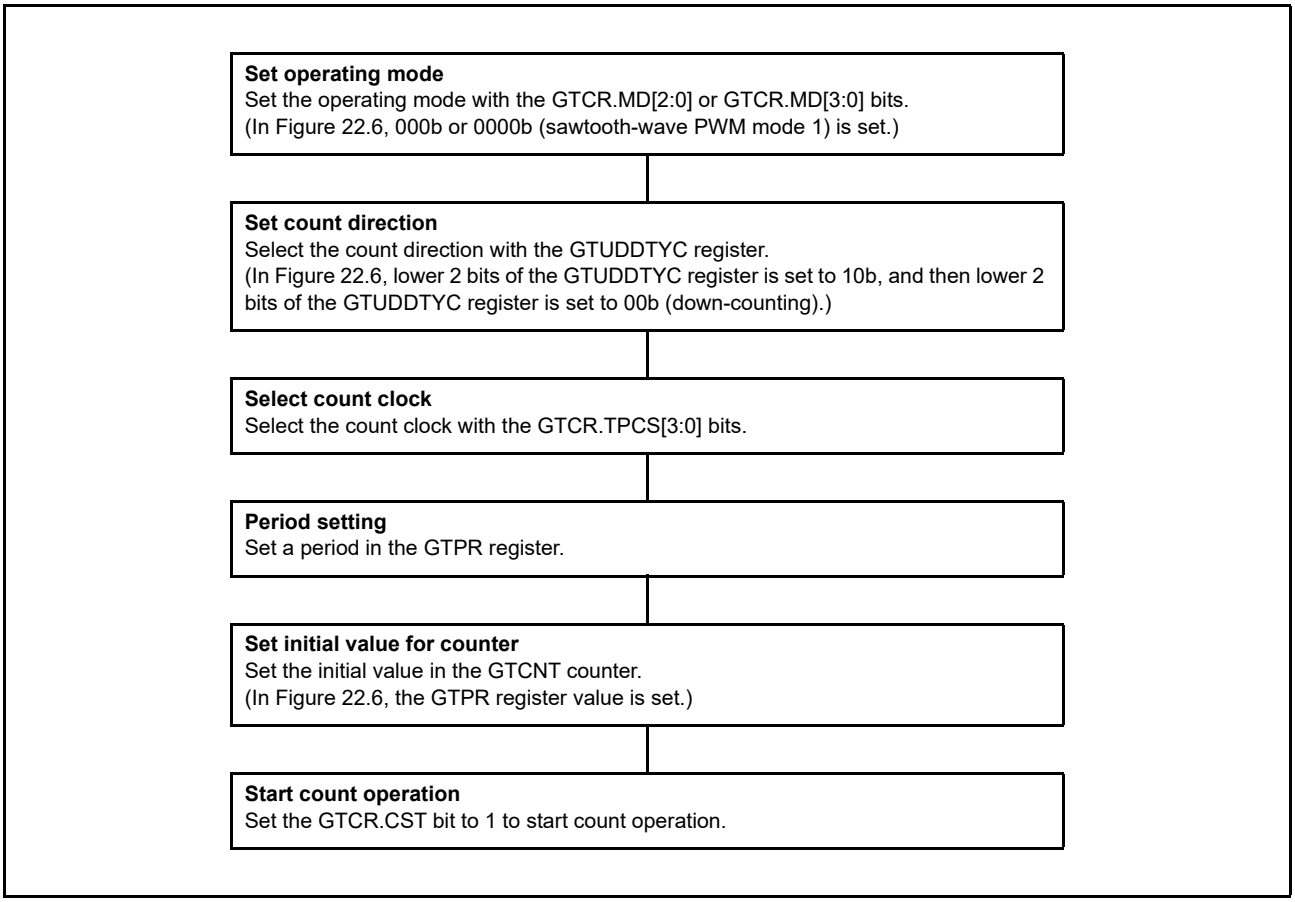


Figure 22.7 Example for Setting Cycle Count Operation (In Down-Counting by the Count Clock)

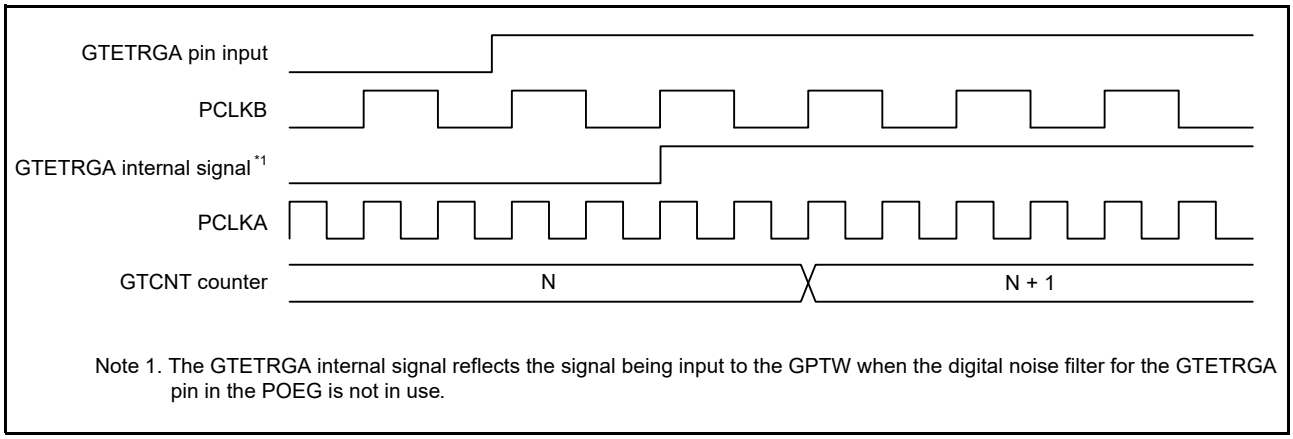
(4) Event Count Operation (In Up-Counting by Hardware Source)

The counter in each channel can start up-counting by hardware source by setting the GTUPSR register. When the GTUPSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value will not change.

Operation due to overflow in up-counting by hardware source is the same as the cycle count operation by the count clock.

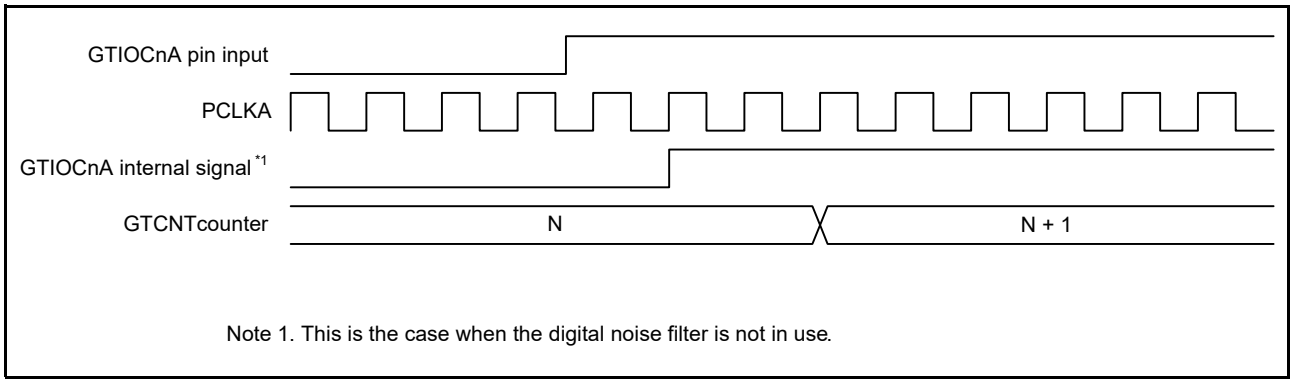
To perform up-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. As the starting the count operation is synchronized with the count clock selected by TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start up-counting operation. To start up-counting after one PCLKA from setting the CST bit to 1, set the TPCS[3:0] bits to 0000b.

Figure 22.8 and Figure 22.9 show examples of up-counting operation by hardware source (edge of GTETRGA pin input and the rising edge of GTIOCnA pin input).



Note 1. The GTETRGA internal signal reflects the signal being input to the GPTW when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 22.8 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTETRGA Pin)



Note 1. This is the case when the digital noise filter is not in use.

Figure 22.9 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTIOCnA Pin)

Figure 22.10 shows an example of event counting by a hardware source (ELC event input).
This is an example when the ELC selectively outputs the event signal output to the ELC by the compare match to the GPTW0.GTCCRA register as the event factor A to the GPTW, and performs the event count operation of the GPTW1.GTCNT counter by the signal.
The GPTW0 compare match A signal synchronized with PCLKA is synchronized with PCLKB in ELC, and GPTW event factor A is output after 1 clock with PCLKB.

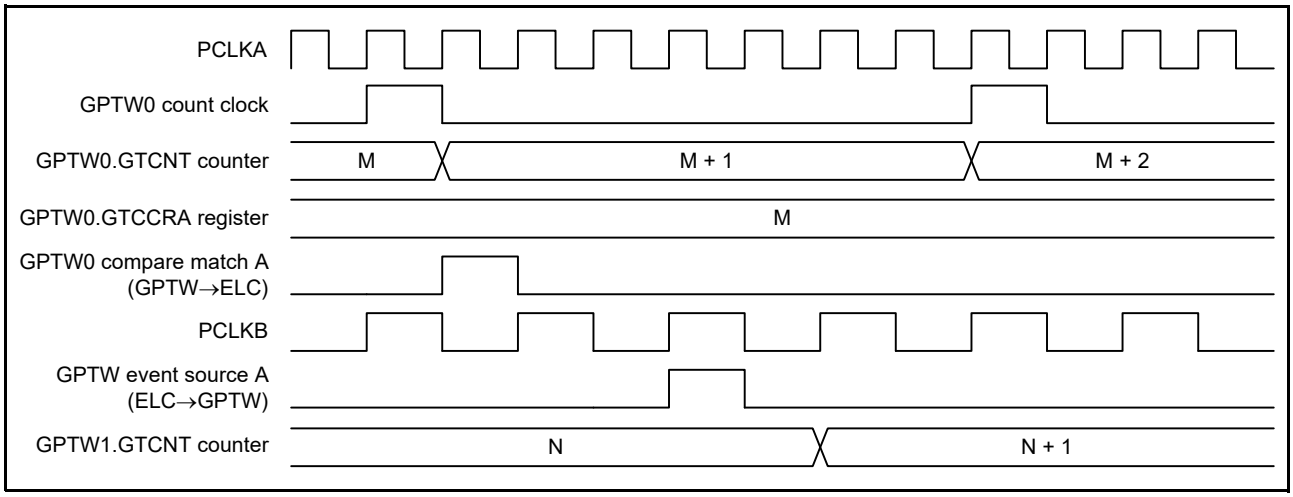


Figure 22.10 Example of Event Count Operation (Up-Counting the Number of Event Signals Input from the ELCA)

Figure 22.11 shows an example for setting cycle count operation in up-counting by hardware source.

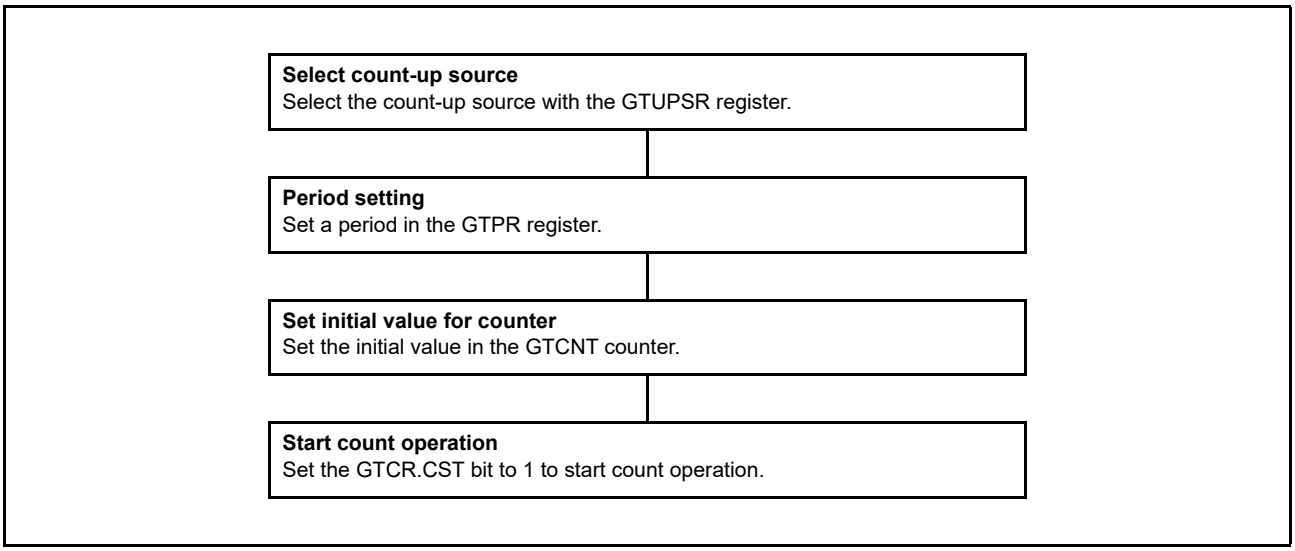


Figure 22.11 Example for Setting Event Count Operation (In Up-Counting by Hardware Source)

(5) Event Count Operation (In Down-Counting by Hardware Source)

The counter in each channel can start down-counting by hardware source by setting the GTDNSR register. When the GTDNSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value will not change.

Operation due to underflow in down-counting by hardware source is the same as the cycle count operation by the count clock.

To perform down-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. As the starting the count operation is synchronized with the count clock selected by TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start down-counting operation. To start down-counting after one PCLKA from setting the CST bit to 1, set the TPCS[3:0] bits to 0000b.

Figure 22.12 shows an example of down-counting operation by hardware source.

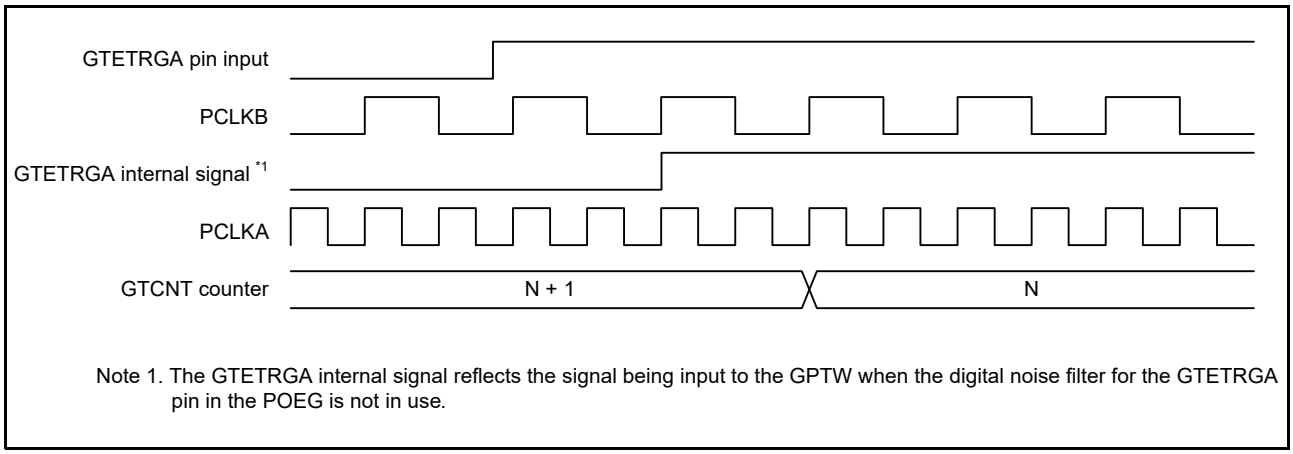


Figure 22.12 Example of Event Count Operation (Down-Counting of Rising Edges of the Input on the GTETRGA Pin)

Figure 22.13 shows an example for setting cycle count operation in down-counting by hardware source.

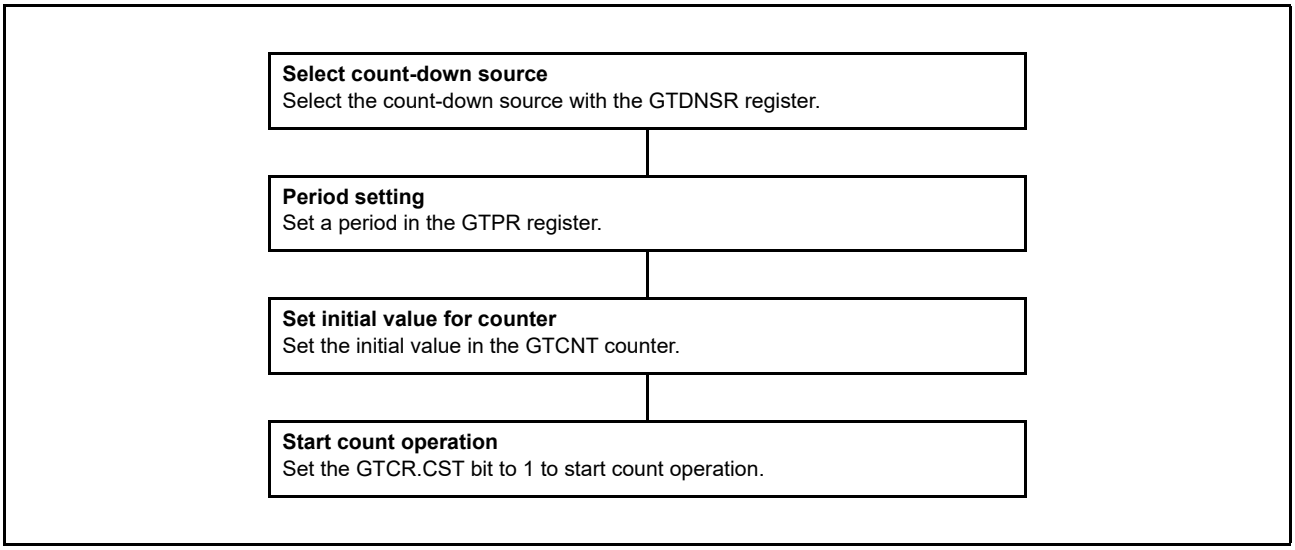


Figure 22.13 Example for Setting Event Count Operation (In Down-Counting by Hardware Source)

(6) Counter Clearing Operation

The counter in each channel can be cleared by the following sources:

- Writing to the GTCNT counter
- Writing 1 to the bit corresponding to the channel number for the GTCLR register while the GTCSR.CCLR bit is 1
- Hardware source specified by the GTCSR register

Write access during counting (when CST = 1) is disabled.
GTCLR register can be written to and cleared by hardware source whether the count operation is performed (GTCR.CST bit = 1) or is stopped (CST bit = 0). When the count direction flag is set as decrement (GTST.TUCF flag = 0) in sawtooth-wave mode (except sawtooth-wave PWM mode 2) selected with GTCR.MD[2:0] or GTCR.MD[3:0] bits, the GTCNT counter value for writing to the GTCLR register and for clearing by hardware source becomes the GTPR register value whether the counter is in operation or is stopped. The GTCNT counter value becomes 0000 0000h for other settings.
When the event count operation is set (at least one of bits for the GTUPSR or the GTDNSR register is set as 1), writing to the GTCLR register and clearing by hardware source are performed right after the clear source is generated (the operation is based on PCLKA). For other settings, clearing operation is synchronized with the count clock selected with the GTCR.TPCS[3:0] bits.

22.3.1.2 Waveform Output by Compare Match

Compare match refers to when the GTCNT counter value matches the GTCCRA or GTCCRB register value. The output of the corresponding GTIOCnA or GTIOCnB pin (n = 0 to 7) can be set to be driven low, high, or toggled in synchronization with the counter clock (including in the case of event counting) after the match.
In addition, the GTIOCnA or GTIOCnB pin output can be driven low, high, or toggled at the “end of the cycle” which is determined by the GTPR register or GTCCRm selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits in sawtooth-wave PWM mode 2 (n = 0 to 7, m = A to F). The end of the cycle is as follows:

- For sawtooth waves (except sawtooth-wave PWM mode 2) in up-counting: When the GTCNT counter value changes from the GTPR register value to 0000 0000h (overflow)
- For sawtooth waves (except sawtooth-wave PWM mode 2) in down-counting: When the GTCNT counter value changes from 0000 0000h to the GTPR register value (underflow)
- For sawtooth-wave PWM mode 2 and GTCCRm register (m = A to F) selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits. When the GTCNT counter value changes from the GTCCRm register value to 0000 0000h
- For sawtooth waves with the GTCNT counter clear
- For triangle waves or complementary PWM mode: When the GTCNT counter value changes from 0000 0000h to 0000 0001h (trough)

(1) Low Output and High Output

Figure 22.14 shows an example of low output and high output operation by a compare match of the GTCNT counter and GTCCRA register and of the GTCNT counter and GTCCRB register.
In this example, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that high is output from the GTIOC0A pin by a GPTW0.GTCCRA register compare match, and low is output from the GTIOC0B pin by a GPTW0.GTCCRB register compare match. The pin level does not change when the specified level and pin level match.

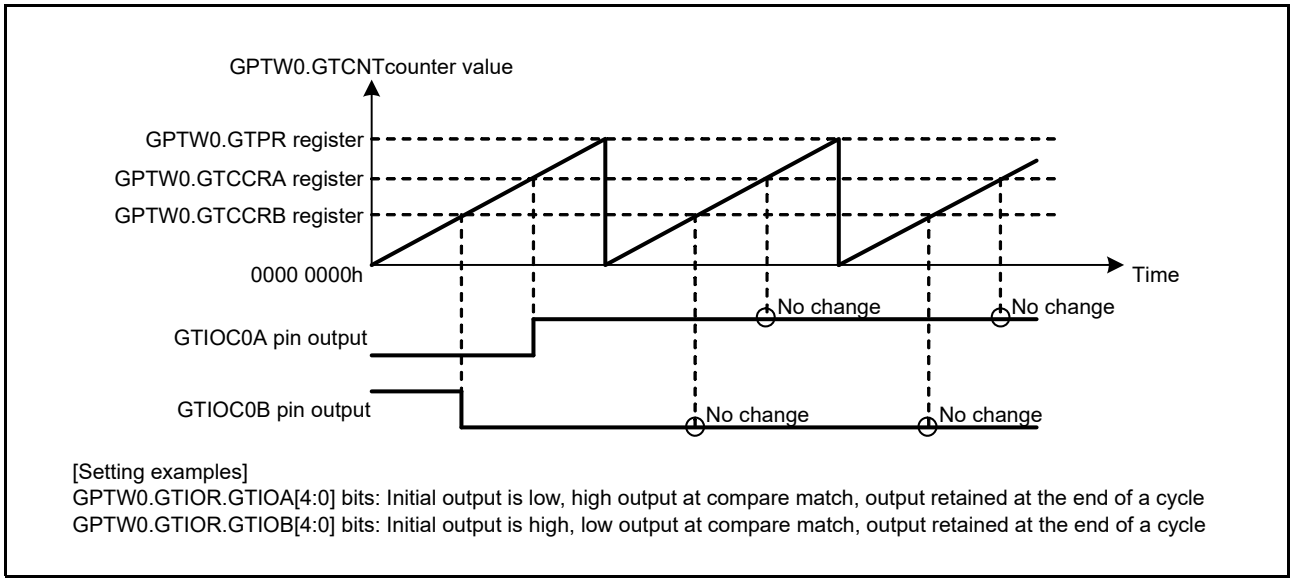


Figure 22.14 Example of Low Output and High Output Operation

Figure 22.15 shows an example for setting Low/High output operation.

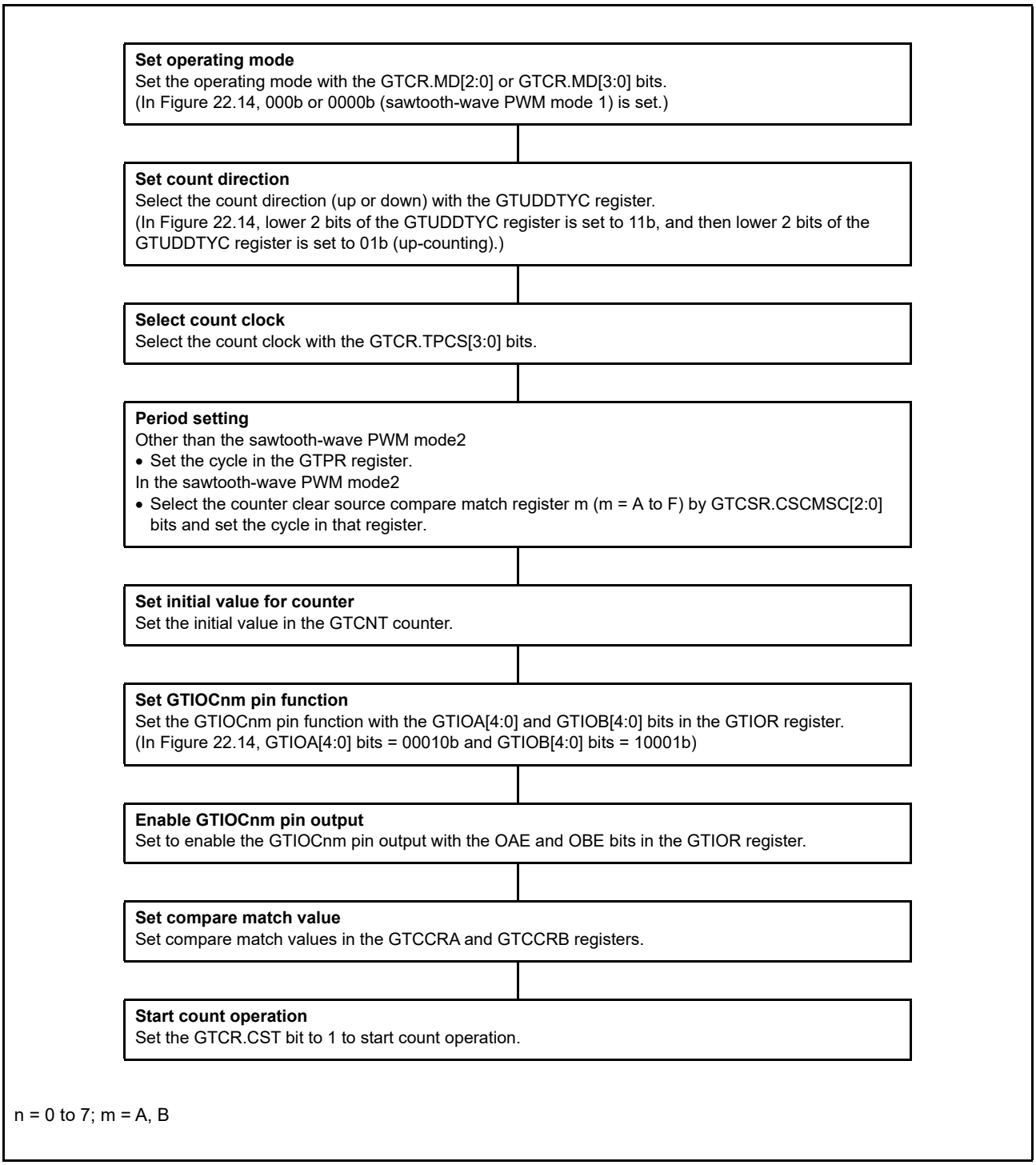


Figure 22.15 Example for Setting Low/High Output Operation

(2) Toggled Output

Figure 22.16 and Figure 22.17 show examples of toggle output operations by compare match between the GPTW0.GTCNT counter and the GTCCRA and GTCCRB registers.

In Figure 22.16, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output by a GPTW0.GTCCRA register compare match and GTIOC0B pin output by a GPTW0.GTCCRB register compare match are toggled.

In Figure 22.17, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output is toggled by a compare match of GPTW0.GTCCRA register and the GTIOC0B pin output is toggled at the end of the cycle.

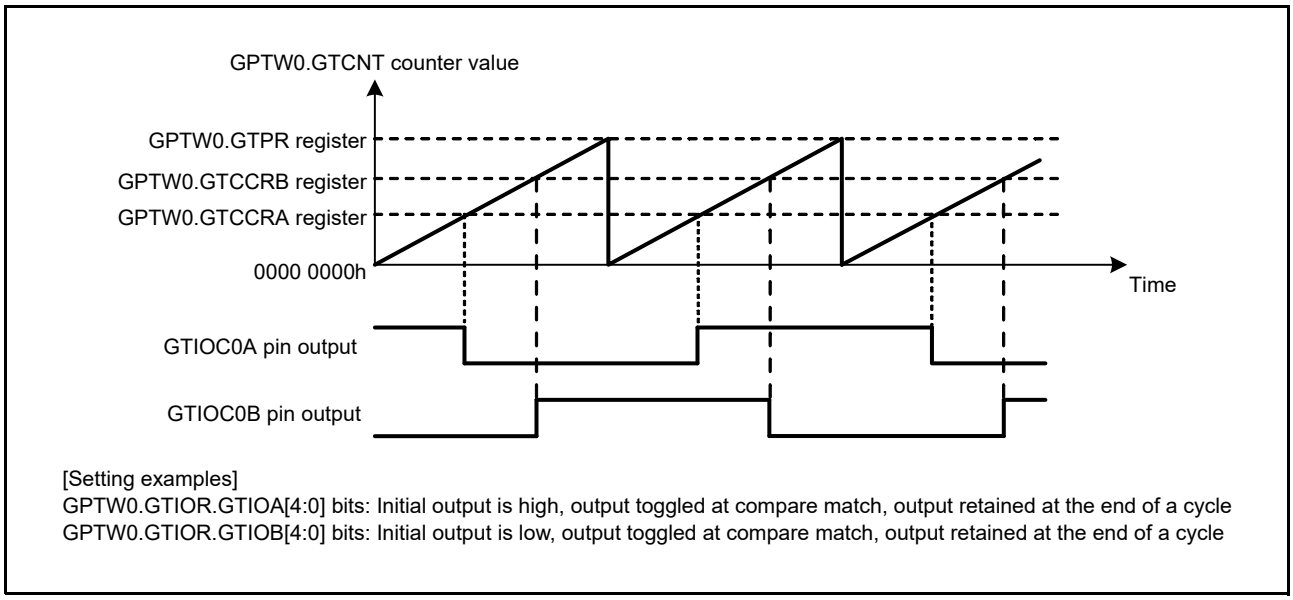


Figure 22.16 Example of Toggled Output Operation (1)

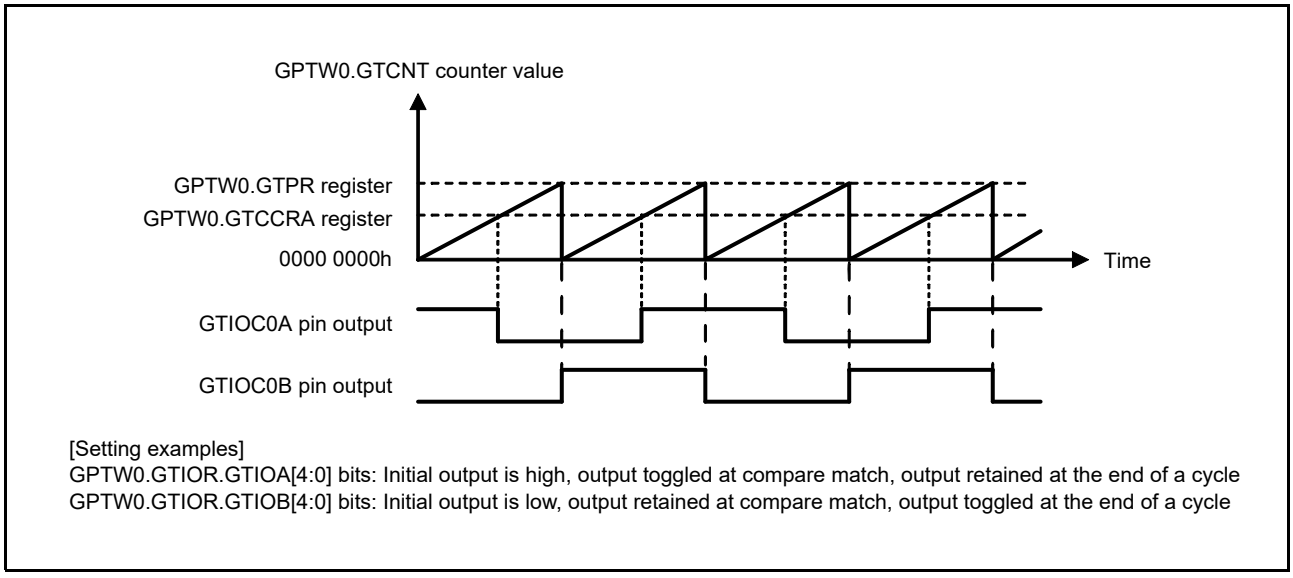


Figure 22.17 Example of Toggled Output Operation (2)

Figure 22.18 shows an example for setting toggled output operation.

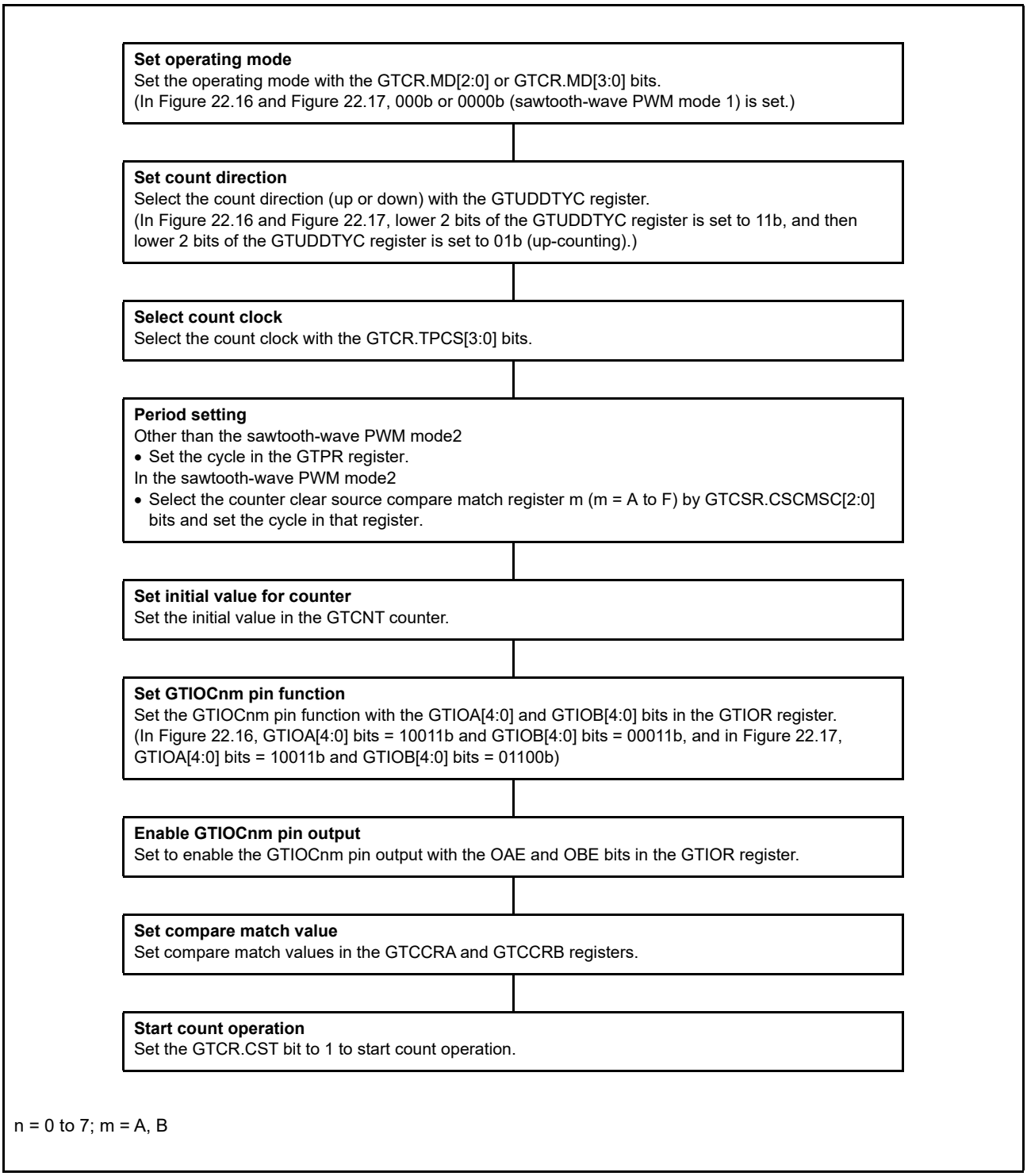


Figure 22.18 Example for Setting Toggled Output Operation

22.3.1.3 Input Capture Function

Through detecting hardware sources selected by the GTICASR and GTICBSR registers, the GTCNT counter values can be transferred to the GTCCRA and GTCCRB registers respectively.

In complementary PWM mode, the GTCCRA and GTCCRB registers do not function as input capture registers.

Figure 22.19 shows an example of input capture function operation.

In this example, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that an input capture is performed at both edges of the GTIOC0A pin input and at the rising edge of the GTIOC0B pin input.

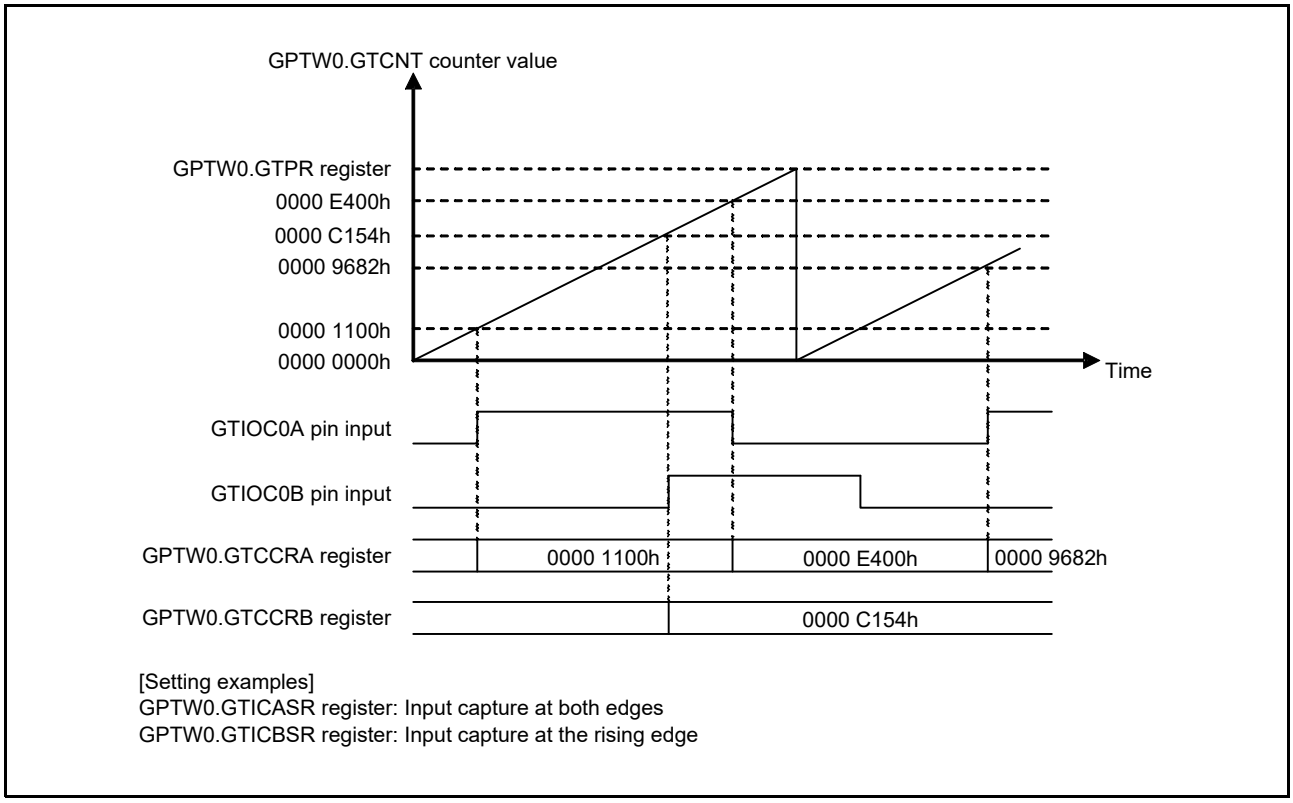


Figure 22.19 Example of Input Capture Operation

Figure 22.20 shows an example for setting input capture operation.

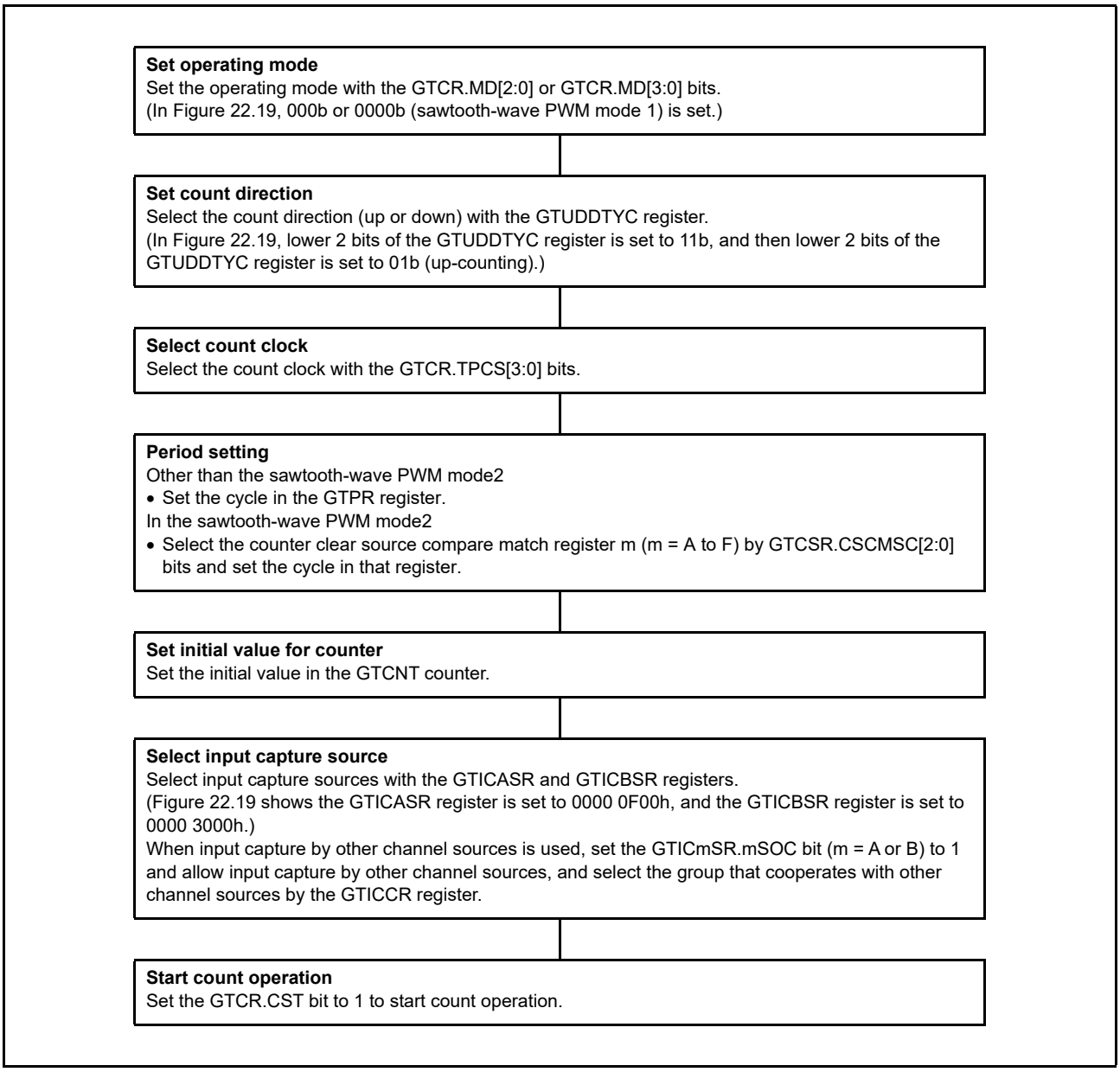
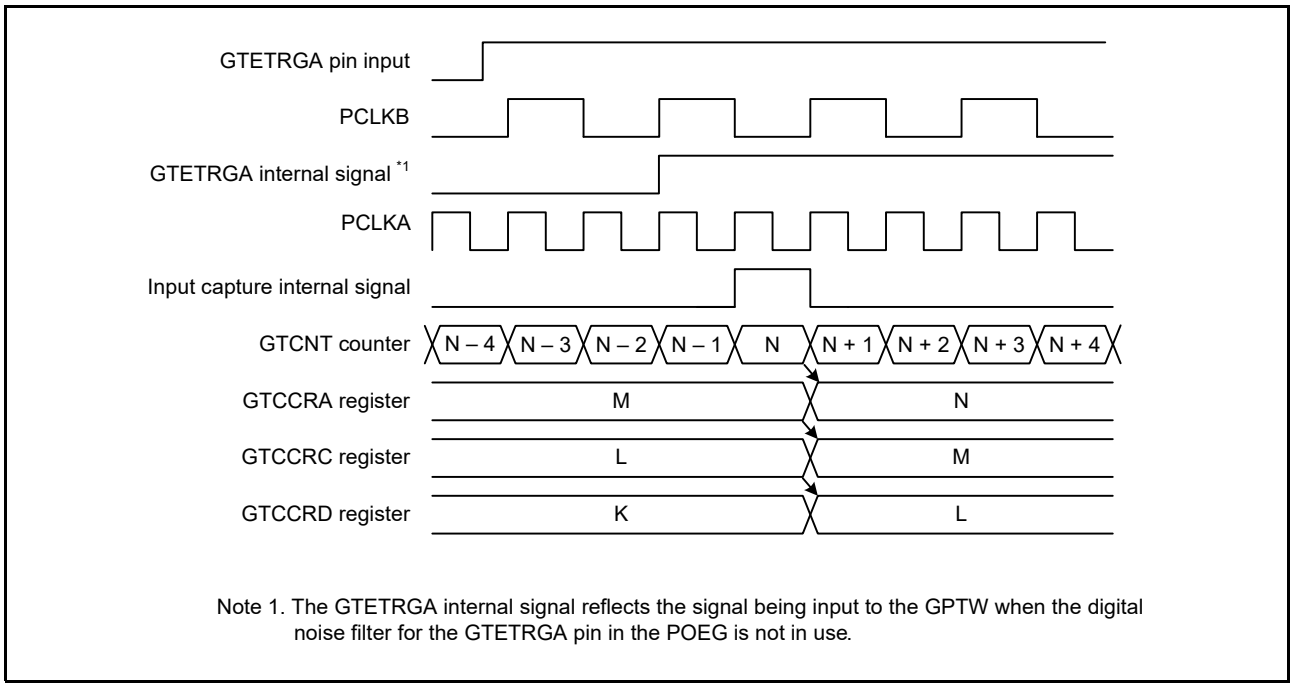


Figure 22.20 Example for Setting Input Capture Operation

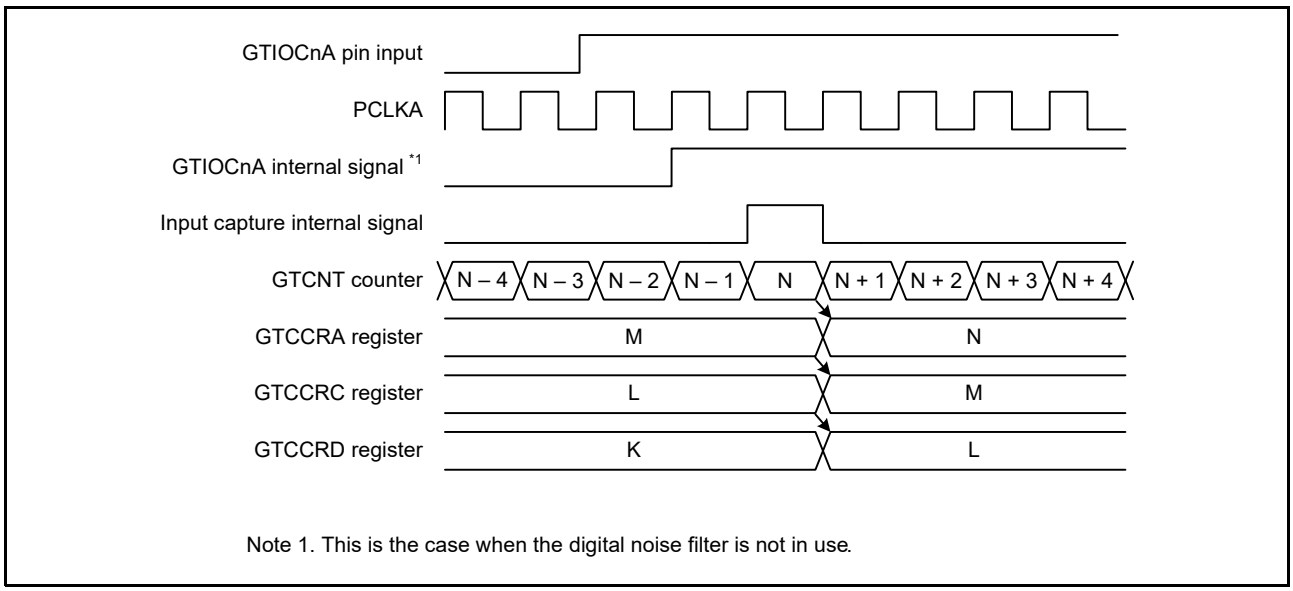
Figure 22.21 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTETRGA pin.



Note 1. The GTETRGA internal signal reflects the signal being input to the GPTW when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 22.21 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 22.22 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 22.22 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 22.23 shows an example of the timing of operations for input capture in response to event input from the ELCA. This is an example of capture of the counter value by the GPTW1.GTCCRA register in response to an input signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with PCLKA. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

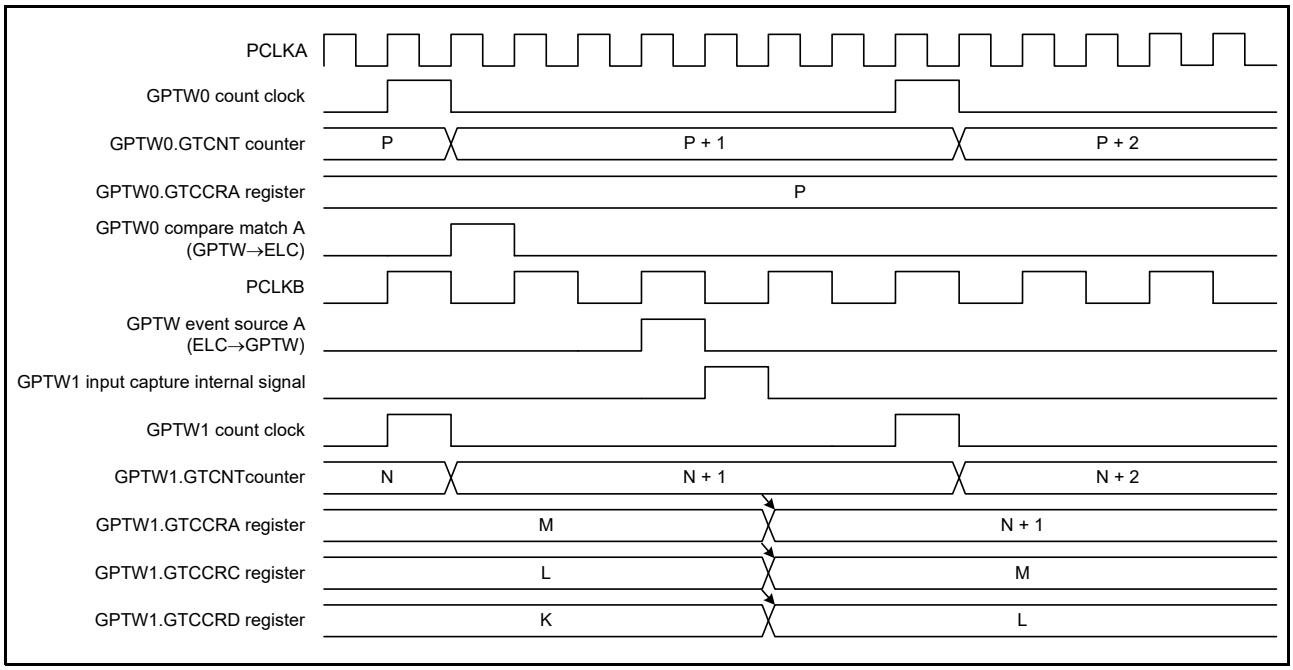


Figure 22.23 Example of the Timing of Operations for Input Capture in Response to Event Input from the ELCA

Figure 22.24 shows the input capture operation timing by the counter clock of other channels.

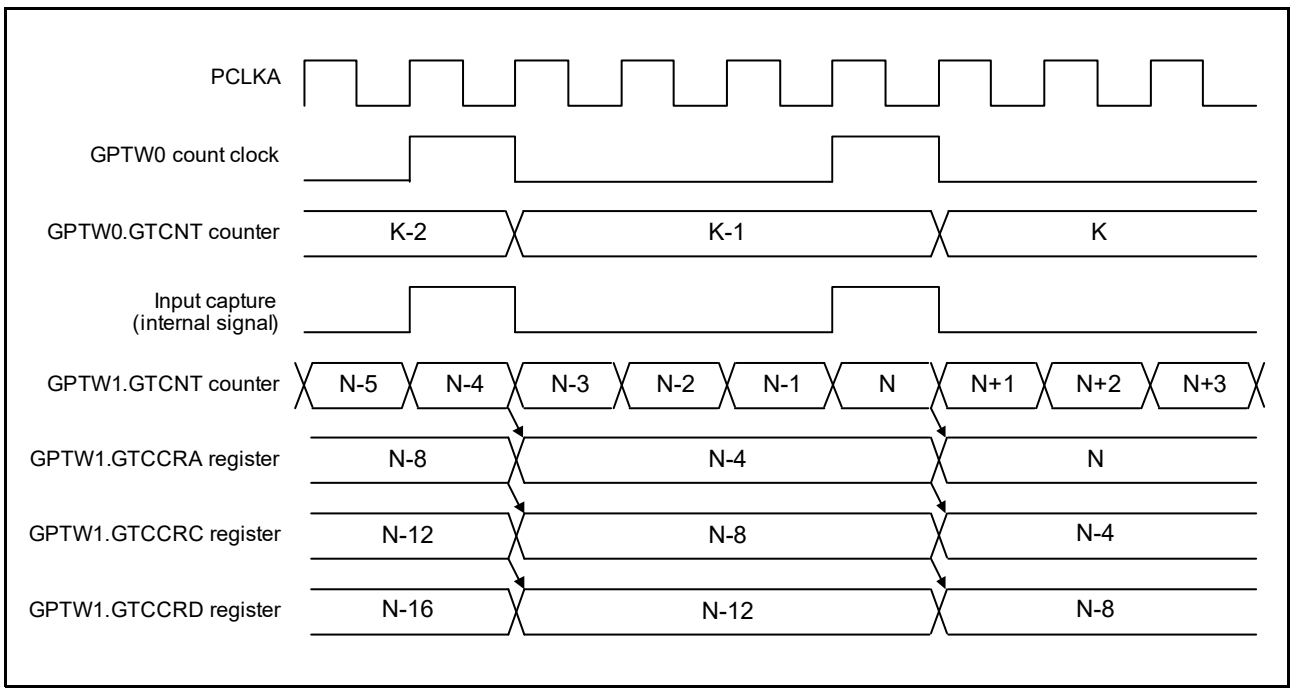


Figure 22.24 Example of the Timing of Input Capture Operation in Response to Count Clock from Other Channel

22.3.2 Buffer Operation

The following buffer operation can be set with the GTBER register.

- Buffer operation with the GTPR, GTPBR, and GTPDBR registers used together
- Buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers used together
- Buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers used together
- Buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers used together
- Buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers used together

22.3.2.1 GTPR Register Buffer Operation

The GTPBR register can function as a buffer register for the GTPR register, and the GTPDBR register can function as a buffer register for the GTPBR register (double buffer register for the GTPR register).
In complementary PWM mode, the buffer transfer from the GTPDBR register to the temporary register P is performed only in the master channel (GPTW0). The temporary register P is transferred to each GTPBR register of master channel, slave channel 1 (GPTW1), and slave channel 2 (GPTW2). Transfer from the GTPBR register to the GTPR register is made concurrently in three channels. Therefore, the same value is stored in the same register of the three channels. The GTPR register of the master channel represents the period of the GTCNT counters of the three channels. In the slave channels, periods are controlled using the GTPR register value and the GTDVU register value.
The setting is invalid in the sawtooth-wave PWM mode 2.
The buffer transfer is performed at an overflow (in up-counting) or underflow (in down-counting) in sawtooth-wave mode or in event counting, or at a trough in triangle-wave mode.
When in sawtooth-wave mode and in event counting, the buffer transfer is performed even when the counter clearing occurs during the count is in operation.

- Hardware source clearing

(Clear source selected with GTCSR.CSGTRGAR, CSGTRGAF, CSGTRGBR, CSGTRGBF, CSGTRGCR, CSGTRGCF, CSGTRGDR, CSGTRGDF, CSCARBL, CSCARBH, CSCAFBL, CSCAFBH, CSCBRAL, CSCBRAH, CSCBFAL, CSCBFAH, CSELCA, CSELCB, CSELCC, CSELCD, CSELCE, CSELCF, CSELCG, CSELCH, CSCMSC[2:0], or CP1CCE bit)

- Clearing by software

(When the GTCLR.CCLRn bit is written to 1 during the GTCSR.CCLR bit is set to 1) (n = 0 to 7)
Figure 22.10 shows the buffer transfer timing in the complementary PWM mode.

Table 22.6 The GTPR Buffer Transfer Timing in Complementary PWM Mode

Mode	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM mode 3 Complementary PWM mode 4
GTPDBR register ↓ Temporary register P	Counter running • One cycle of PCLKA after writing to the GTCCRD register for slave channel 2 (GPTW2) Counter stopped • One cycle of PCLKA after writing to the GTPDBR register	Counter running • One cycle of PCLKA after writing to the GTCCRD register for slave channel 2 (GPTW2) Counter stopped • One cycle of PCLKA after writing to the GTPDBR register	Counter running • One cycle of PCLKA after writing to the GTCCRD register for slave channel 2 (GPTW2) Counter stopped • One cycle of PCLKA after writing to the GTPDBR register
Temporary register P ↓ GTPBR register	When data is transferred to the temporary register P during up-counting middle • One cycle of PCLKA after transfer of data to the temporary register P When data is transferred to the temporary register P during a section other than up-counting middle • At the end of trough	When data is transferred to the temporary register P during down-counting middle • One cycle of PCLKA after transfer of data to the temporary register P When data is transferred to the temporary register P during a section other than down-counting middle • At the end of crest	When data is transferred to the temporary register P middle • One cycle of PCLKA after transfer of data to the temporary register P When data is transferred to the temporary register P during a section other than middle • At the end of crest/trough
GTPBR register ↓ GTPR register	At the end of crest • Counter clearing in an up-counting middle or crest section, including counter clearing enabled by setting the GTCSR.CP1CCE bit	At the end of trough • Counter clearing in a down-counting middle or trough section	At the end of crest At the end of trough • Counter clearing

To use a GTPR register in GPTW3 to GPTW7 as a buffer, set the GTBER.PR[0] bit to 1. Otherwise, set the given bit to 0.

To set the GTPR register for GPTW0 to GPTW2 to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.

In complementary PWM mode, complementary PWM mode-specific buffer operation is performed regardless of the GTBER.PR[1:0] bit setting.

Figure 22.25 to Figure 22.27 show examples of the GTPR register buffer operation and Figure 22.31 shows an example for setting the GTPR register buffer operation.

For details of operation settings in complementary PWM mode, refer to section 22.3.3, PWM Output Operating Mode

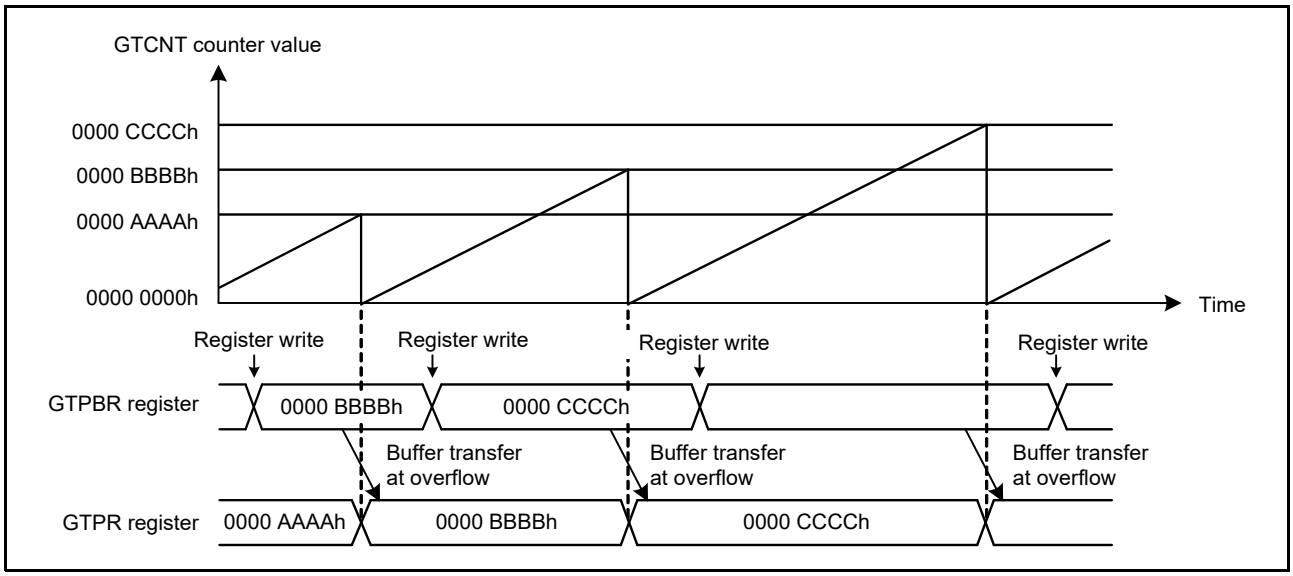


Figure 22.25 Example of the GTPR Register Buffer Operation (Sawtooth Waves in Up-Counting)

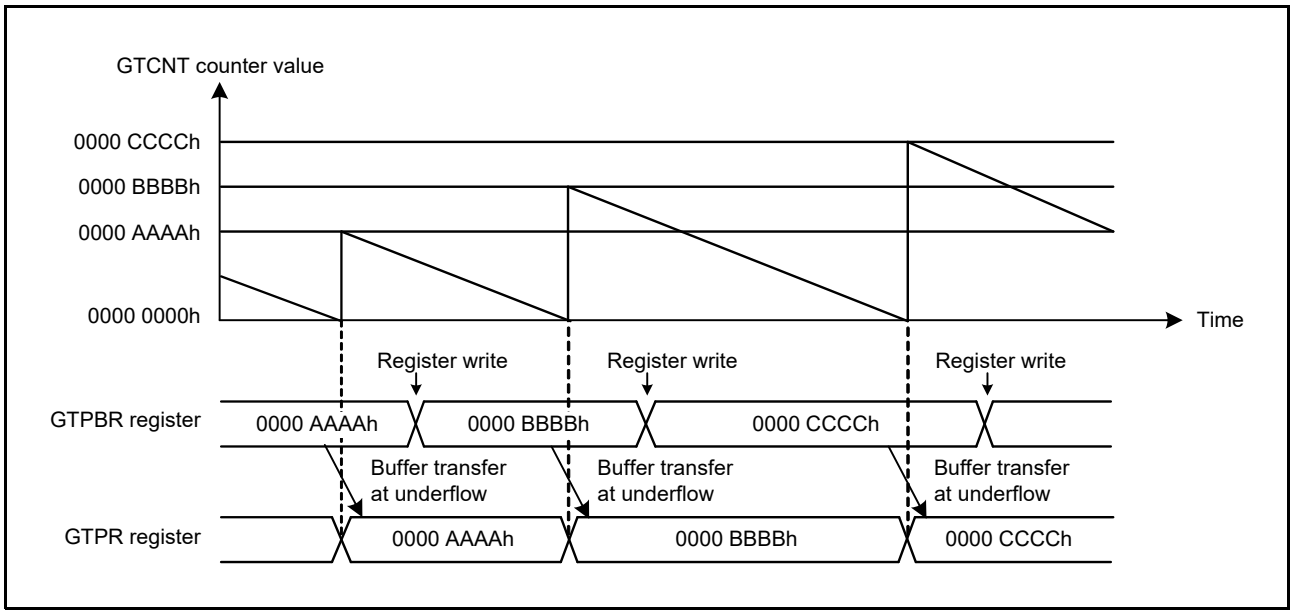


Figure 22.26 Example of the GTPR Register Buffer Operation (Sawtooth Waves in Down-Counting)

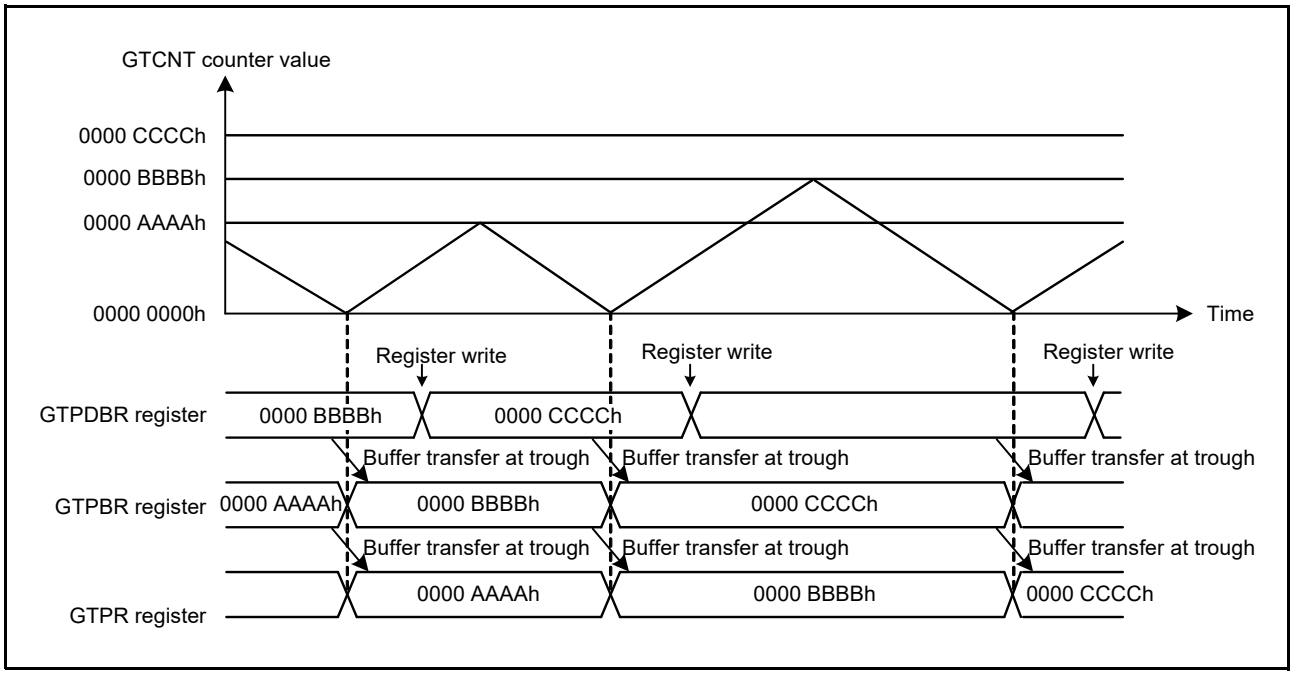


Figure 22.27 Example of the GTPR Register Double Buffer Operation (Triangle Waves)

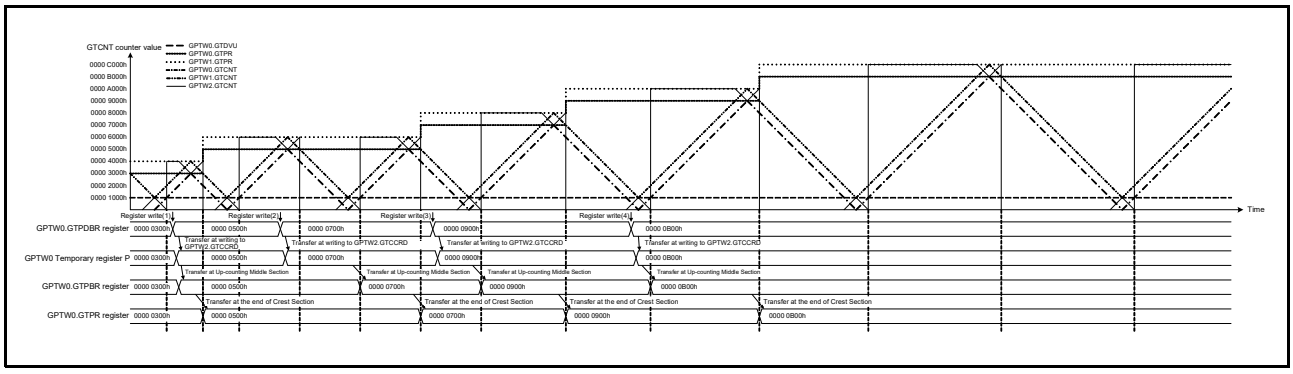


Figure 22.28 Example of GTPR Double Buffer Operation (Complementary PWM Mode 1)

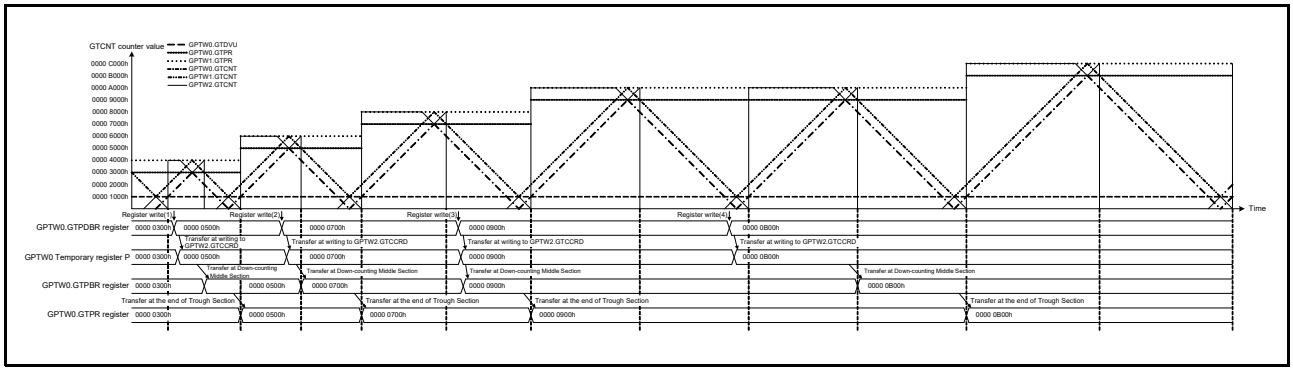


Figure 22.29 Example of GTPR Double Buffer Operation (Complementary PWM Mode 2)

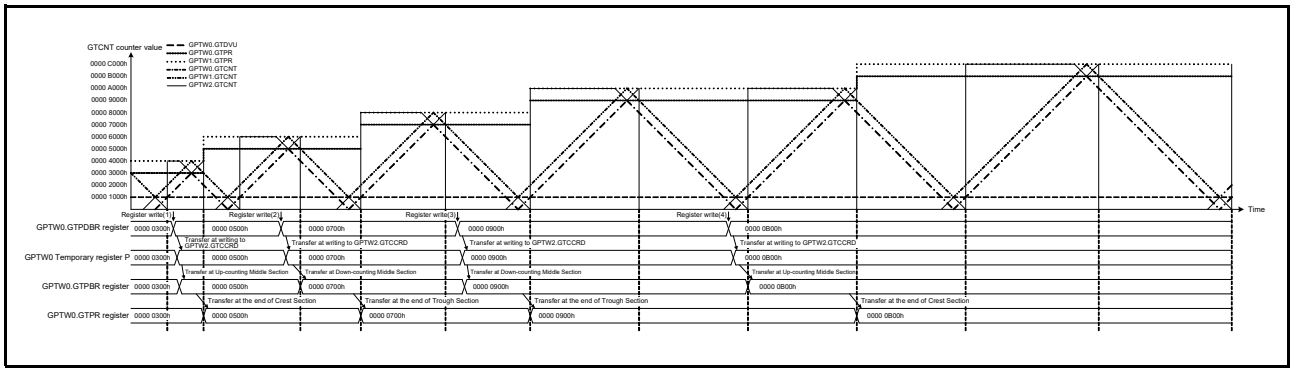


Figure 22.30 Example of GTPR Double Buffer Operation (Complementary PWM Mode 3, 4)

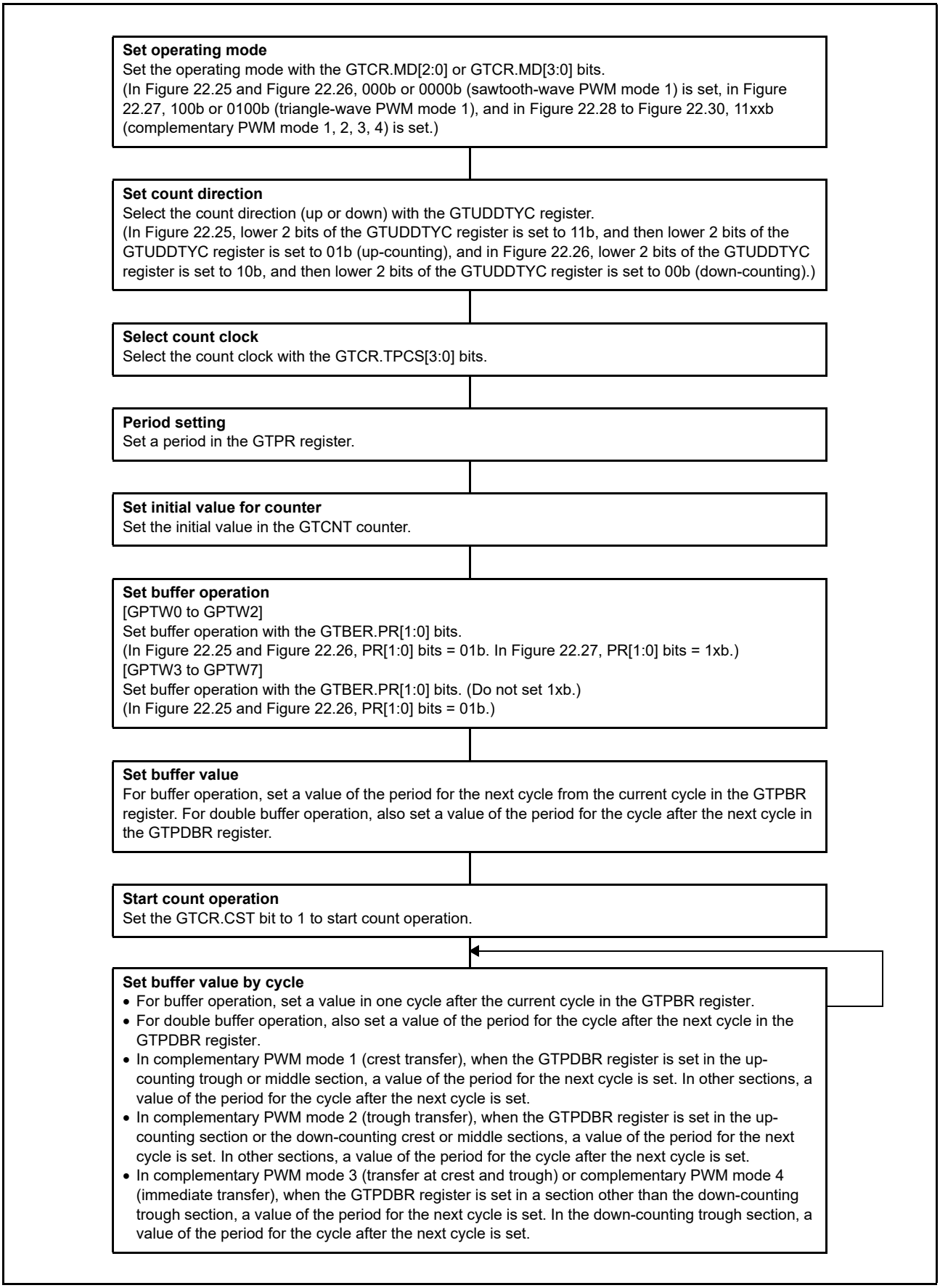


Figure 22.31 Example for Setting the GTPR Register Buffer Operation

22.3.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers

The GTCCRC register can function as the GTCCRA buffer register and the GTCCRD register can function as the GTCCRC buffer register (double buffer register for the GTCCRA register). Similarly, the GTCCRE register can function as the GTCCRB buffer register and the GTCCRF register can function as the GTCCRE buffer register (double buffer register for the GTCCRB register).
To set the GTCCRA or GTCCRB register to function as a double buffer, set the GTBER.CCRA[1:0] or GTBER.CCRB[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.
The following describes buffer operation during output compare and input capture operation.
In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA[1:0] bits and GTBER.CCRB[1:0] bits.

(1) When the GTCCRA or GTCCRB Register Functions as Output Compare Register

In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA[1:0] bits and GTBER.CCRB[1:0] bits. For details, refer section 22.3.3, PWM Output Operating Mode. Other than above PWM output operation mode, buffer transfer is performed for the following three cases:

- Buffer transfer at an overflow (in up-counting) and underflow (in down-counting)
  Buffer transfer is performed at an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode and in event counting, and at a trough (triangle-wave PWM mode 1) or at a crest/trough (triangle-wave PWM mode 2) in triangle-wave mode.
- Buffer transfer by counter clearing
  In sawtooth-wave mode and in event counting, the buffer transfer by similar counter clearing sources in counting operation and in section 22.3.2.1, GTPR Register Buffer Operation is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).
  In triangle-wave mode, buffer transfer by counter clearing is not performed.
- Forcible buffer transfer
  In both sawtooth-wave mode and triangle-wave mode, buffer transfer of the GTCCRA and GTCCRB registers is forcibly performed by writing 1 to the GTBER.CCRSWT bit while the counting is stopped.
  In sawtooth-wave one-shot pulse mode and triangle-wave PWM mode 3, buffer transfer from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B is also performed using the forcible buffer operation.

Figure 22.32 to Figure 22.34 show examples of the GTCCRA and GTCCRB registers buffer operations, and Figure 22.35 shows an example for setting the GTCCRA and GTCCRB registers buffer operations.

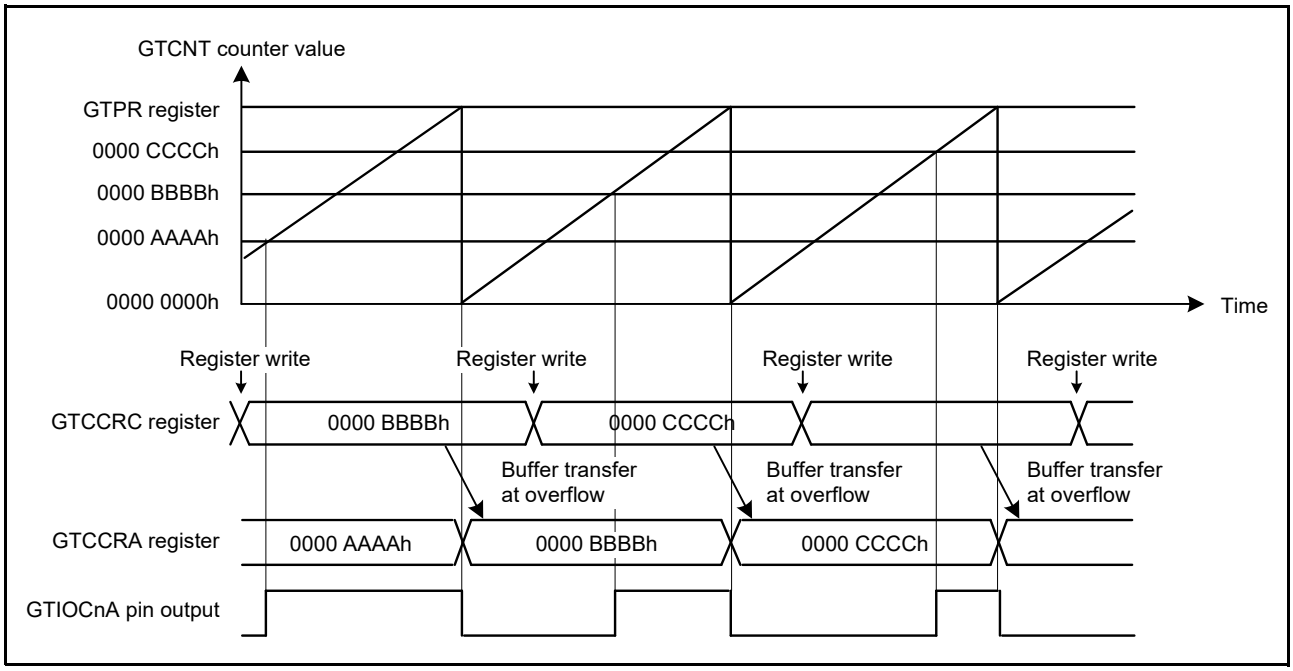


Figure 22.32 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Sawtooth-Waves, Sawtooth-Wave PWM mode 1 in Up-Counting, High Output at GTCCRA Register Compare Match, Low Output at the End of the Cycle) (n = 0 to 7)

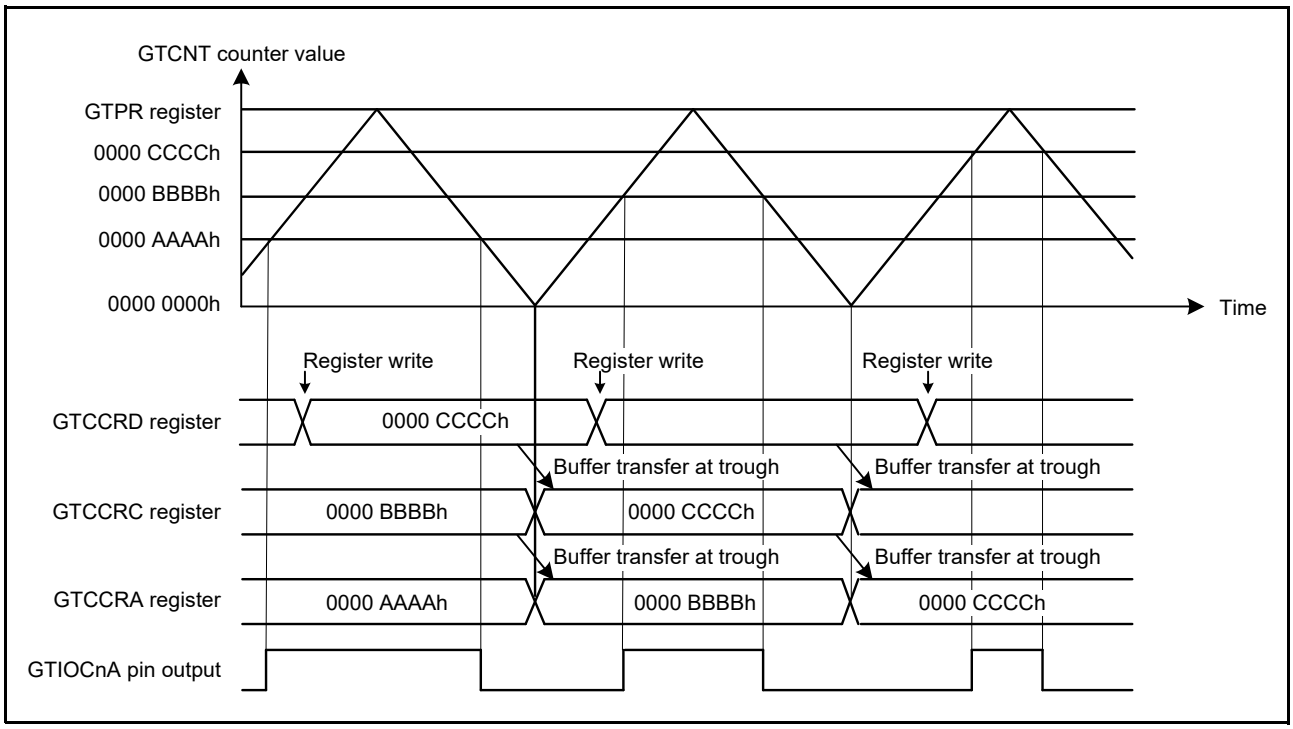


Figure 22.33 Example of GTCCRA and GTCCRB Registers Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Trough, Output Toggled at GTCCRA Register Compare Match, Output Retained at the End of the Cycle) (n = 0 to 7)

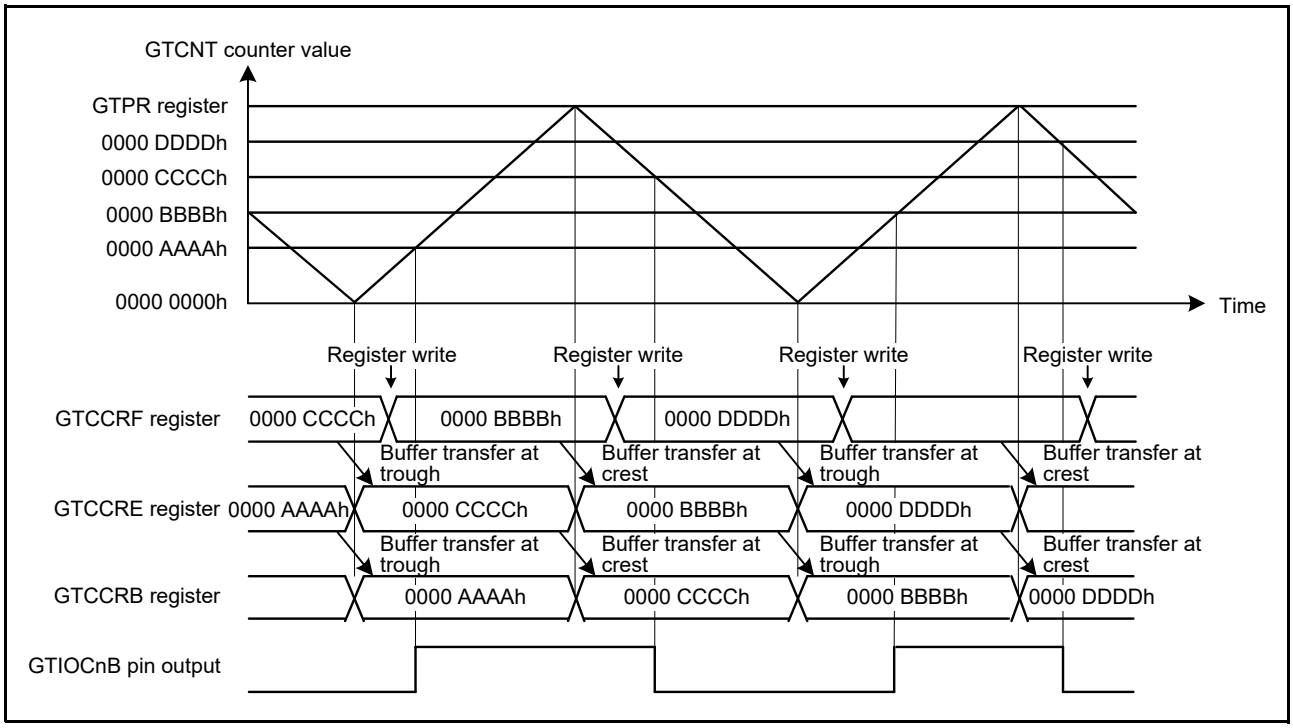


Figure 22.34 Example of GTCCRA and GTCCRB Registers Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Both Troughs and Crests, Output Toggled at GTCCRB Register Compare Match, Output Retained at the End of the Cycle) (n = 0 to 7)

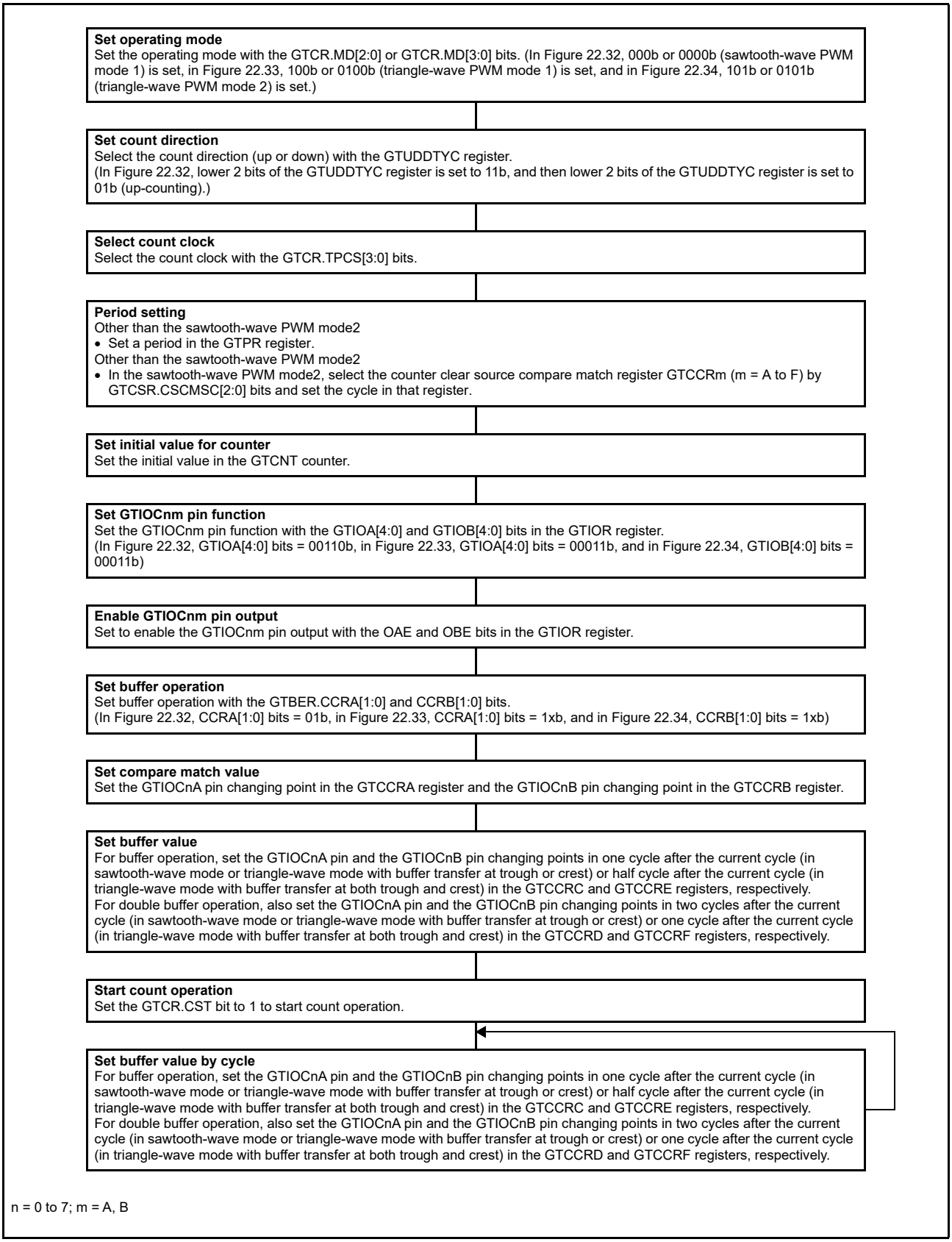


Figure 22.35 Example for Setting Buffer Operation of the GTCCRA and GTCCRB Registers (for Output Compare)

(2) When the GTCCRA or GTCCRB Register Functions as Input Capture Register

The timing of transfer to the buffers is when an input capture is generated. When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB registers and the stored GTCCRA and GTCCRB registers values are transferred to buffer registers. Buffer transfer by counter clearing is not performed at input capture. Figure 22.36 and Figure 22.37 show examples of buffer operation of the GTCCRA and GTCCRB registers, and Figure 22.38 shows an example for setting buffer operation of the GTCCRA and GTCCRB registers.

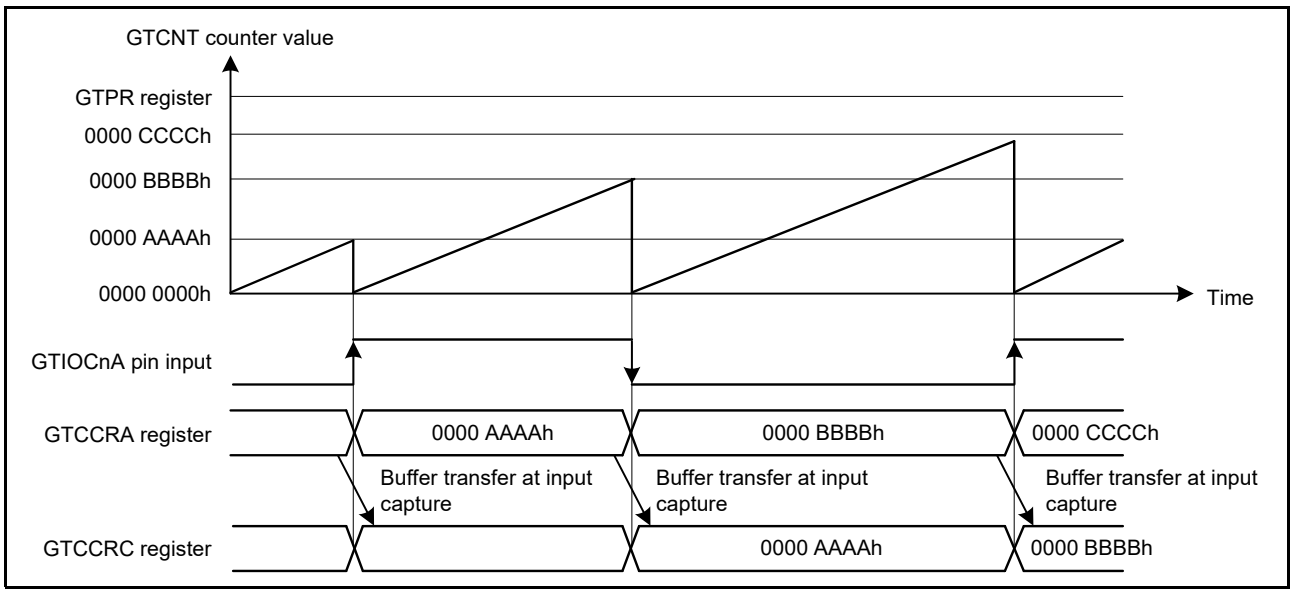


Figure 22.36 Example of Buffer Operation of the GTCCRA and GTCCRB Registers (Counting up to Produce a Sawtooth Wave, and Clearing the GTCNT Counter and Input Capture on Both Edges of the Input on the GTIOCnA Pin) (n = 0 to 7)

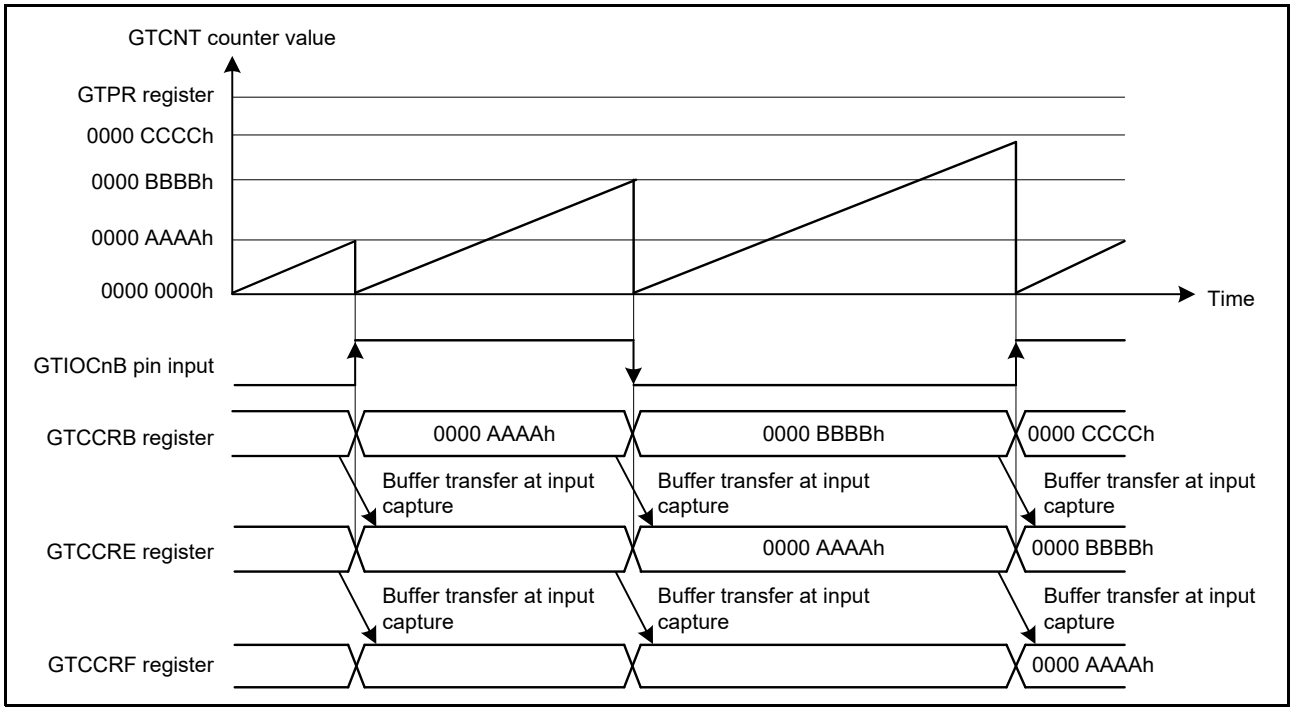


Figure 22.37 Example of Double Buffer Operation of the GTCCRA and GTCCRB Registers (Counting up to Produce a Sawtooth Wave, and Clearing the GTCNT Counter and Input Capture on Both Edges of the Input on the GTIOCnB Pin) (n = 0 to 7)

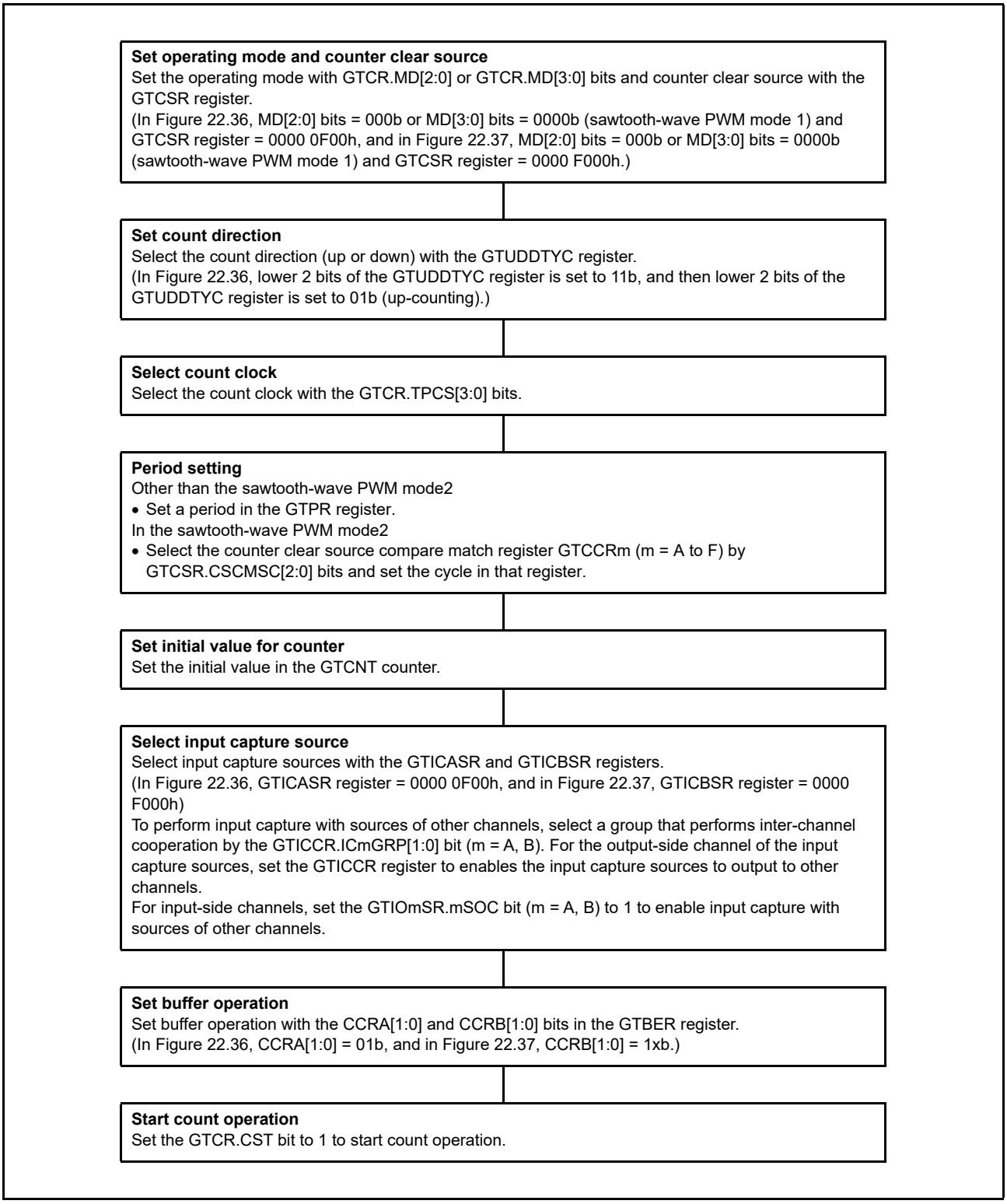


Figure 22.38 Example for Setting Buffer Operation of the GTCCRA and GTCCRB Registers (for Input Capture)

22.3.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to11b for both crest and trough in triangle-wave mode or complementary PWM mode.

In sawtooth-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 22.3.2.1, GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

In complementary PWM mode, the buffer transfer is performed after one PCLK cycle from the GTCCRD register write of slave channel 2.

Figure 22.39 to Figure 22.42 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Figure 22.43 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

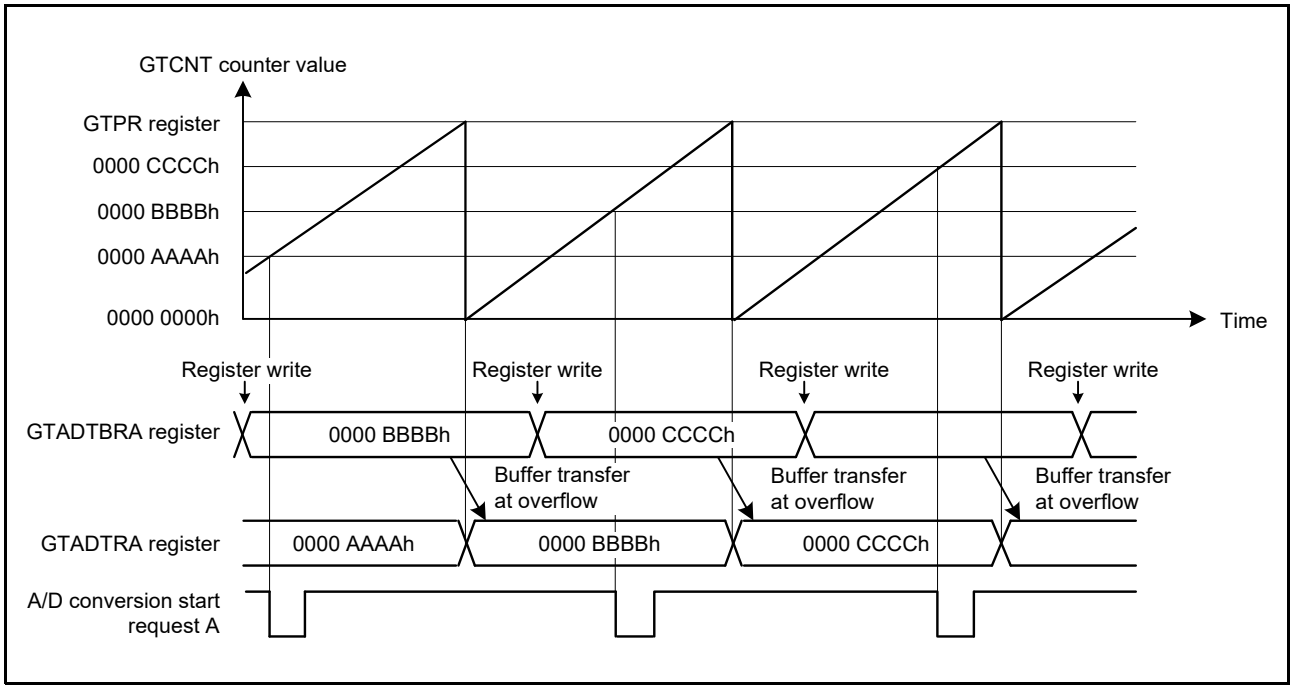


Figure 22.39 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Sawtooth Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)

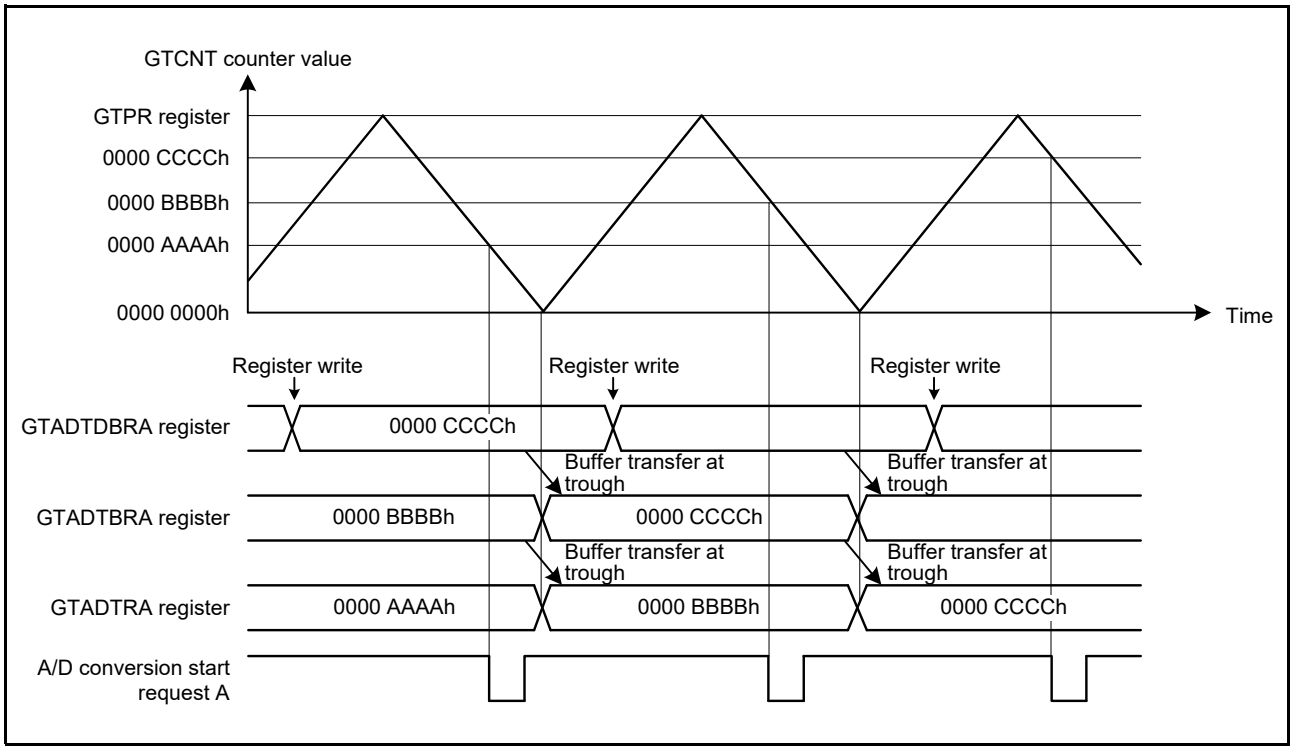


Figure 22.40 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Operation at Trough, A/D Conversion Start Request Generated by Down-Counting)

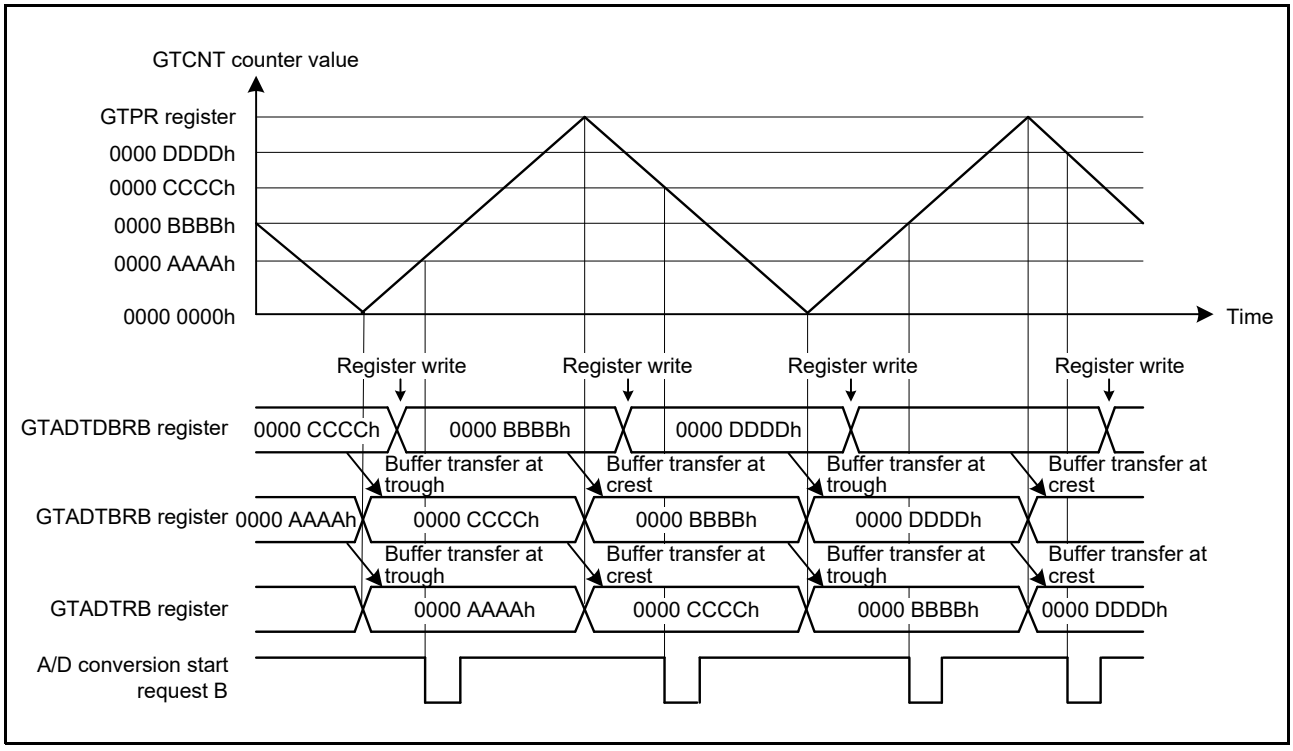


Figure 22.41 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request Generated by Both Up- and Down-Counting)

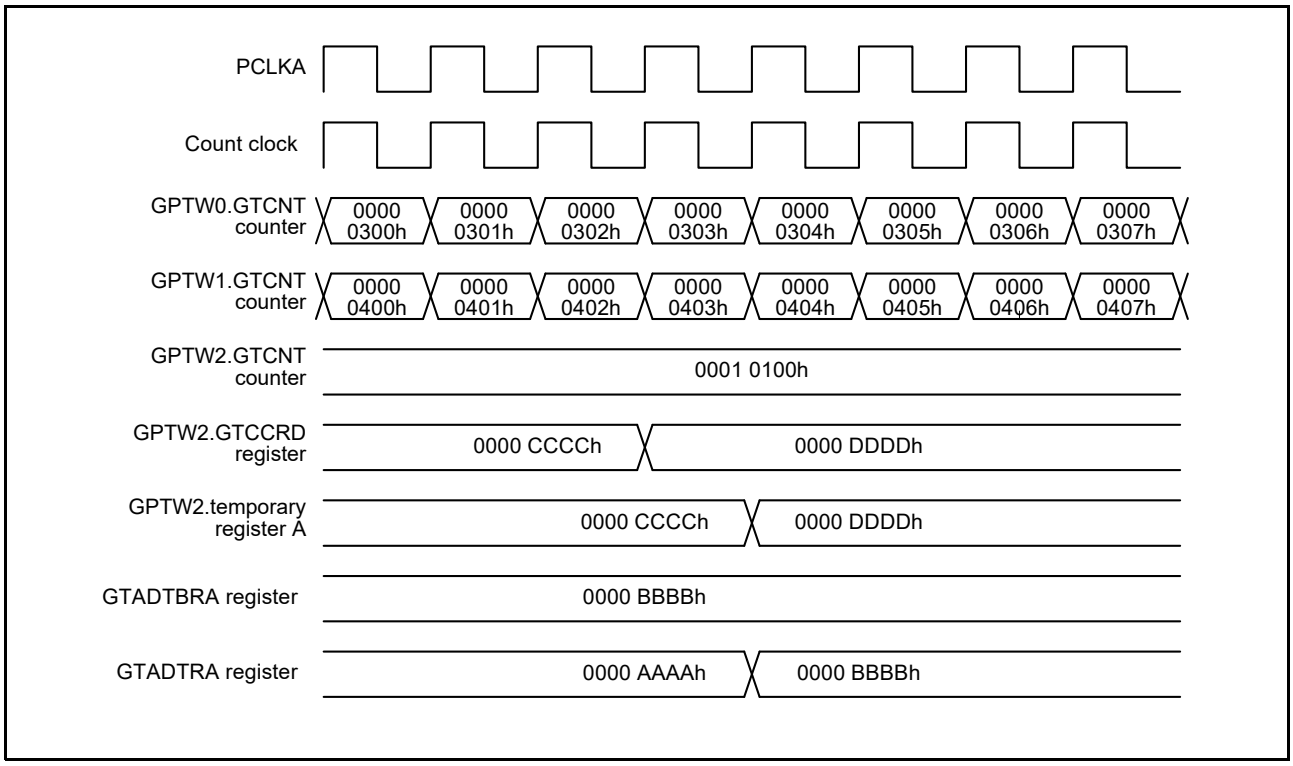


Figure 22.42 Example of Buffer Operation of the GTADTRA and GTADTRB Registers at the GTCCRD Register of Slave Channel 2 Updating in Complementary PWM Mode

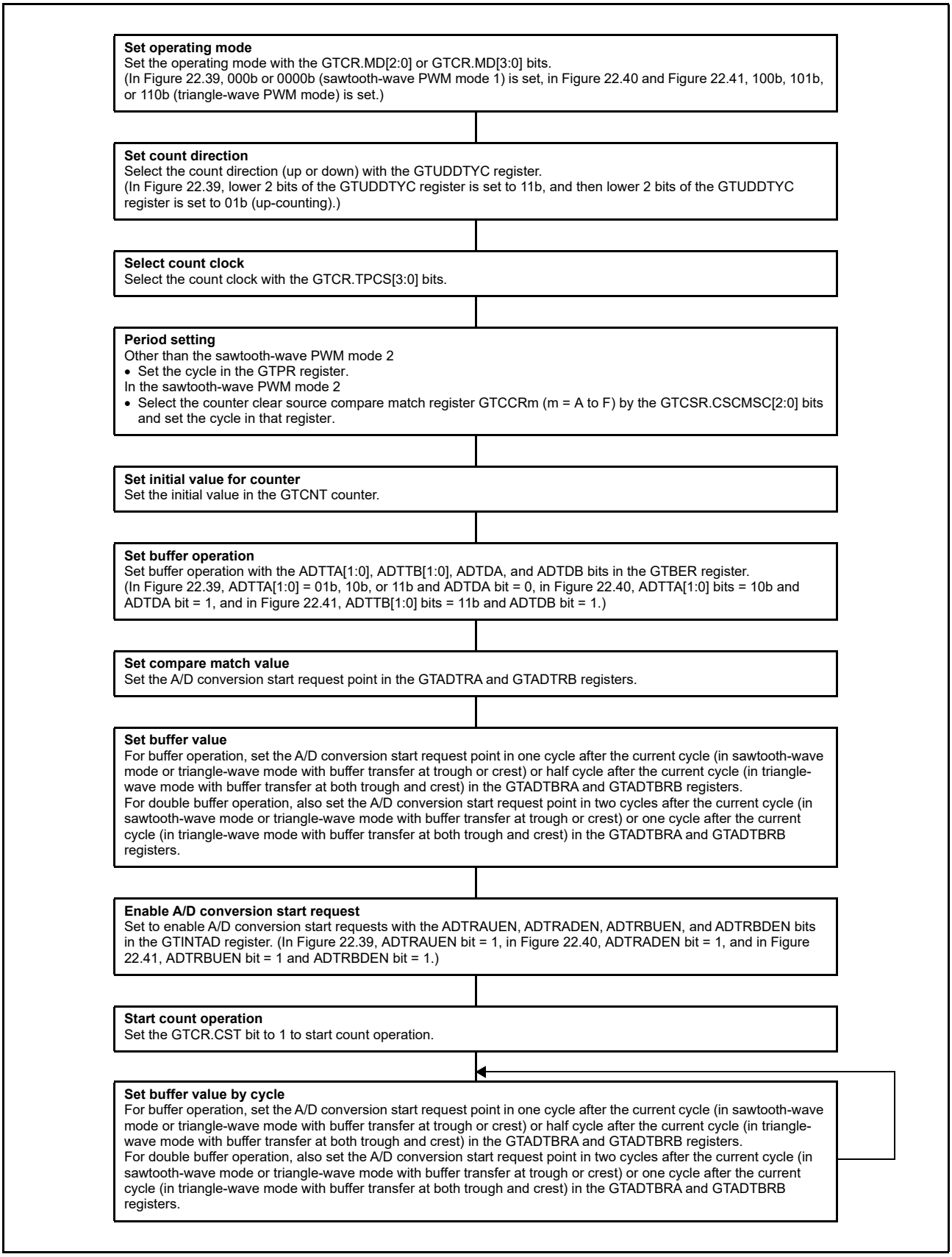


Figure 22.43 Example for Setting Buffer Operation of the GTADTRA and GTADTRB Registers

22.3.3 PWM Output Operating Mode

The GPTW can output PWM waveforms to the GTIOCnA pin or GTIOCnB pin (n = 0 to 7) by a compare match between the GTCNT counter and the GTCCRA or GTCCRB register.
By setting the GTDTCR and GTDVU registers, the compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
In complementary PWM mode, it is possible to output PWM waveforms (positive phase, negative phase) with dead time that guarantees the linearity of the PWM output pulse width near 0% and 100% duty.
In sawtooth-wave mode other than sawtooth-wave PWM mode 2, or triangle-wave mode, or the master channel of complementary PWM mode, the signal synchronized with the PWM cycle can be output from the GTCPPO0 output terminal by setting the GTIOR.PSYE bit to 1. The GTCPPO0 output is toggled at the end of cycle in sawtooth-wave mode or at the timing of crest/trough/GTCNT counter clearing in triangle-wave mode or complementary PWM mode. The initial output of GTCPPO0 is low, and it becomes high when the count starts.

(1) Sawtooth-Wave PWM Mode 1

In sawtooth-wave PWM mode 1, the GTCNT counter performs sawtooth-wave (half-wave) operation by setting the period in the GTPR register and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
When 0 is set in the GTIOR.OmEOCD (m = A, B) bit and the timing of the end of cycle and the timing of the GTCCRm register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOm[3:2] bits.
When 1 is set in the GTIOR.OmEOCD bit, GTIOCnm output is retained.
Figure 22.44 shows an example of sawtooth-wave PWM mode 1 operation, and Figure 22.45 shows an example for setting sawtooth-wave PWM mode 1.

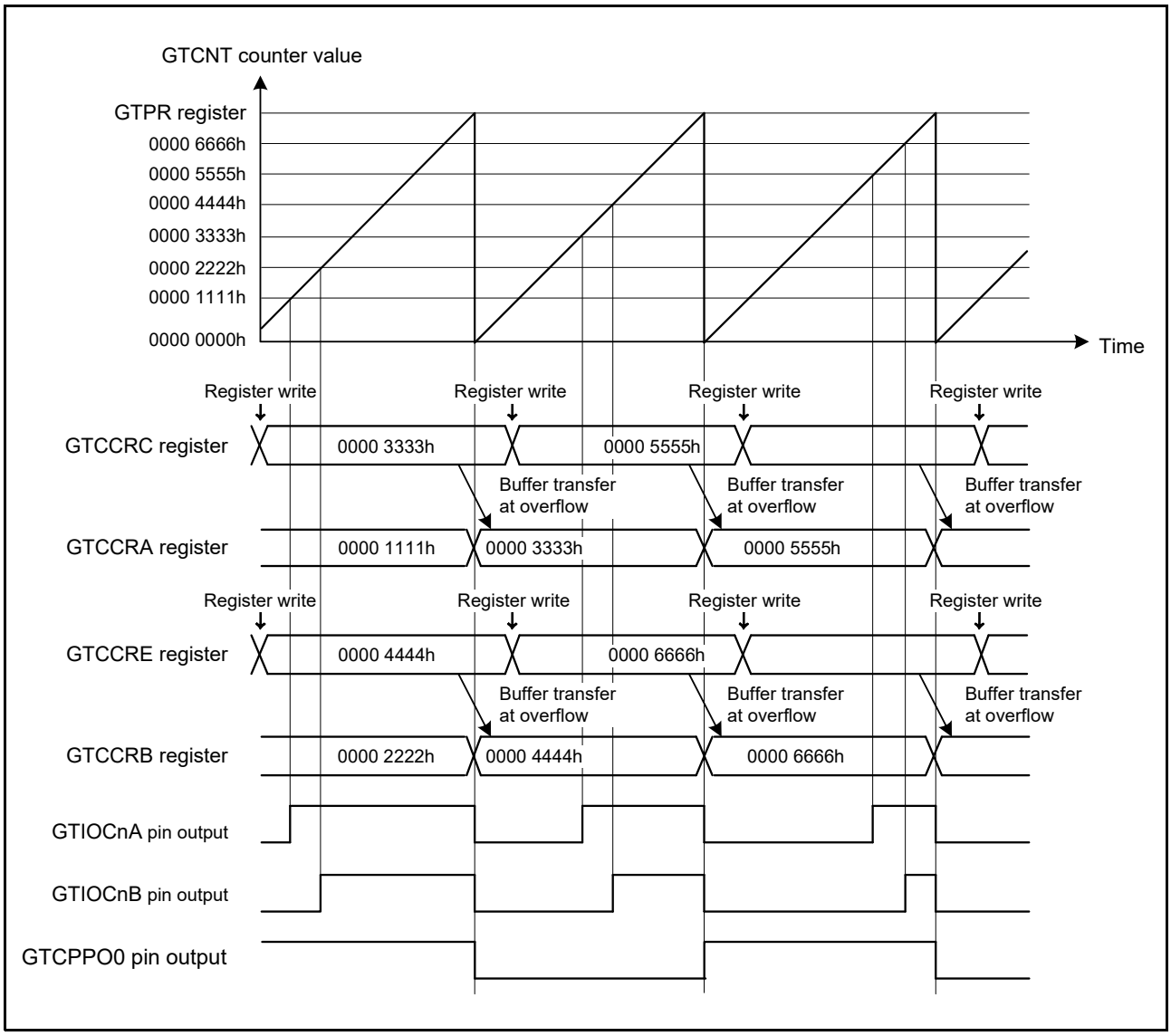


Figure 22.44 Example of Sawtooth-Wave PWM Mode 1 Operation (Up-Counting, Buffer Operation, High Output on Compare Match between the GTCCRA and GTCCRB Registers, Low Output at the End of the Cycle (n = 0 to 7), GPTW0.GTIOR.PSYE bit = 1 in GPTW0)

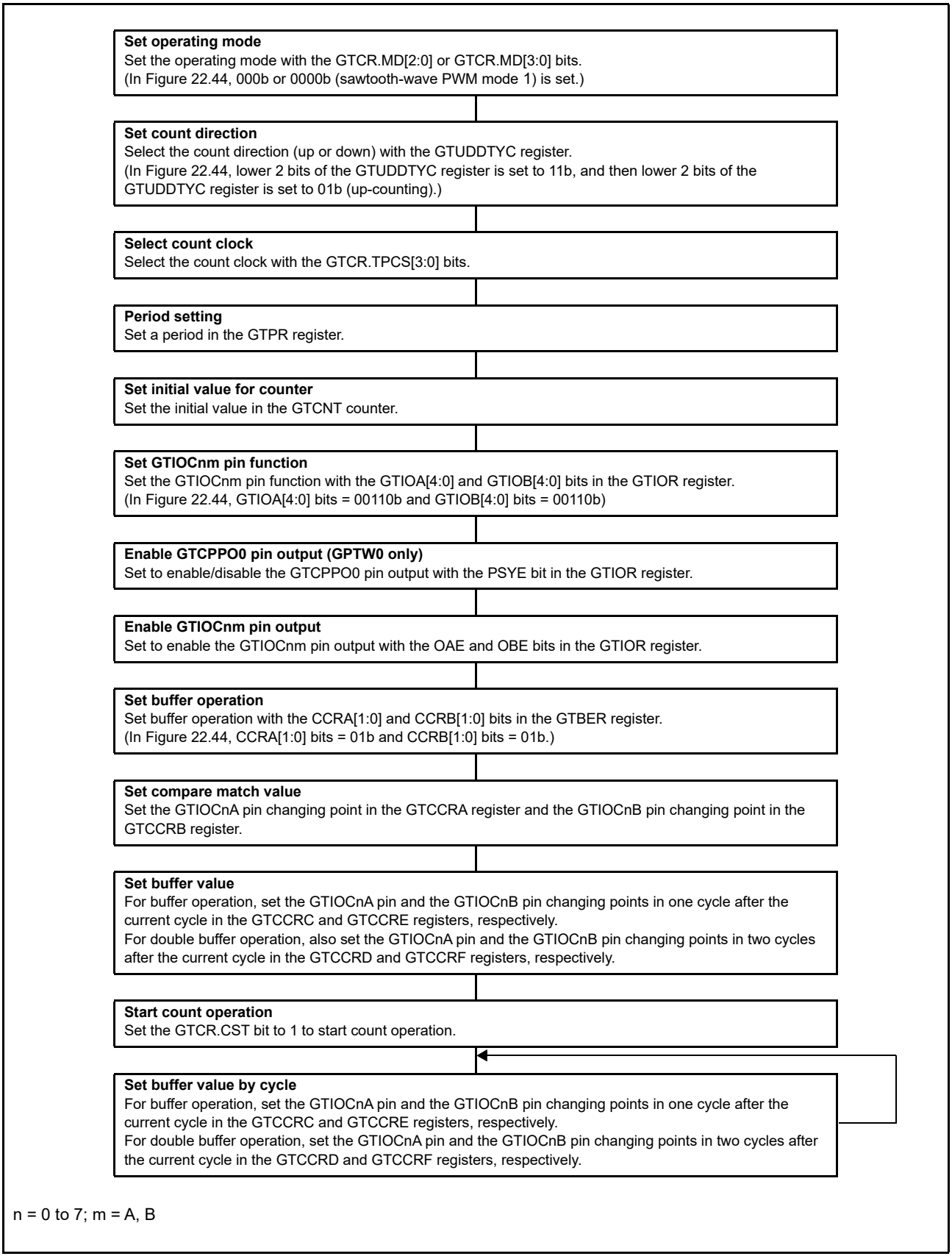


Figure 22.45 Example for Setting Sawtooth-Wave PWM Mode 1

(2) Sawtooth-Wave PWM Mode 2

The sawtooth-wave PWM mode 2 is a mode in which the GTCNT counter is operated as a sawtooth wave by up-counting without using the GTPR register, and the PWM waveform is output by the compare match of the GTCCRA and GTCCRB registers.

The pin output level can be selected from low output, high output, or toggle output separately for a compare match according to the GTIOR register setting.

The GTIOCnA (n = 0 to 2, 4 to 6) pin is used as an output pin. Use the GTIOR.GTIOB[1:0] bits for setting the GTIOCnA pin output at a compare match of the GTCCRB register.

When a counter clear occurs due to the GTCNT counter clearing source selected in the GTCSR register, this is handled at the end of cycle and PWM output operation at the end of the cycle selected with the GTIOR.GTIOA[3:2] bits is performed.

If a counter clear (at the end of cycle) conflicts with a PWM output change due to a GTCCRm (m = A, B) register compare match, PWM output operation is performed at the end of cycle (in the case of the GTIOR.OmEOCD bit = 0) or the PWM output is retained (in the case of GTIOR.OmEOCD bit = 1).

Figure 22.46 to Figure 22.48 show examples of sawtooth-wave PWM mode 2 operation. Figure 22.49 shows an example for setting sawtooth-wave PWM mode 2.

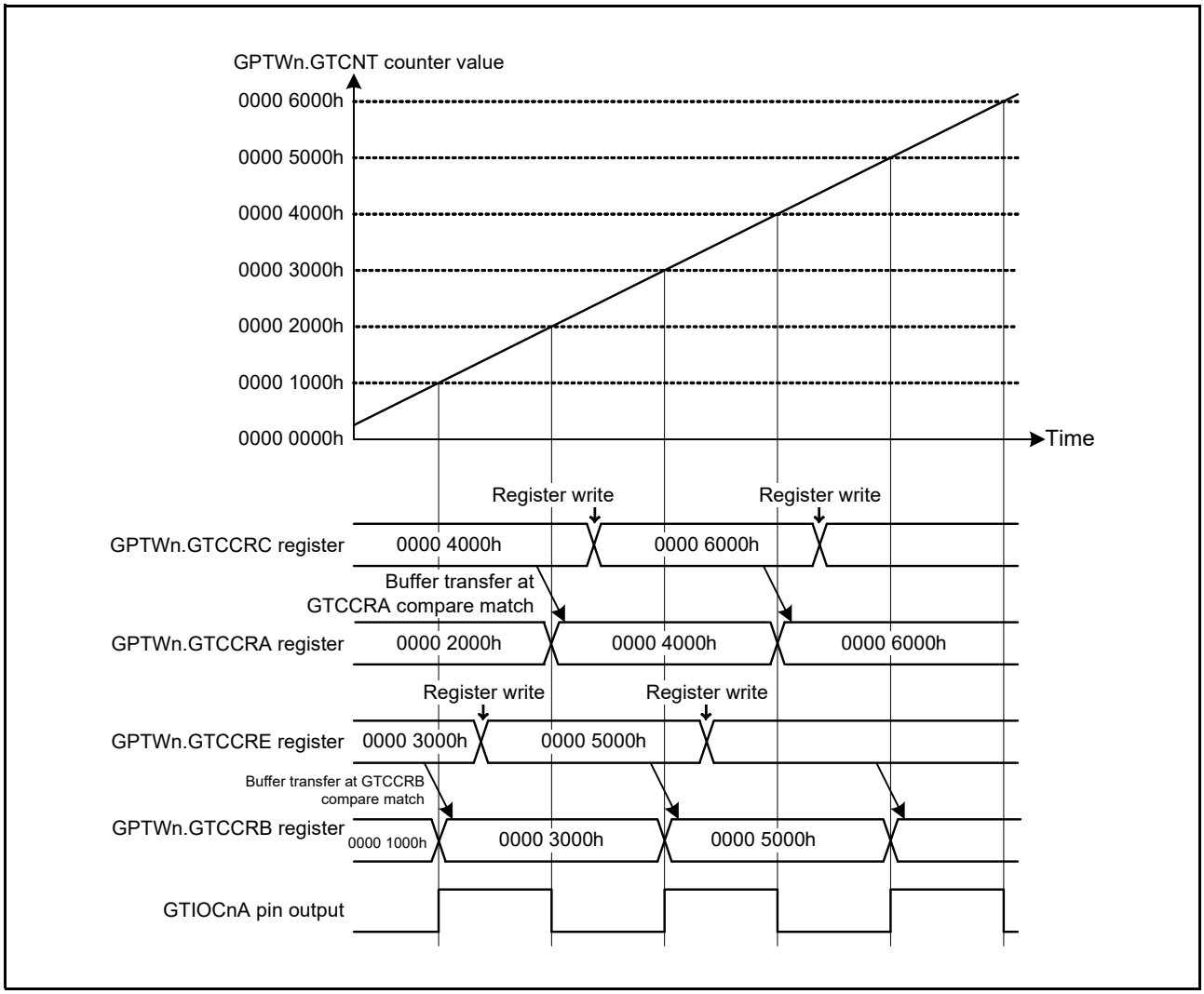


Figure 22.46 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Single Buffer Operation, and No Clear Setting) (n = 0 to 2)

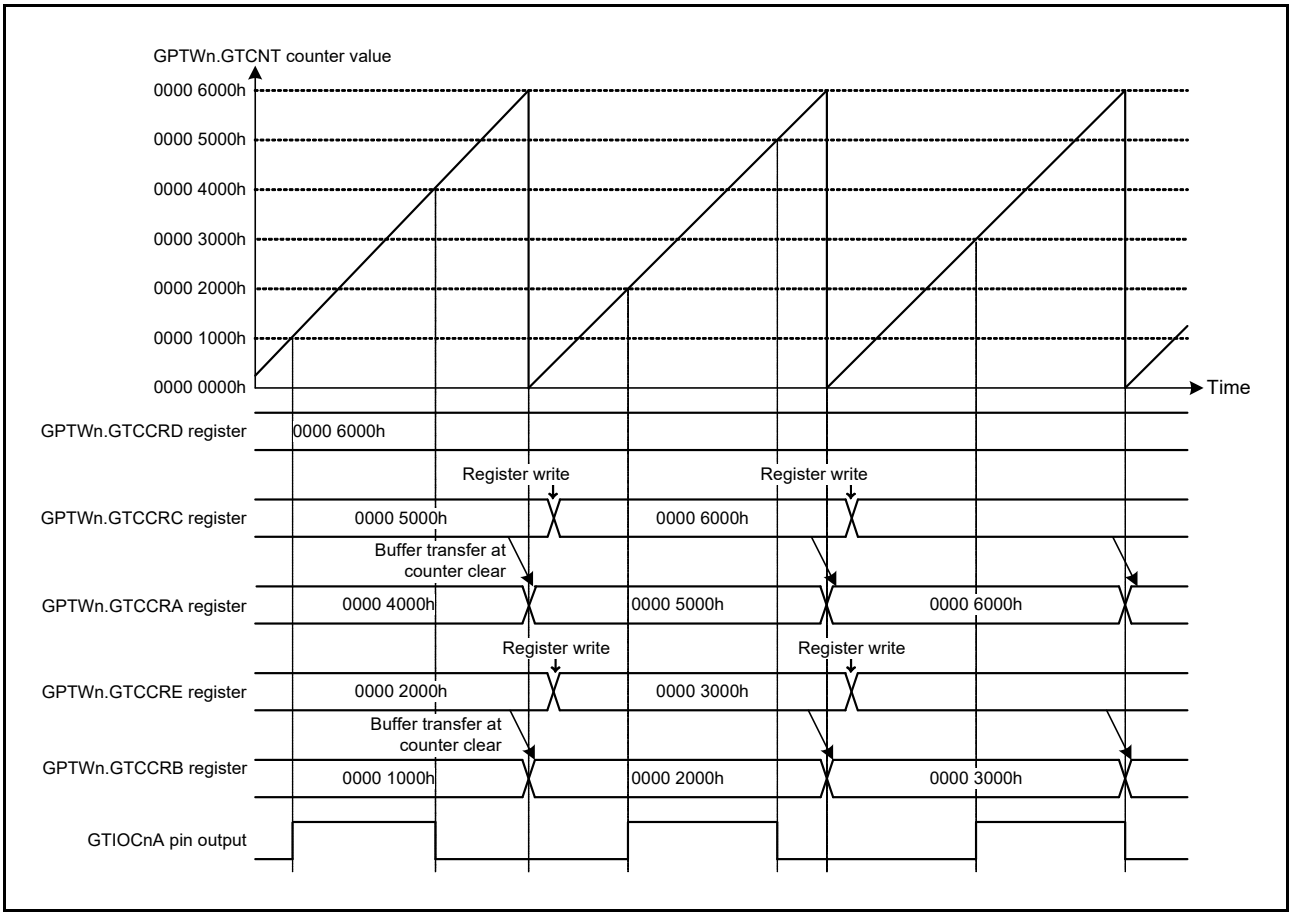


Figure 22.47 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output at the GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Low Output at the End of the Cycle, Single Buffer Operation, Cleared at the GTCCRD Register Compare Match, and the GTIOR.OAEOCD Bit = 0) (n = 0 to 2)

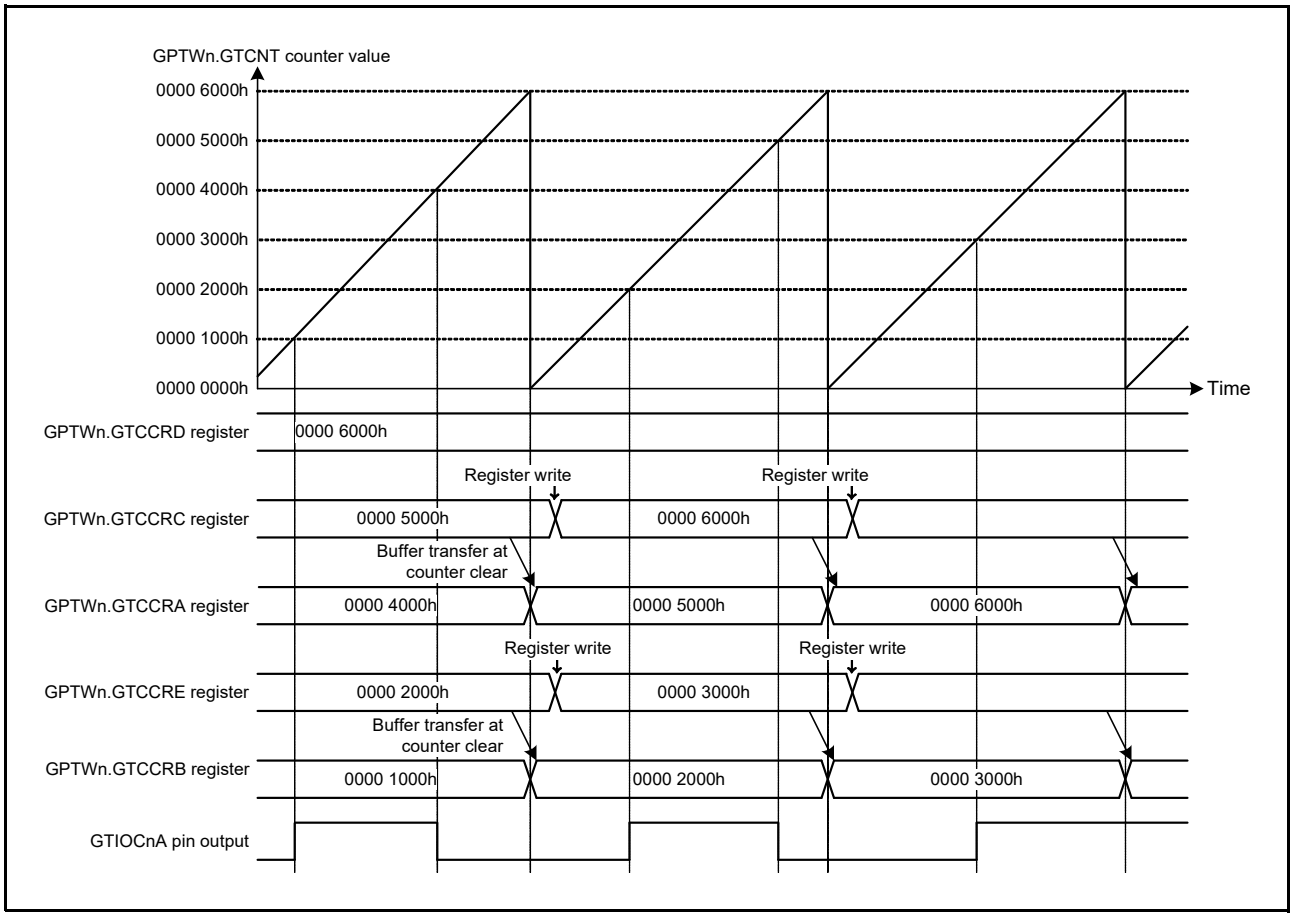


Figure 22.48 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output at the GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at the GTCCRD Register Compare Match, and GTIOR.OAEOCD Bit = 1) (n = 0 to 2)

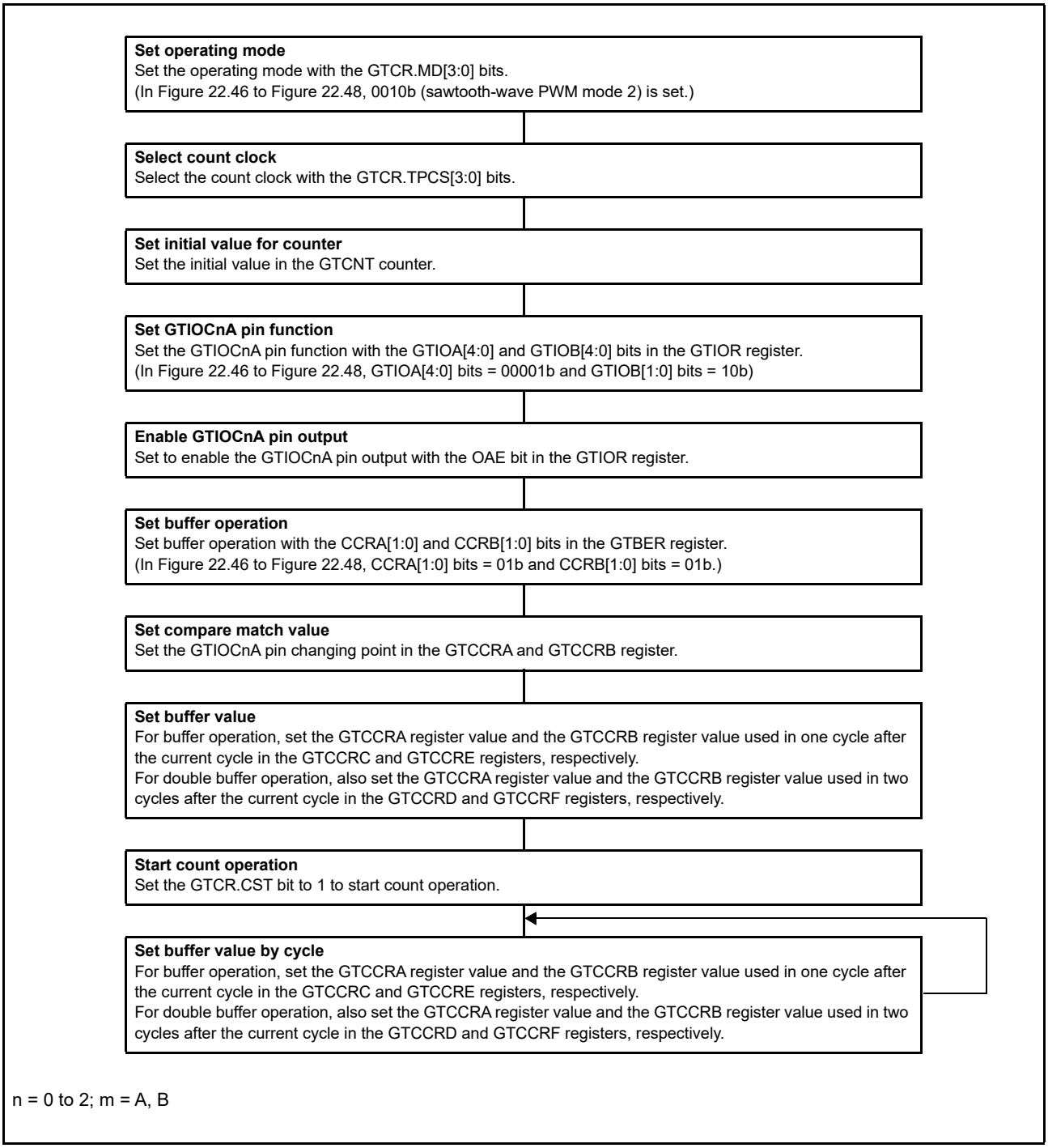


Figure 22.49 Example for Setting Sawtooth-Wave PWM Mode 2

(3) Sawtooth-Wave One-Shot Pulse Mode

The sawtooth-wave one-shot pulse mode is a mode in which the period is set in the GTPR register, the GTCNT counter performs sawtooth-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed (n = 0 to 7).

Buffer operation in sawtooth-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to the temporary register A, and from the GTCCRF register to the temporary register B at the end of the cycle, and from the temporary register A to the GTCCRA register at a GTCCRA register compare match and from the temporary register B to the GTCCRB register at a GTCCRB register compare match. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the end of the cycle according to the GTIOR register setting.

The temporary register A and the temporary register B can perform forcible buffer transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B respectively by writing 1 to the GTBER.CCRSWT bit while the count is stopped.

By setting the GTDTCR and GTDVU registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 22.50 shows an example of sawtooth-wave one-shot pulse mode operation, and Figure 22.51 shows an example for setting sawtooth-wave one-shot pulse mode.

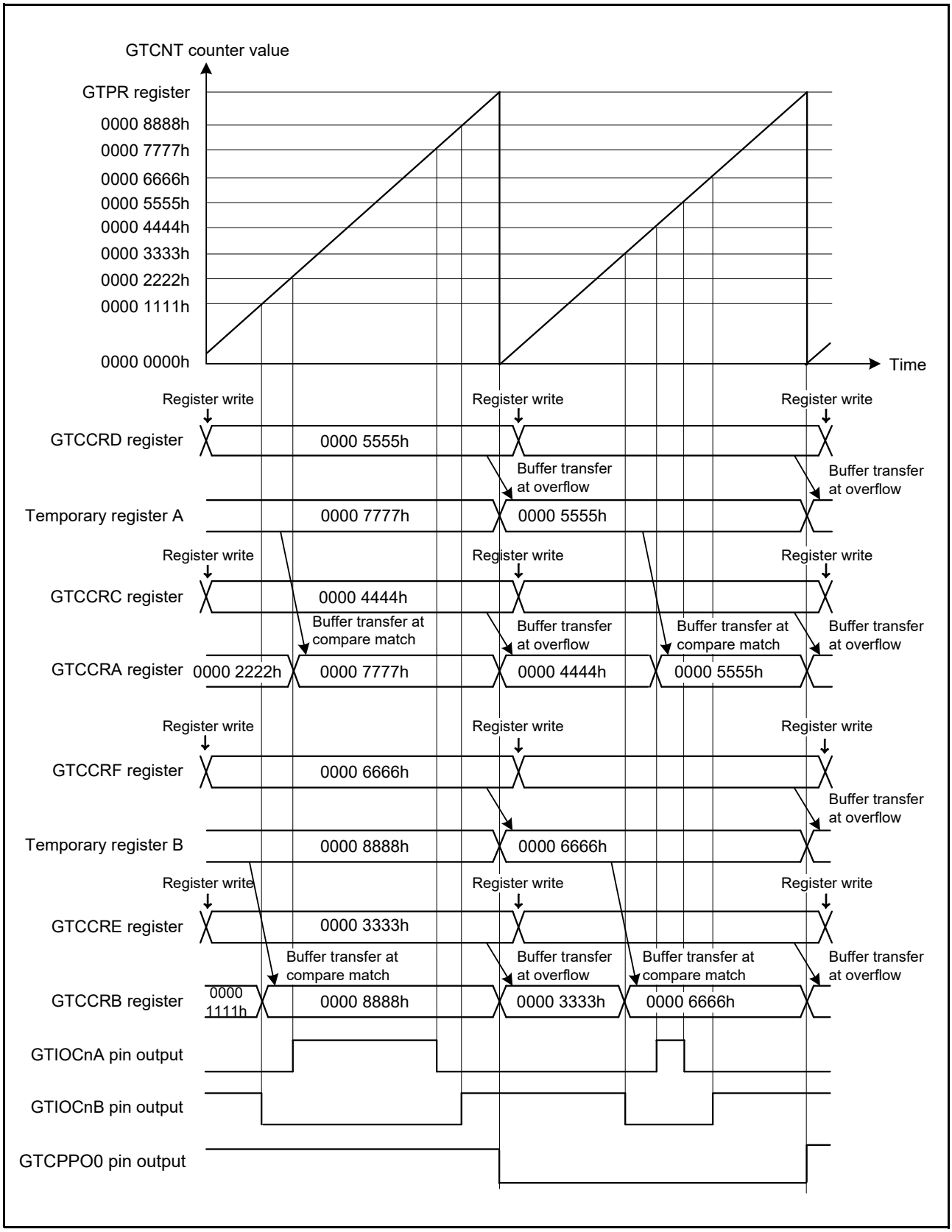


Figure 22.50 Example of Sawtooth-Wave One-Shot Pulse Mode Operation (Up-Counting, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle (n = 0 to 7), GPTW0.GTIOR.PSYE bit = 1 (GPTW0 only))

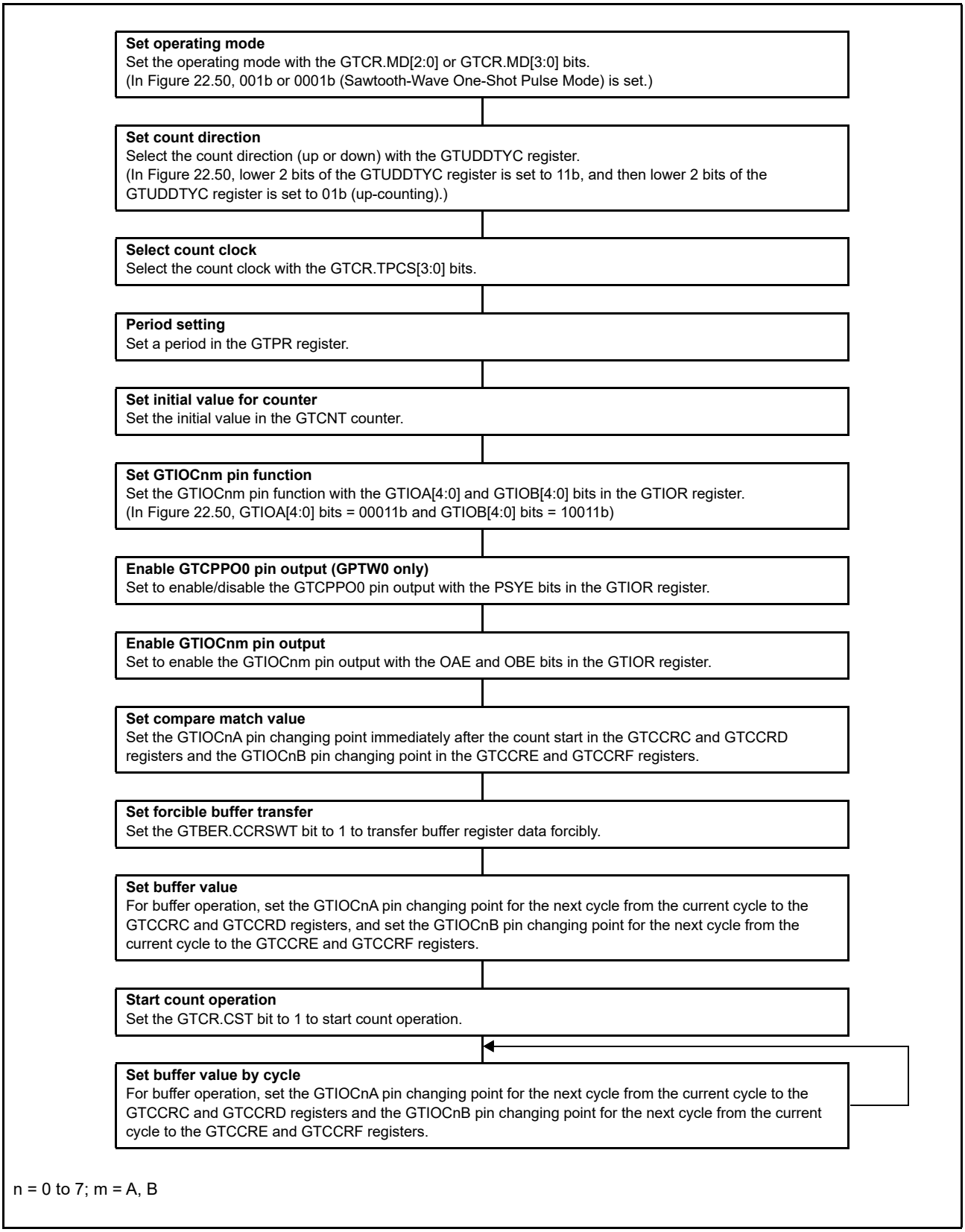


Figure 22.51 Example for Setting Sawtooth-Wave One-Shot Pulse Mode

(4) Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs (n = 0 to 7).

Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR and GTDVU registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 22.52 shows an example of triangle-wave PWM mode 1 operation, and Figure 22.53 shows an example for setting triangle-wave PWM mode 1.

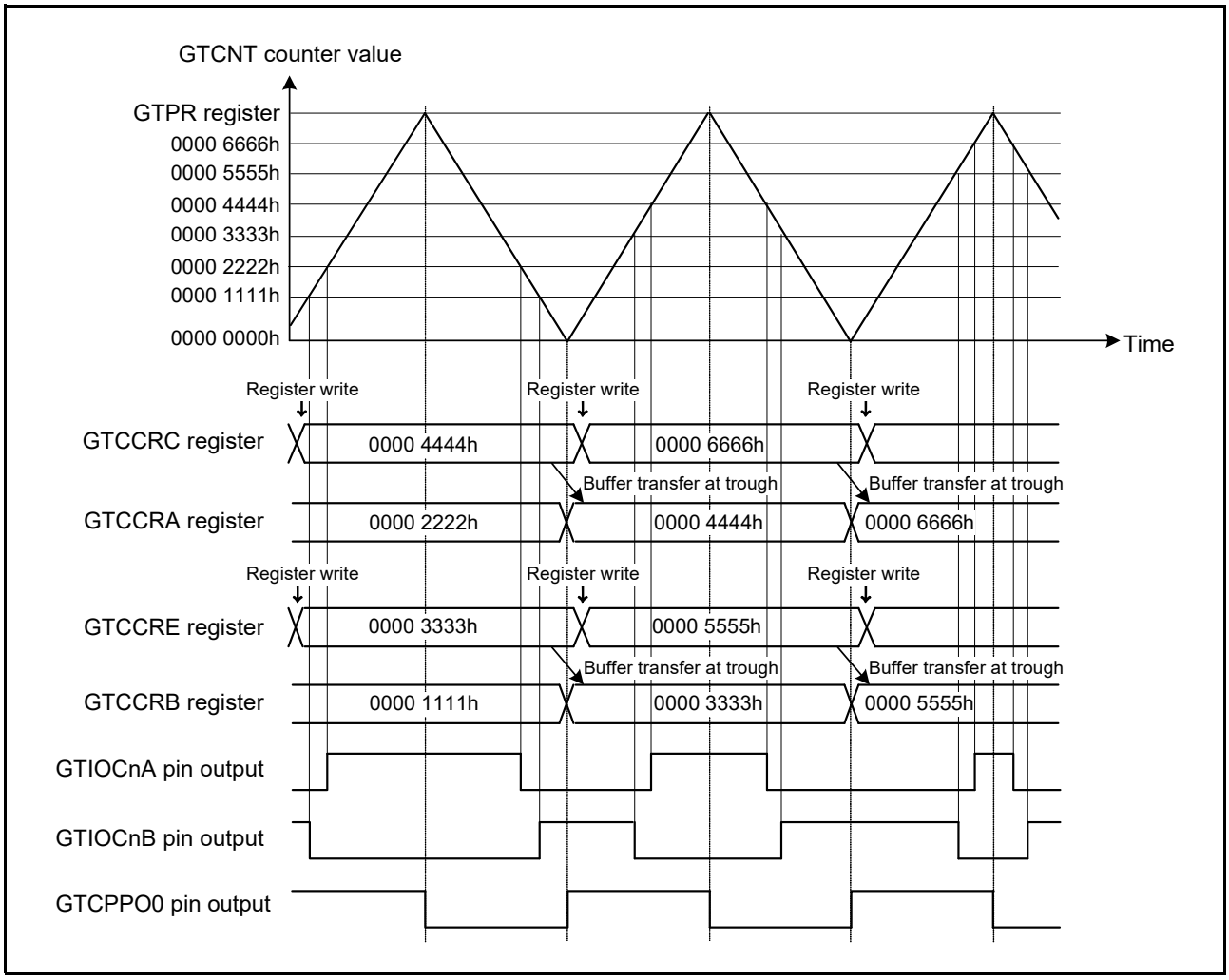


Figure 22.52 Example of Triangle-Wave PWM Mode 1 Operation (Buffer Operation, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle (n = 0 to 7), GPTW0.GTIOR.PSYE bit = 1 (GPTW0 only))

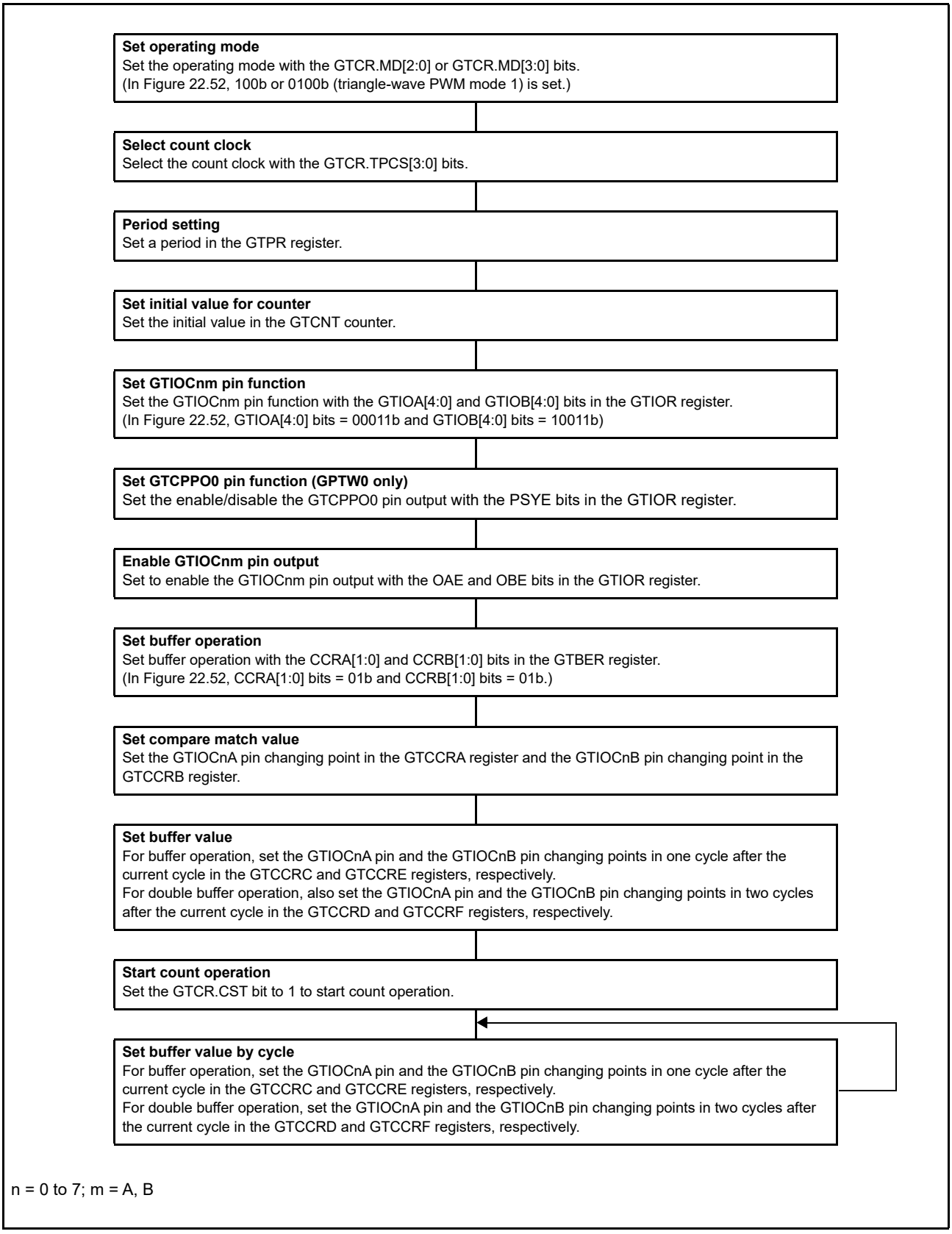


Figure 22.53 Example for Setting Triangle-Wave PWM Mode 1

(5) Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs (n = 0 to 7). The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
By setting the GTDTCR and GTDVU registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
Figure 22.54 shows an example of triangle-wave PWM mode 2 operation, and Figure 22.55 shows an example for setting triangle-wave PWM mode 2.

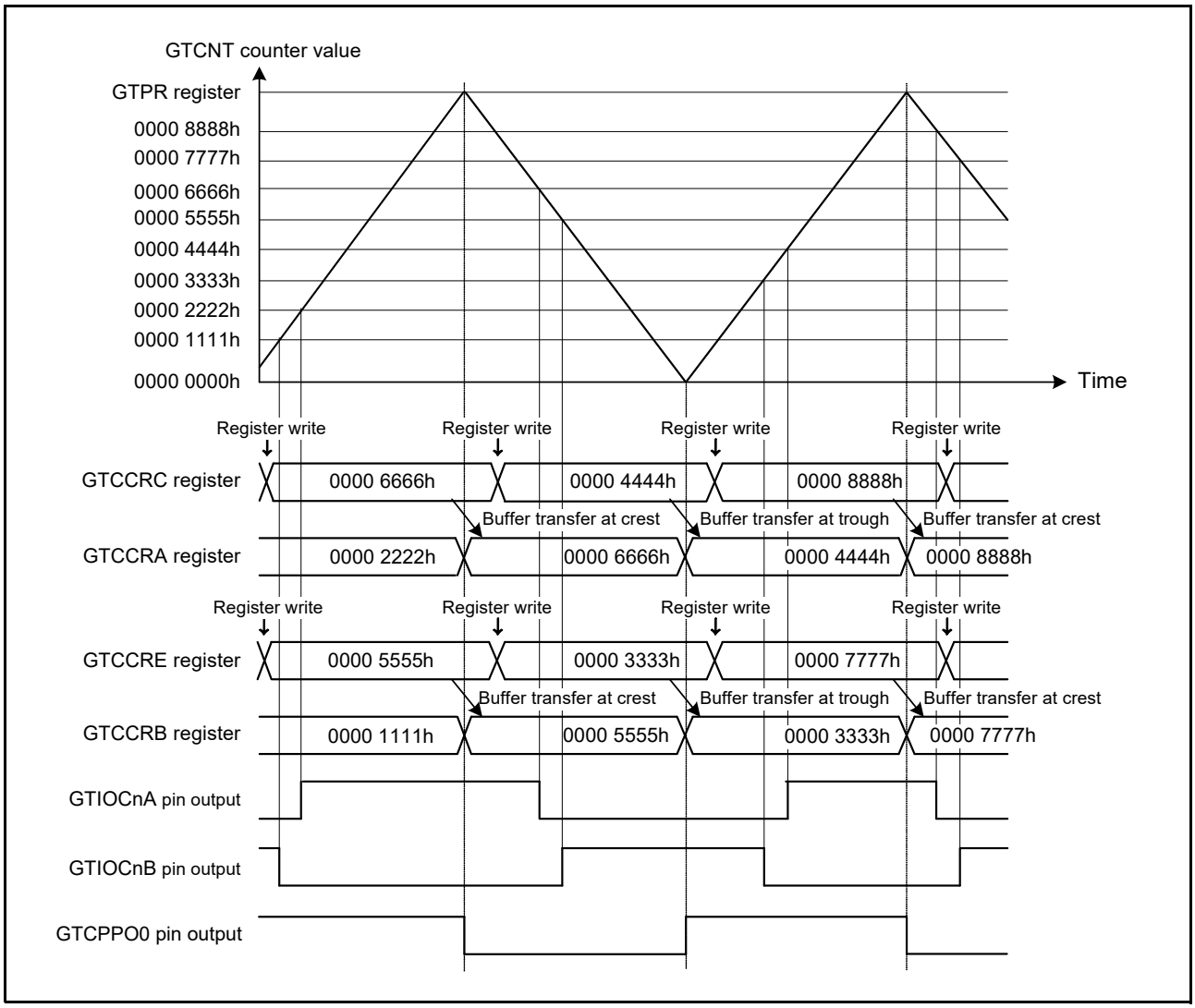


Figure 22.54 Example of Triangle-Wave PWM Mode 2 Operation (Buffer Operation, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle (n = 0 to 7), GPTW0.GTIOR.PSYE bit = 1 (GPTW0 only))

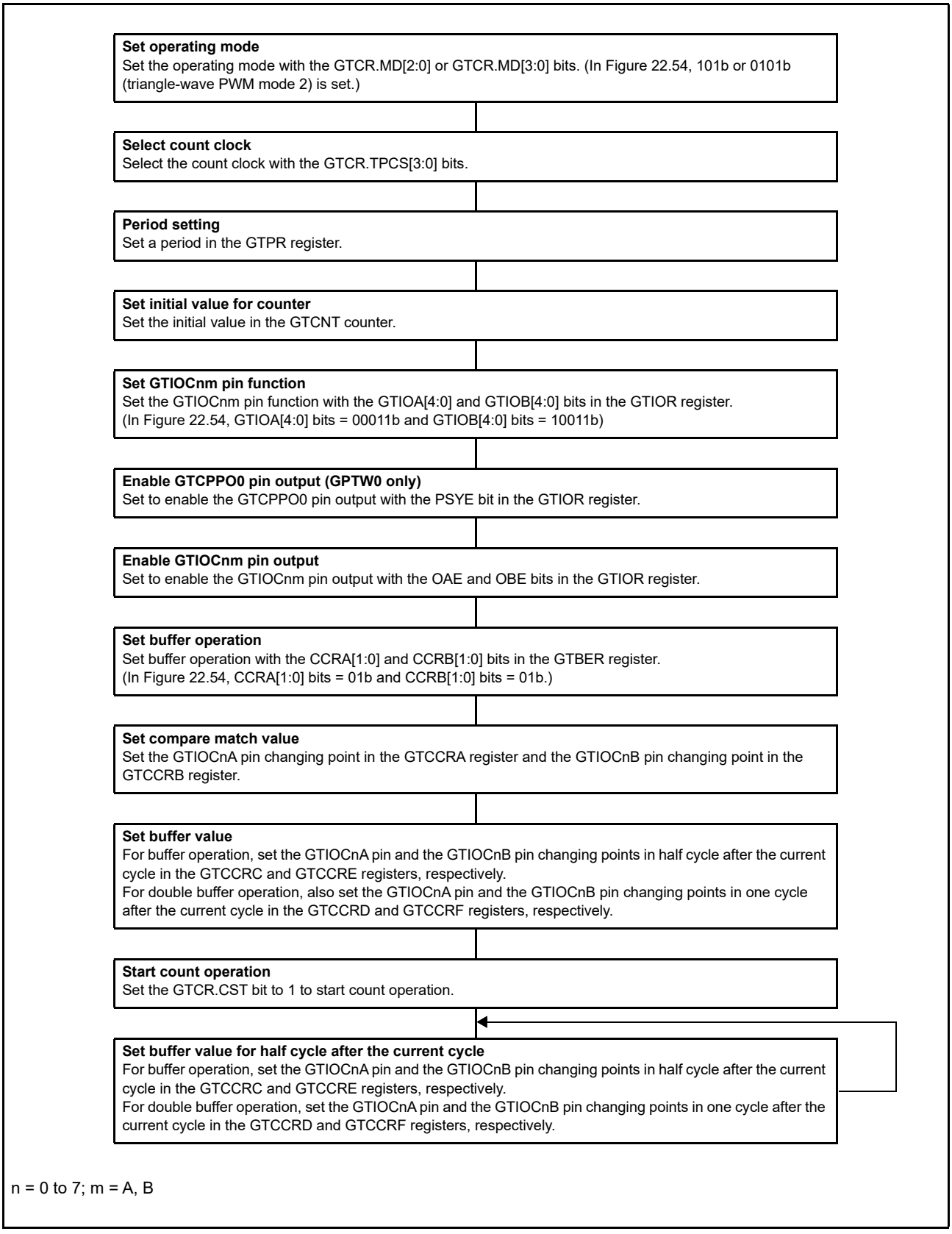


Figure 22.55 Example for Setting Triangle-Wave PWM Mode 2

(6) Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed (n = 0 to 7). Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to the temporary register A, and from the GTCCRF register to the temporary register B at the trough, and from the temporary register A to the GTCCRA register and from the temporary register B to the GTCCRB register at the crest. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR and GTDVU registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 22.56 shows an example of triangle-wave PWM mode 3 operation, and Figure 22.57 shows an example for setting triangle-wave PWM mode 3.

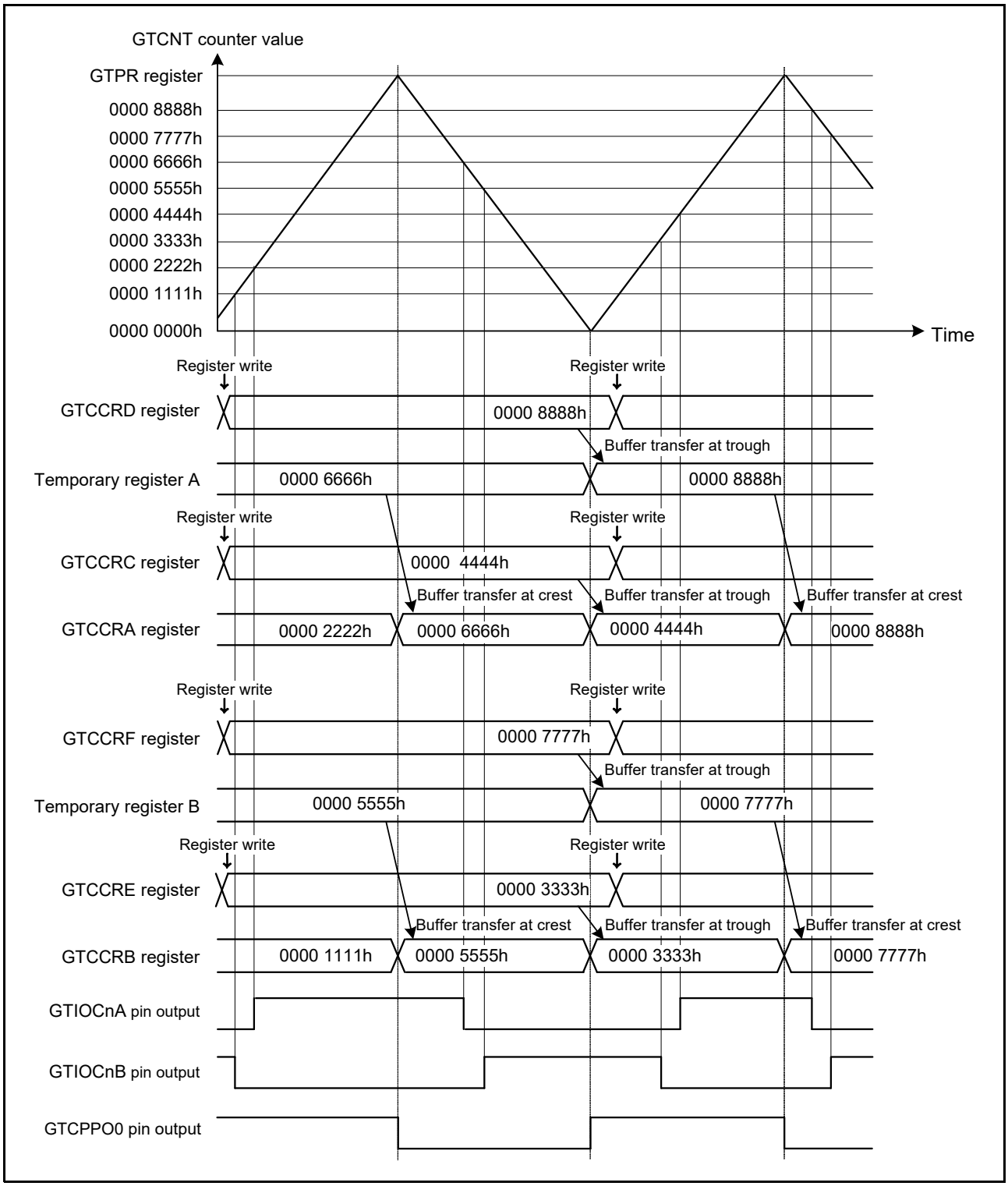


Figure 22.56 Example of Triangle-Wave PWM Mode 3 Operation (Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle (n = 0 to 7), GPTW0.GTIOR.PSYE bit = 1 (GPTW0 only))

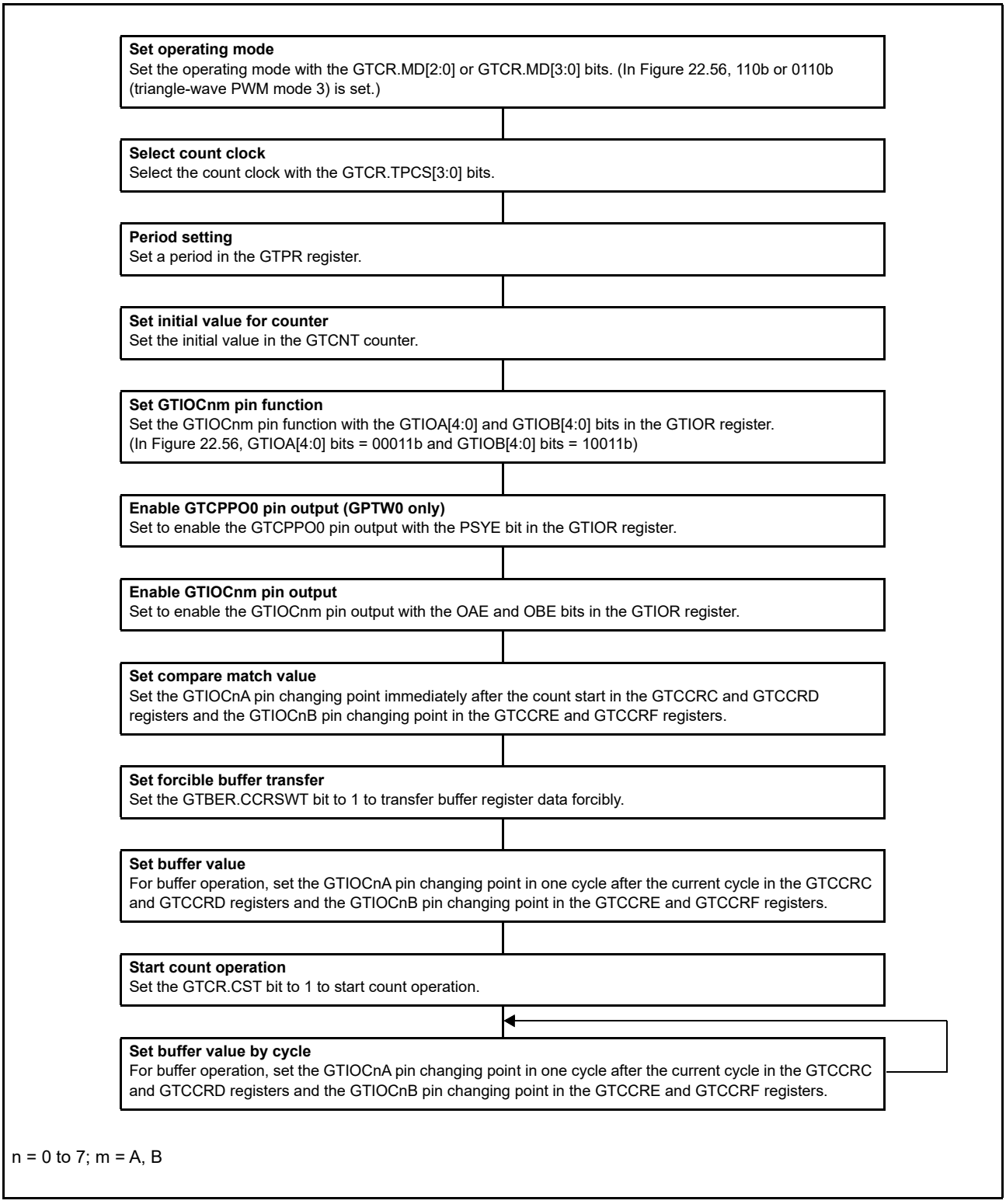


Figure 22.57 Example for Setting Triangle-Wave PWM Mode 3

(7) Complementary PWM mode 1, 2, 3

In complementary PWM mode, a three-phase PWM waveform with dead time that ensures the linearity in the vicinity of duty 0% and 100% can be output using the GTCNT counter of consecutive three channels. There are four modes depending on differences in buffer operation: (1) complementary PWM mode 1 (transfer at crests), (2) complementary PWM mode 2 (transfer at troughs), (3) complementary PWM mode 3 (transfer at crests and troughs), and (4) complementary PWM mode 4 (immediate transfer).

Figure 22.58 shows the block diagram in complementary PWM modes 1 to 3.

Among consecutive three channels, the lowest channel is referred to as master channel, and the adjacent upper two channels are referred to as slave channel 1 (lower) and slave channel 2 (upper).

The GTCNT counter of each channel performs individual count operation under the cycle operation by the master channel.

In each channel, compare match with the GTCCRA register is performed selecting one of the three GTCNT counters in each operation section, and a positive-phase waveform and a negative-phase waveform are output from the GTIOCnA pin (n = 0 to 2) and the GTIOCnB pin (n = 0 to 2) respectively with a non-overlapping section of the dead time value set in the GTDVU register of the master channel.

The GTCCRA register performs buffer operation by the GTCCRC register, temporary register A, and GTCCRD register.

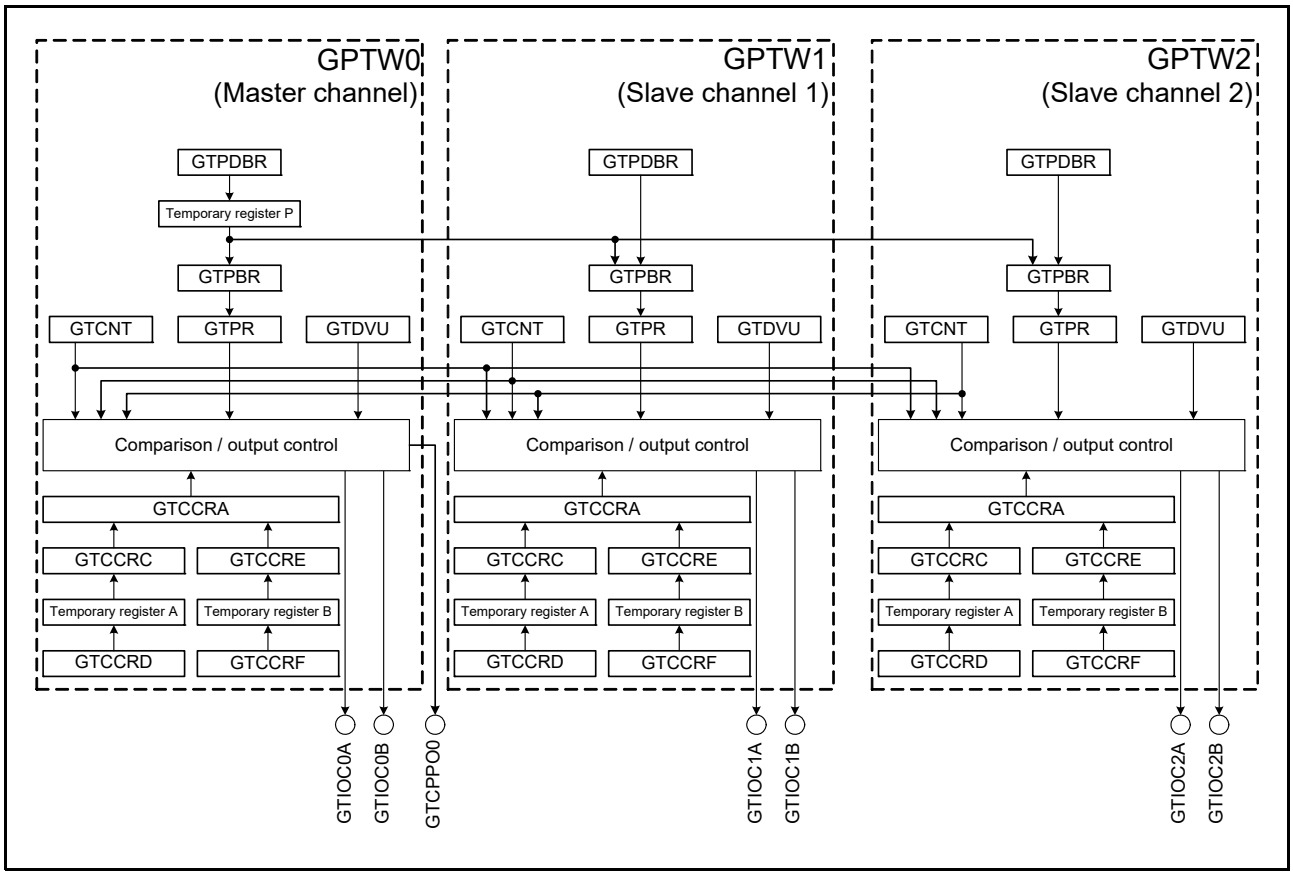


Figure 22.58 Block Diagram in Complementary PWM Mode 1, 2, 3

The GPTW0.GTCNT counter performs count operation for triangle waves using the GPTW0.GTPR register as a cycle register. A section where the GPTW0.GTCNT counter value is not larger than the dead time value is referred to as trough section.

The GPTW1.GTCNT counter performs count operation with the value (GPTW0.GTCNT counter value + dead time value set in the GPTW0.GTDVU register). A section where the GPTW1.GTCNT counter value is larger than the GPTW0.GTPR register value is referred to as crest section.

Crest section and trough section are classified into up-counting crest section, down-counting crest section, up-counting

trough section, and down-counting trough section according to counting direction. A section between trough section and crest section is referred to as up-counting middle section or down-counting middle section according to counting direction.
A section equivalent to the up-counting trough section after starting count operation is referred to as initial output section where operation differs partially from other up-counting trough sections.
The GPTW2.GTCNT counter functions as a counter to ensure the linearity in the vicinity of duty 0% and 100%. In a crest section, this counter performs count operation for a triangle wave (up-counting after down-counting) with the value (GPTW0.GTPR register value + dead time value) as an initial value and the GPTW0.GTPR register value as a trough. This counter is cleared to 0 at the end of the crest section, and then stops counting until the next trough section. In a trough section, this counter performs count operation for a triangle wave with an initial value of 0 and dead time value as a crest.
This counter becomes the value (GPTW0.GTPR register value + dead time value) at the end of the trough section, and then stops counting until the next crest section. In the initial output section, however, this counter counts up with an initial value of 0 until the dead time value, and then becomes the value (GPTW0.GTPR register value + dead time value).
If the counter stops and then restarts in complementary PWM mode, the counter of each channel returns to the initial value after starting count operation, and then starts counting from the initial output section.
Table 22.7 and Table 22.8 show count operation (counting direction/counting range) in each section. In these tables, registers with no channel identification indicate that the same value is stored in them of the master channel, slave channel 1, and slave channel 2.

Table 22.7 Count Operation in Complementary PWM Mode (1)

Counter	Initial Value	Initial Output Section (After Start)	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section
GPTW0.GTCNT	0	Up-Counting 0 to GTDVU	Up-Counting GTDVU + 1 to GTPR – GTDVU	Up-Counting GTPR – GTDVU + 1 to GTPR	Down-Counting GTPR – 1 to GTPR – GTDVU
GPTW1.GTCNT	GTDVU	Up-Counting GTDVU to GTDVU × 2	Up-Counting GTDVU × 2 + 1 to GTPR	Up-Counting GTPR + 1 to GTPR + GTDVU	Down-Counting GTPR + GTDVU – 1 to GTPR
GPTW2.GTCNT	0	Up-Counting 0 to GTDVU	Stop GTPR + GTDVU	Down-Counting GTPR + GTDVU – 1 to GTPR	Up-Counting GTPR + 1 to GTPR + GTDVU

Table 22.8 Count Operation in Complementary PWM Mode (2)

Counter	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
GPTW0.GTCNT	Down-Counting GTPR – GTDVU – 1 to GTDVU	Down-Counting GTDVU – 1 to 0	Up-Counting 1 to GTDVU
GPTW1.GTCNT	Down-Counting GTPR – 1 to GTDVU × 2	Down-Counting GTDVU × 2 – 1 to GTDVU	Down-Counting GTDVU + 1 to GTDVU × 2
GPTW2.GTCNT	Stop 0	Up-Counting 1 to GTDVU	Down-Counting GTDVU – 1 to 0

In complementary PWM mode, the GTCCRA register buffer operation is different from normal buffer operation.
Data transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B are performed at the same time in three channels by writing a value to the GTCCRD register of the GPTW2 channel.
Data transfers from the temporary register A and temporary register B to the GTCCRC and GTCCRE registers vary depending on the transfer timing to the temporary register A and temporary register B. Data transfers from the GTCCRC and GTCCRE registers to the GTCCRA register are performed according to each complementary PWM mode name (crest transfer, trough transfer, and crest/trough transfer).
Buffer operation of the GTPR register in complementary PWM mode is described in section 22.3.2.1, GTPR Register Buffer Operation. Do not perform buffer operation for the GTDVU register in complementary PWM mode.
Table 22.9 shows buffer transfer timing during single buffer operation in complementary PWM modes 1 to 3.

Table 22.9 Single Buffer Transfer Timing in Complementary PWM Mode 1, 2, 3

Buffer Transfer	Complementary PWM Mode 1	Complementary PWM Mode 2	Complementary PWM Mode 3
GTCCRD ↓ Temporary register A	After one PCLKA cycle from the GTCCRD register write of slave channel 2 (GPTW2)	After one PCLKA cycle from the GTCCRD register write of slave channel 2 (GPTW2)	After one PCLKA cycle from the GTCCRD register write of slave channel 2 (GPTW2)
Temporary register A ↓ GTCCRC	(1) When transferring to the temporary register A at the middle section of the up-count • After 1 PCLKA after for transferred to the temporary register A (2) When transferring to the temporary register A at other than the middle section of the up-count • The end of the trough	(1) When transferring to the temporary register A at the middle section of the down-count • After 1 PCLKA after for transferred to the temporary register A (2) When transferring to the temporary register A at other than the middle section of the down-count • The end of the crest	(1) When transferring to the temporary register A at the middle section • After 1 PCLKA after for transferred to the temporary register A (2) When transferring to the temporary register A at other than the middle section • The end of the crest/trough
GTCCRC ↓ GTCCRA	• The end of the trough • Counter clear in the up-count intermediate section and crest section	• The end of the trough section excluding the initial output section • Counter clear in the down-count intermediate section and trough section	• The end of the trough • The end of the trough section excluding the initial output section • Counter clear

The change in the output level of the positive phase waveform output from the GTIOCnA pin (n = 0 to 2) and the negative phase waveform output from the GTIOCnB pin (n = 0 to 2) is determined by the compare match by the combination of the counter and register in each operation section. In the middle section, the compare match between the GPTW0.GTCNT counter and the GTCCRA register changes the positive phase waveform output level, and the compare match between the GPTW1.GTCNT counter and the GTCCRA register changes the negative phase waveform output level. The crest and trough sections use the GPTW2.GTCNT counter, GTCCRC register, and GTCCRE register to perform a compare match operation to ensure linearity near 0% and 100% duty.

When the comparison match value is greater than or equal to the GPTW0.GTPR register value, the duty is 0% (positive phase waveform OFF, negative phase waveform ON). When the comparison match value is 0, the duty is 100% (positive phase waveform ON, negative phase waveform OFF).

Table 22.10 shows the counter and register combinations used in the comparative match operation to generate a positive-phase waveform and a negative-phase waveform in each operating section.

Table 22.10 Combinations of Counters and Registers for Compare Match Operation in Complementary PWM Mode

	Up-Counting Middle Section	Up-Counting Crest Section	Up-Counting Crest Section	Up-Counting Middle Section	Up-Counting Trough Section	Up-Counting Trough Section
Negative-Phase OFF	GPTW1.GTCNT	GPTW1.GTCNT	—	GPTW2.GTCNT*1	GPTW2.GTCNT	GPTW1.GTCNT
	GTCCRA	GTCCRA	—	GTCCRC	GTCCRC	GTCCRC
Positive-Phase ON	GPTW0.GTCNT	GPTW0.GTCNT	GPTW2.GTCNT	—	GPTW2.GTCNT*1	GPTW2.GTCNT
	GTCCRA	GTCCRA	GTCCRA	—	GTCCRC	GTCCRC
Negative-Phase OFF	GPTW2.GTCNT*1	GPTW2.GTCNT	GPTW0.GTCNT	GPTW0.GTCNT	GPTW0.GTCNT	—
	GTCCRC	GTCCRC	GTCCRC	GTCCRA	GTCCRA	—
Positive-Phase ON	—	GPTW1.GTCNT*1	GPTW1.GTCNT	GPTW1.GTCNT	GPTW1.GTCNT	GPTW2.GTCNT
	—	GTCCRC	GTCCRC	GTCCRA	GTCCRA	GTCCRA

Note 1. Compare match is performed only at the time of final count in the target section, but is not performed at count values other than the final count.

In the case of normal complementary PWM mode waveform, a PWM waveform change occurs in the order of negative-phase OFF → positive-phase ON → positive-phase OFF → negative-phase ON. However, this order may vary depending on operation section and register values. In this case, OFF takes precedence in trough sections and ON takes precedence in crest sections (for negative-phase waveforms), and ON takes precedence in trough sections and OFF takes precedence in crest sections (for positive-phase waveforms). A lower-priority compare match that occurs at the same time or after a higher-priority compare match is ignored.

In the initial output section, the initial output set in the GTIOR register is retained. In the case that the GTCCRA register value is larger than the GTDVU register value at the end of the initial output section, negative phase is enabled. In the case that the GTCCRA register value is not larger than the GTDVU register value, positive phase is enabled.

As operation examples of normal complementary PWM mode waveform where compare match operation occurs in the middle section, Figure 22.59 and Figure 22.60 show complementary PWM mode 1, Figure 22.61 and Figure 22.62 show complementary PWM mode 2, Figure 22.63 and Figure 22.64 show single buffer complementary PWM mode 3 Figure 22.65 to Figure 22.76 show complementary PWM mode waveforms where compare match operation occurs in crest sections and trough sections and differences due to compare match occurrence order.

Figure 22.77 and Figure 22.78 show examples of initial output operation according to the GTCCRA register value.

Figure 22.79 shows an example for setting complementary PWM modes 1 to 3.

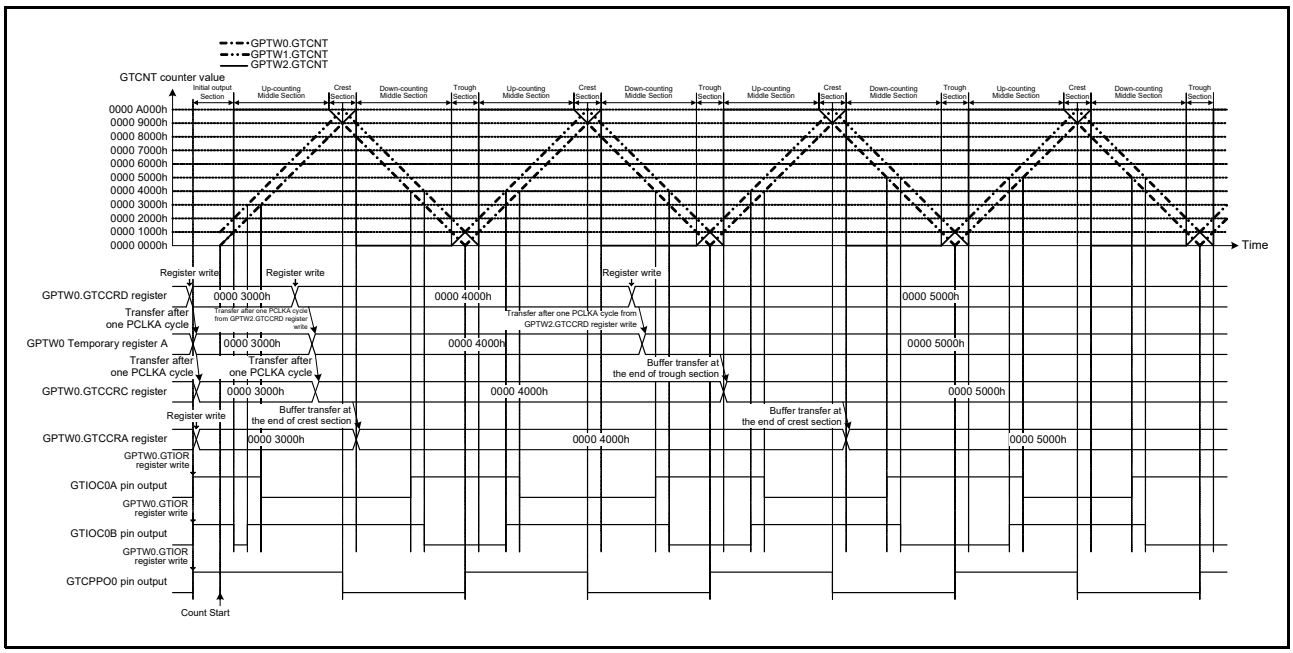


Figure 22.59 Example of Complementary PWM Mode 1 Operation (1)
(GTIOC0A pin = High/GTIOC0B pin = High as initial output,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section)

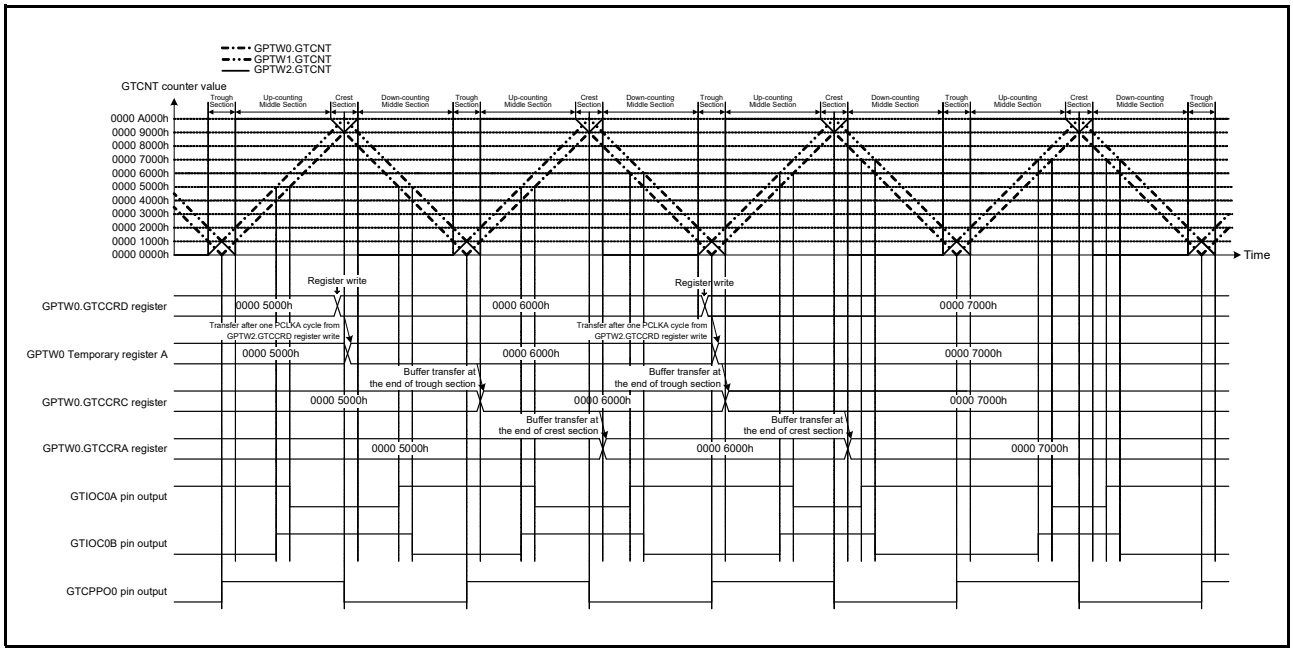


Figure 22.60 Example of Complementary PWM Mode 1 Operation (2)
(GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections)

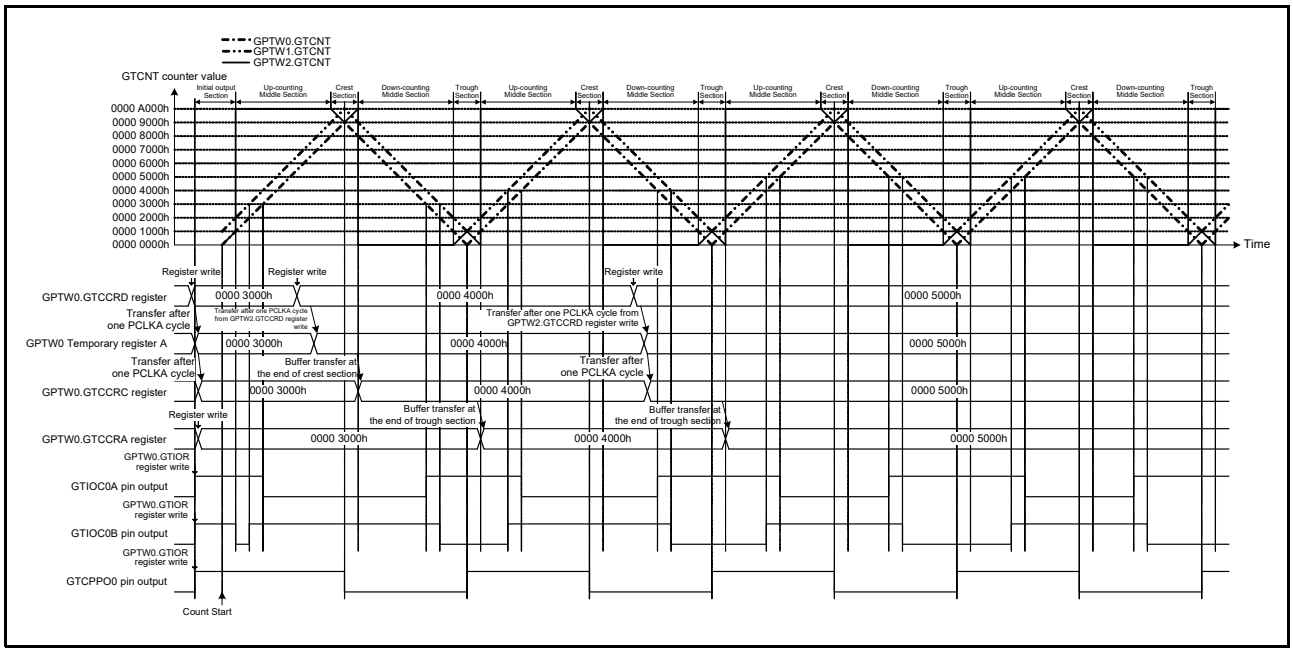


Figure 22.61 Example of Complementary PWM Mode 2 Operation (1)
(GTIOC0A pin = High/GTIOC0B pin = High as initial output,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section)

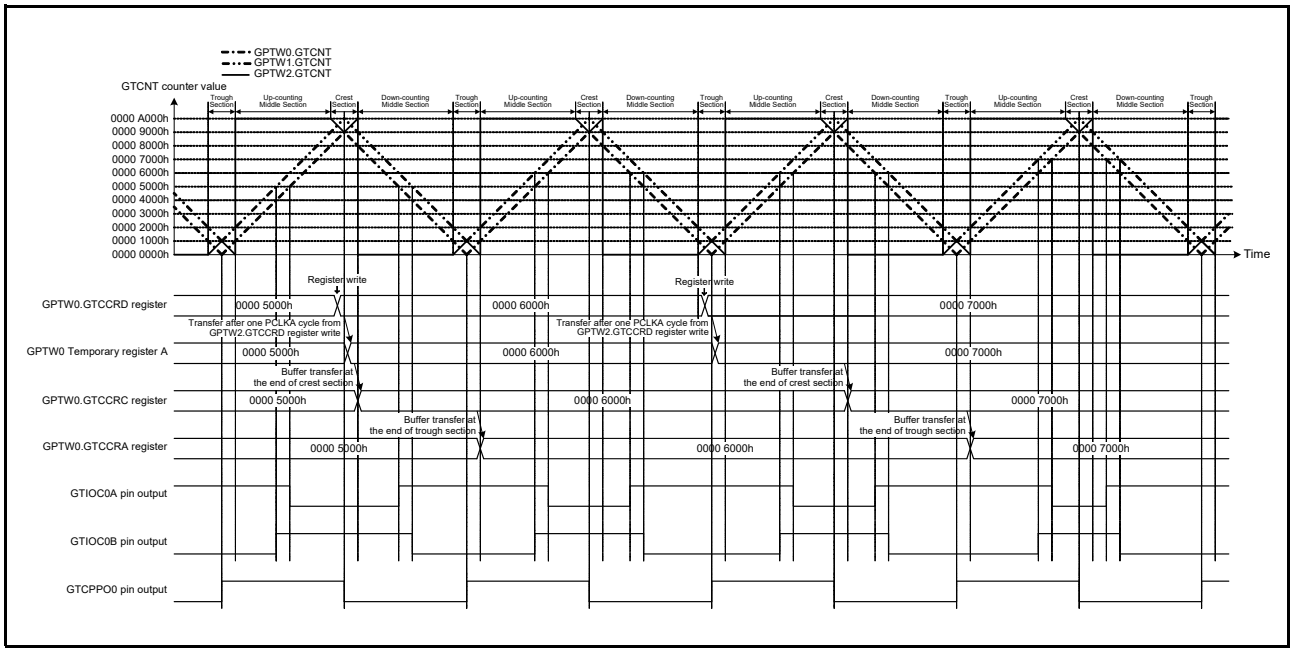


Figure 22.62 Example of Complementary PWM Mode 2 Operation (2)
(GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections)

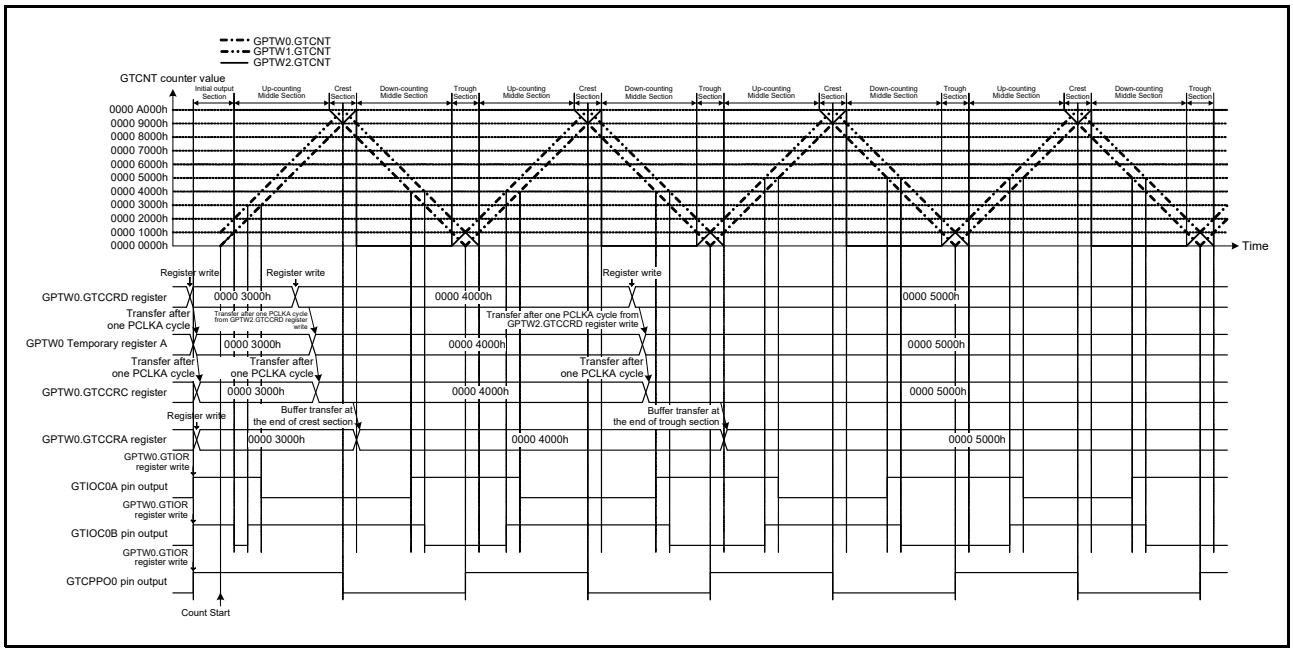


Figure 22.63 Example of Complementary PWM Mode 3 Operation (1)
(Single buffer operation, GTIOC0A pin = High/GTIOC0B pin = High as initial output,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section)

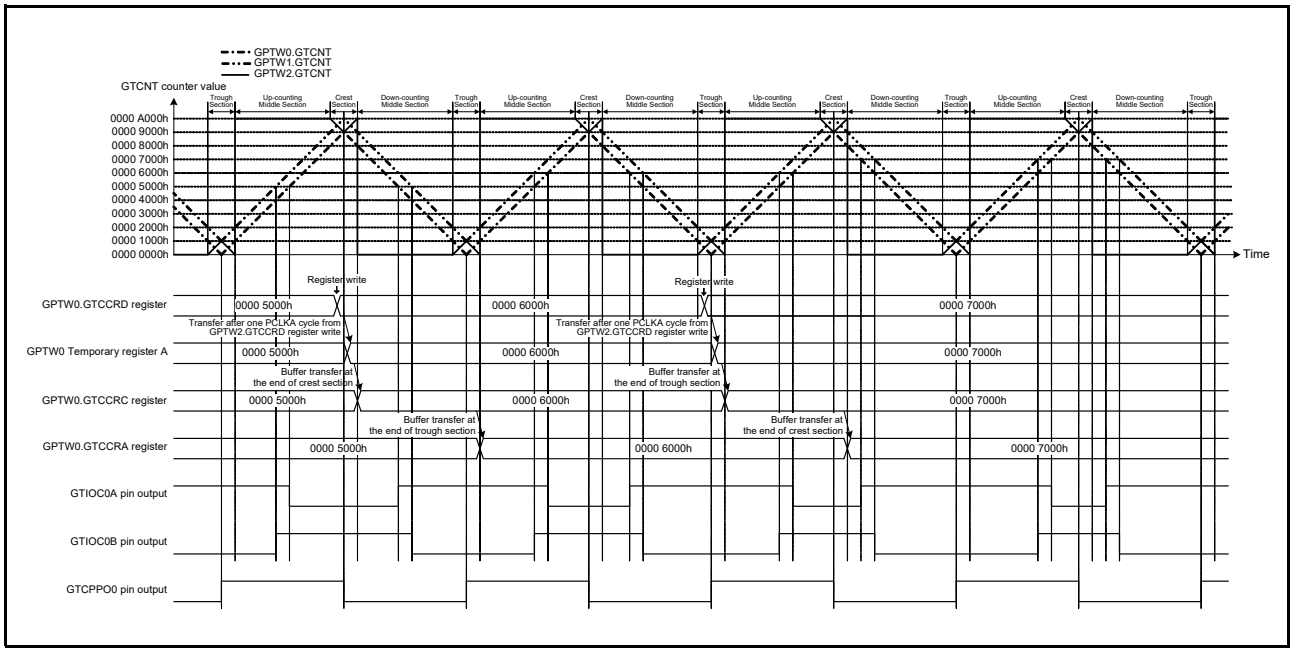


Figure 22.64 Example of Complementary PWM Mode 3 Operation (2)
(Single buffer operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections)

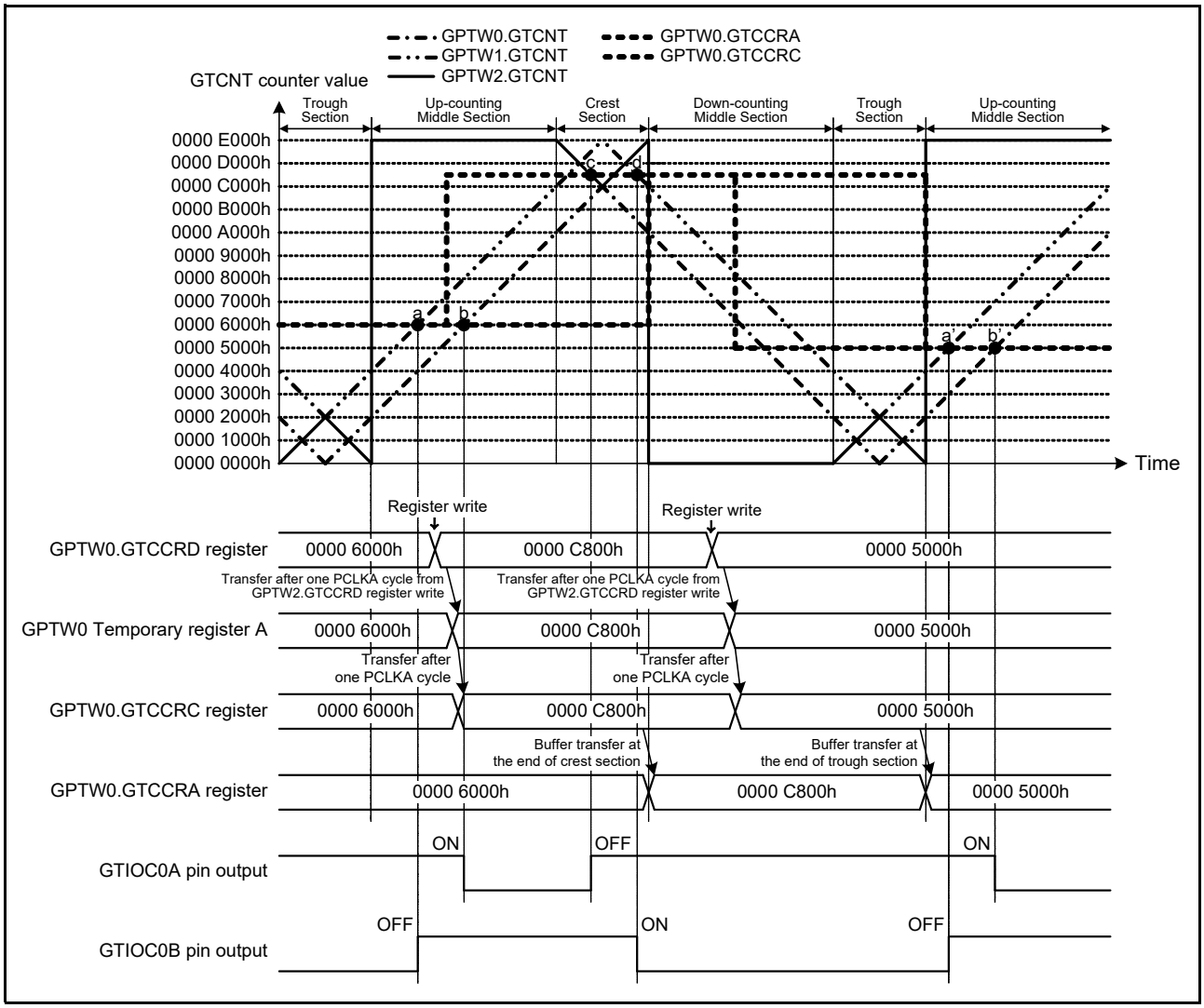


Figure 22.65 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (1) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → b → c → d)

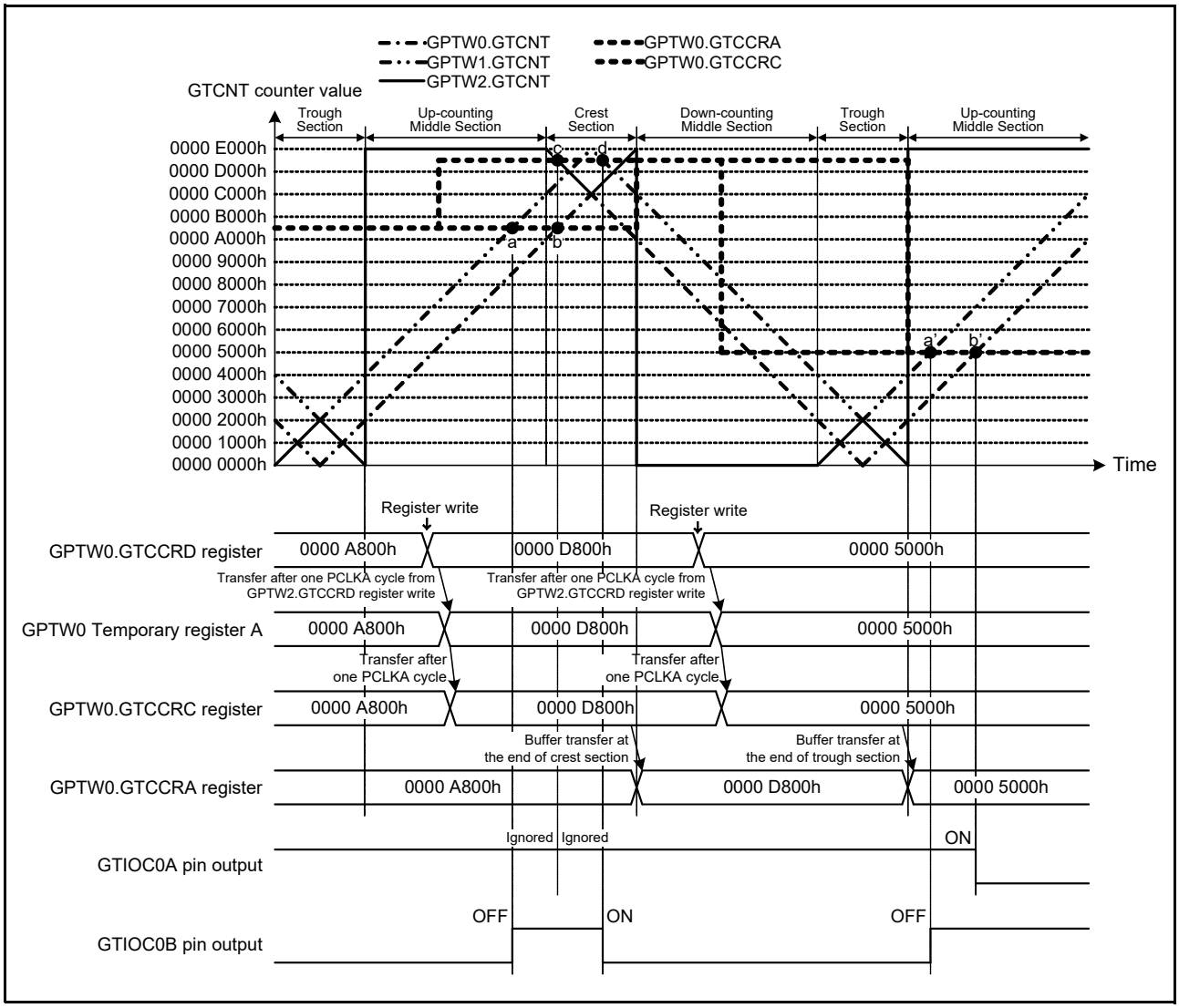


Figure 22.66 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (2) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → (b, c) → d)

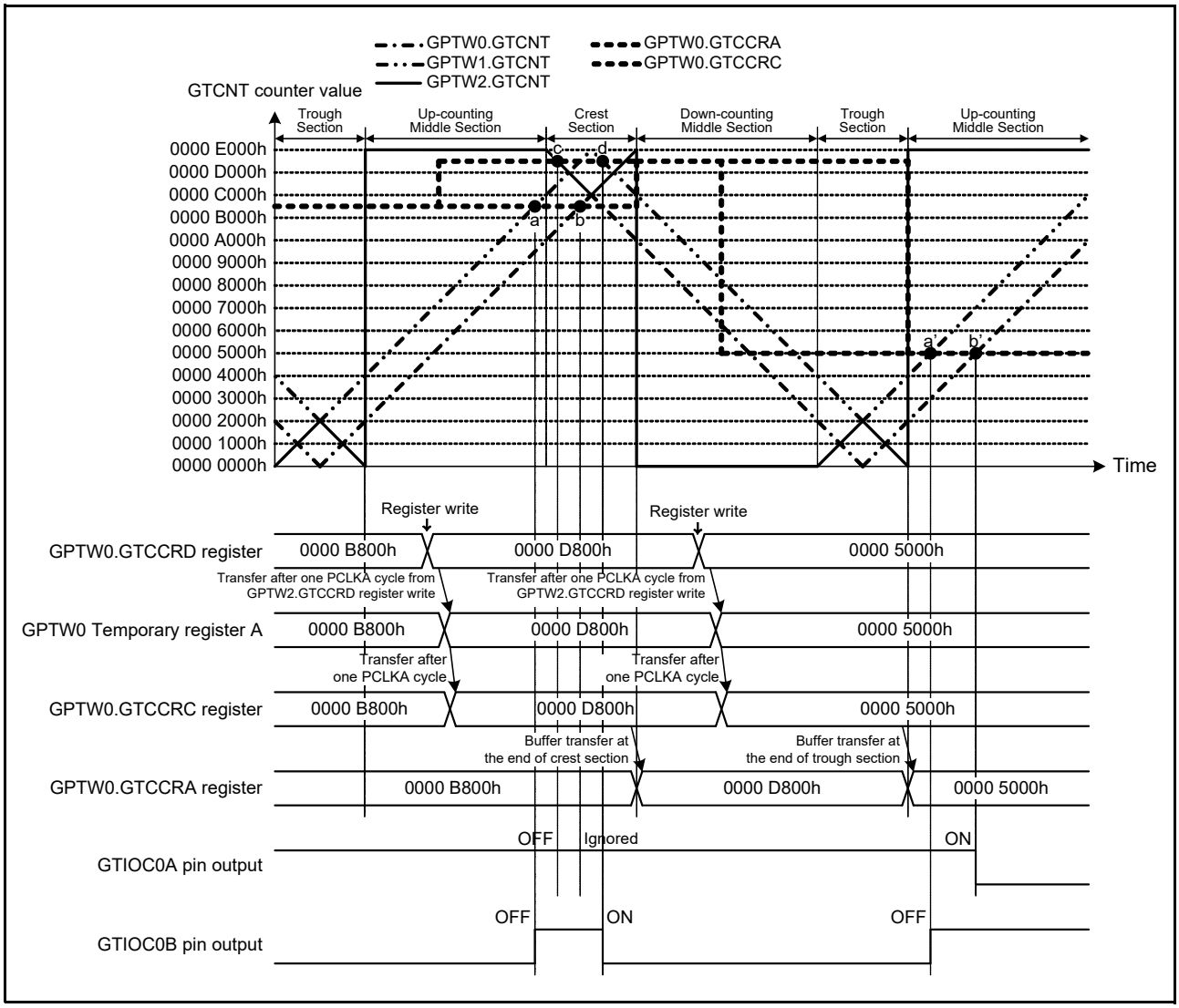


Figure 22.67 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (3) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → c → b → d)

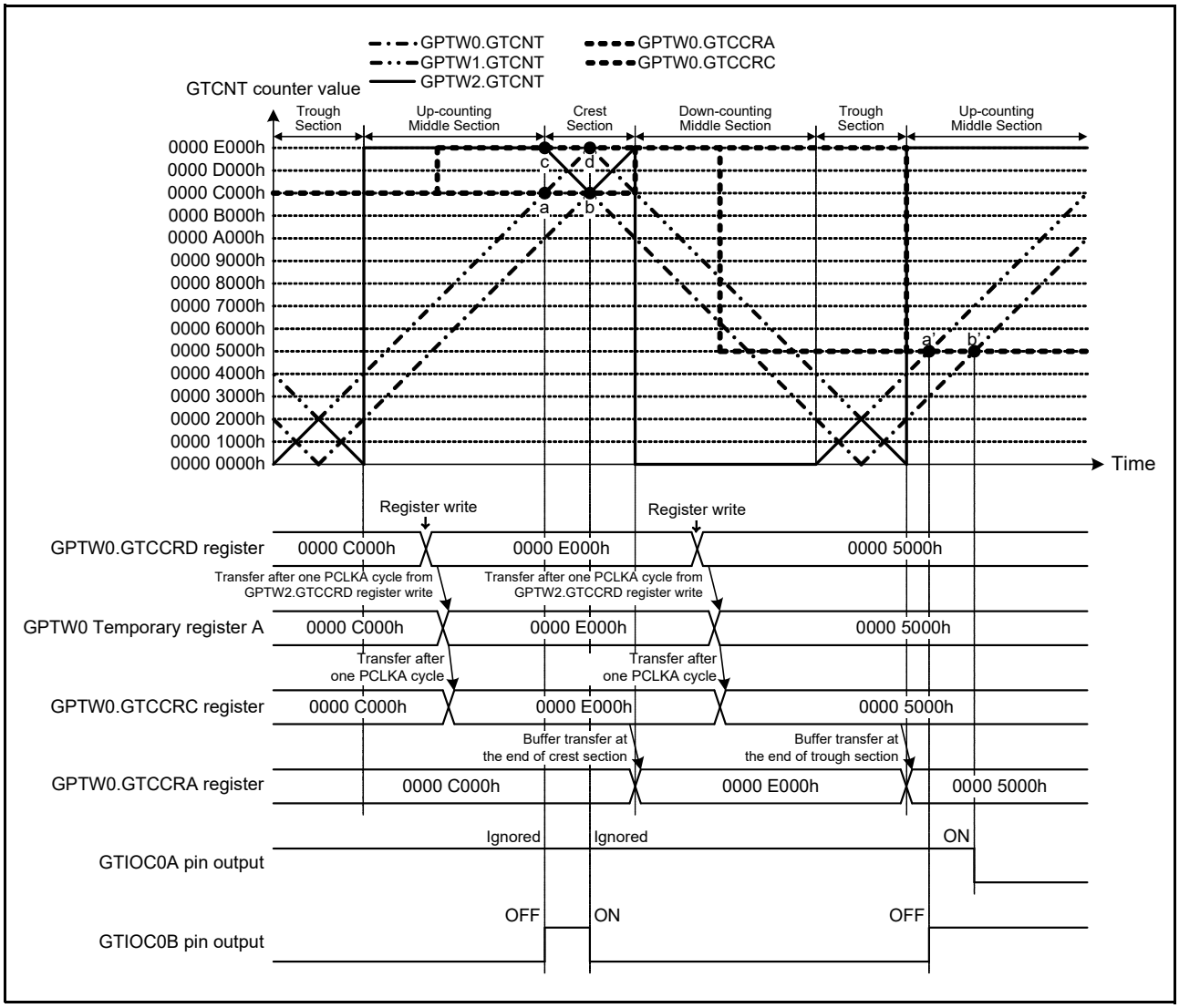


Figure 22.68 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (4) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: (a, c) → (b, d))

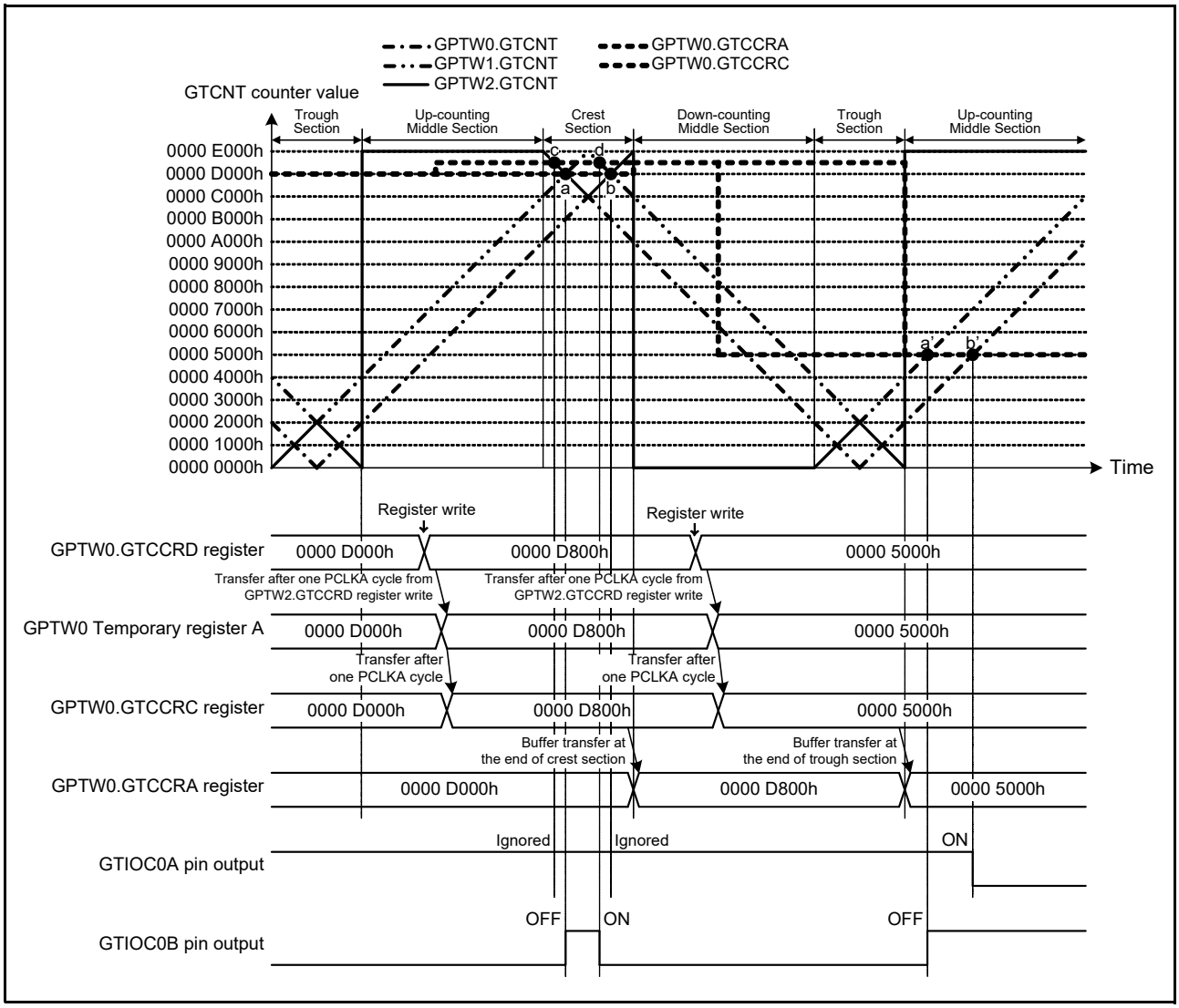


Figure 22.69 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (5) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → a → d → b)

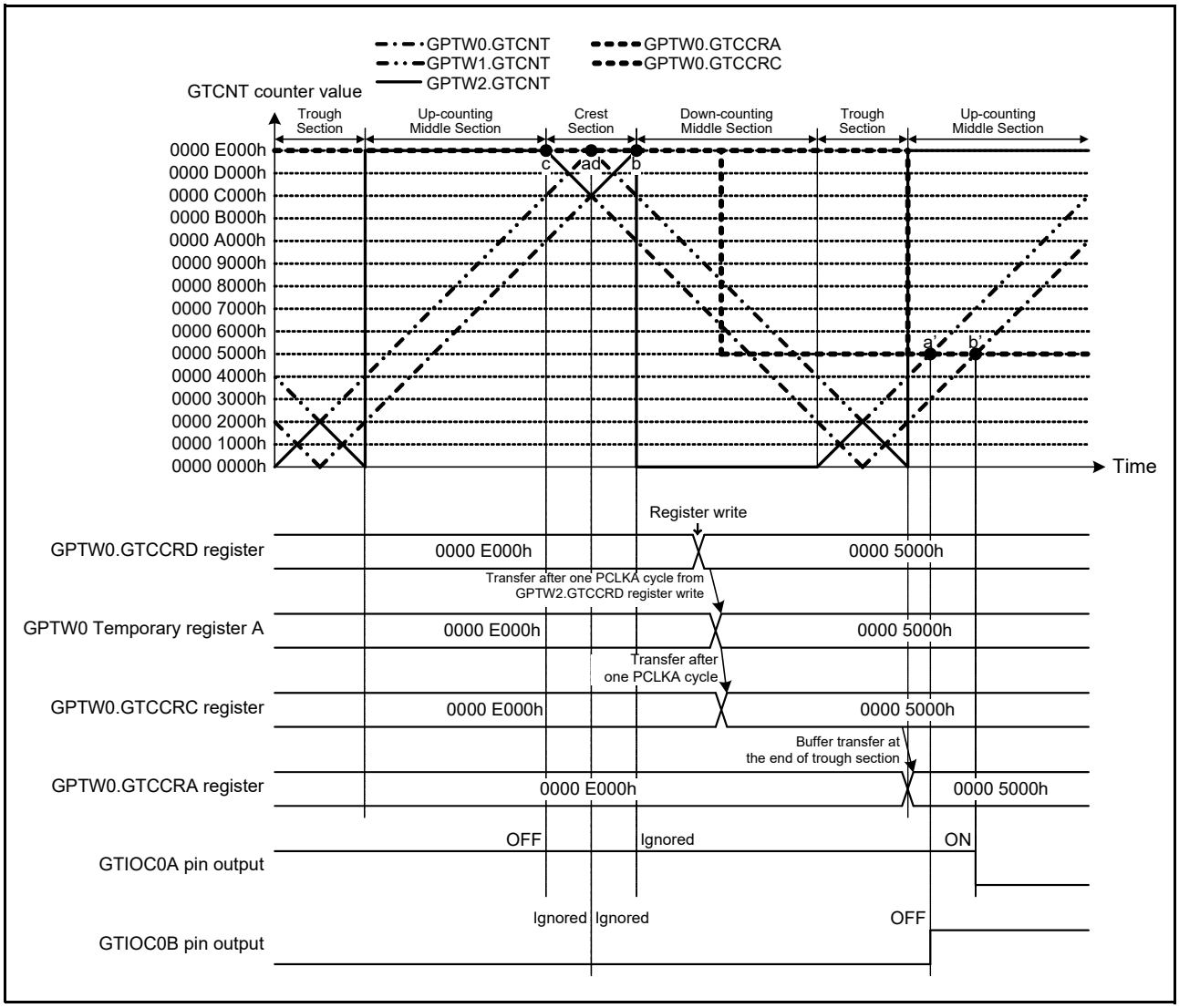


Figure 22.70 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (6) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → (a, d) → b)

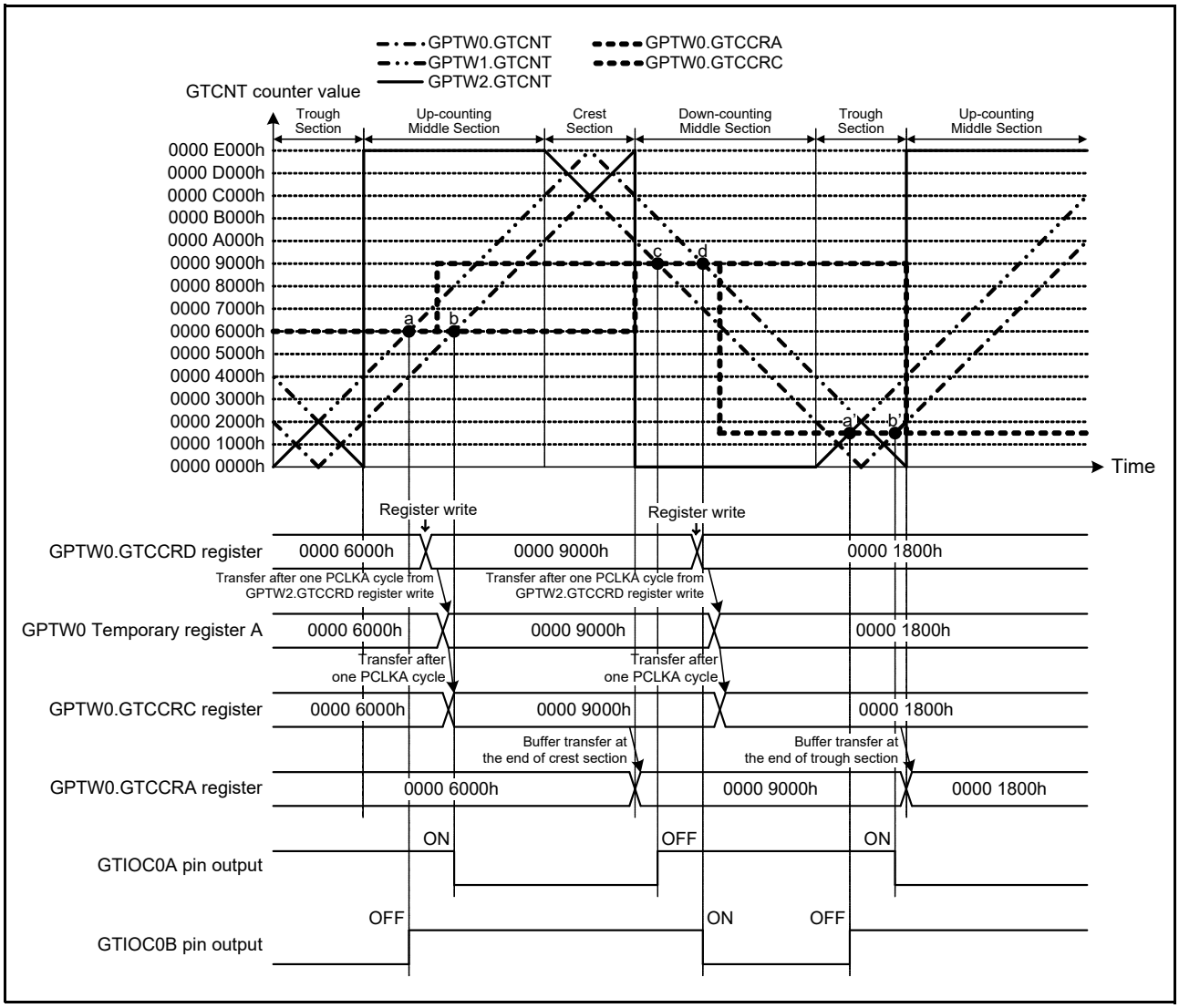


Figure 22.71 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (7) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → d → a' → b')

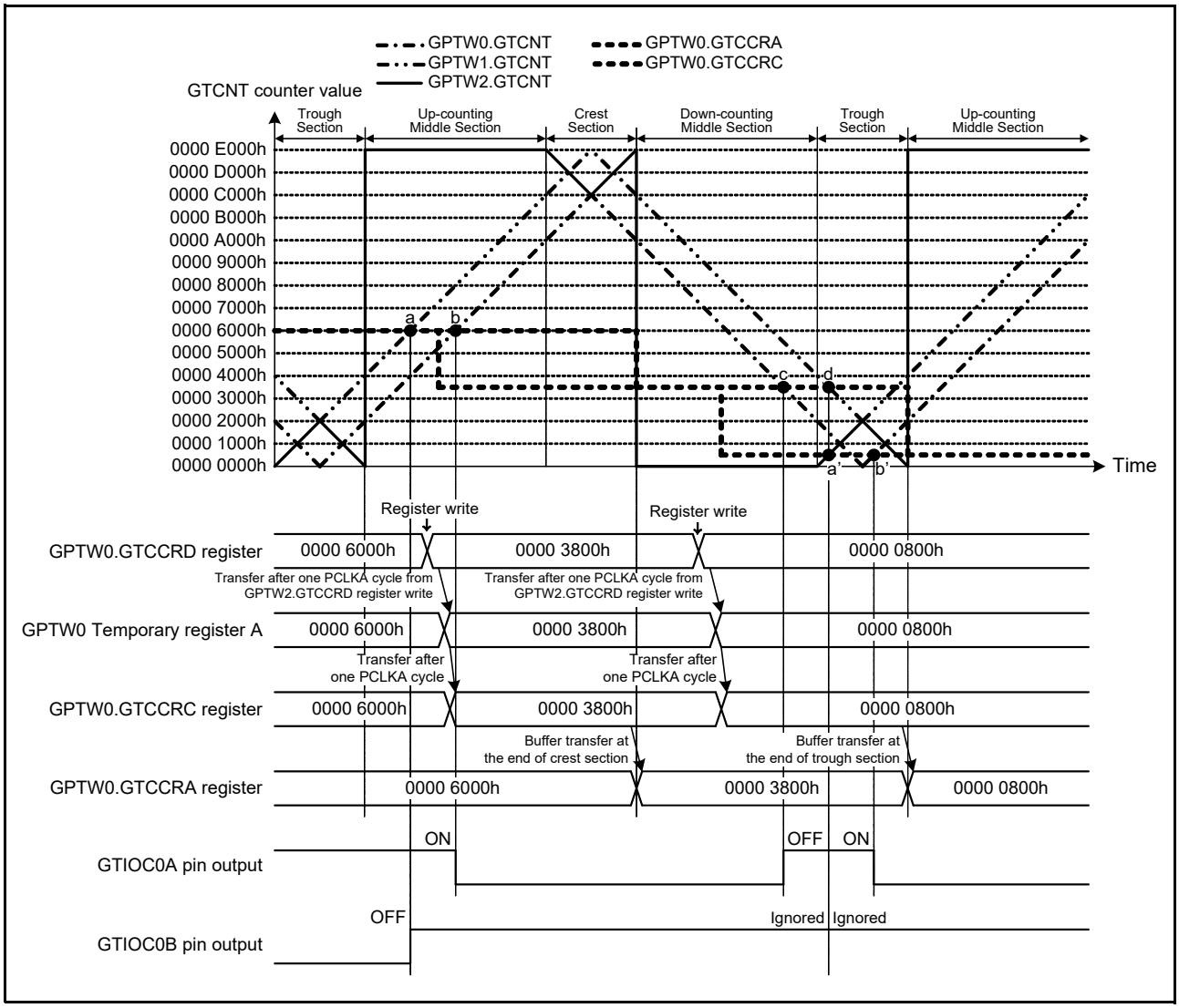


Figure 22.72 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (8) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → (d, a') → b')

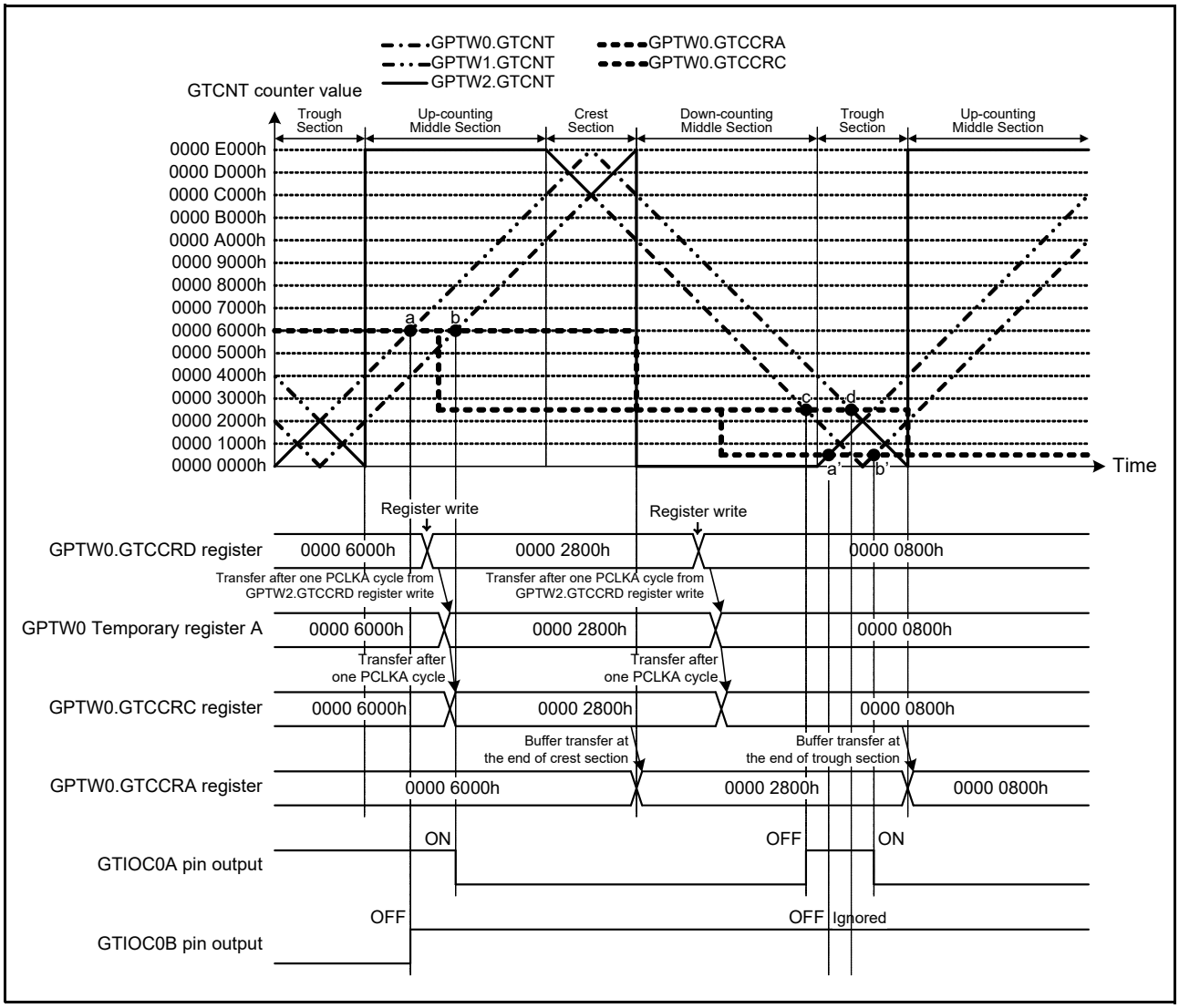


Figure 22.73 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (9) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → a' → d → b')

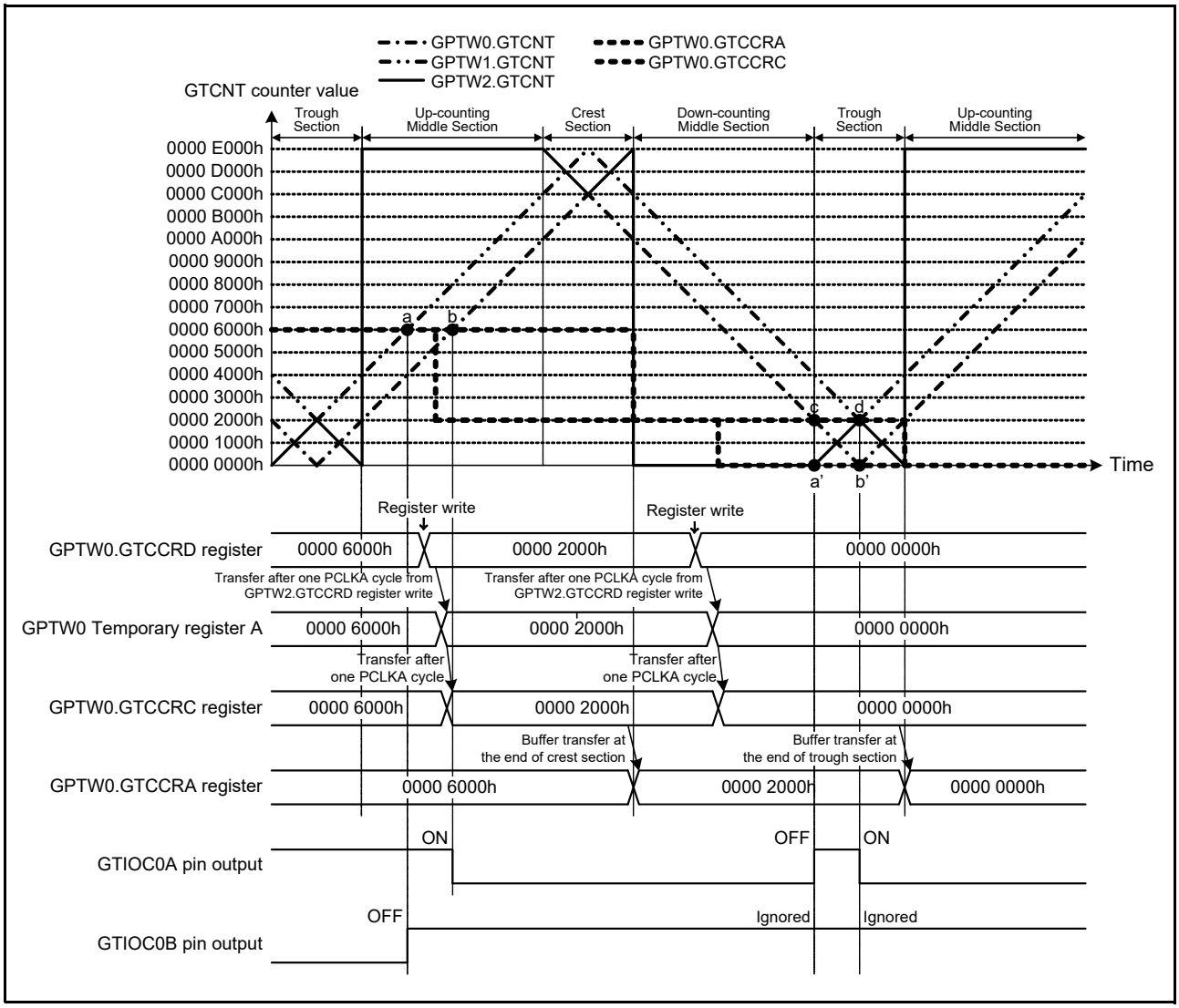


Figure 22.74 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (10) (Complementary PWM mode 3 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: (c, a') → (d, b'))

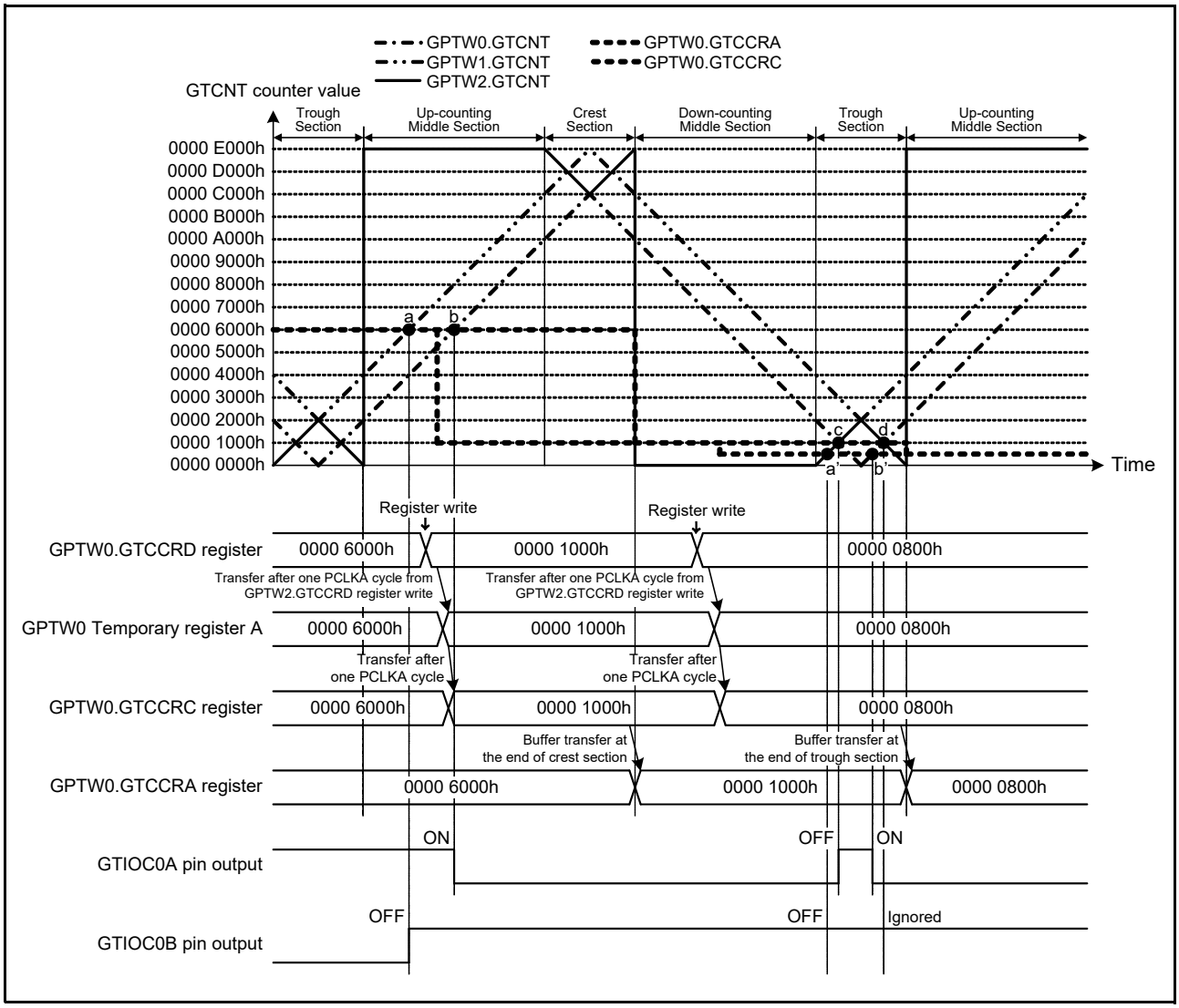


Figure 22.75 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (11)
(Complementary PWM mode 3 single buffer operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 2000h, compare match generation order: a' → c → b' → d)

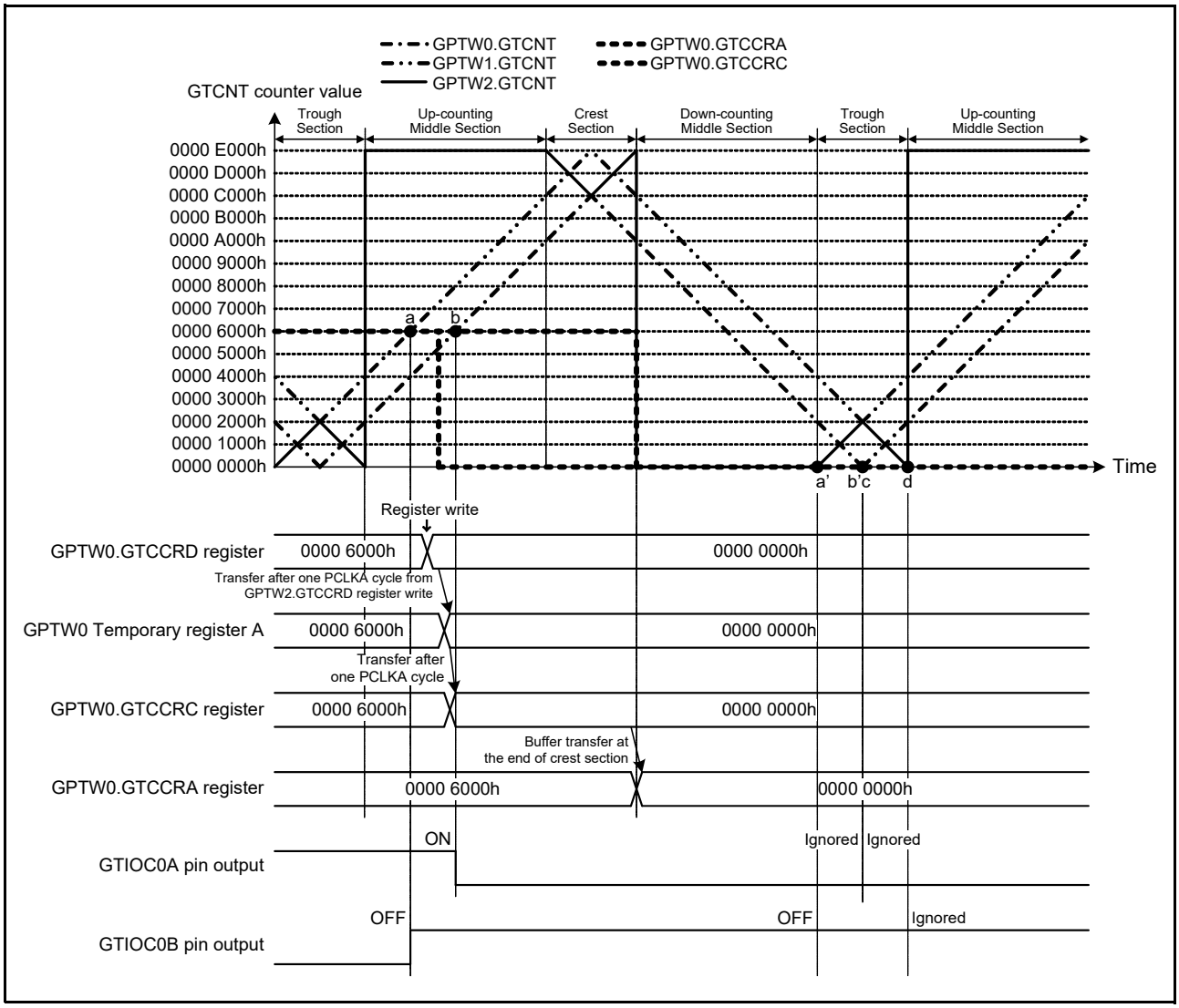


Figure 22.76 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (12)
(Complementary PWM mode 3 single buffer operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 2000h, compare match generation order: a' → (b', c) → d)

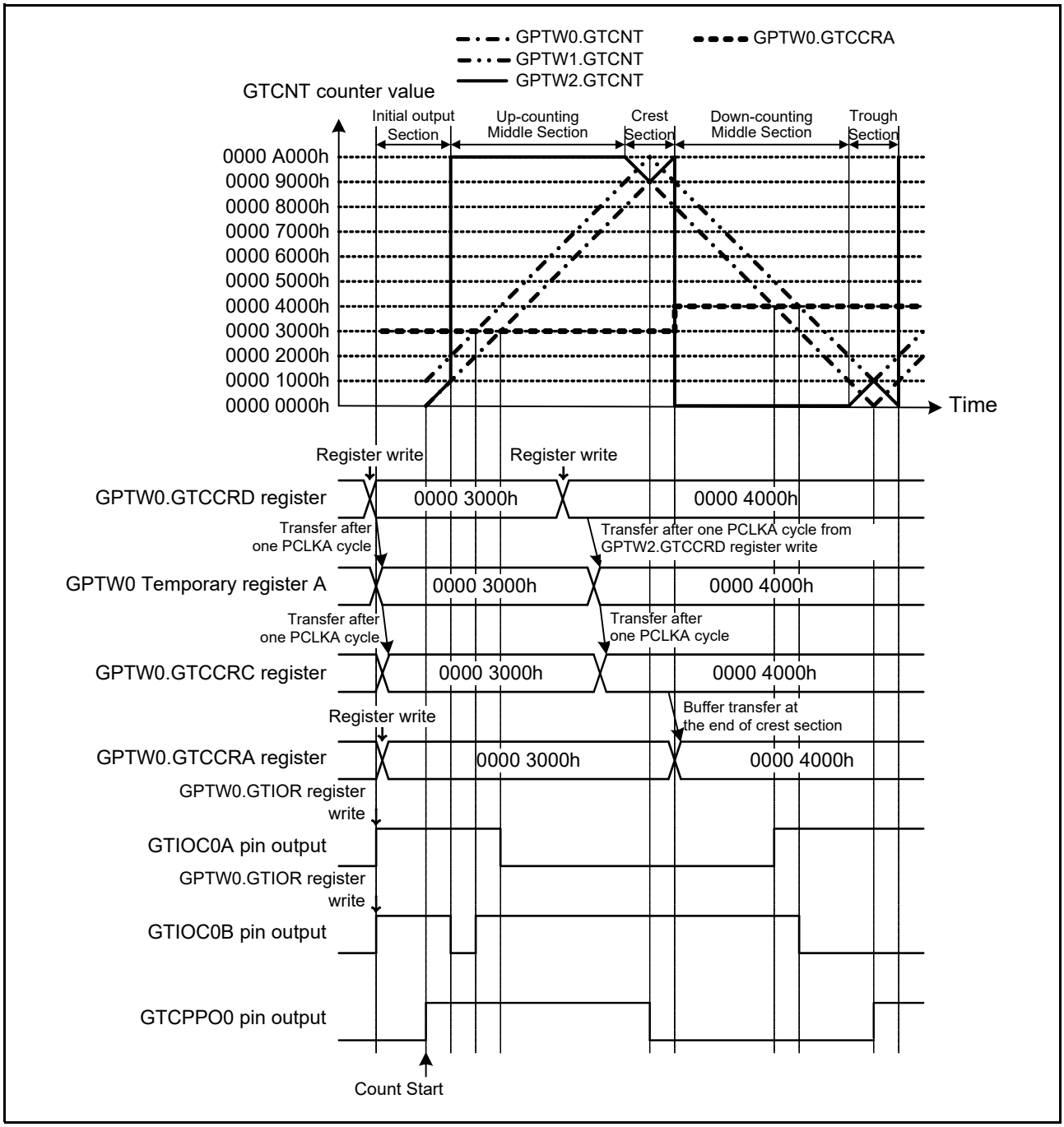


Figure 22.77 Example of Complementary PWM Mode Initial Output Operation (1)
(Complementary PWM mode 1 operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h, in the case that the initial GTCCRA register value is larger than the dead time value)

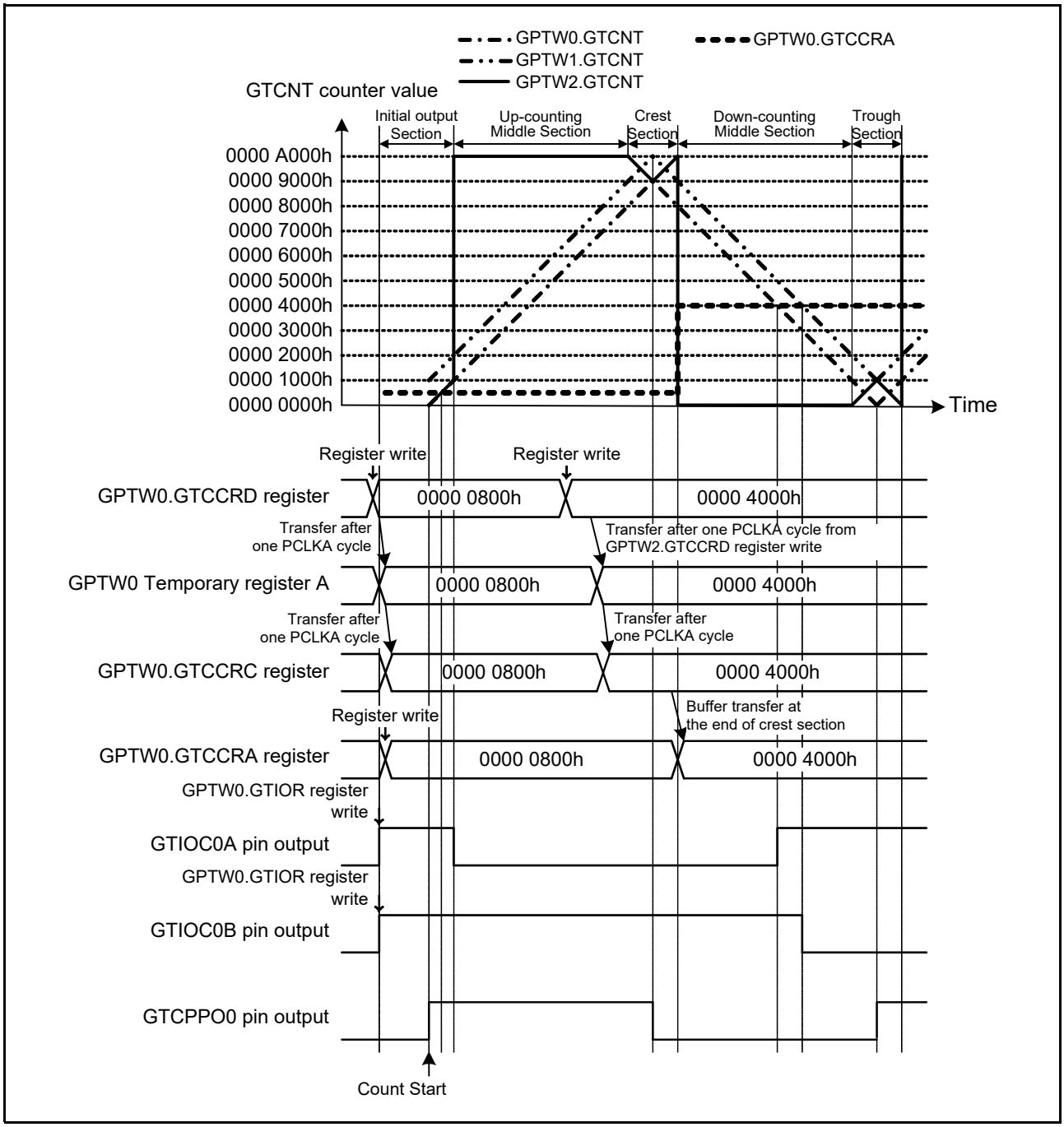


Figure 22.78 Example of Complementary PWM Mode Initial Output Operation (2)
(Complementary PWM mode 1 operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h, in the case that the initial GTCCRA register value is equal to or less than the dead time value)

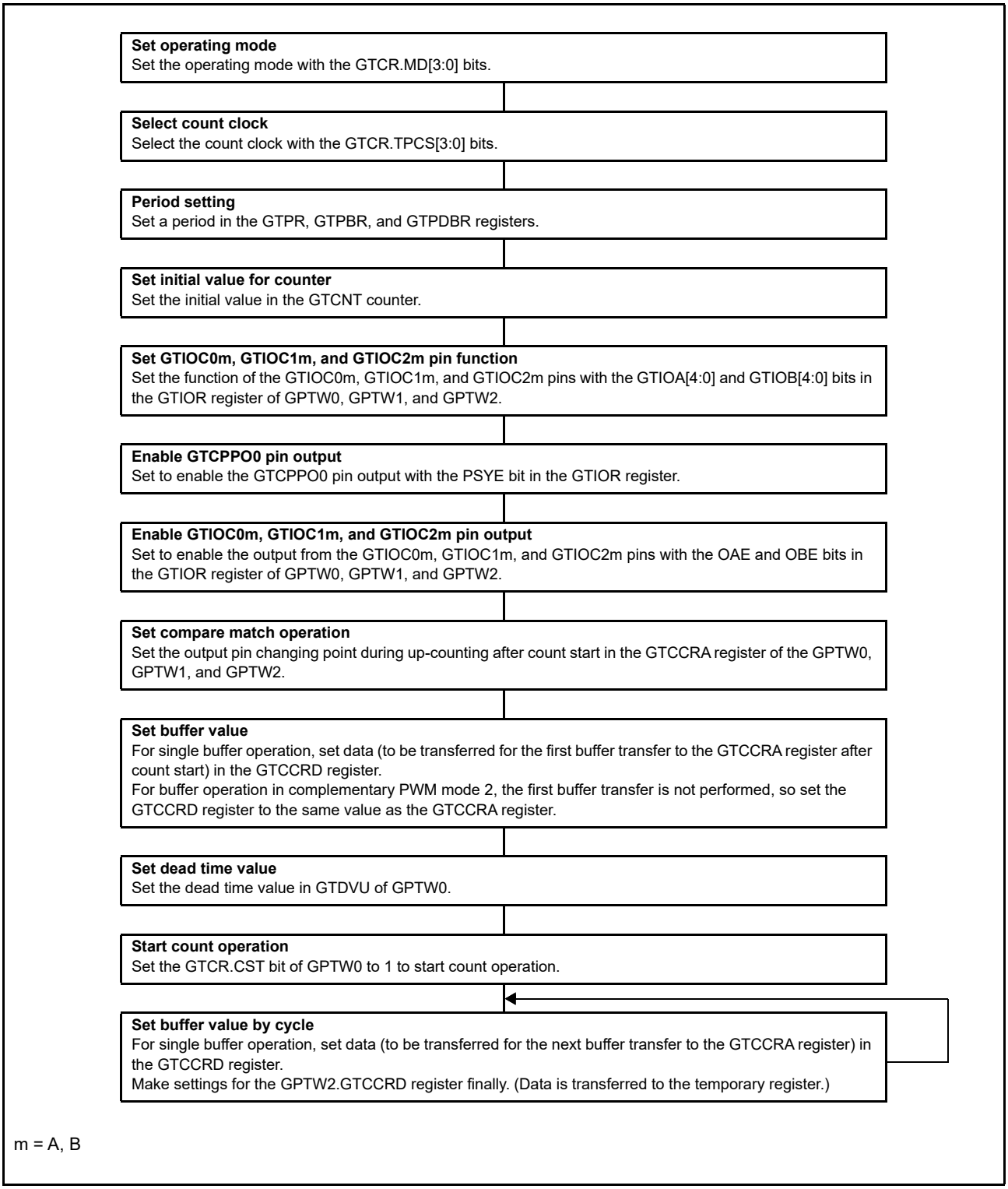


Figure 22.79 Example for Setting Complementary PWM Mode 1, 2, 3

(8) Complementary PWM Mode 4

In complementary PWM mode 4, the value written to the GTCCRD and GTCCRF registers is immediately applied to compare match operation by transferring data also to the GTCCRA register during buffer transfer to a temporary register before crest or trough transfer timing.

Figure 22.80 shows the block diagram in complementary PWM mode 4.

In the configuration of complementary PWM mode 4, a buffer transfer path from the GTCCRD register to the GTCCRC and GTCCRA registers and a buffer transfer path from the GTCCRF register to the GTCCRE and GTCCRA registers are added to other complementary PWM modes shown in Figure 22.58.

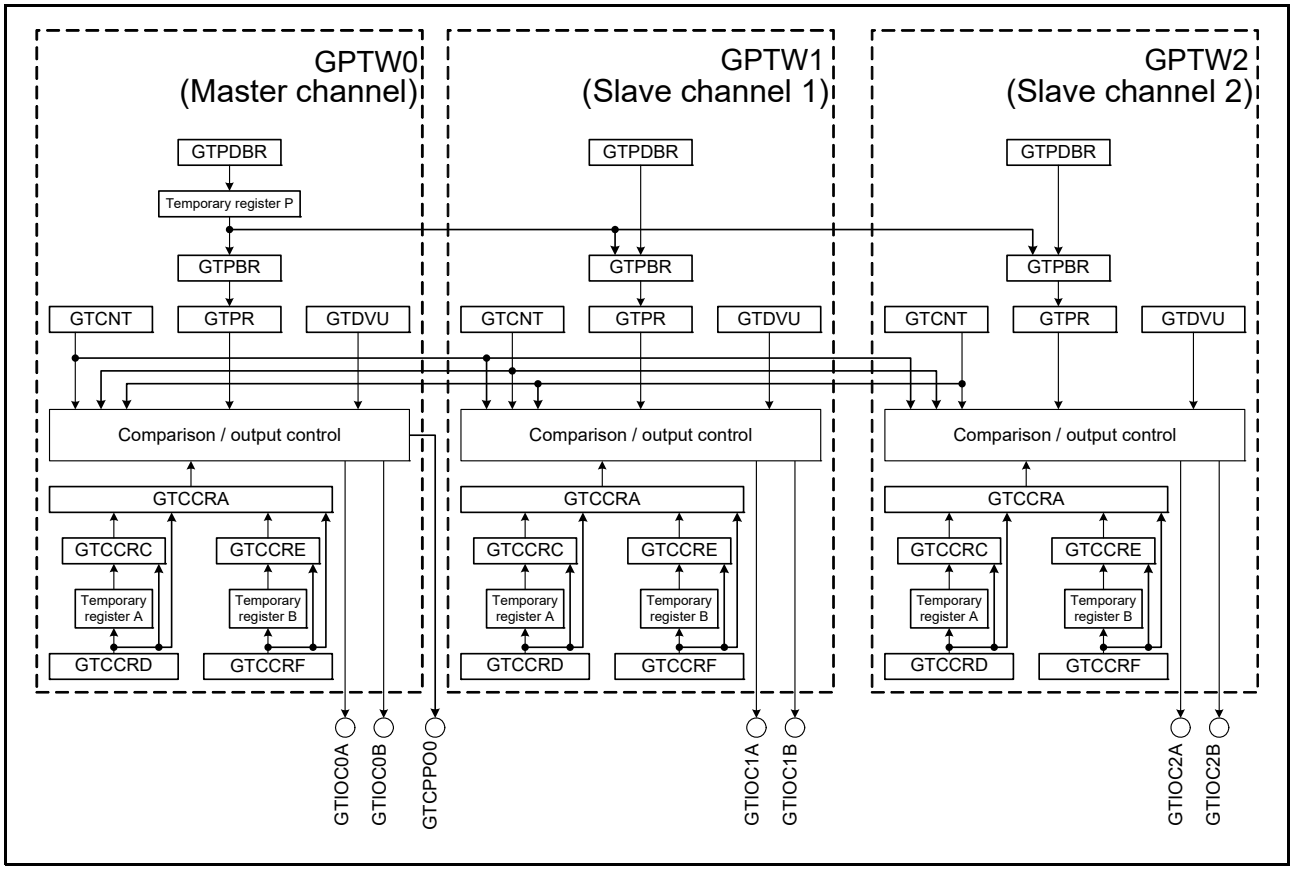


Figure 22.80 Block Diagram in Complementary PWM Mode 4

Count operation is performed in the same way as complementary PWM modes 1 to 3. See Table 22.7 and Table 22.8.

With respect to buffer operation and PWM waveform changes in complementary PWM mode 4, buffer transfer (shown in Figure 22.80) from the GTCCRD and GTCCRF registers is added based on the complementary PWM mode 3 operation.

Buffer transfer and PWM waveform are controlled according to operation section, state of comparison with the GTCCRA register, and write value.

In single buffer operation, a compare match value is written only to the GTCCRD register, and transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Transfer from the GTCCRD register to the temporary register A and transfer from the GTCCRF register to the temporary register B are performed in the same way as other complementary PWM modes. Transfers are concurrently performed in three channels by writing a value to the GPTW2.GTCCRD register. Transfer from the GTCCRD register to the GTCCRC register, GTCCRA register, temporary register B, and GTCCRE register and transfer from the GTCCRF register to the GTCCRE register and the GTCCRA register are performed at the same time as the above-mentioned transfer to the temporary register.

Table 22.11 and Table 22.12 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPTW2.GTCCRD register) to the GTCCRC and GTCCRA registers by writing a value to the GTCCRD register during single buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the temporary register A to the GTCCRC register, and from the GTCCRC register to the GTCCRA register) other than those shown in Table 22.11 and Table 22.12 are the same as single buffer transfer in complementary PWM mode 3 shown in Table 22.9.

Table 22.11 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (1)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	• In the case of GTCCRD > GPTW1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTW1.GTCNT GPTW1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	No transfer
	After up-counting compare match	GTCCRD	No transfer
Up-counting crest section	Before up-counting compare match	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	Before down-counting compare match • In the case of GTCCRD > GPTW1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTW1.GTCNT GPTW1.GTCNT Negative-phase OFF After down-counting dead time start No transfer
	Up-counting dead time period	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	No transfer
	After up-counting compare match	Before down-counting compare match • In the case of GTCCRD < GPTW2.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTW2.GTCNT GPTW2.GTCNT Negative-phase OFF After down-counting dead time start No transfer	No transfer
Down-counting crest section	Before down-counting compare match	Up-counting dead time period • In the case of GTCCRD < GPTW1.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTW1.GTCNT GPTW1.GTCNT Negative-phase OFF After up-counting compare match • In the case of GTCCRD < GPTW0.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTW0.GTCNT GPTW0.GTCNT Positive-phase OFF	No transfer
	Down-counting dead time period	No transfer	No transfer
	After down-counting compare match	No transfer	No transfer

Table 22.12 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	• In the case of GTCCRD < GPTW0.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTW0.GTCNT GPTW0.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	No transfer
	After down-counting compare match	GTCCRD	No transfer
Down-counting trough section	Before down-counting compare match	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	Before up-counting compare match • In the case of GTCCRD < GPTW0.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTW0.GTCNT GPTW0.GTCNT Positive-phase OFF After up-counting dead time start No transfer
	Down-counting dead time period	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	No transfer
	After down-counting compare match	Before up-counting compare match • In the case of GTCCRD > GPTW2.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTW2.GTCNT GPTW2.GTCNT Negative-phase OFF After up-counting dead time start No transfer	No transfer
Up-counting trough section	Before up-counting compare match	Down-counting dead time period • In the case of GTCCRD > GPTW0.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTW0.GTCNT GPTW0.GTCNT Positive-phase ON After down-counting compare match • In the case of GTCCRD > GPTW1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTW1.GTCNT GPTW1.GTCNT Negative-phase OFF	No transfer
	Up-counting dead time period	No transfer	No transfer
	After up-counting compare match	No transfer	No transfer

Figure 22.81 to Figure 22.85 show examples of single buffer operation for each operation section in complementary PWM mode 4.

- Up-Counting Middle Section: Figure 22.81
- Up-Counting Crest Section: Figure 22.82
- Down-Counting Crest Section: Figure 22.83

- Down-Counting Middle Section: Figure 22.84
- Down-Counting Trough Section: Figure 22.85

Figure 22.86 shows an example for setting complementary PWM mode 4.

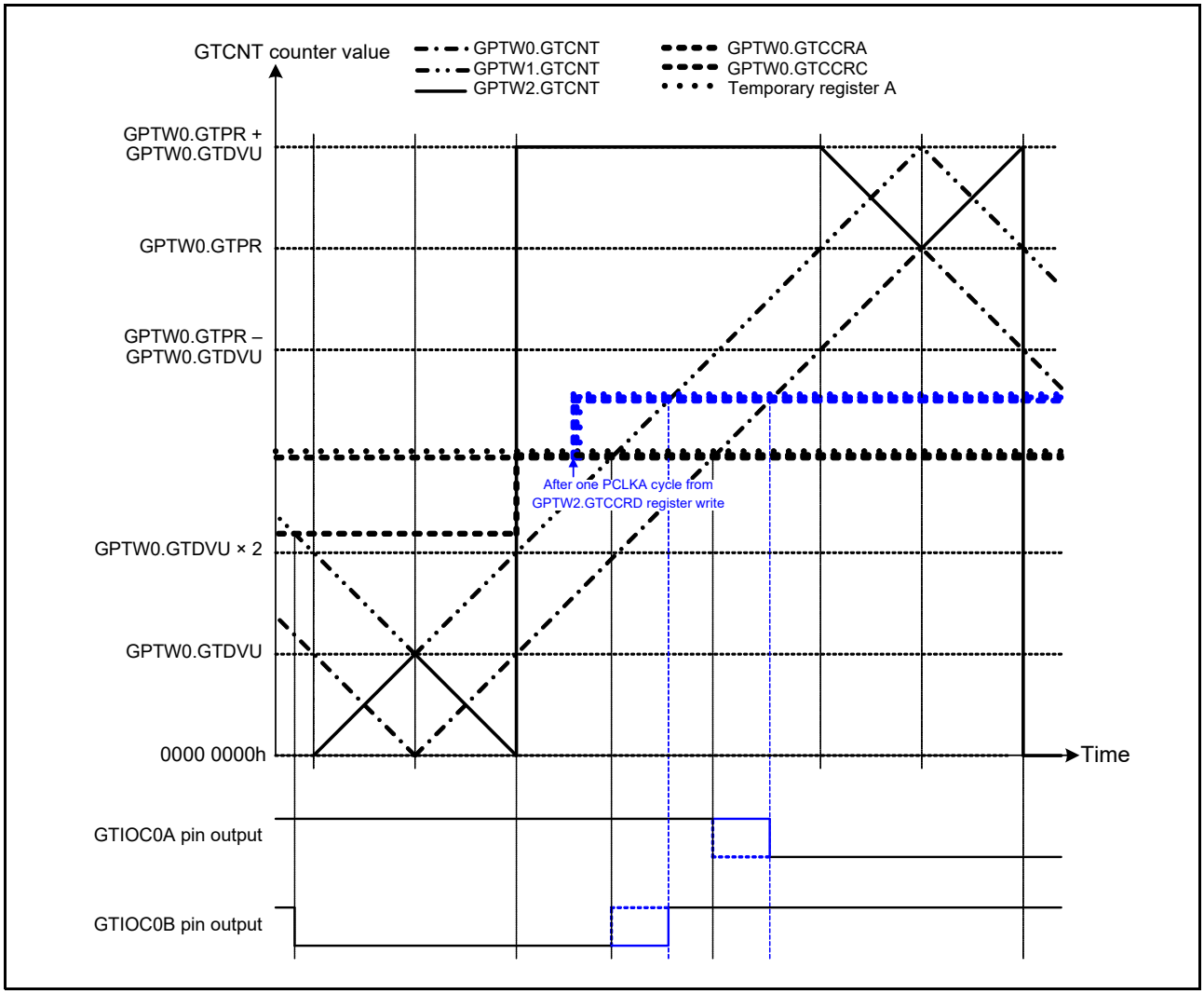


Figure 22.81 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPTW1.GTCNT value is written to the GTCCRD register before up-counting compare match)

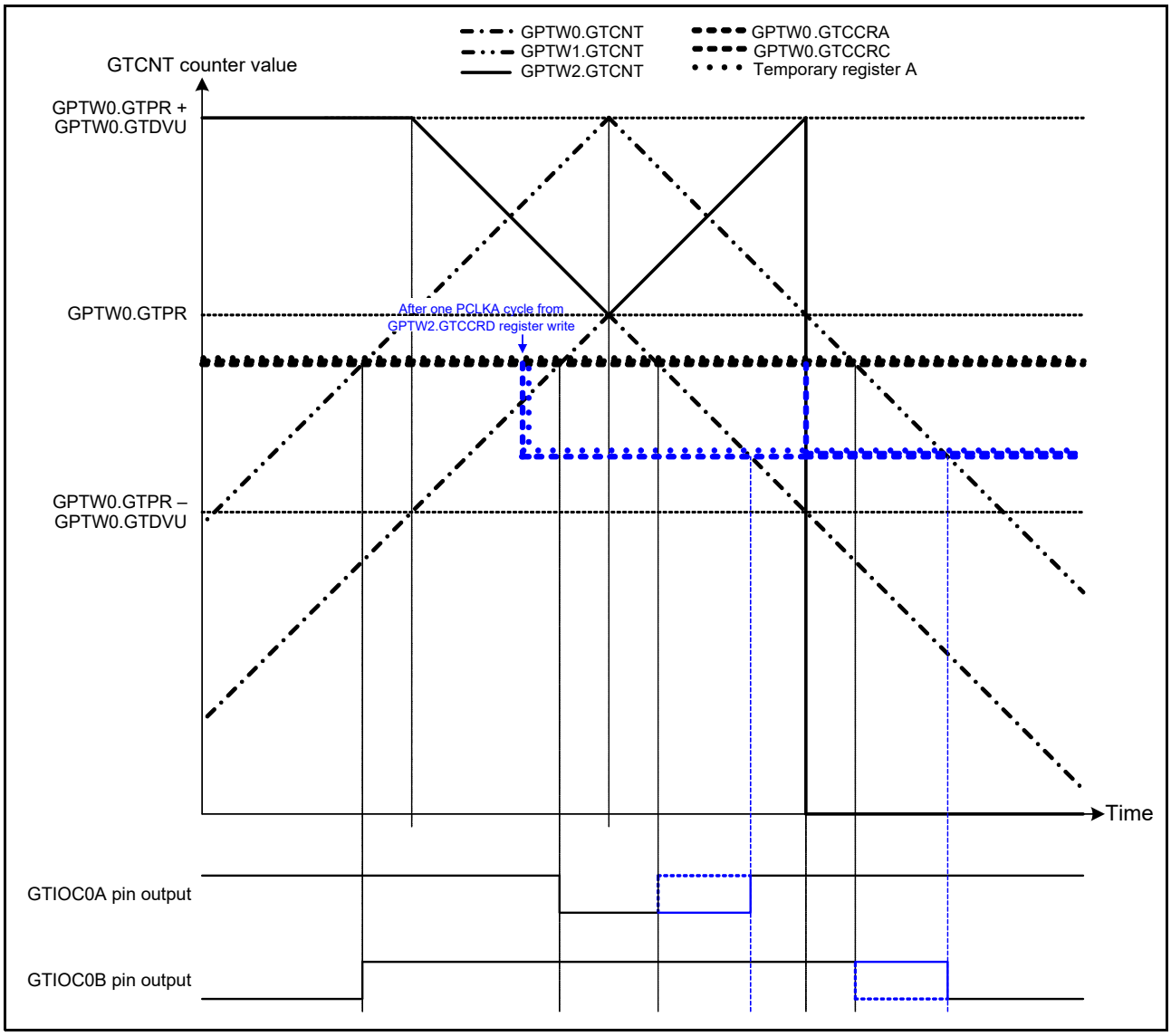


Figure 22.82 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting, GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting, when a value is written to the GTCCRD register during up-counting dead time)

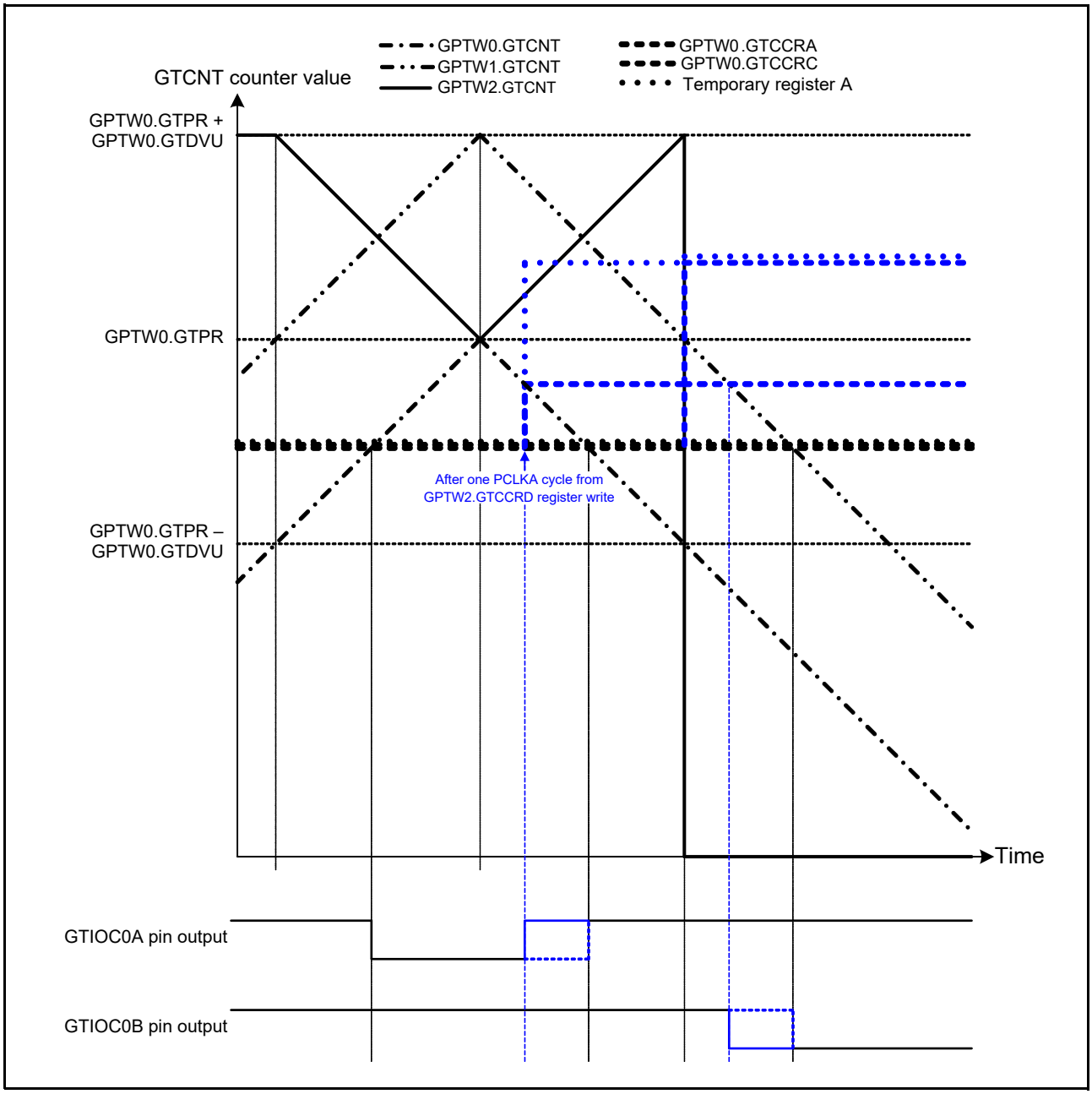


Figure 22.83 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Crest Section)
(Complementary PWM mode 4 single buffer operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
when a value large than and equal to GPTW0.GTCNT value is written to the GTCCRD register before down-counting compare match and after up-counting compare match)

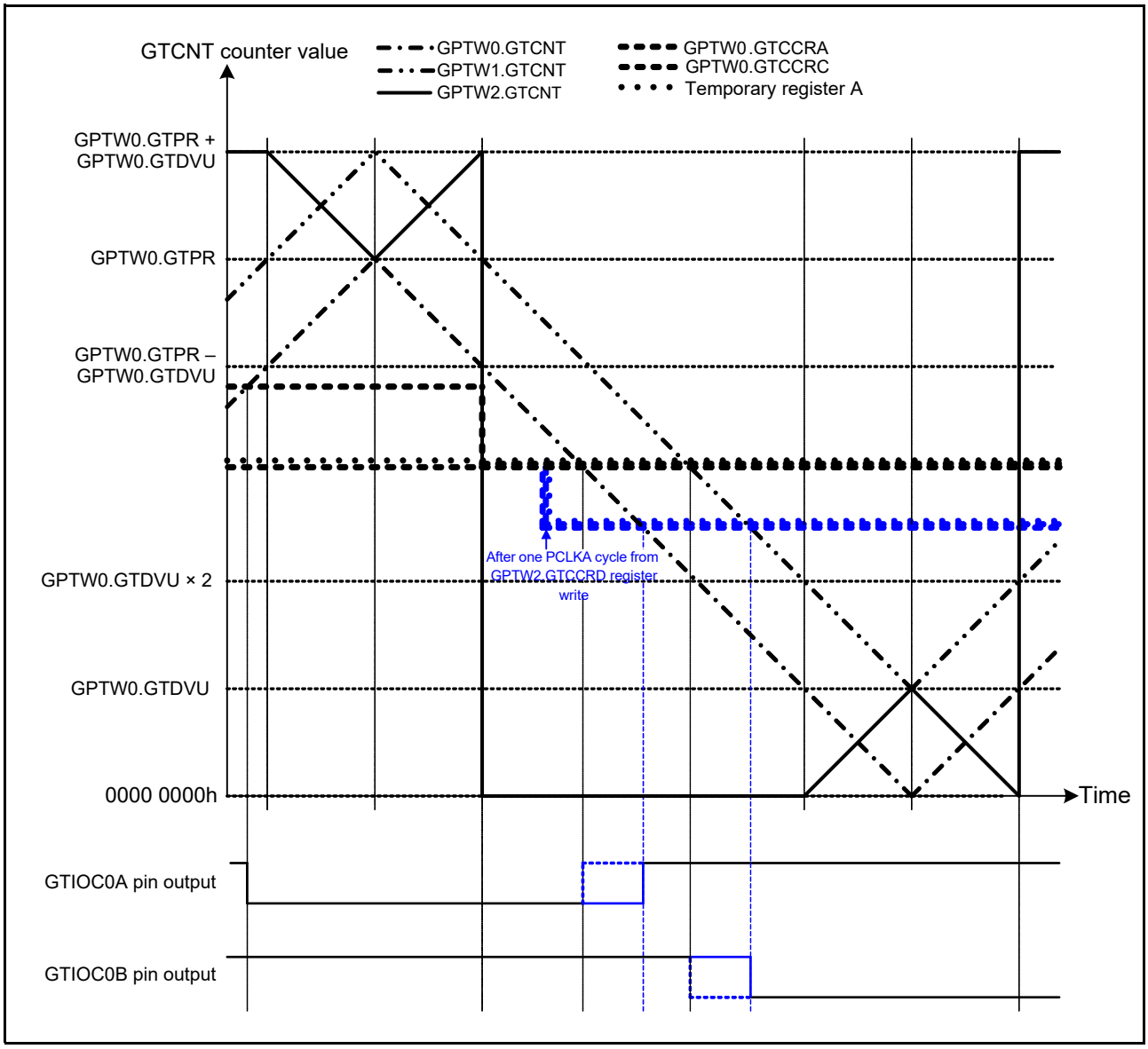


Figure 22.84 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 single buffer operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
When a value less than GPTW0.GTCNT value is written to the GTCCRD register before down-counting compare match)

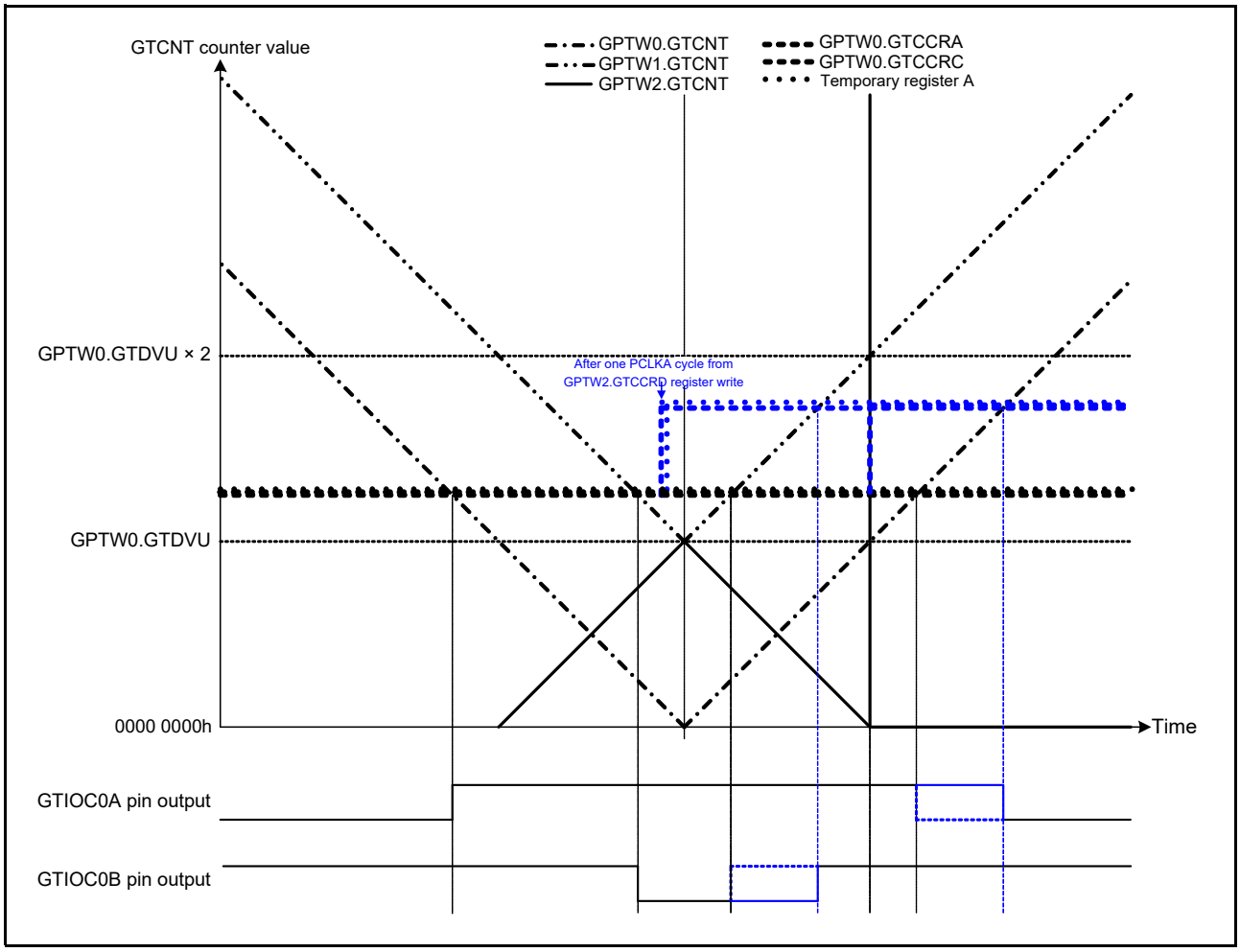


Figure 22.85 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Trough Section)
(Complementary PWM mode 4 single buffer operation,
GTIOC0A pin = Low/GTIOC0B pin = High at GTCCRA register compare match during up-counting,
GTIOC0A pin = High/GTIOC0B pin = Low at GTCCRA register compare match during down-counting,
When a value large than GPTW2.GTCNT value is written to the GTCCRD register after down-counting compare match)

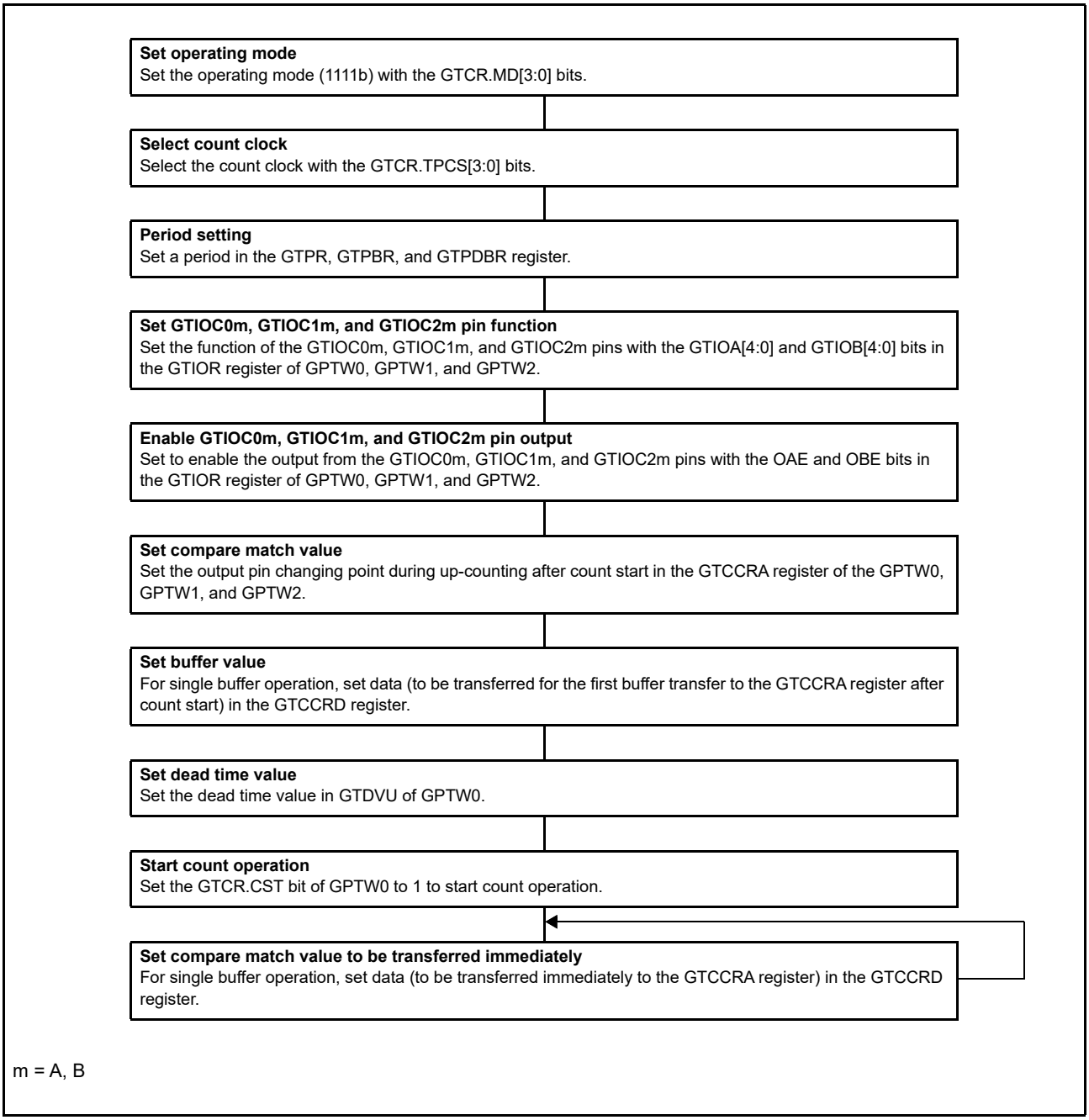


Figure 22.86 Example for Setting Complementary PWM Mode 4

22.3.4 Automatic Dead Time Setting Function

By setting the GTDTCR register, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (the GTCCRA register value) and specified dead time values (the GTDVU register value) can automatically be set to the GTCCRB register.
The automatic dead time setting function can be used in sawtooth-wave one-shot pulse mode and all the triangle PWM modes.
Dead time for the changing point of a negative waveform is set in the GTDVU register.
The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to the GTCCRB register is prohibited when the automatic dead time setting function is used.

Do not set the dead-time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 22.13. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.
If the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.
Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 22.13 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

PWM Output Operating Mode	Count Direction	Period	Condition for Dead-Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	GTCCRA – GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVU > GTPR	GTPR – GTDVU	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR – GTDVU	GTPR
		Second half	GTCCRA – GTDVU < 0	GTDVU	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA – GTDVU≤0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA – GTDVU < 0	GTDVU	0

Figure 22.87 to Figure 22.90 show examples of automatic dead time setting function operation. Figure 22.91 and Figure 22.92 show the setting examples.

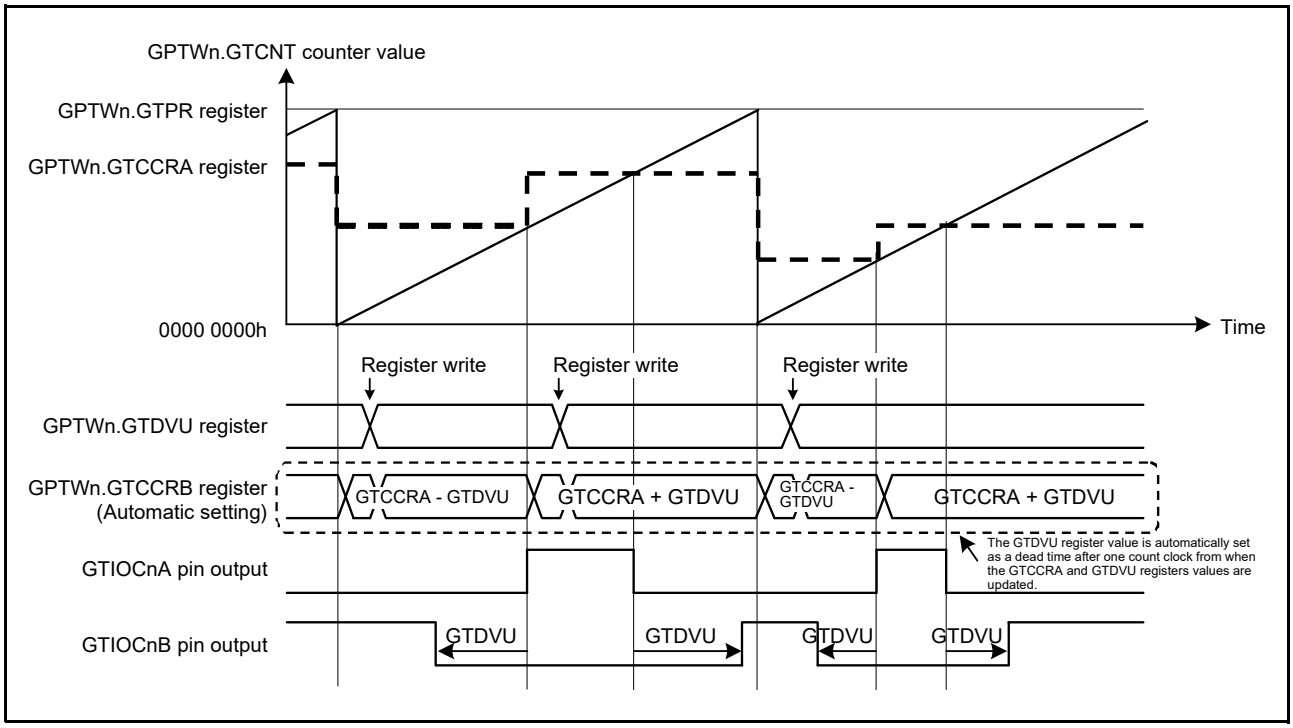


Figure 22.87 Example of Automatic Dead Time Setting Function Operation (Sawtooth-Wave One-Shot Pulse Mode, Up-Counting, Active Level: High) (n = 0 to 2)

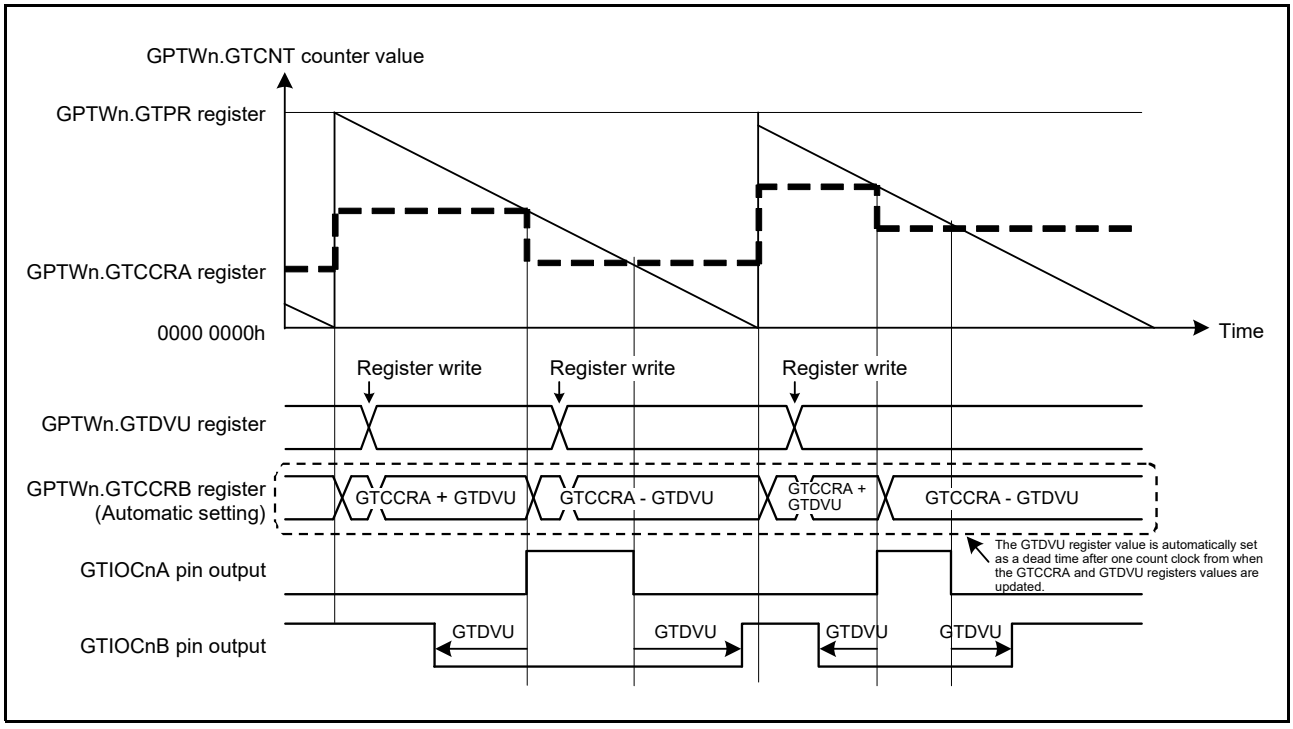


Figure 22.88 Example of Automatic Dead Time Setting Function Operation (Sawtooth-Wave One-Shot Pulse Mode, Down-Counting, Active Level: High) (n = 0 to 2)

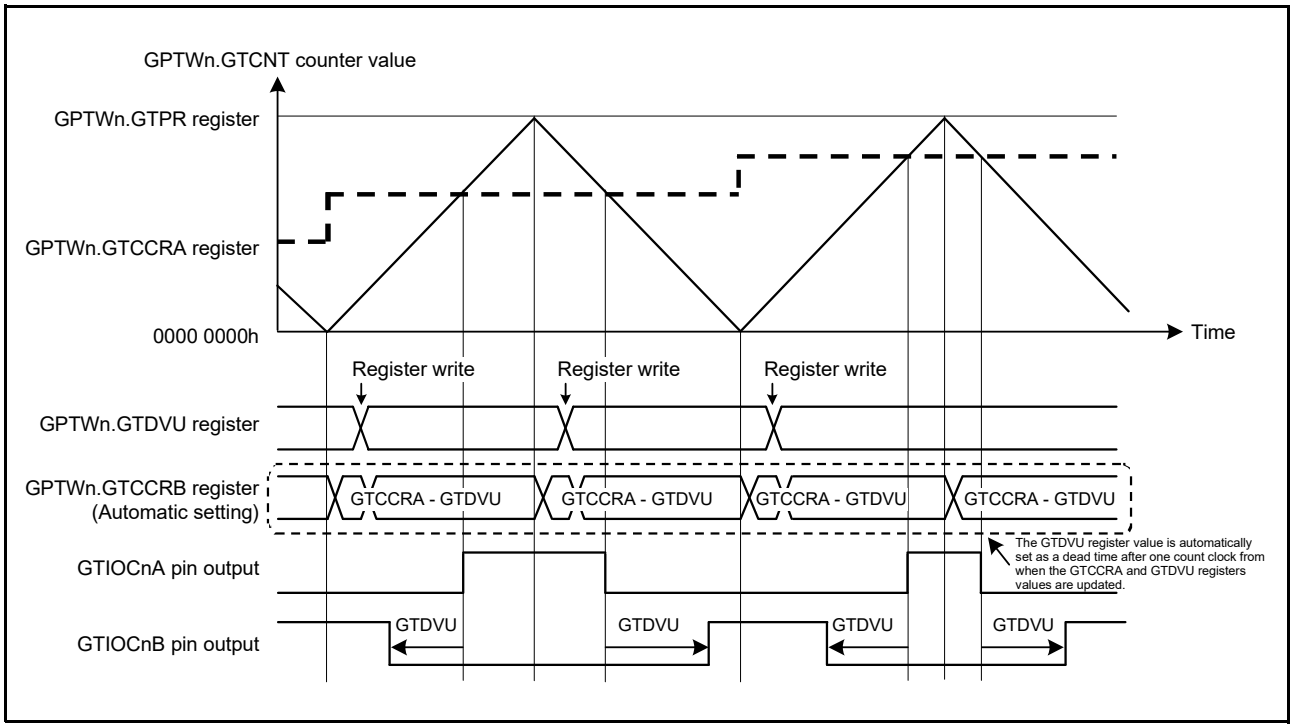


Figure 22.89 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 1, Active Level: High) (n = 0 to 2)

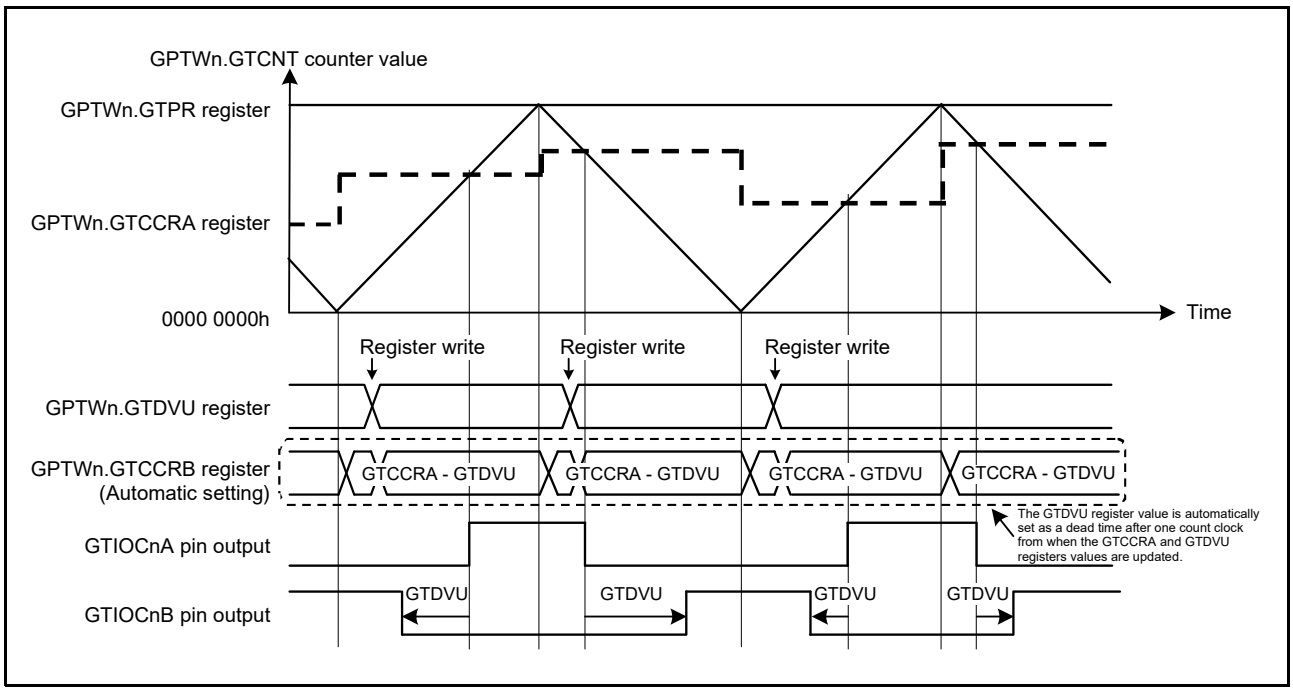


Figure 22.90 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 2 or 3, Active Level: High) (n = 0 to 2)

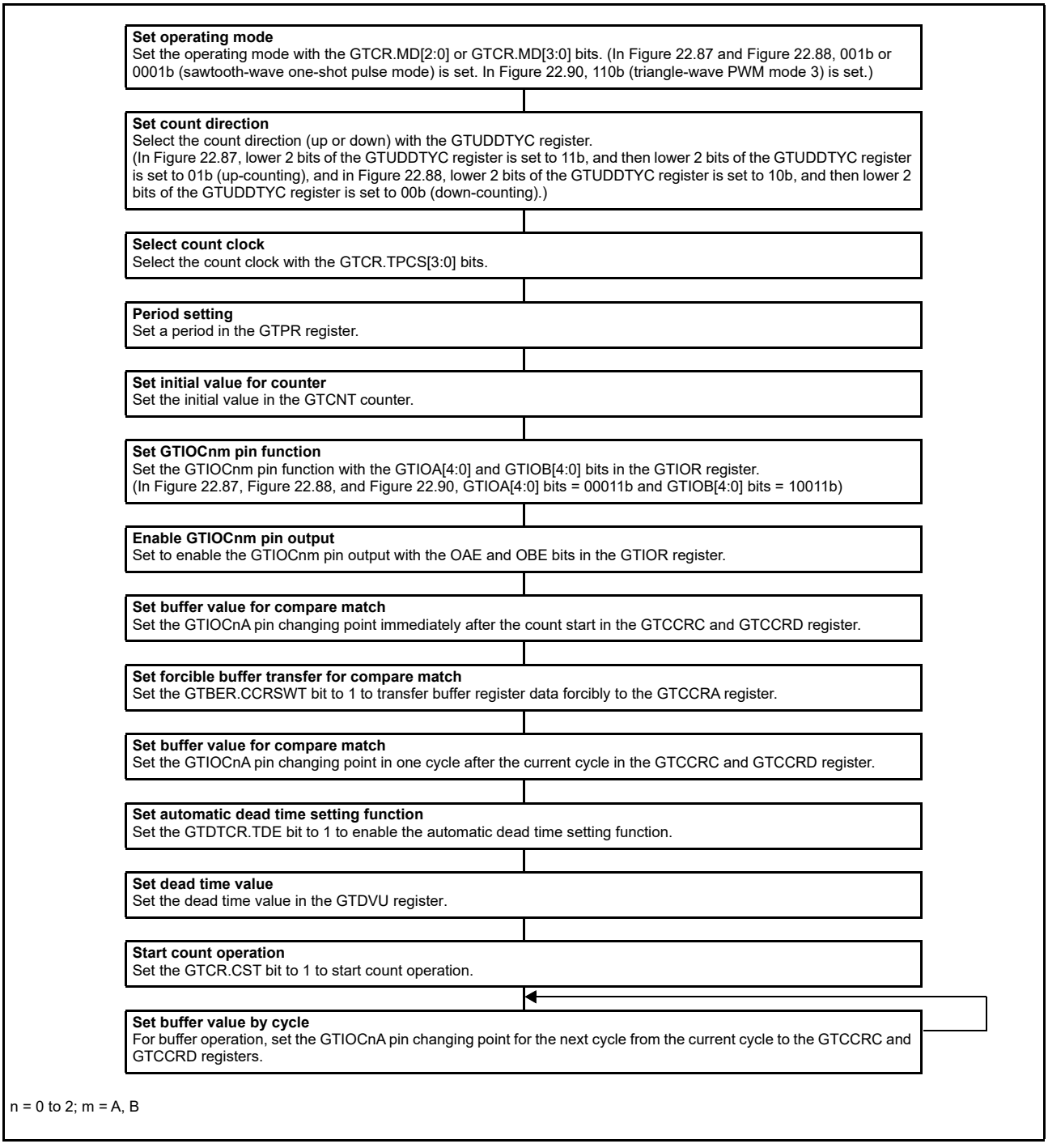


Figure 22.91 Example for Setting Automatic Dead Time Setting Function (Sawtooth-Wave One-Shot Pulse Mode, Triangle-Wave PWM Mode 3)

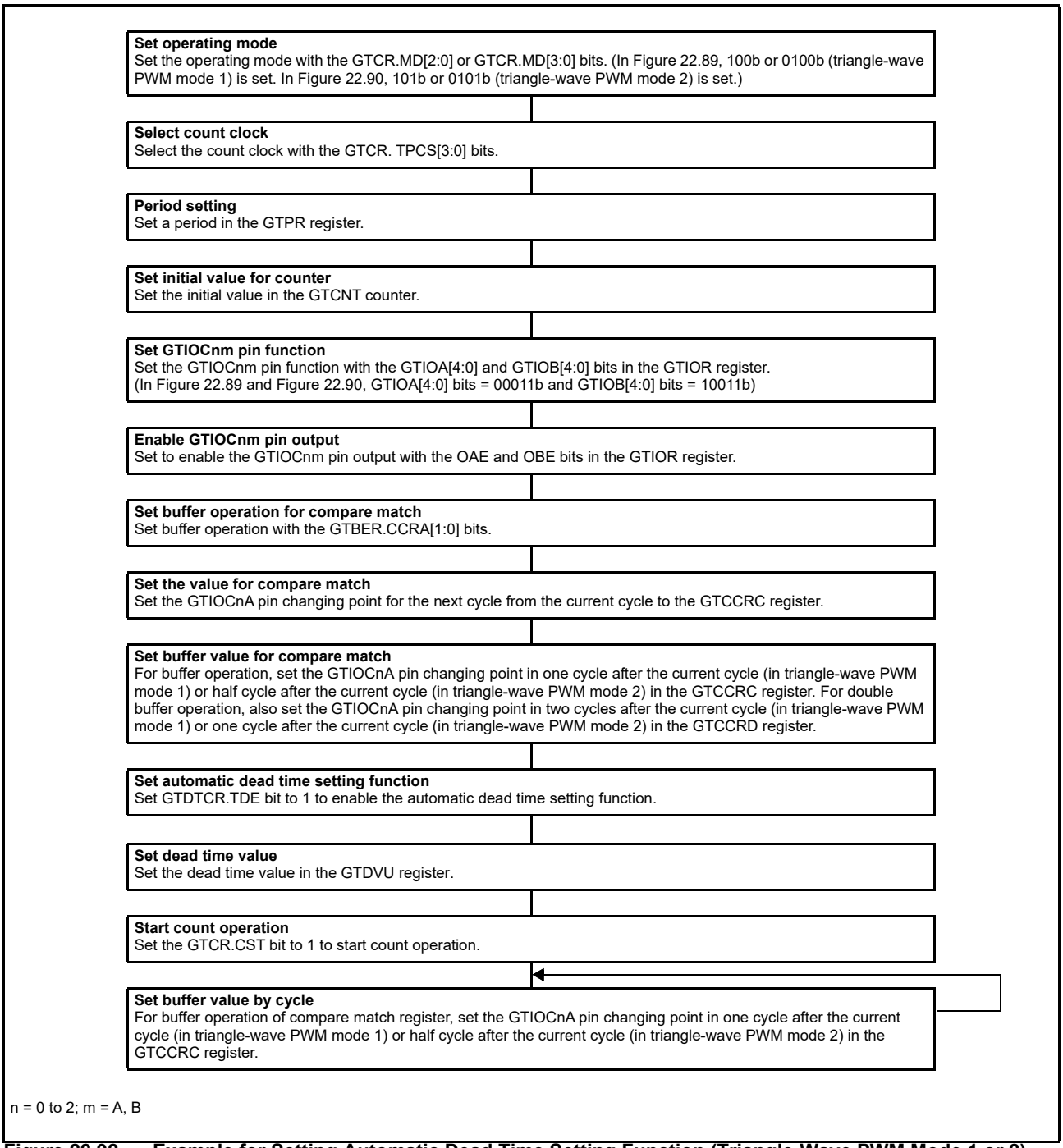


Figure 22.92 Example for Setting Automatic Dead Time Setting Function (Triangle-Wave PWM Mode 1 or 2)

22.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the GTUDDTYC.UD bit.
In sawtooth-wave mode, if the UD bit is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the UD bit is modified while count operation is stopped and the GTUDDTYC.UDF bit is 0, the UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.
In triangle-wave mode, changing the value of the GTUDDTYC.UD bit during counting does not switch the direction of counting. Likewise, changing the value of the GTUDDTYC.UD bit while the GTUDDTYC.UDF bit is 0 and counting is stopped does not actually update the value of the bit. If 1 is written to the GTUDDTYC.UDF bit while counting is stopped, the value of the GTUDDTYC.UD bit at that time is reflected from the time counting is started.
When counting direction is switched during count operation in sawtooth-wave mode, the value of the GTPR register after start of up-counting is reflected to the count period in up-counting operation, and the value of the GTPR register after the start of down-counting is reflected to the count period in down-counting operation.
Figure 22.93 shows an example of count direction changing function operation.

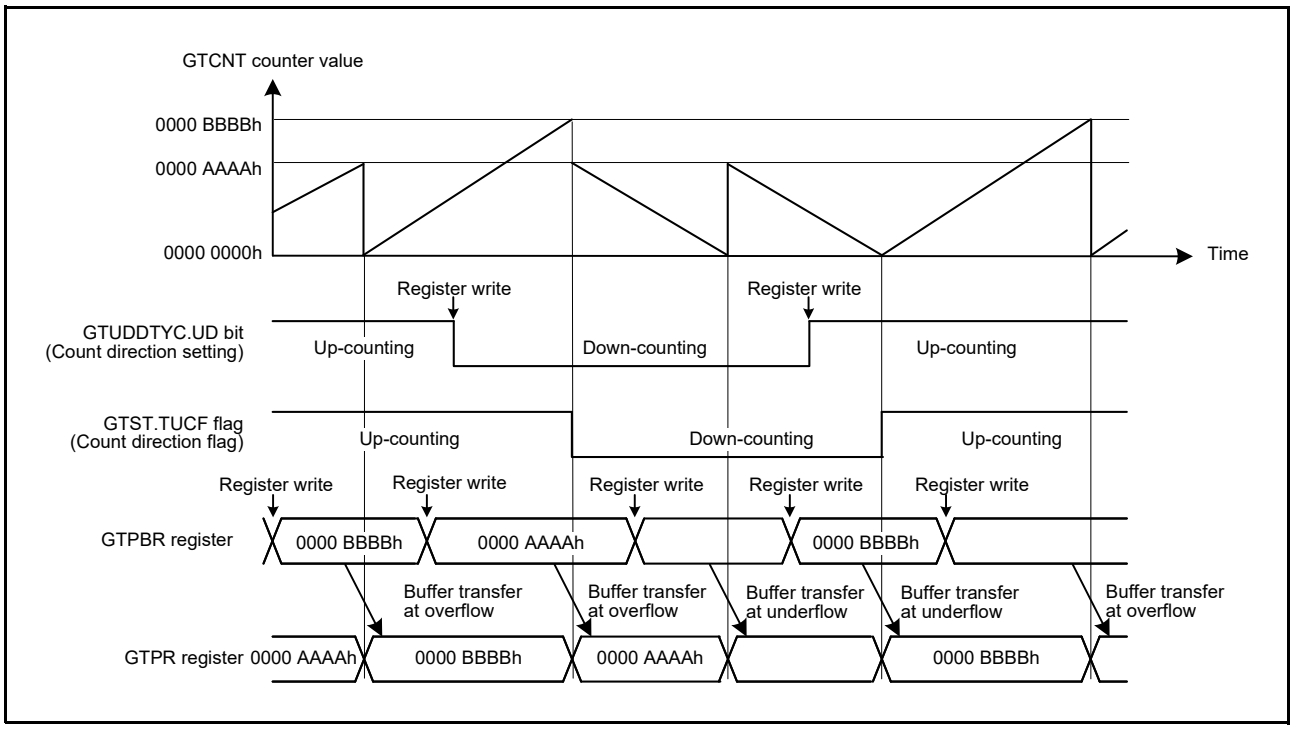


Figure 22.93 Example of Count Direction Changing Function Operation (during Buffer Operation)

22.3.6 Duty Cycle 0%/100% Output Function

Changing the value of the GTUDDTYC.OADTY[1:0] bits and GTUDDTYC.OBDTY[1:0] bits specifies the output duty setting on the GTIOCnA and GTIOCnB pins to 0% or 100%. (n = 0 to 7)

This function is invalid in sawtooth-wave PWM mode 2 or complementary PWM mode.

In sawtooth-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting is reflected at an overflow (when changed during up-counting) or an underflow (when changed during down-counting). When the GTUDDTYC.OADTYF bit or GTUDDTYC.OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at an overflow or underflow is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

In triangle-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting changed at an underflow is reflected.

When the OADTYF bit or OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at an underflow is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

During the operation under the settings of duty cycle for 0%/100%, compare match operation inside the GPTW continues to perform interrupt outputs and buffer operations.

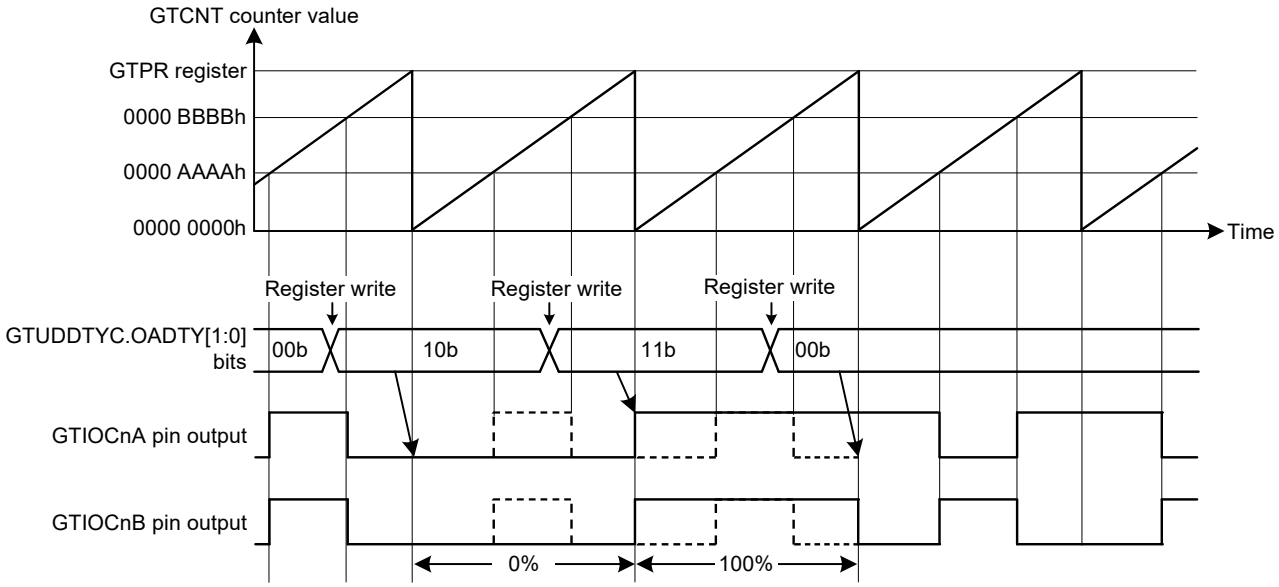
When the output duty setting is changed from 0% or 100% due to a compare match, the GTIOR.GTIOA[3:2] bits and the OADTYR bits determine the level output on the GTIOCnA pin and GTIOR.GTIOB[3:2] bits and OBDTYR bits determine the level output on the GTIOCnB pin at the end of the cycle. When the GTIOA[3:2], GTIOB[3:2] bits are set as 01b, output is driven Low at the end of the cycle, and when set as 10b, output is driven High at the end of the cycle. The output is retained at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 00b, and toggled at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 11b. A value to be output is selectable in the OADTYR or OBDTYR bit when the corresponding GTIOA[3:2] or GTIOB[3:2] bits are either 00b or 11b. Table 22.14 lists the output values at the end of the cycle when the output setting is changed from duty 0% or 100% to compare match.

Table 22.14 Output Value after Release of Duty 0%/100%

GTIOR.GTIOm[3:2]	Value at Compare Match Output at the End of the Cycle in the Case of Masking by a Duty Cycle 0% or 100%	The GTUDDTYC.OmDTYR Bit at Duty 0% Setting		The GTUDDTYC.OmDTYR Bit at Duty 100% Setting	
		0	1	0	1
00b (Output retained at the end of the cycle)	0	0	0	1	0
	1	0	1	1	1
01b (Low output at the end of the cycle)	—	0	0	0	0
10b (High output at the end of the cycle)	—	1	1	1	1
11b (Toggle output at the end of the cycle)	0	1	1	0	1
	1	1	0	0	0

m = A, B

Figure 22.94 shows an example of operation of the output of the duty cycle 0% or 100% (n = 0 to 7).



[Setting examples]

GTIOR.GTIOA[4:0] bits: 00011b
The initial output is Low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OADTYR bit: 0
The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.

GTIOR.GTIOB[4:0] bits: 00011b
The initial output is Low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OBDTYR bit: 1
The function of the GTIOB[3:2] bits is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 22.94 Example of Operation of Output of Duty Cycle 0% or 100% (n = 0 to 7)

22.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by hardware sources in this MCU. There are 3 types of hardware sources, including external trigger input, ELC event input, and GTIOCnm pin input (n = 0 to 7; m = A, B)

22.3.7.1 Hardware Start Operation

The GTCNT counter can be started by a hardware source. Select a hardware source to start counting using the GTSSR register, and enable to start counting.
Figure 22.95 shows an example of count start operation by a hardware source. Figure 22.96 shows the setting example.

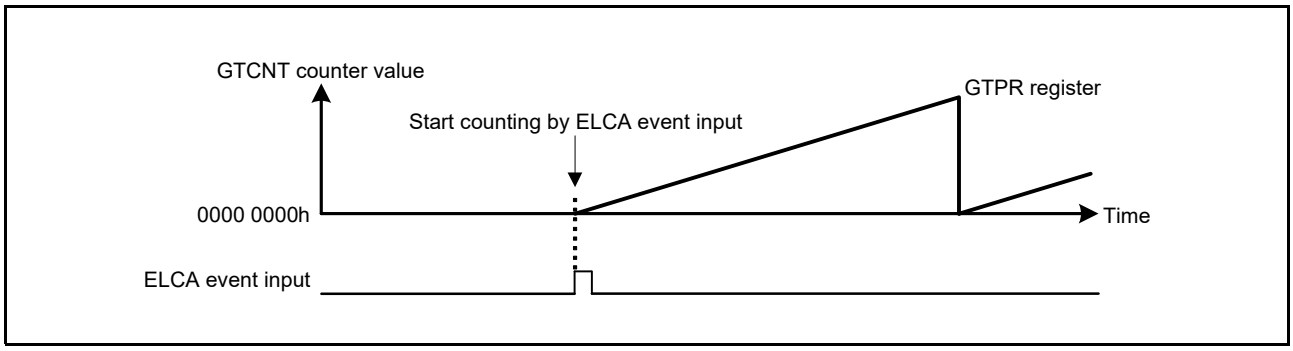


Figure 22.95 Example of Count Start Operation by Hardware Source (At starting by ELCA event)

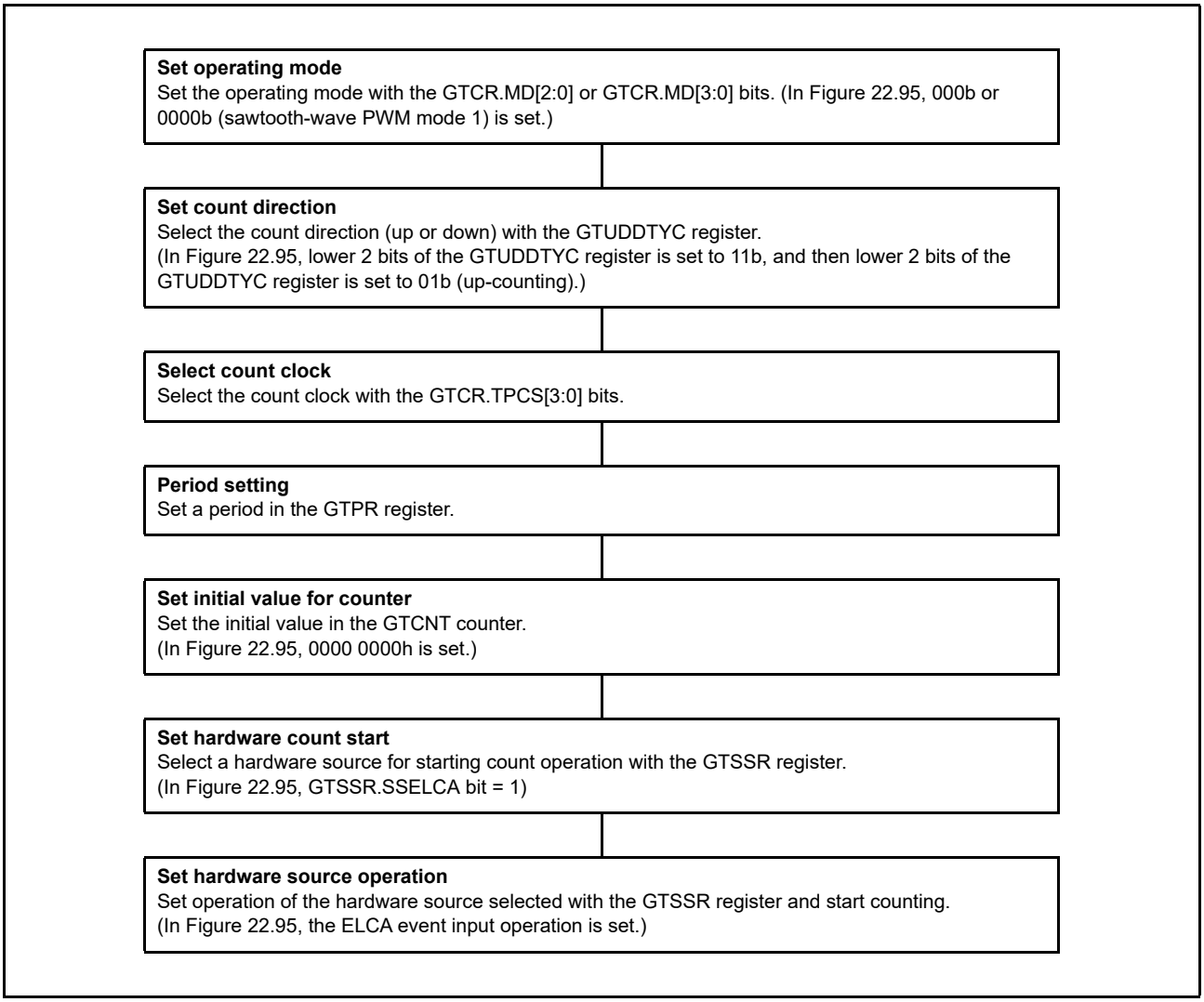


Figure 22.96 Example for Setting Count Start Operation by Hardware Source

Figure 22.97 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTETRGA pin.

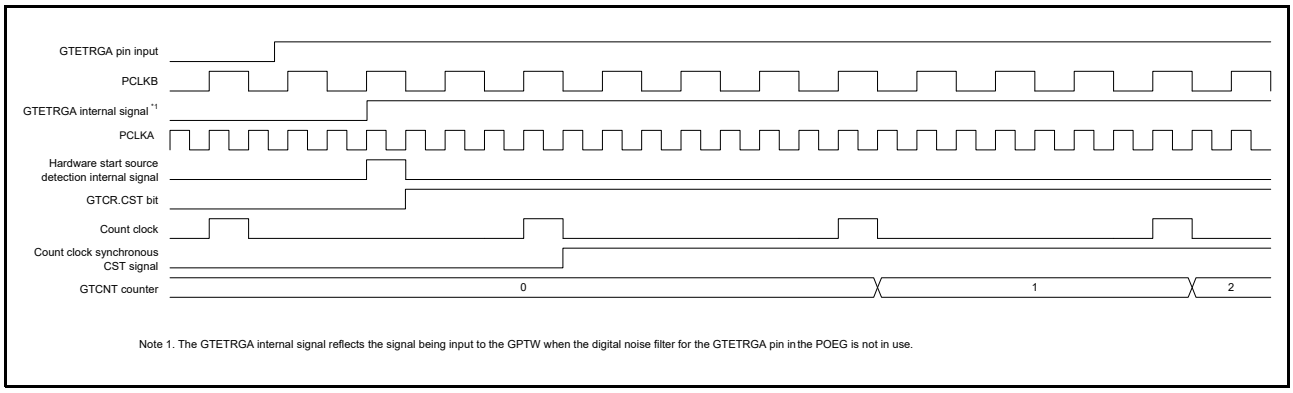


Figure 22.97 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 22.98 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTIOCnA pin.

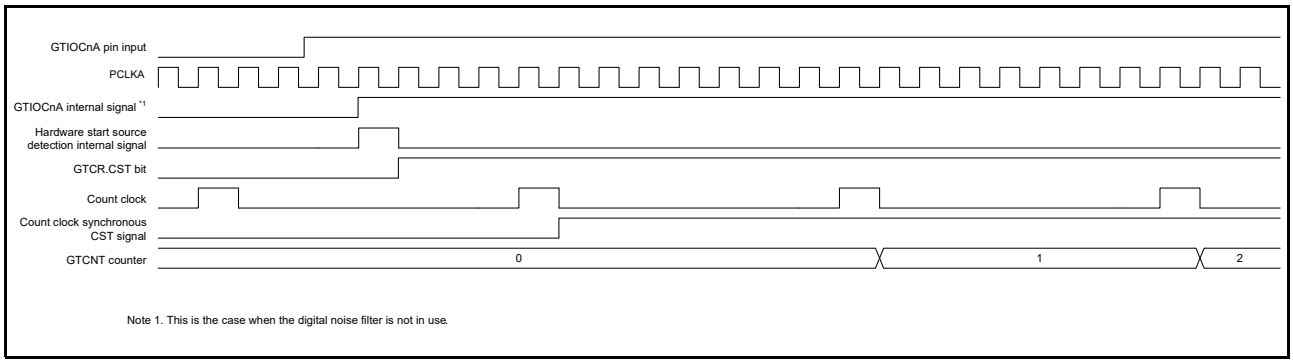


Figure 22.98 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 22.99 shows an example of timing of operations to start counting in response to event input from the ELCA. This is an example of operations to start counting by the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with PCLKA. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

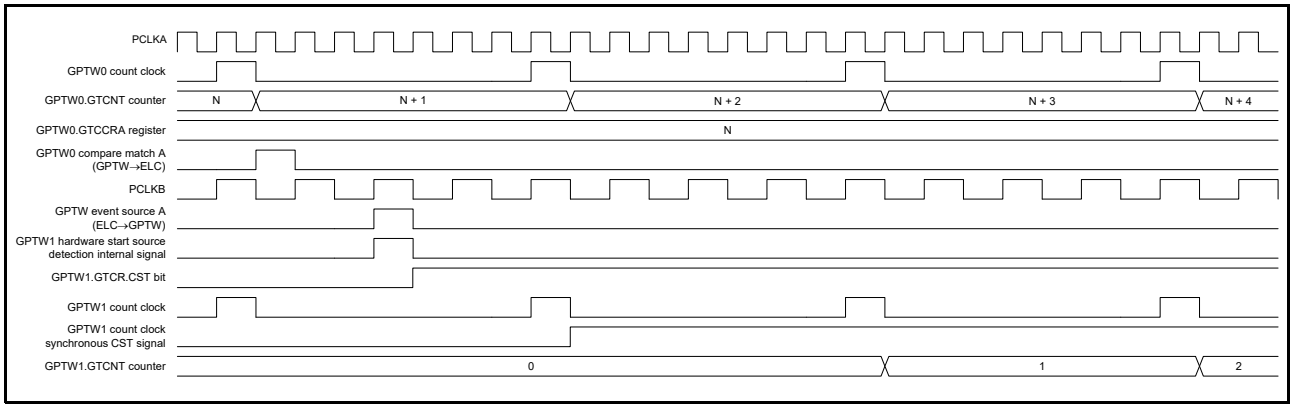


Figure 22.99 Example of Timing of Operations to Start Counting in Response to Event Input from the ELCA

22.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by a hardware source. Select a hardware source to stop counting with the GTPSR register, and enable to stop counting.

Figure 22.100 shows an example of count stop operation by a hardware source. Figure 22.101 shows the setting example. In this example, the count operation is stopped by the ELCA event input, and is restarted by the ELCB event input.

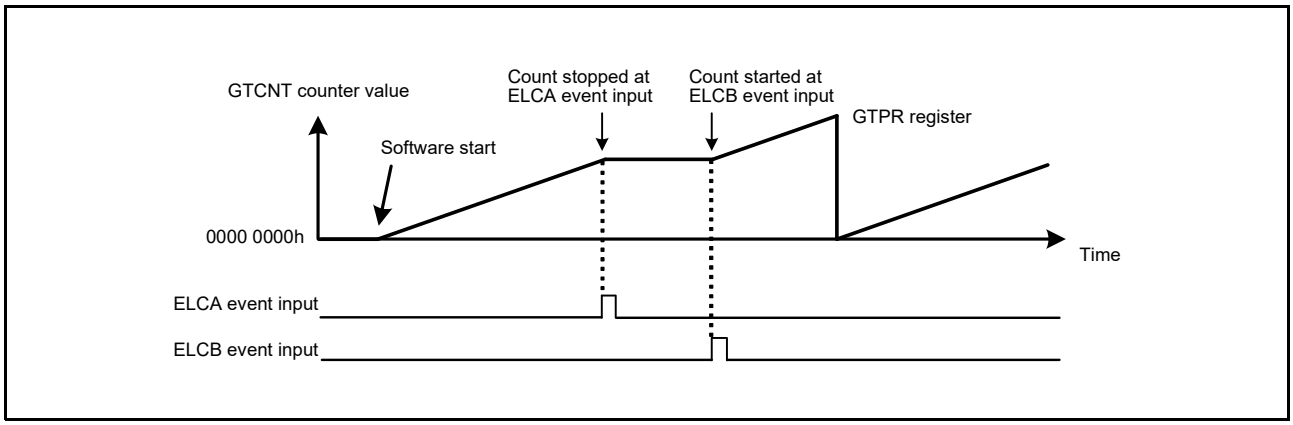


Figure 22.100 Example of Count Stop Operation by Hardware Source (Started by Software, Stopped at ELCA event input, and Restarted at ELCB event input)

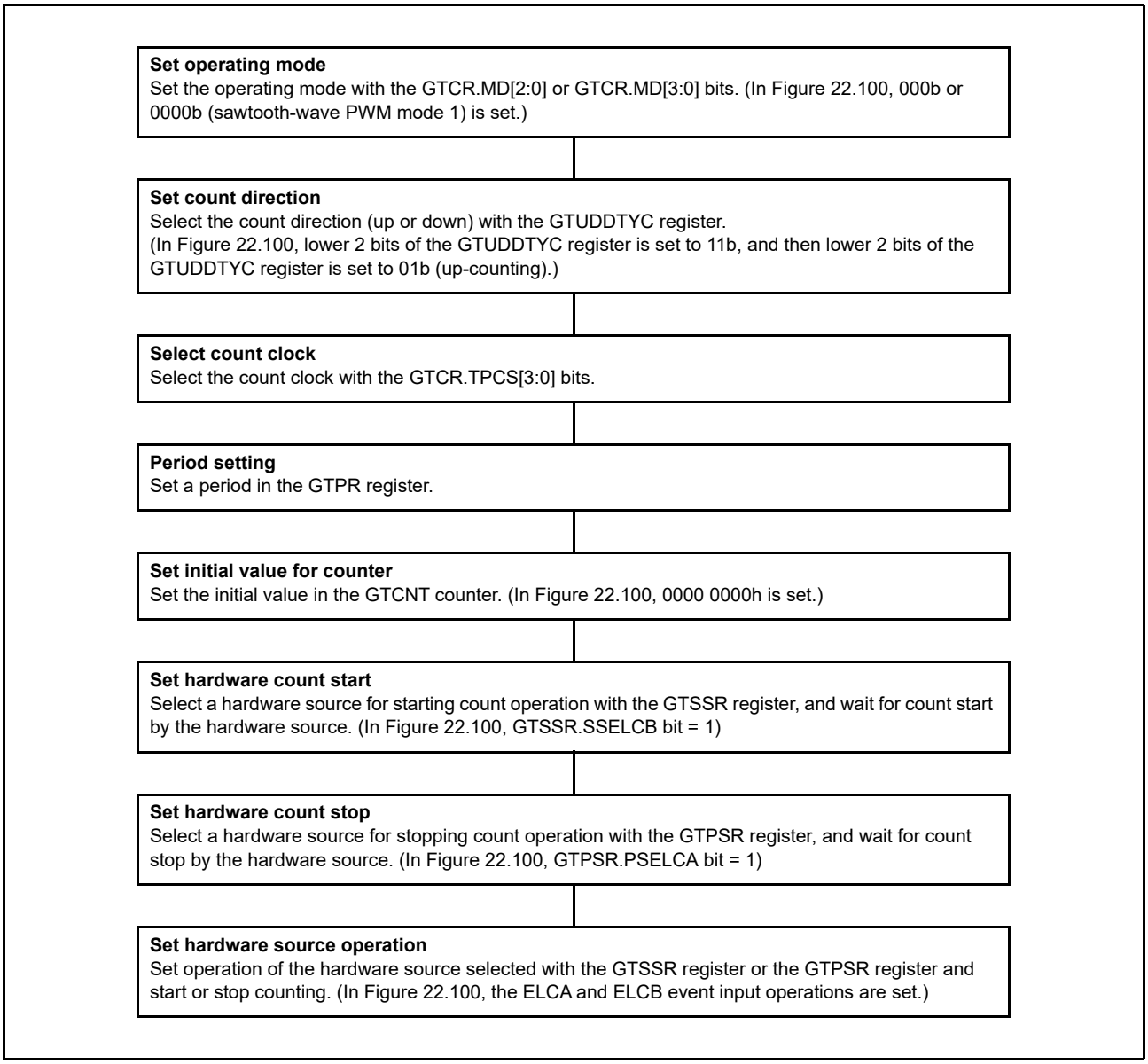


Figure 22.101 Example for Setting Count Stop Operation by Hardware Source

Figure 22.102 shows an example of count start/stop operation by a hardware source. Figure 22.103 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA pin.

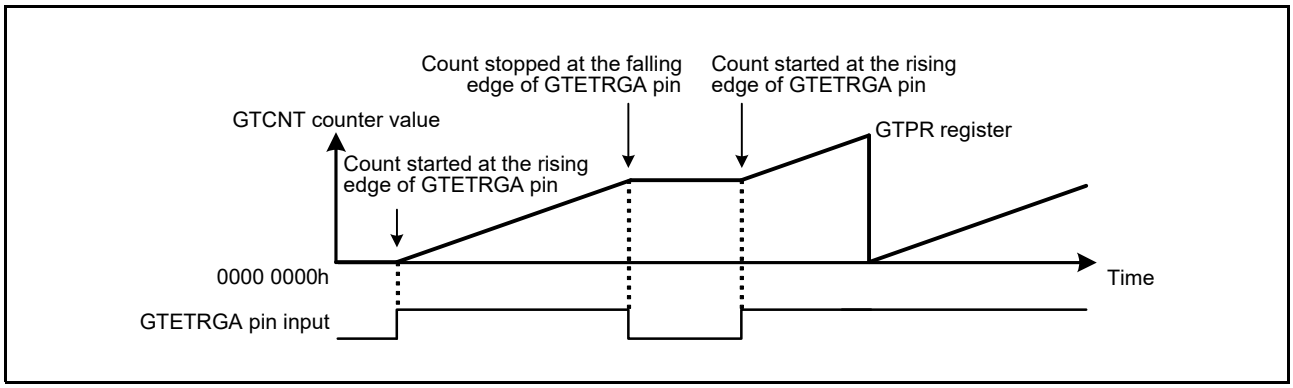


Figure 22.102 Example of Count Start/Stop Operation by Hardware Source (Started at Rising Edge of GTETRGA Pin Input, Stopped at Falling Edge of GTETRGA Pin Input)

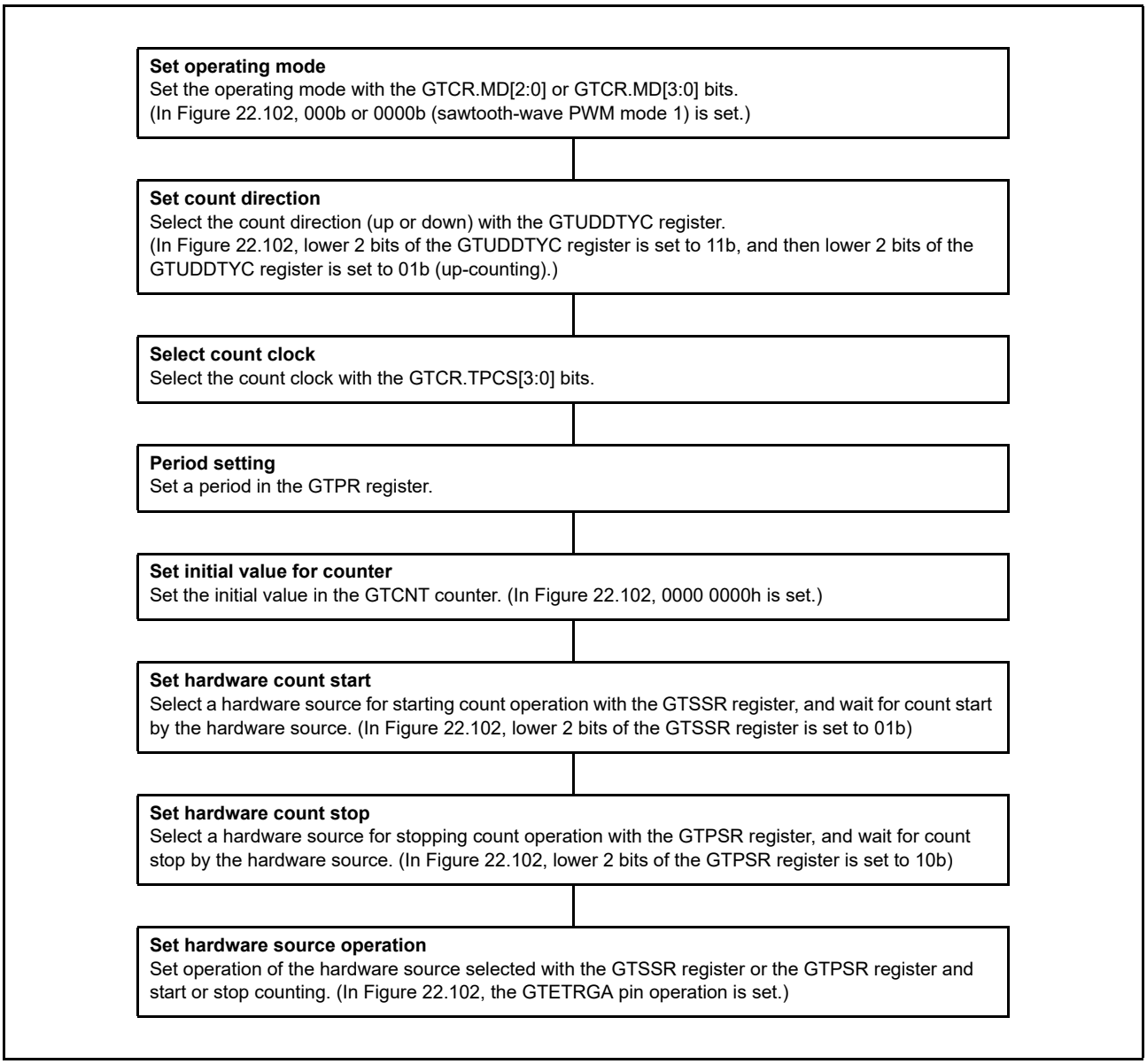


Figure 22.103 Example for Setting Count Start/Stop Operation by Hardware Source

Figure 22.104 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTETRGA pin.

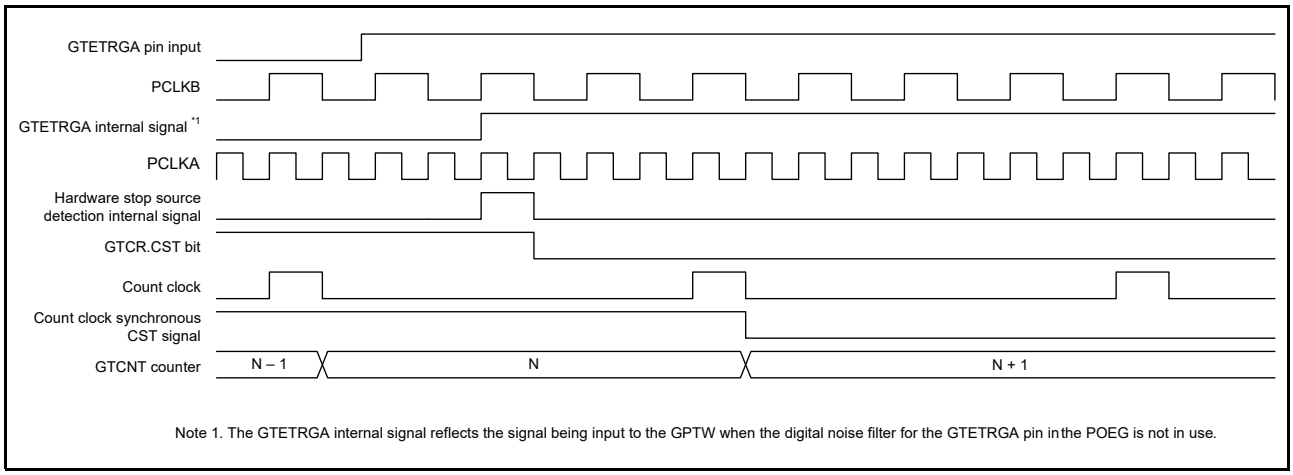


Figure 22.104 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 22.105 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTIOCnA pin.

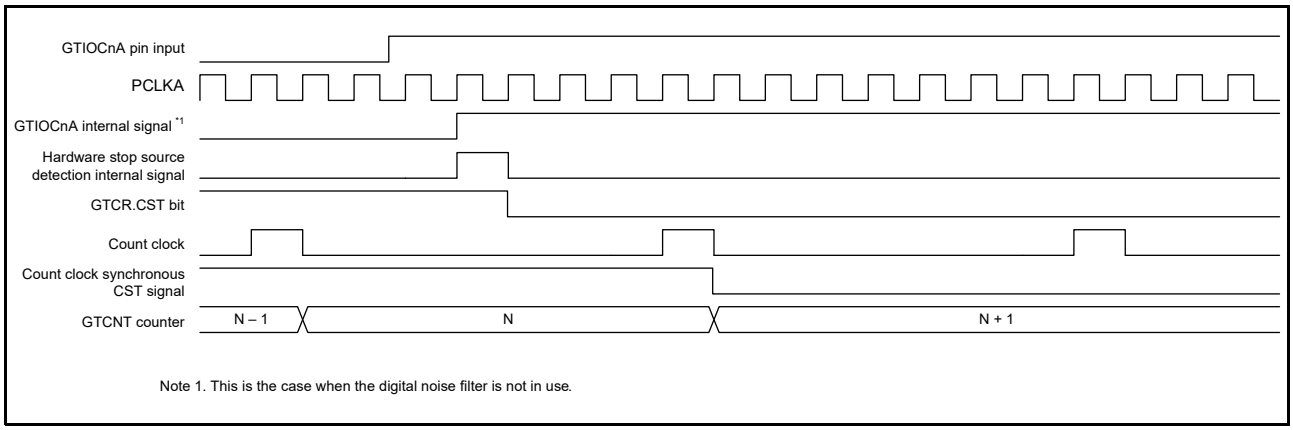


Figure 22.105 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 22.106 shows an example of timing of operations to stop counting in response to event input from the ELCA. This is an example of operations to stop counting by the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with PCLKA. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

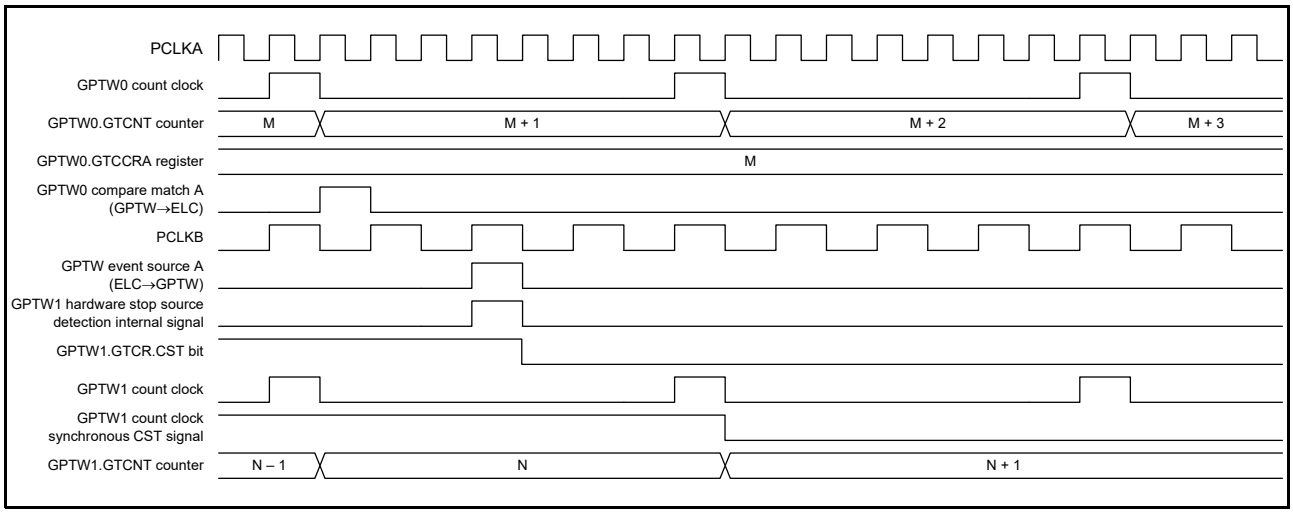


Figure 22.106 Example of Timing of Operations to Stop Counting in Response to Event Input from the ELCA

22.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by a hardware source. Select a hardware source to clear the counter with the GTCSR register, and enable clearing the counter.

Note that the GTCIV/GTCIU interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 22.107 and Figure 22.108 show examples of the GTCNT counter clearing operation by a hardware source. Figure 22.109 shows the setting example. In this example, the GTCNT counter is started at the ELCA event input, and is stopped/cleared at the ELCB event input.

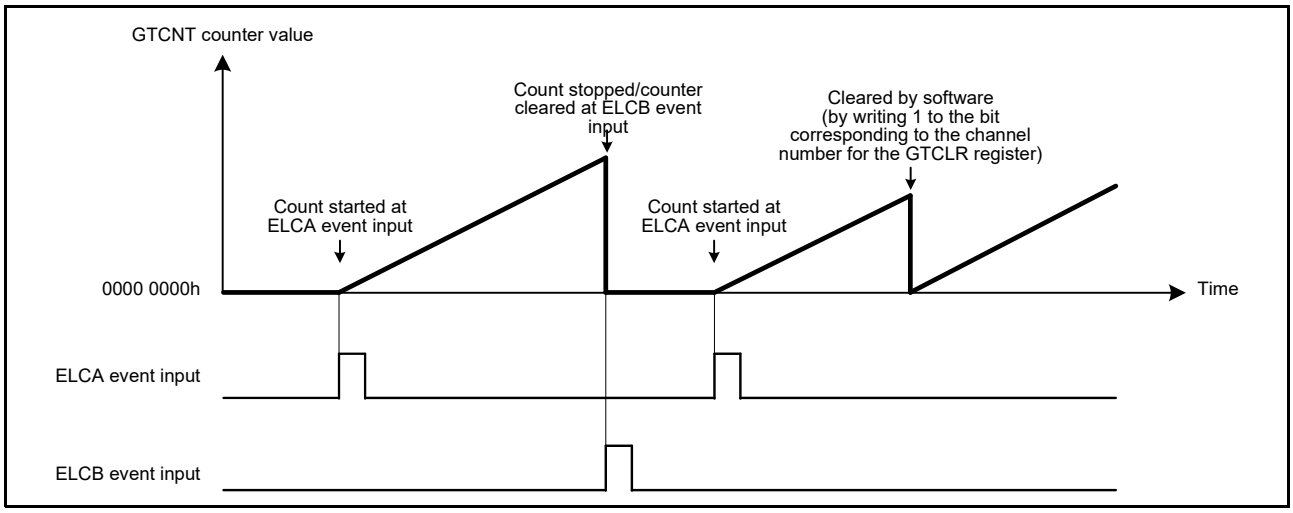


Figure 22.107 Examples of Counter Clearing Operation by Hardware Source (Sawtooth-Wave Up-Counting, Started at ELCA event input, and Count Stopped/Counter Cleared at ELCB event input)

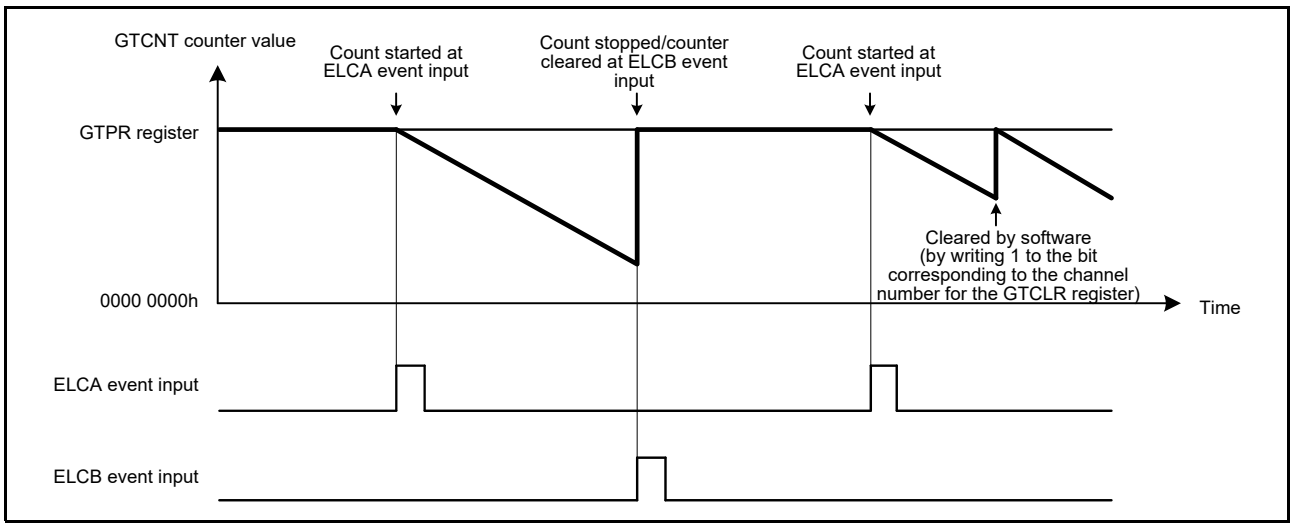


Figure 22.108 Examples of Counter Clearing Operation by Hardware Source (Sawtooth-Wave Down-Counting, Started at ELCA event input, and Count Stopped/Counter Cleared at ELCB event input)

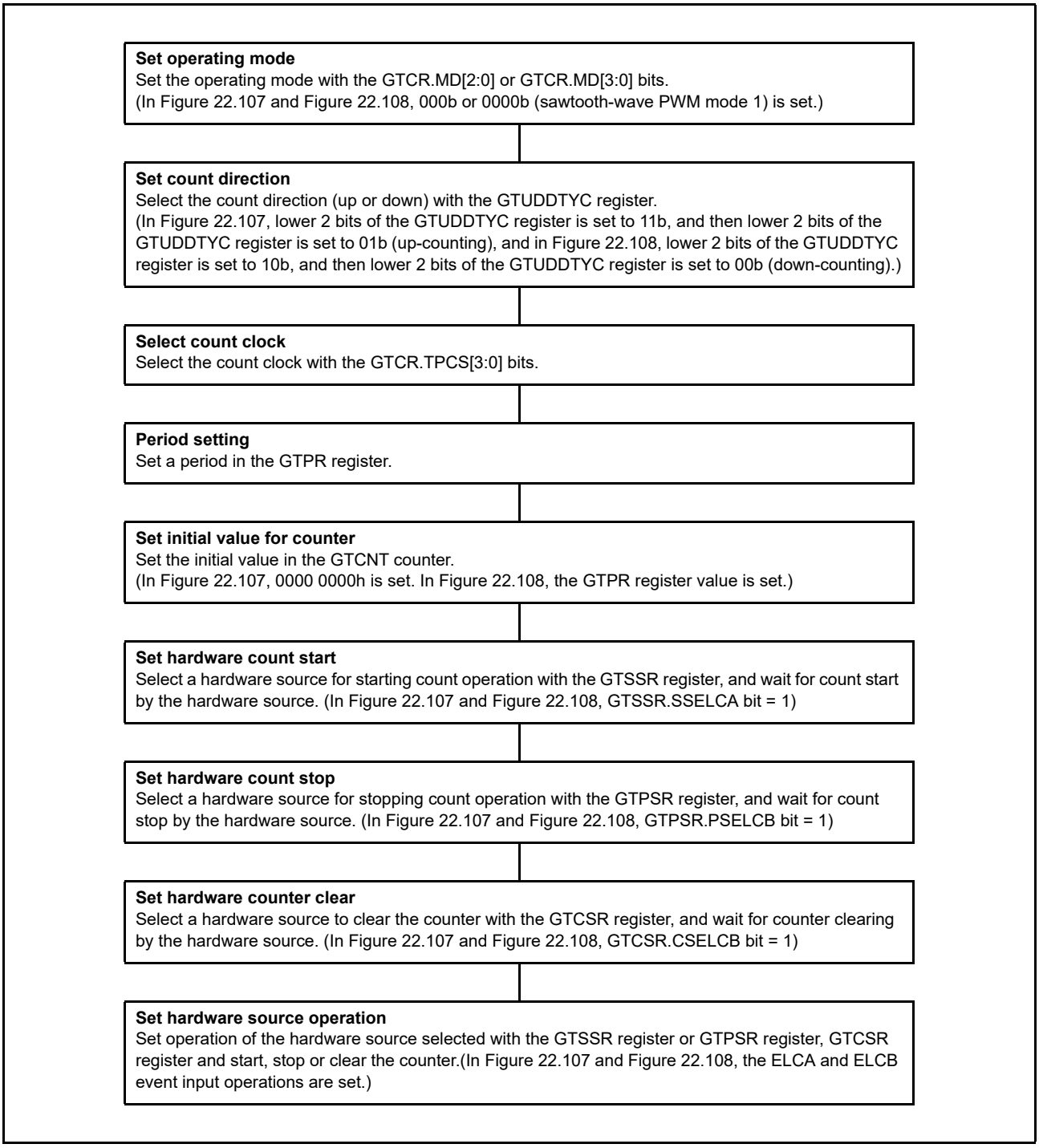


Figure 22.109 Example for Setting Counter Clearing Operation by Hardware Source

The GTCIV/GTCIU interrupt (overflow/underflow interrupt) does not occur when the counter is cleared by a hardware source. In the same way, the GTCIV/GTCIU interrupt does not occur when the counter is cleared by software.
Figure 22.110 shows the relationship between counter clearing by a hardware source and the GTCIV interrupt.

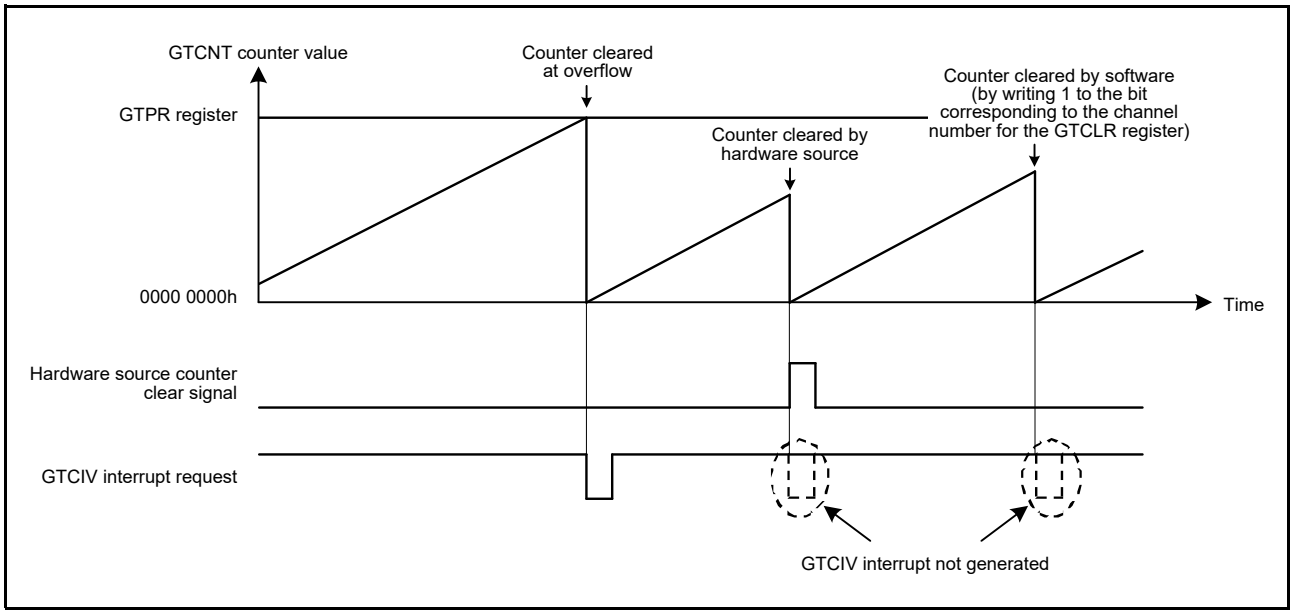


Figure 22.110 Relationship between Counter Clearing by Hardware Source and GTCIV Interrupt

Figure 22.111 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the PCLKA signal is used as the counter clock for the GTCNT counter.
The GTCNT counter is cleared when counting is in progress after the GPTW has detected the internal clearing signal.

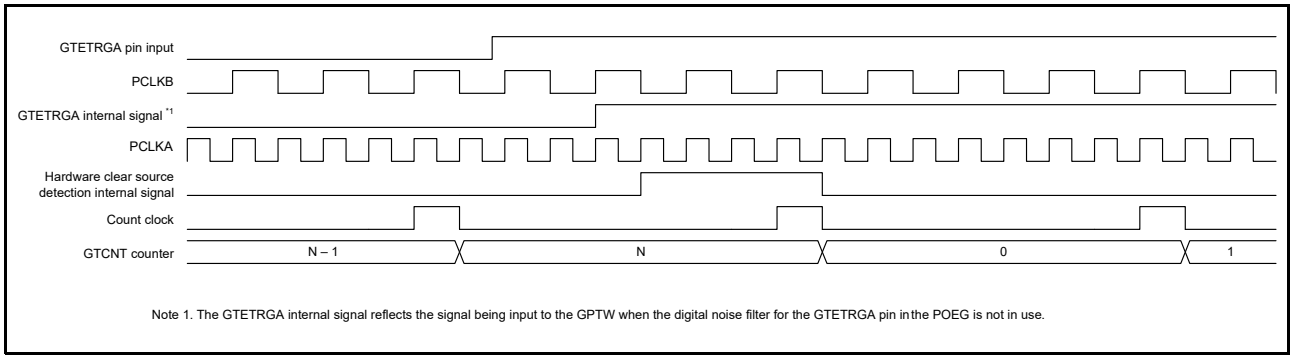


Figure 22.111 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the PCLKA Frequency)

Figure 22.112 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin when a clock signal produced by frequency-dividing the PCLKA signal is used as the counter clock for the GTCNT counter.
The GTCNT counter is cleared when counting is in progress after the GPTW has detected the internal clearing signal.

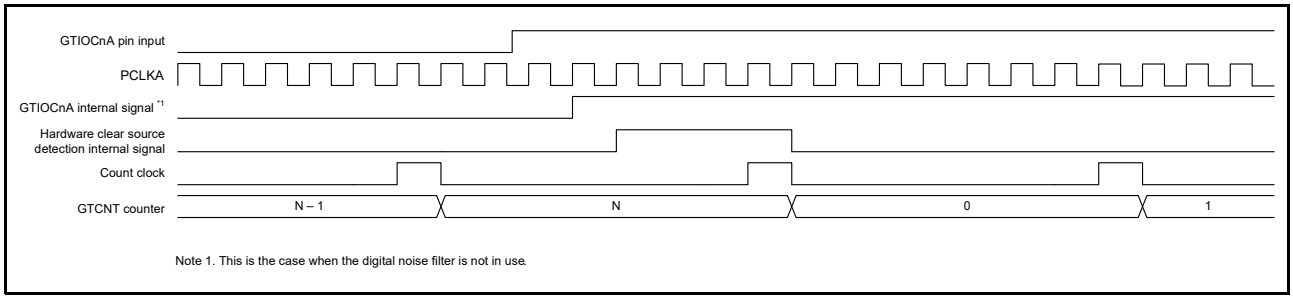


Figure 22.112 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the PCLKA Frequency)

Figure 22.113 shows an example of the timing of operations to clear the counter in response to event input from the ELCA when a clock signal produced by frequency-dividing the PCLKA signal is used as the counter clock for the GTCNT counter.
This is an example of operations to clear the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with PCLKA. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is cleared when counting is in progress after the GPTW has detected the internal clearing signal.

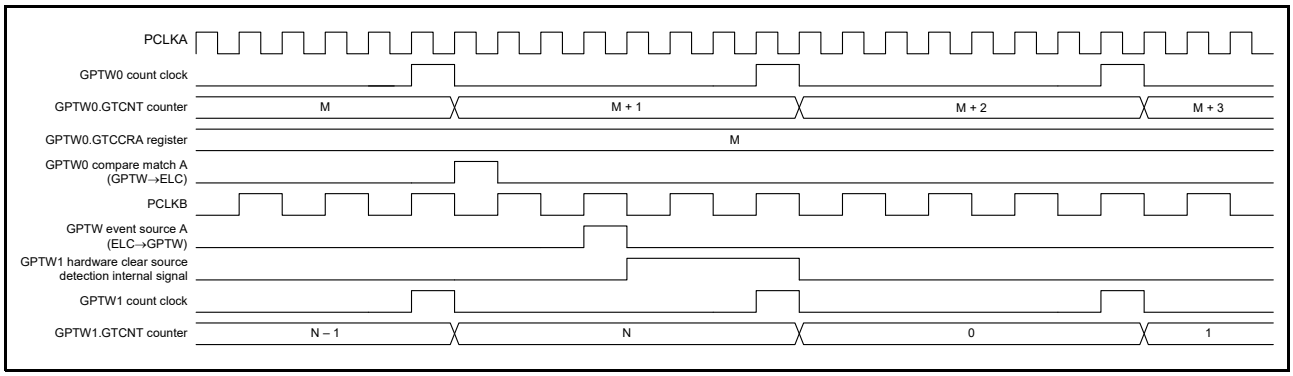


Figure 22.113 Example of the Timing of Operations for Counter Clearing in Response to Event Input from the ELCA (During the Counting of Cycles of Clock Signal Produced by Dividing the PCLKA Frequency)

Figure 22.114 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin input when counting is triggered by a hardware source.
The GTCNT counter is cleared in synchronization with PCLKA after the GPTW has detected the internal clearing signal.

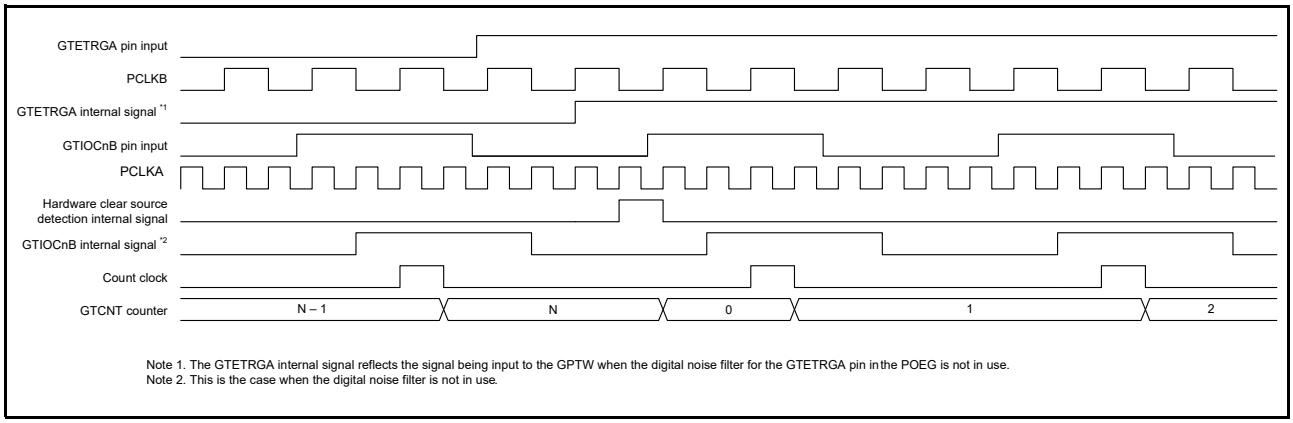


Figure 22.114 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During Counting Triggered by Hardware Source)

Figure 22.115 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin input when counting is triggered by a hardware source.
The GTCNT counter is cleared in synchronization with PCLKA after the GPTW has detected the internal clearing signal.

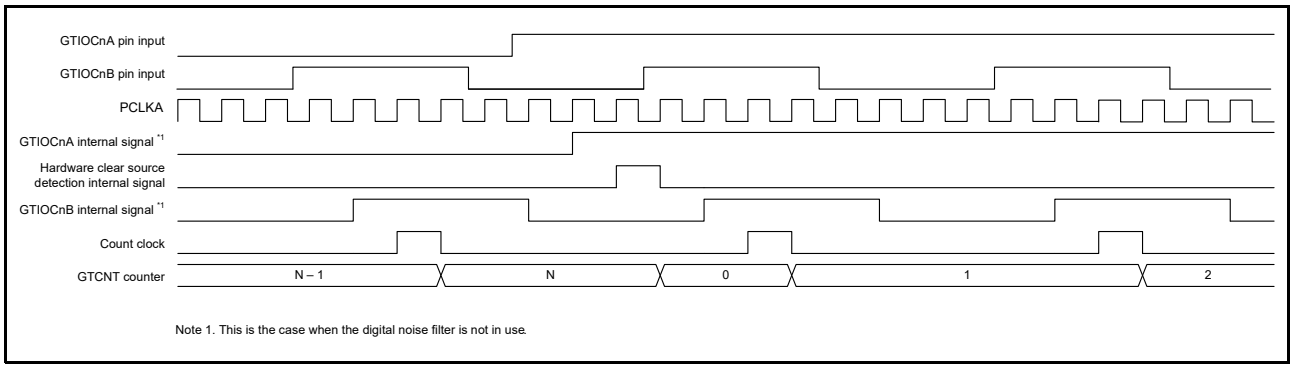


Figure 22.115 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During Counting Triggered by Hardware Source)

Figure 22.116 shows an example of the timing of operations for clearing the counter in response to the input of an event signal from the ELCA when counting is triggered by a hardware source.
This is an example of operations to clear the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with PCLKA. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is cleared in synchronization with PCLKA after the GPTW has detected the internal clearing signal.

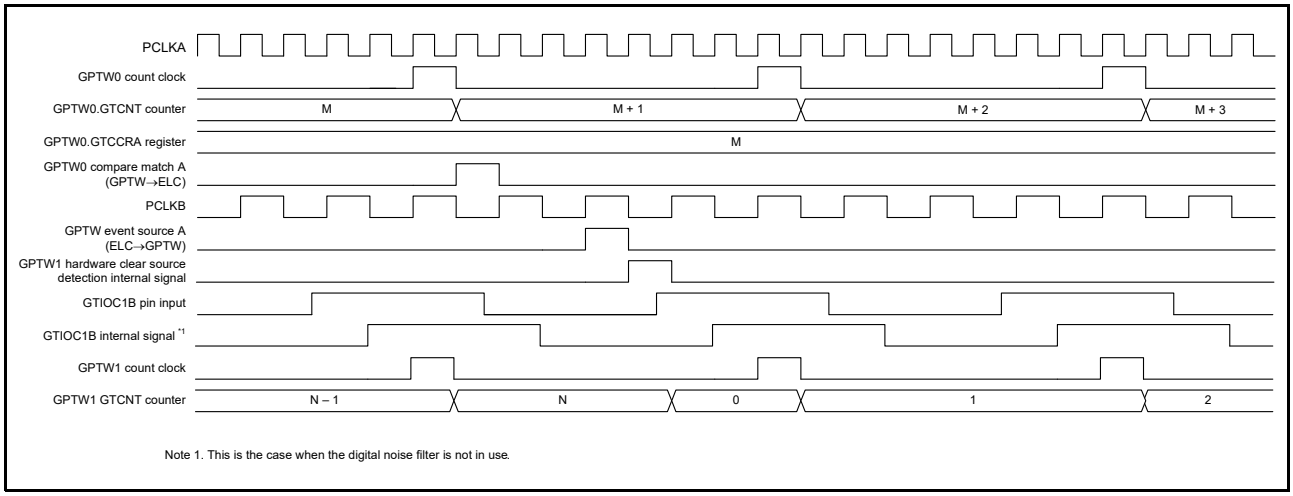


Figure 22.116 Example of the Timing of Operations for Counter Clearing in Response to Event Input from the ELCA (During Counting Triggered by Hardware Source)

22.3.8 Synchronous Operation

Synchronous operation on channels (synchronous start/stop/clear operation) can be performed.

22.3.8.1 Synchronous Operation by Software

The count operations for respective channels can be started, stopped and cleared simultaneously by setting multiple bits of the GTSTR, GTSTP, and GTCLR registers to 1 at the same time.
Count start with phase differences among channels is possible firstly by setting the GTCNT counter value before counting starts, and then, by setting simultaneously multiple bits in the GTSTR register to 1.
The GTCNT counter value settings are synchronized by setting the GTCR.SSCEN bit to 1, and can be written simultaneously to multiple channels set in the same group by the GTCR.SSCGRP[1:0] bits. Synchronous sets are not valid for channels that are set to complementary PWM mode.
When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to value and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.
Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channels are not exact same.
The Figure 22.117 shows an example of simultaneous start/stop/clearing of four channels by software, and Figure 22.118 shows an example of phase shift start among four channels by software. Figure 22.119 to Figure 22.121 show examples of simultaneous start/stop/clearing with different count period.

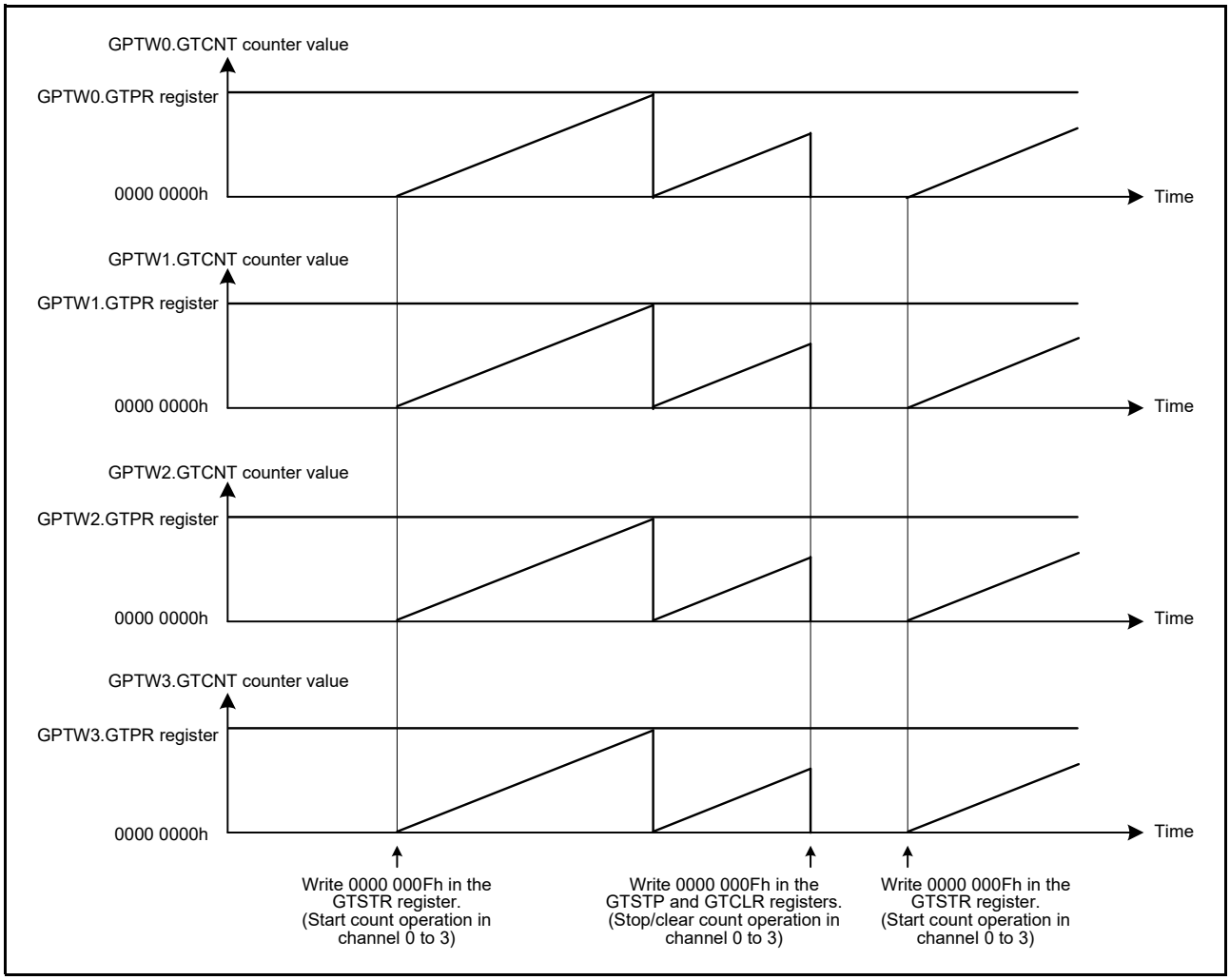


Figure 22.117 Example of Simultaneous Start/Stop/Clearing Operation by Software (with the Same Count Period (GTPR Register Value))

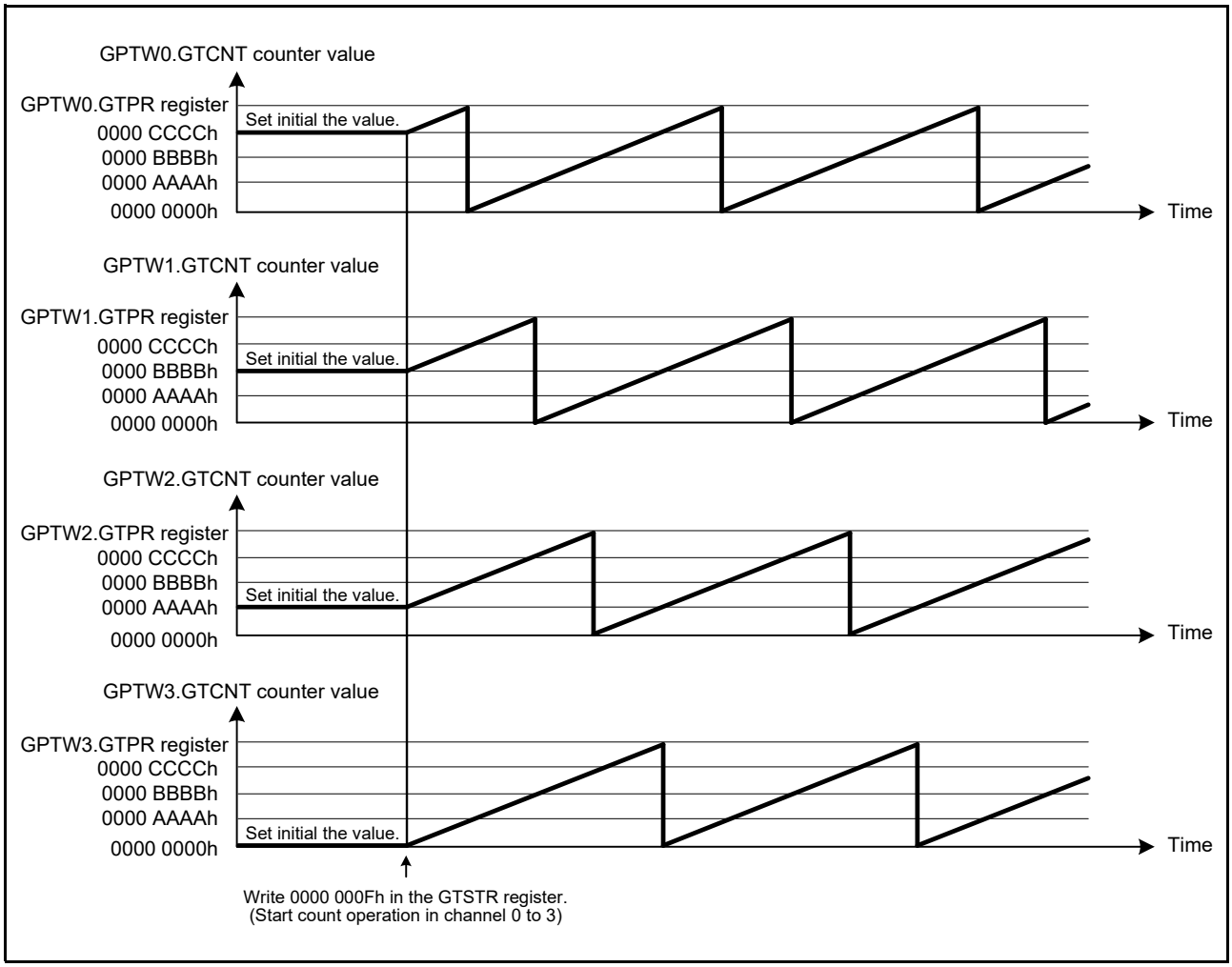


Figure 22.118 Example of Software Phase Shift Start (with Same Count Period (GTPR Register Value))

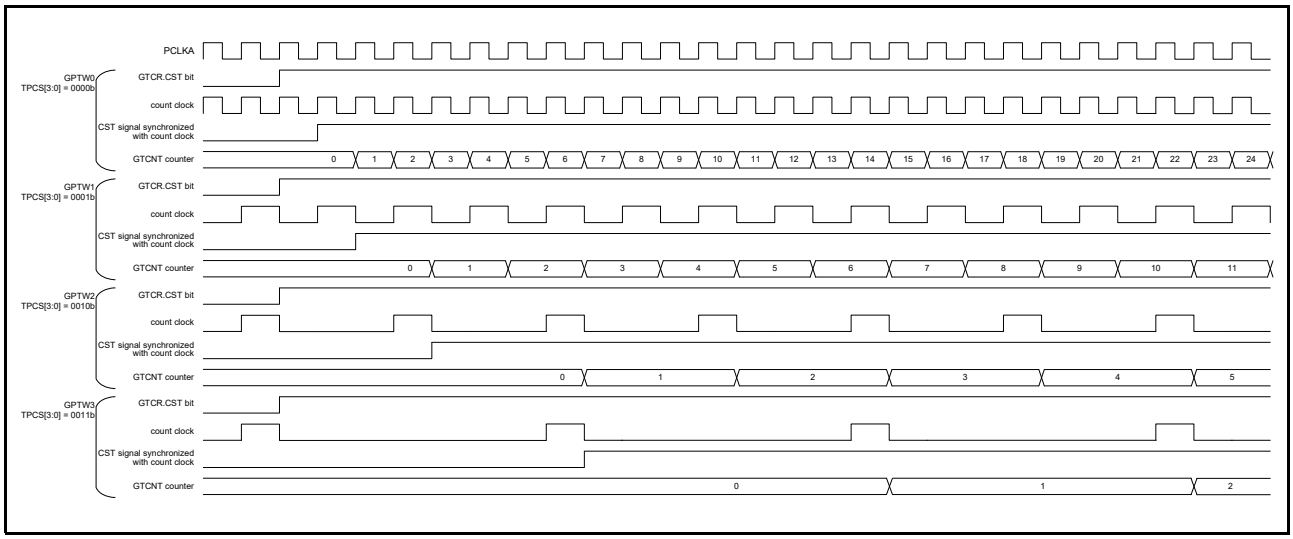


Figure 22.119 Example of Simultaneous Start Operation by Software (with Different Count Period)

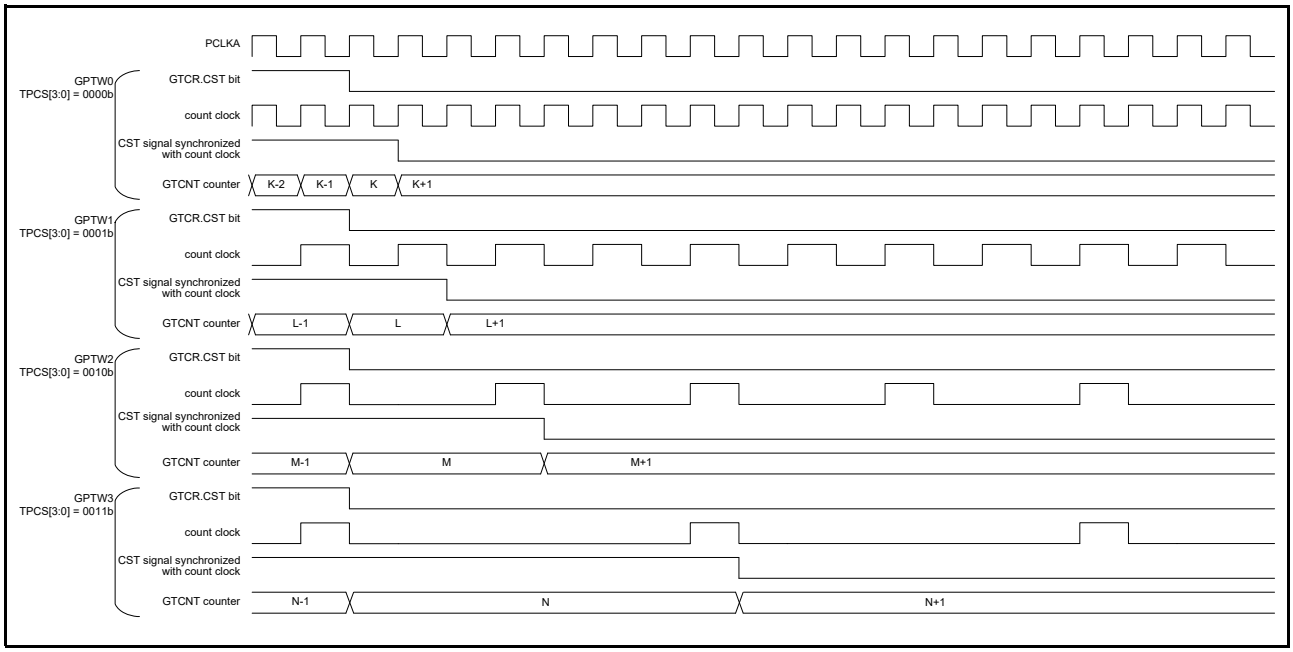


Figure 22.120 Example of Simultaneous Stop Operation by Software (with Different Count Period)

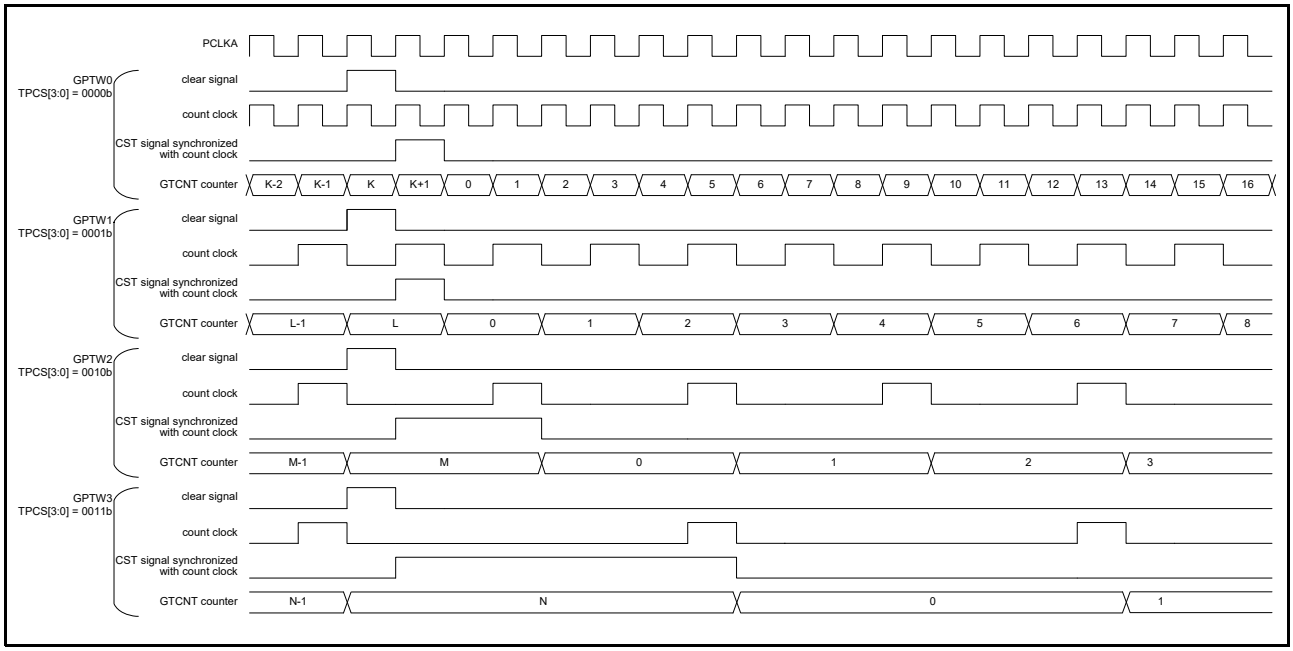


Figure 22.121 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

22.3.8.2 Synchronous Operation by Hardware Source

The count operations for respective channels can be started/stopped/cleared simultaneously by following hardware sources: external trigger input, and ELC event input.

Synchronous operation by the GTIOCnA and GTIOCnB pin inputs (n = 0 to 7) can be performed by setting the ELC event by input capture as a hardware source.

Figure 22.122 shows an example of simultaneous start/stop/clearing operation of four channels by hardware source.

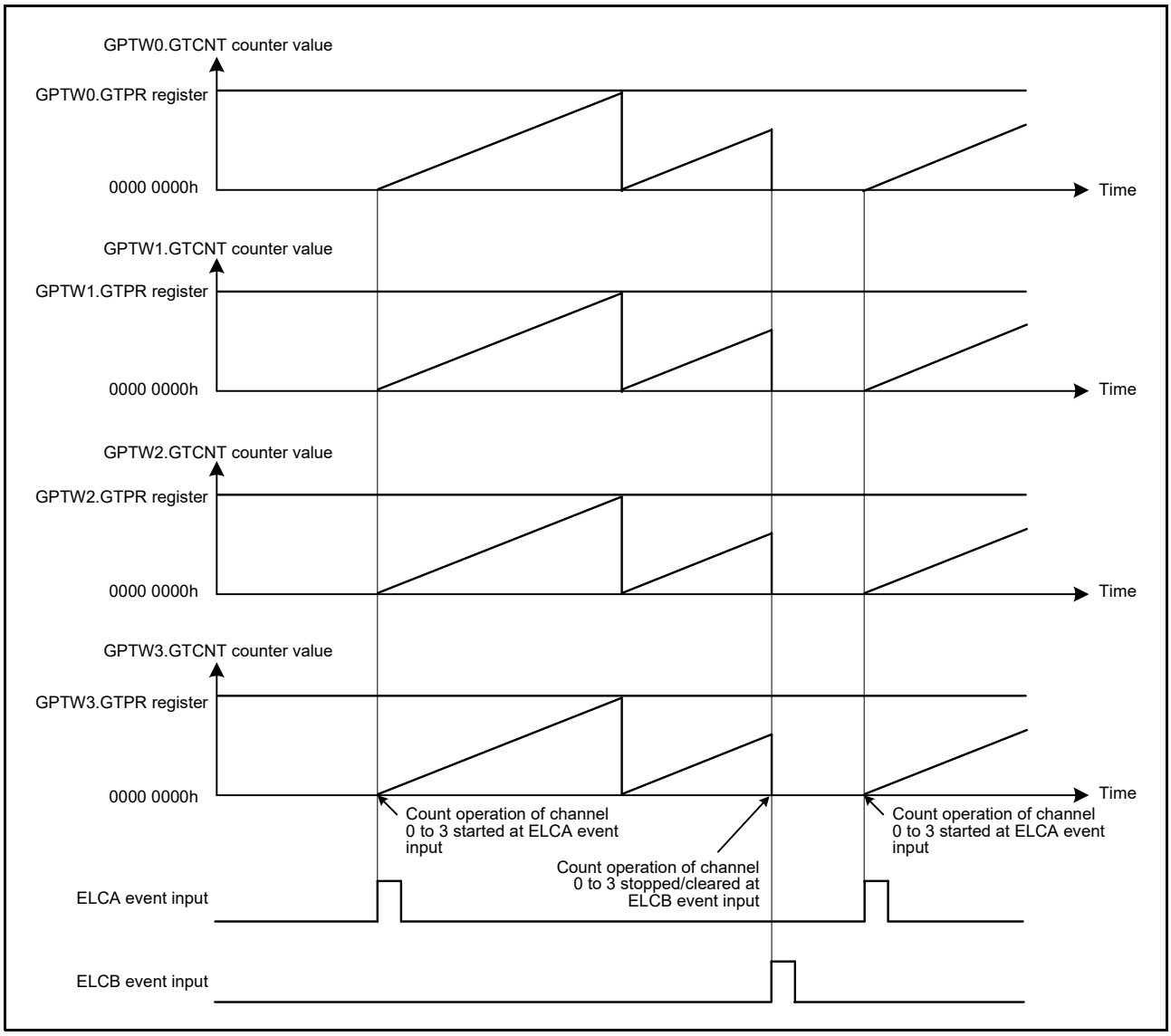


Figure 22.122 Example of Simultaneous Start/Stop/Clearing Operation by Hardware Source (with Same Count Period (GTPR Register Value))

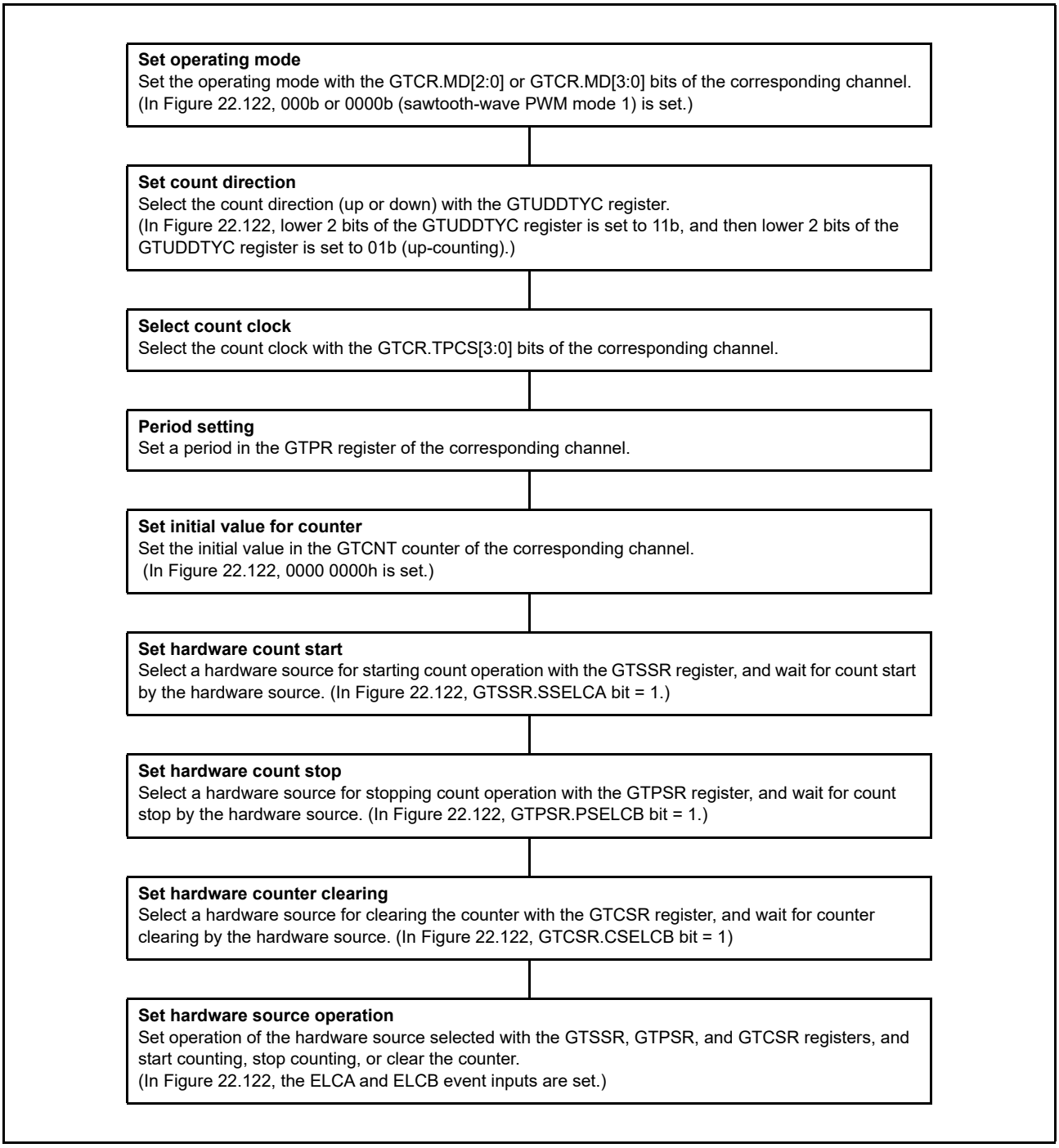


Figure 22.123 Example for Setting Simultaneous Start by Hardware Source

22.3.8.3 Synchronous Clear Operation by Inter Channel Cooperation

The counters of other channels can be cleared at the same time with the counter clear of some channel caused by the compare match, input capture, sawtooth wave up-count overflow, sawtooth wave down-count underflow, and the GTIOCnA/GTIOCnB pin input selected in the GTCSR register.

The counter clear caused by GTCLR register, external trigger input and the ELC input can perform as synchronous clear when the same counter clear factor is set to target channels of synchronous clear. Therefore, these factors are not prepared for the synchronous clear operation by inter channel cooperation.

Set the channel that generates the synchronization clear source and the channel that is synchronized clear in the same synchronous set/clear group with the GTCR.SSCGRP[1:0] bits.

Similar to the synchronous operation in section 22.3.8.1, Synchronous Operation by Software, if the count clock selected by the GTCR.TPCS[3:0] bits is different for each channel, the synchronous clear operation cannot be performed at exact same timing. Similar to the example of simultaneous clearing operation by software in Figure 22.121, if the count clock is different for each channel, the synchronous clear factor is synchronized with the count clock of each channel before counter clearing.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Table 22.15 lists the settings of the registers used to set sources for synchronous clearing. Figure 22.124 shows an example of synchronous clear operation by inter channel cooperation and Figure 22.125 shows an example for setting synchronous clear operation by inter channel cooperation.

Table 22.15 Settings of the Registers Used to Set Sources for Synchronous Clearing

Sources for Synchronous Clearing by Inter Channel Cooperation	Register.Bit(s)	Value
GTCCRA register compare match/input capture	GTINTAD.SCFA	1b
	GTCSR.CSCMSC[2:0]	001b
GTCCRB register compare match/input capture	GTINTAD.SCFB	1b
	GTCSR.CSCMSC[2:0]	010b
GTCCRC register compare match	GTINTAD.SCFC	1b
	GTCSR.CSCMSC[2:0]	011b
GTCCRD register compare match	GTINTAD.SCFD	1b
	GTCSR.CSCMSC[2:0]	100b
GTCCRE register compare match	GTINTAD.SCFE	1b
	GTCSR.CSCMSC[2:0]	101b
GTCCRF register compare match	GTINTAD.SCFF	1b
	GTCSR.CSCMSC[2:0]	110b
Sawtooth wave up-count overflow	GTINTAD.SCFPO	1b
Sawtooth wave down-count underflow	GTINTAD.SCFPU	1b
Clearing due to input on the GTIOCnA or GTIOCnB pin	GTCR.SCGTIOC	1b

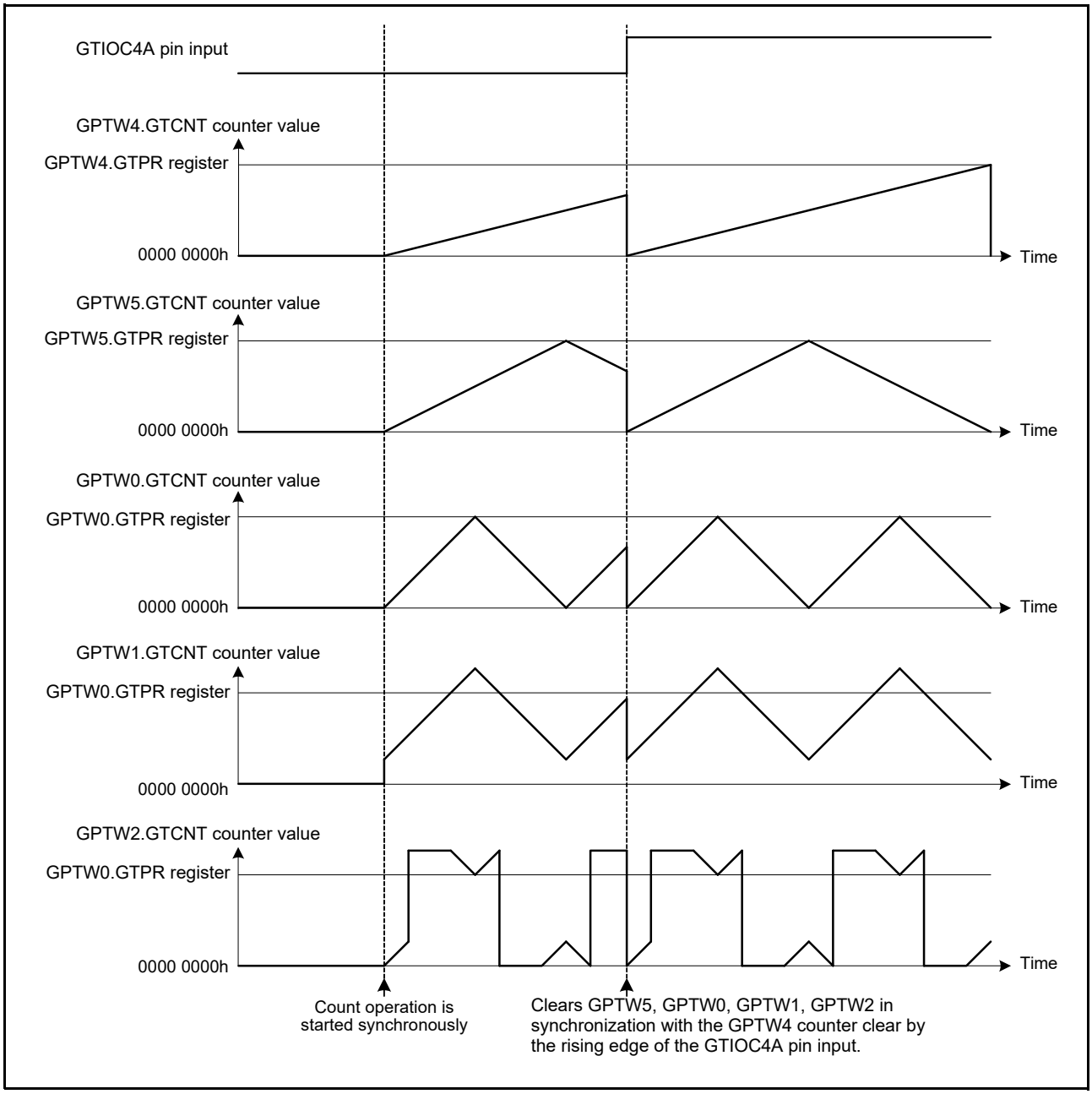


Figure 22.124 Example of Synchronous Clear Operation by Inter Channel Cooperation (GPTW4 is sawtooth wave and counter is cleared by the rising edge of the GTIOC4A, GPTW5 is triangle wave, GPTW0, GPTW1, GPTW2 are complementary PWM mode. GPTW0, GPTW1, GPTW2, GPTW4, GPTW5 are the same synchronous set/clear group)

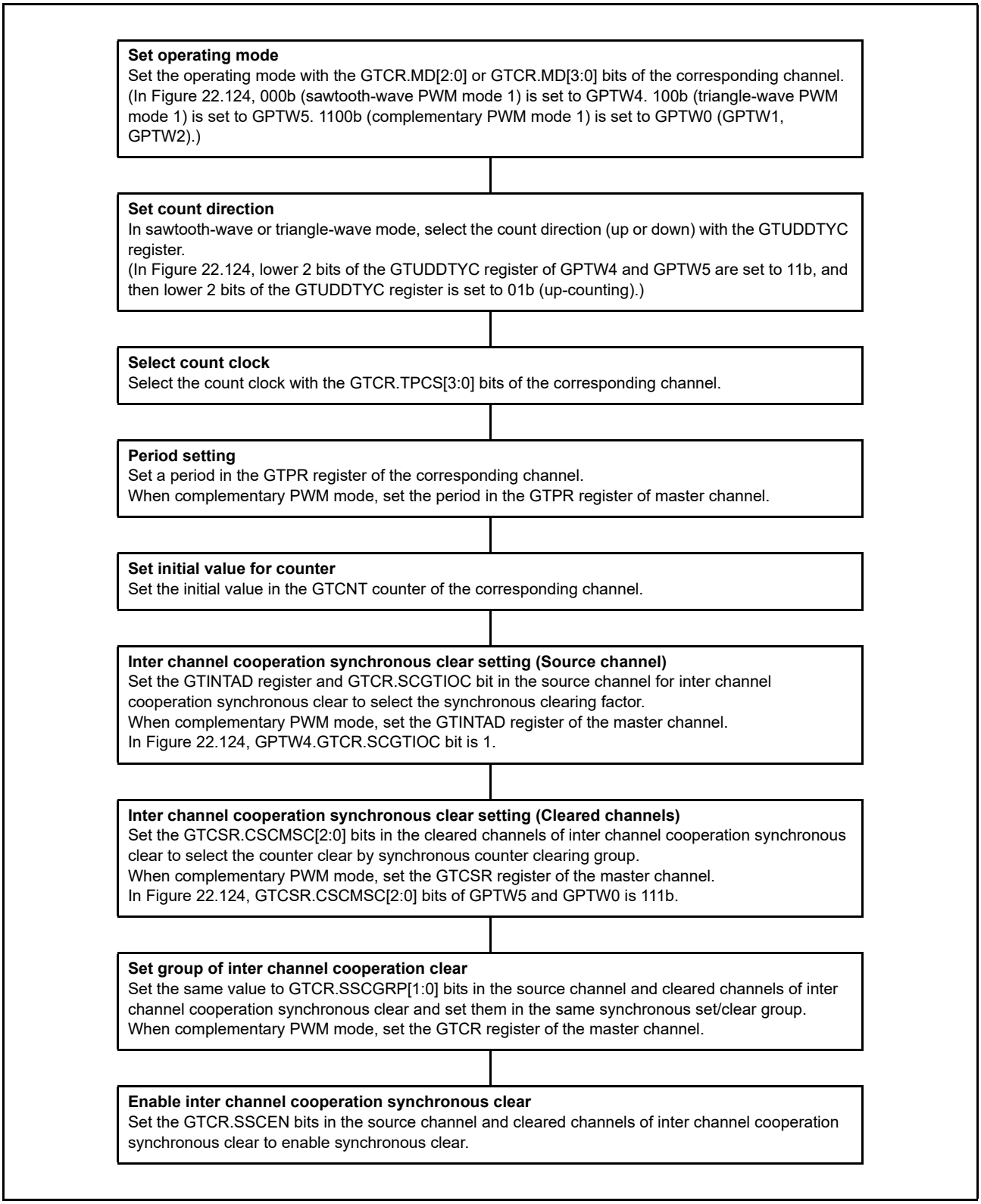


Figure 22.125 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation

22.3.8.4 Input Capture Operation by Inter Channel Cooperation

The events of compare match, input capture, sawtooth wave up-count overflow, sawtooth wave down-count underflow, the crest and trough of triangle wave and complementary PWM mode and the count clock can be used as the input capture factor for the GTCCRm register (m = A or B) of other channels.

The input capture factor by inter channel cooperation can be set by the GTICCR register of the channel that generates the input capture factor. And input capture by other channel factors can be enable by GTICmSR.mSOC bit (m = A or B) in the captured channels. The channels that generate input capture factors and the channels that are captured are set to the same input capture group by GTICCR.ICmGRP[1:0] bit (m = A or B).

Figure 22.126 shows an example of input capture operation by inter channel cooperation and Figure 22.127 shows an example for setting input capture operation by inter channel cooperation.

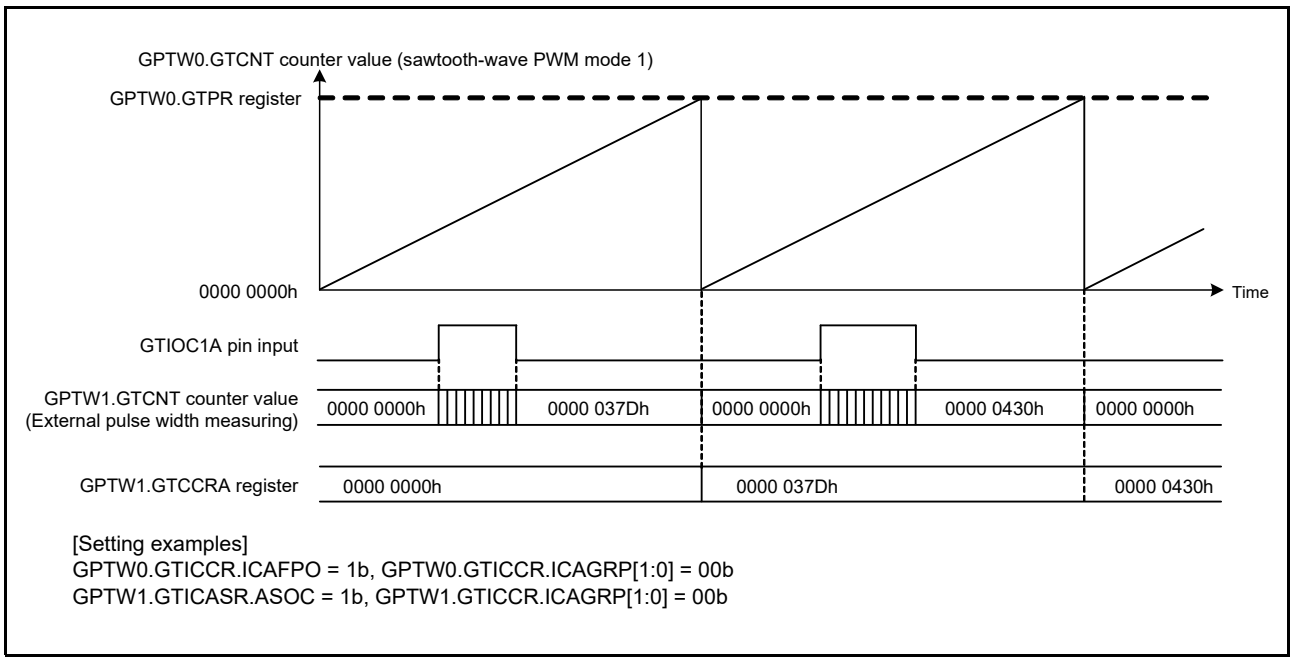


Figure 22.126 Example of Input Capture Operation by Inter Channel Cooperation(Channel 1 is captured by the overflow of Channel 0)

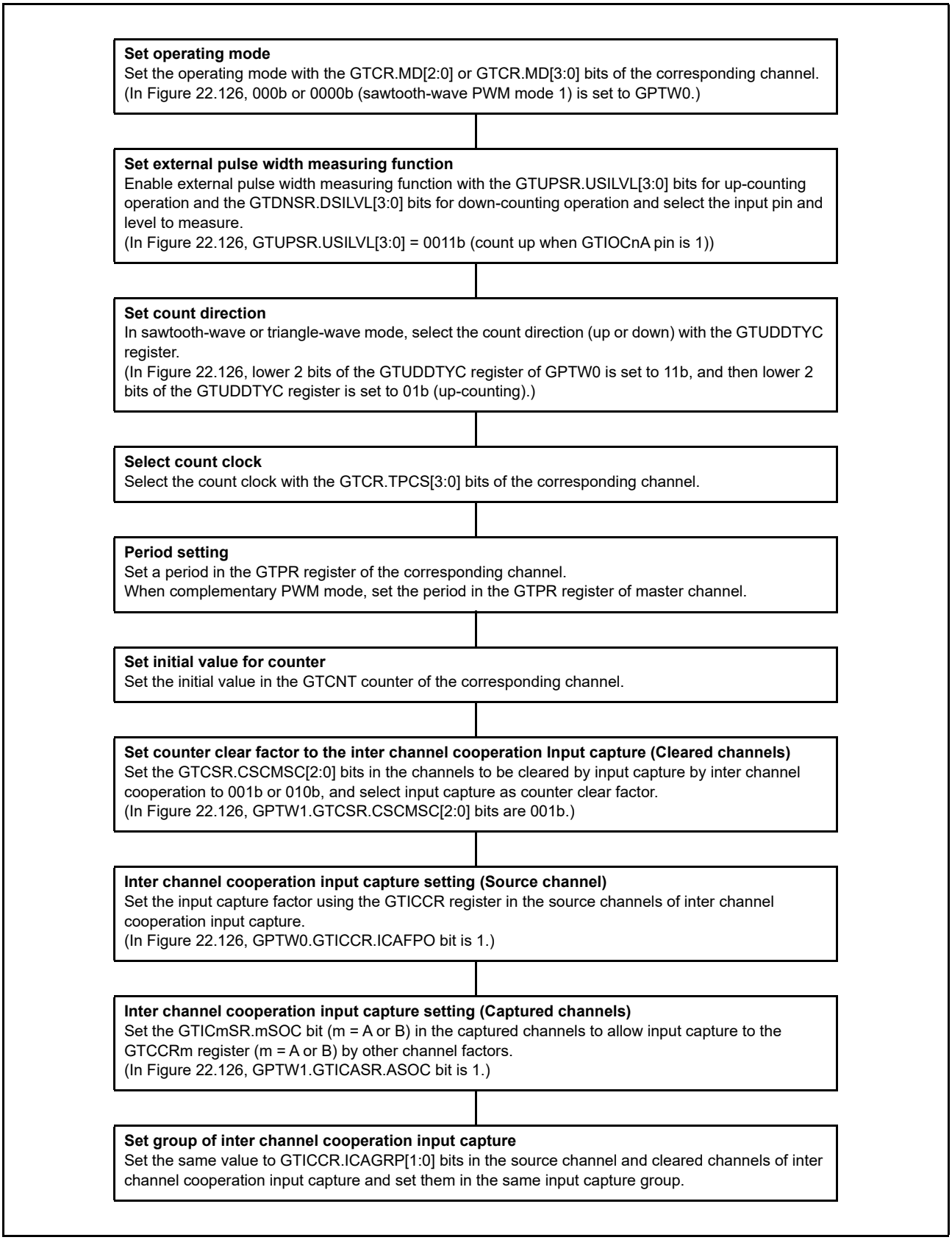


Figure 22.127 Example for Setting Input Capture Operation by Inter Channel Cooperation

22.3.9 PWM Output Operation Examples

(1) Synchronous PWM Output

The GPTW can output two phases of linked PWM waveforms for one channel, or 16 phases of linked PWM waveforms for a maximum of 8 channels by synchronizing operation of the channels.

Figure 22.128 shows an example in which four channels perform synchronous operation in sawtooth-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA pin (n = 0 to 3) is set so that it will output low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output low as the initial output, high at a GTCCRB register compare match, and low at the end of the cycle.

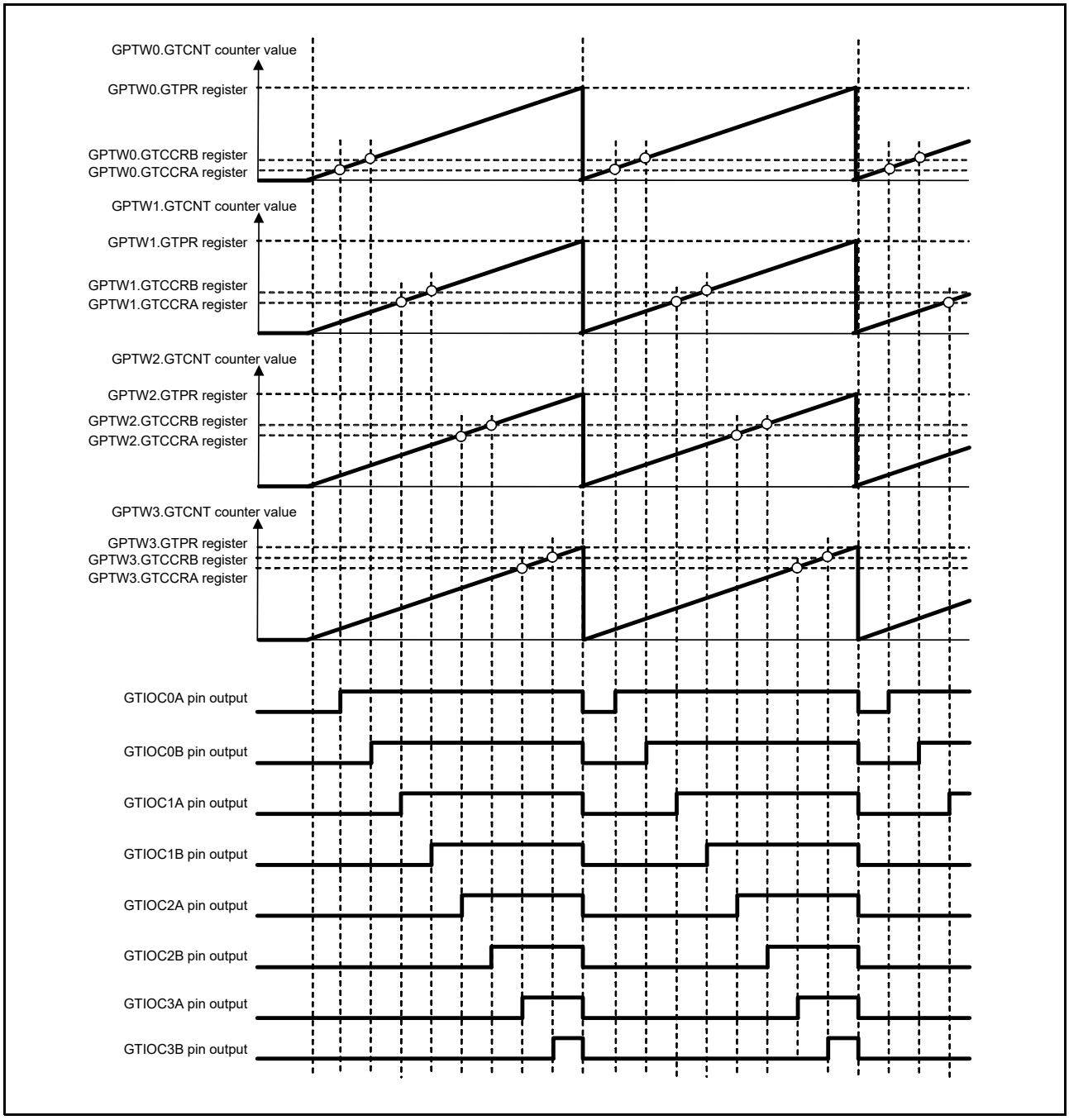


Figure 22.128 Example of Synchronous PWM Output

(2) Three-Phase Sawtooth-Wave Complementary PWM Output

Figure 22.129 shows an example in which three channels perform synchronous operation in sawtooth-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, low at a GTCCRB register compare match, and high at the end of the cycle.

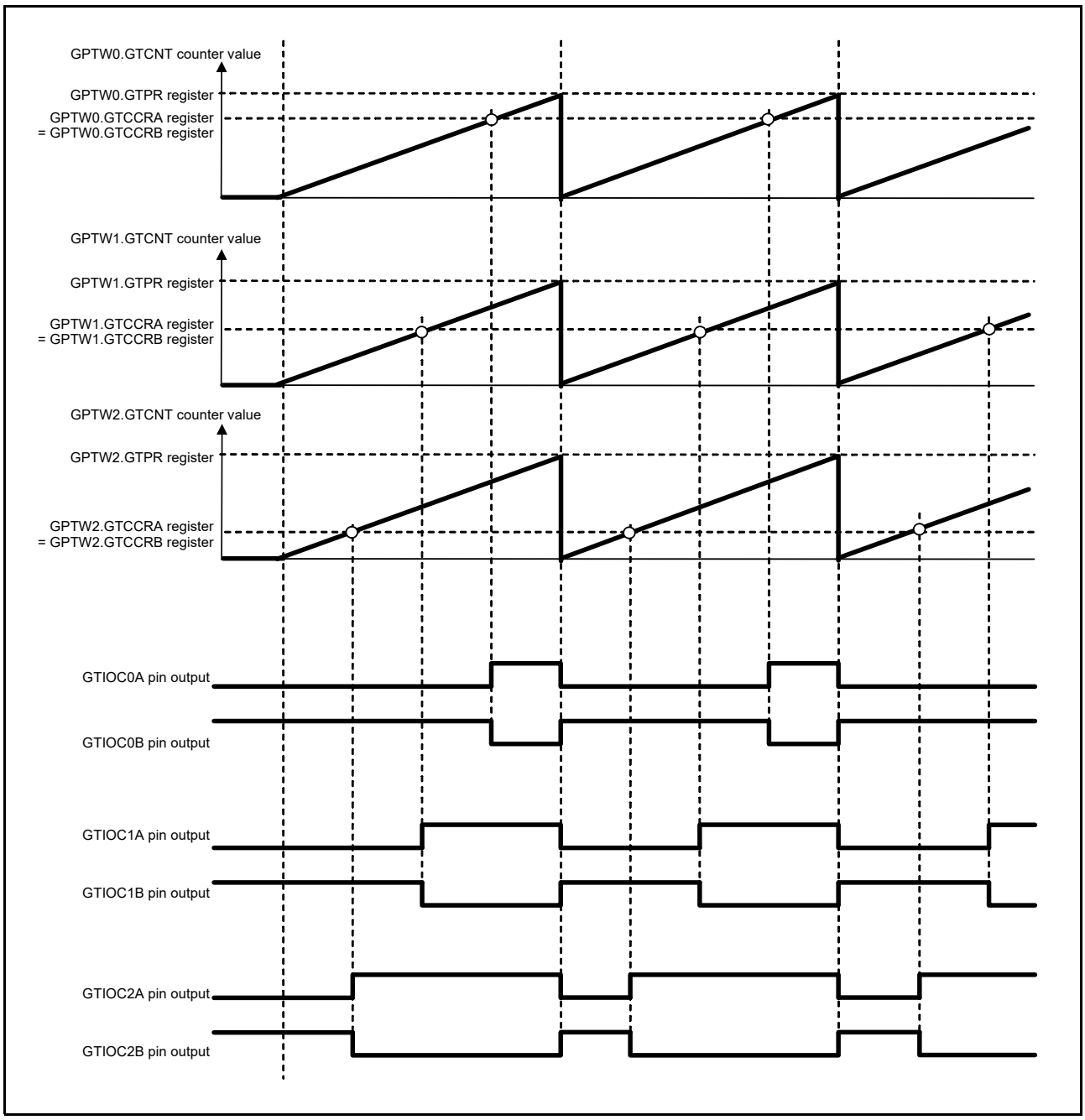


Figure 22.129 Example of Three-Phase Sawtooth-Wave Complementary PWM Output

(3) Three-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 22.130 shows an example in which three channels perform synchronous operation in sawtooth-wave one-shot pulse mode with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

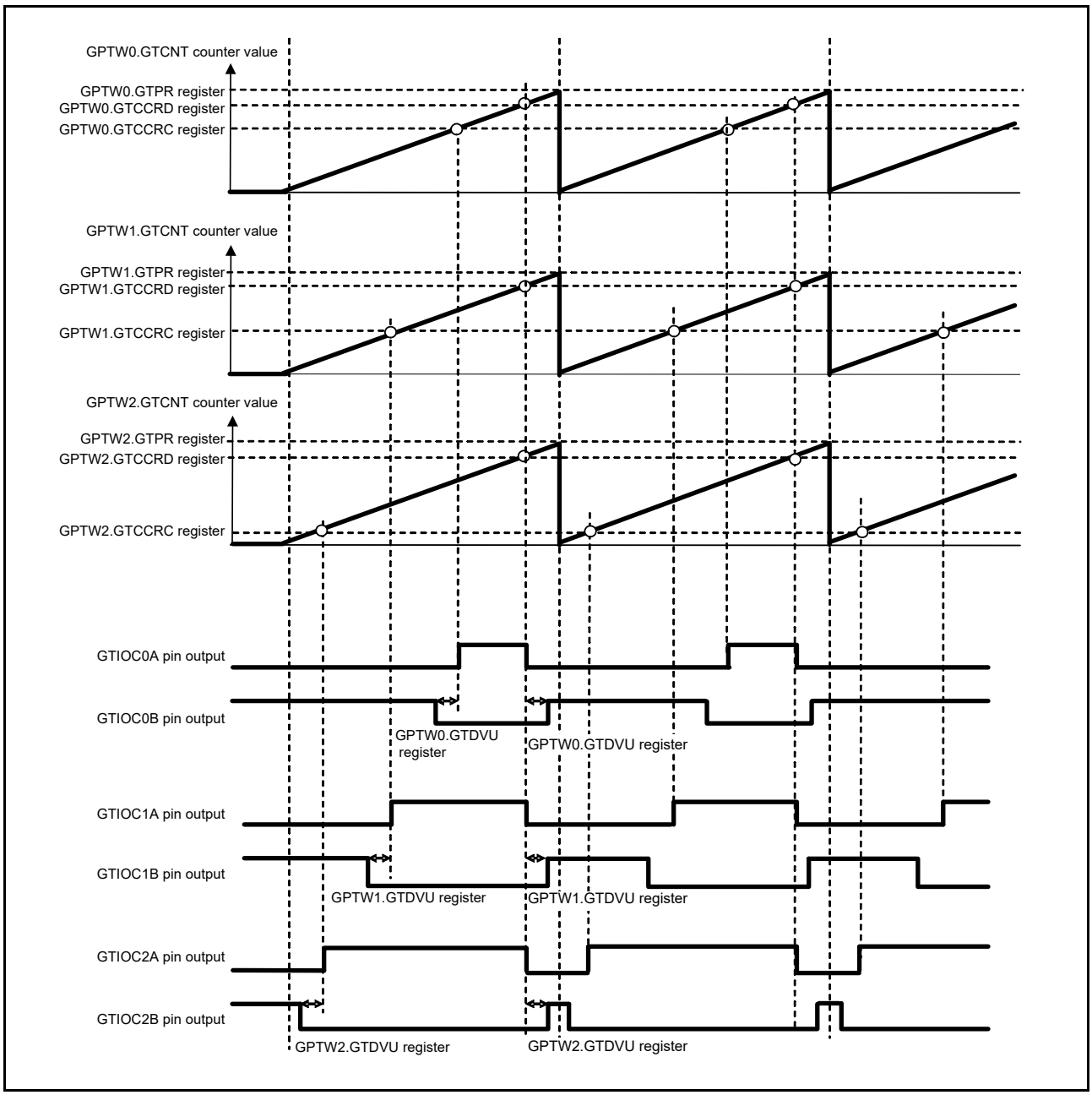


Figure 22.130 Example of Three-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

(4) Three-Phase Triangle-Wave Complementary PWM Output

Figure 22.131 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

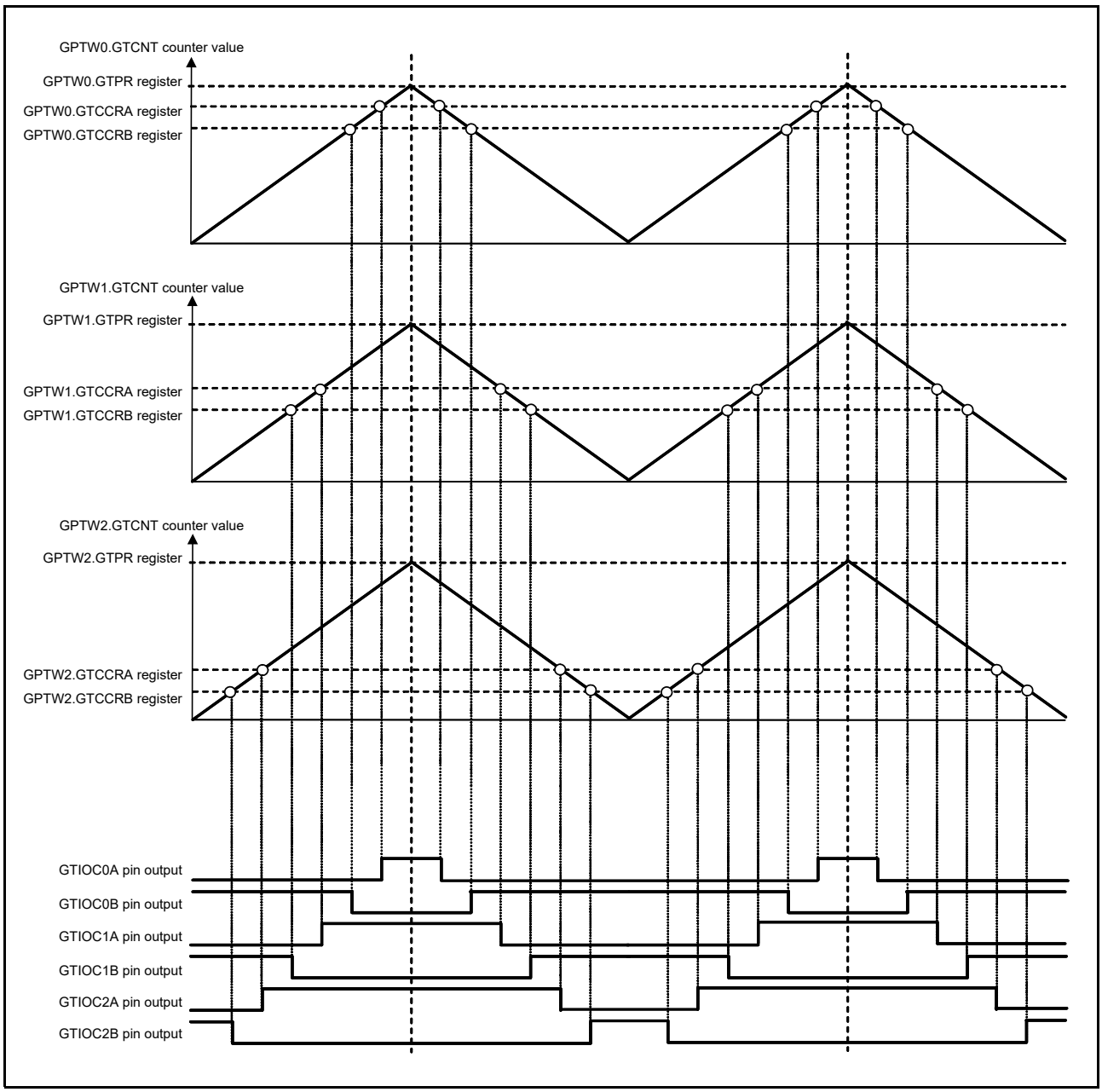


Figure 22.131 Example of Three-Phase Triangle-Wave Complementary PWM Output

(5) Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 22.132 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

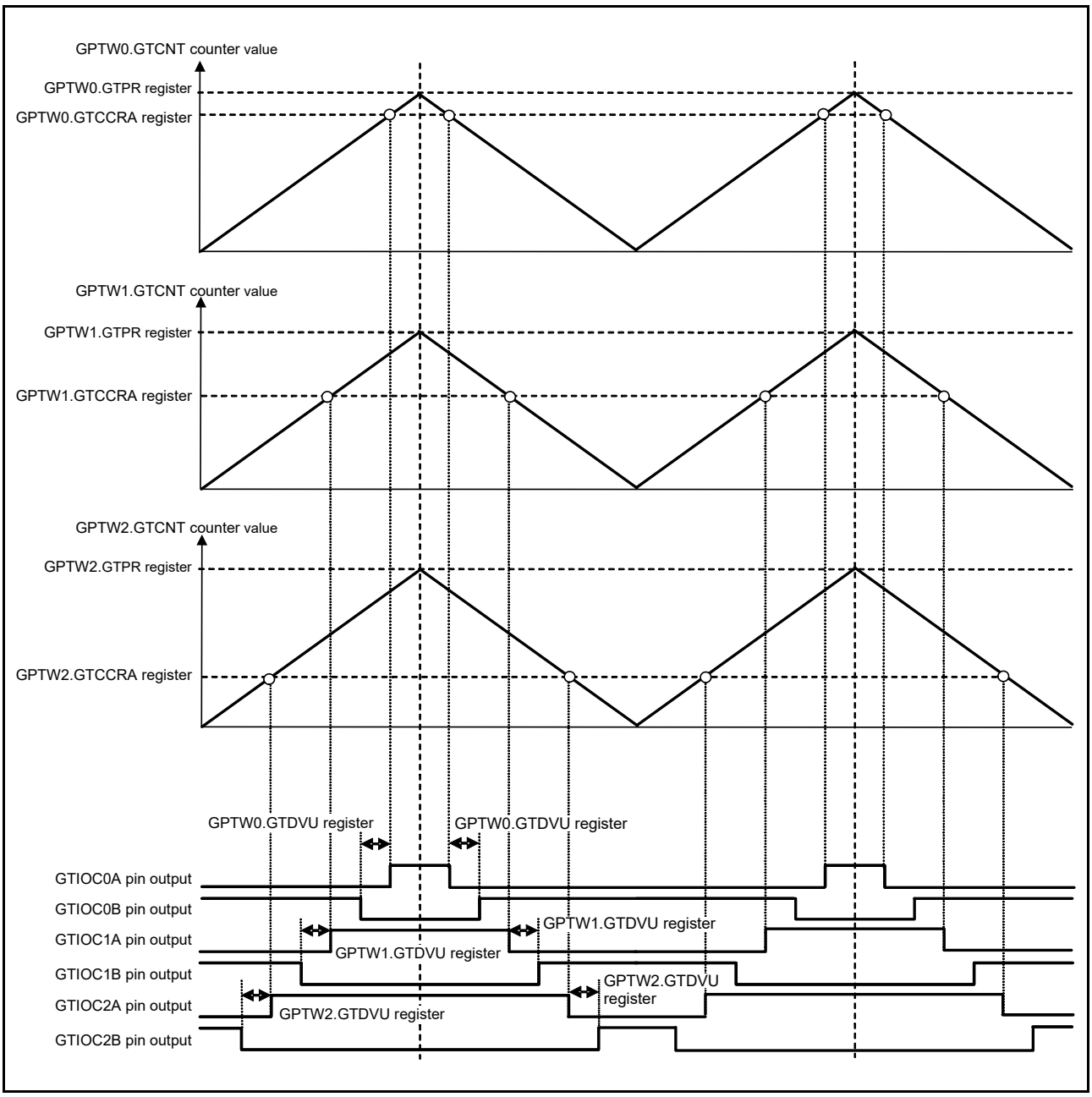


Figure 22.132 Example of Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

22.3.10 Phase Counting Mode

The GTCNT counter can be up-counting or down-counting by detecting the phase difference between the GTIOCnA pin input (n = 3 to 7) and the GTIOCnB pin input.
Detection of the phase difference can be performed by setting the relationship between the edge and the level of the GTIOCnA pin input and the GTIOCnB pin input using the GTUPSR and GTDNSR registers for input combination. Refer to section 22.3.1.1, Counter Operation for count operation.
Figure 22.133 to Figure 22.142 show the examples of setting procedure for the phase counting mode 1 to 5, and the Table 22.16 to Table 22.25 show the setting of the GTUPSR and GTDNSR registers and up-counting and down-counting conditions.

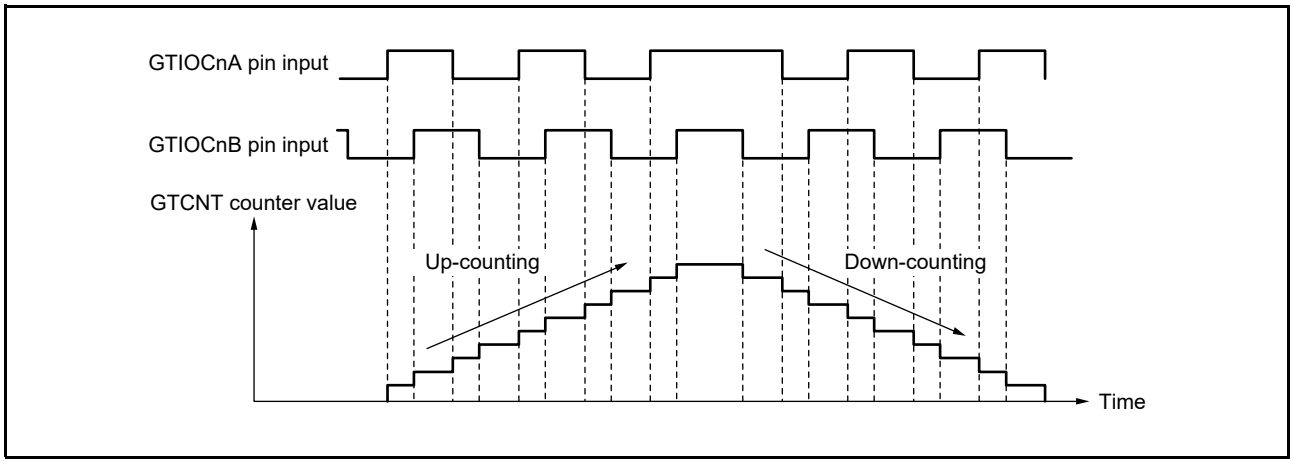


Figure 22.133 Example of setting procedure for the phase counting mode 1 (n = 3 to 7)

Table 22.16 Up-counting and down-counting conditions for the phase counting mode 1 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0000 6900h GTDNSR = 0000 9600h
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

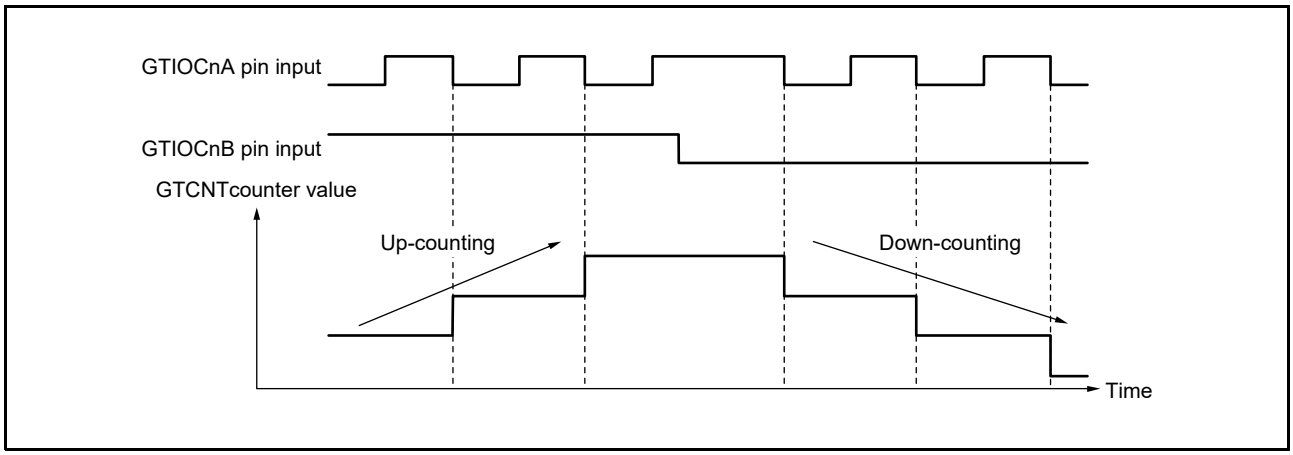


Figure 22.134 Example of setting procedure for the phase counting mode 2 (n = 3 to 7)

Table 22.17 Up-counting and down-counting conditions for the phase counting mode 2 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 0400h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

↑: Rising edge

↓: Falling edge

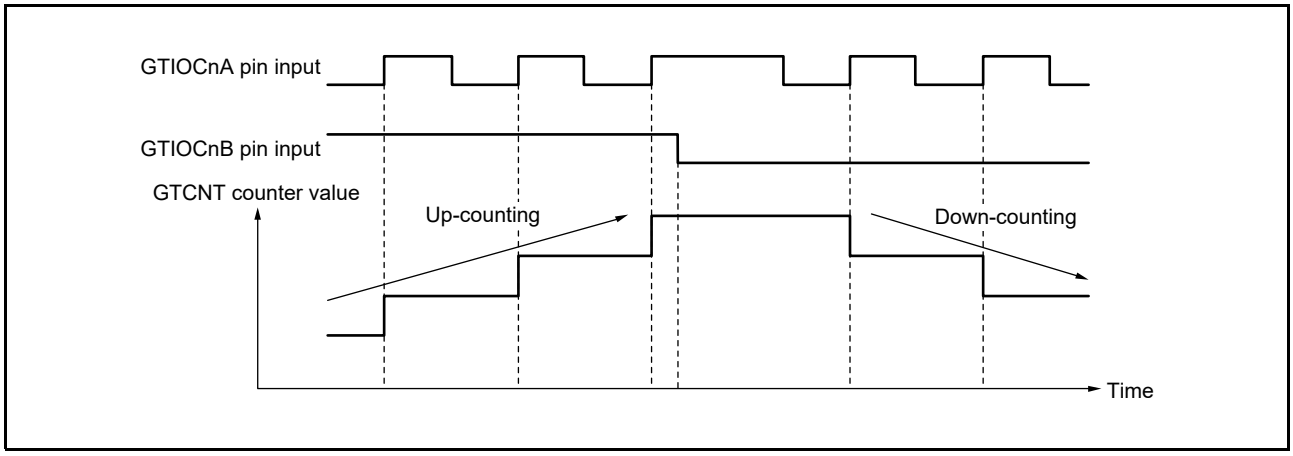


Figure 22.135 Example of setting procedure for the phase counting mode 2 (n = 3 to 7)

Table 22.18 Up-counting and down-counting conditions for the phase counting mode 2 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0200h GTDNSR = 0000 0100h
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

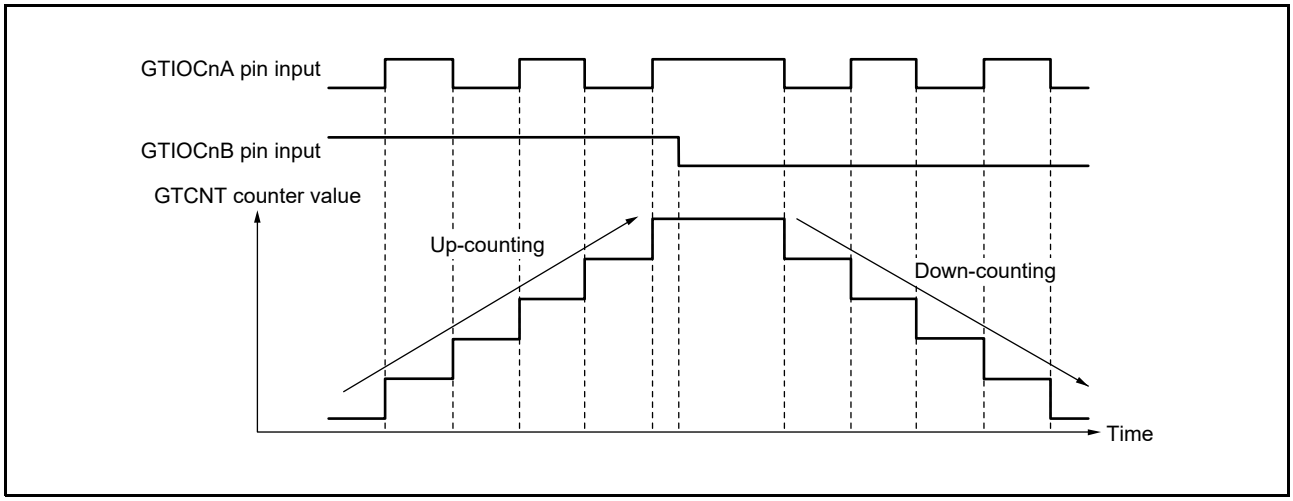


Figure 22.136 Example of setting procedure for the phase counting mode 2 (n = 3 to 7)

Table 22.19 Up-counting and down-counting conditions for the phase counting mode 2 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0A00h GTDNSR = 0000 0500h
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

↑: Rising edge

↓: Falling edge

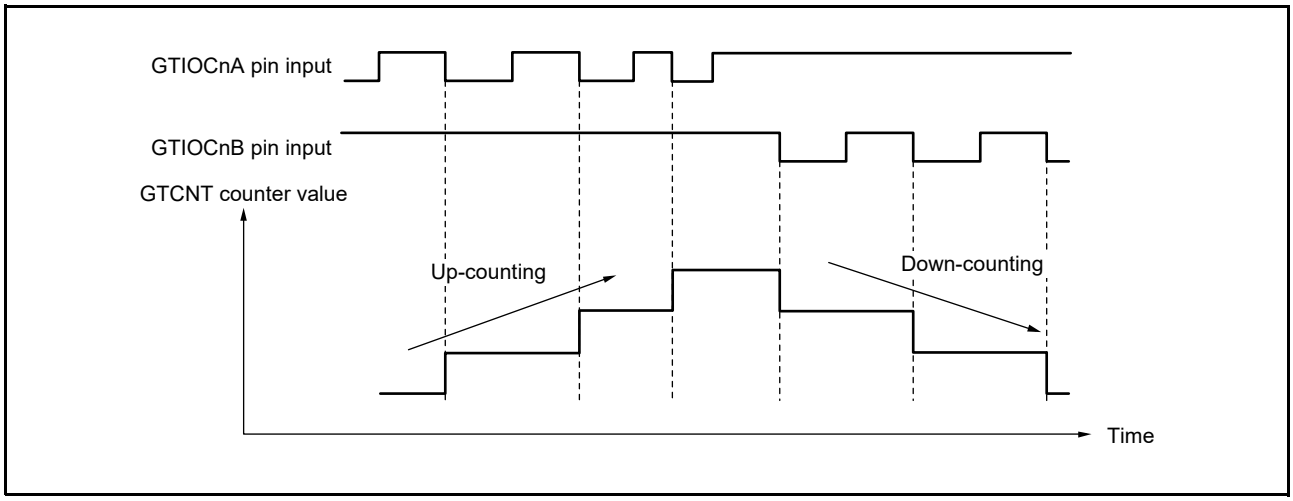


Figure 22.137 Example of setting procedure for the phase counting mode 3 (n = 3 to 7)

Table 22.20 Up-counting and down-counting conditions for the phase counting mode 3 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 8000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

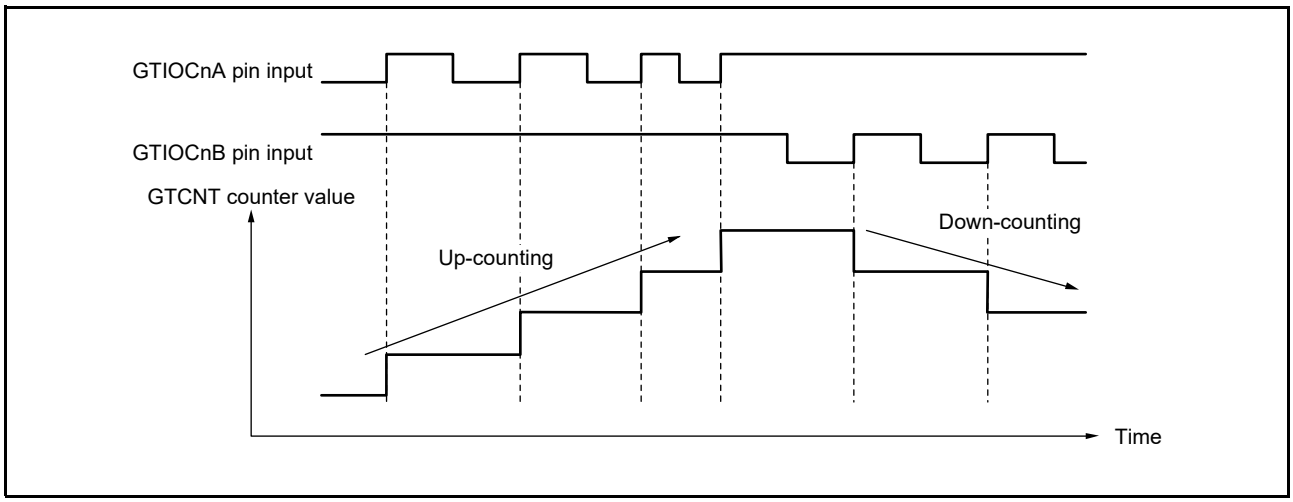


Figure 22.138 Example of setting procedure for the phase counting mode 3 (n = 3 to 7)

Table 22.21 Up-counting and down-counting conditions for the phase counting mode 3 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0000 0200h GTDNSR = 0000 2000h
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

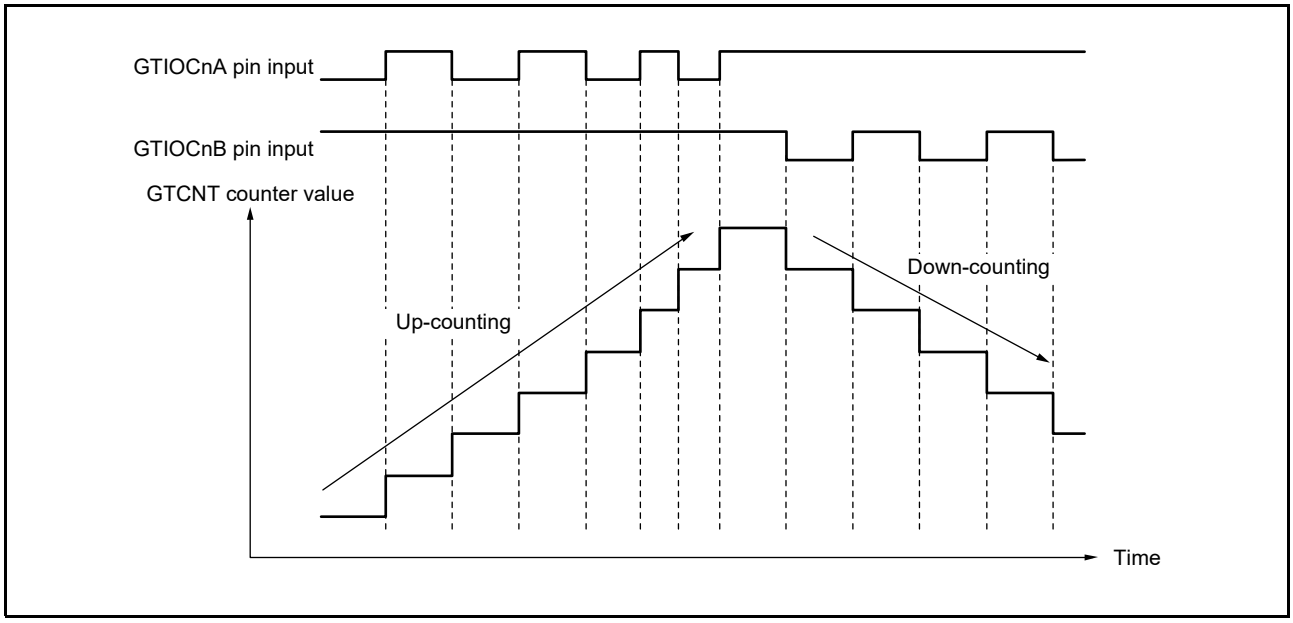


Figure 22.139 Example of setting procedure for the phase counting mode 3 (n = 3 to 7)

Table 22.22 Up-counting and down-counting conditions for the phase counting mode 3 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0000 0A00h GTDNSR = 0000 A000h
Low	↓	Don't care	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

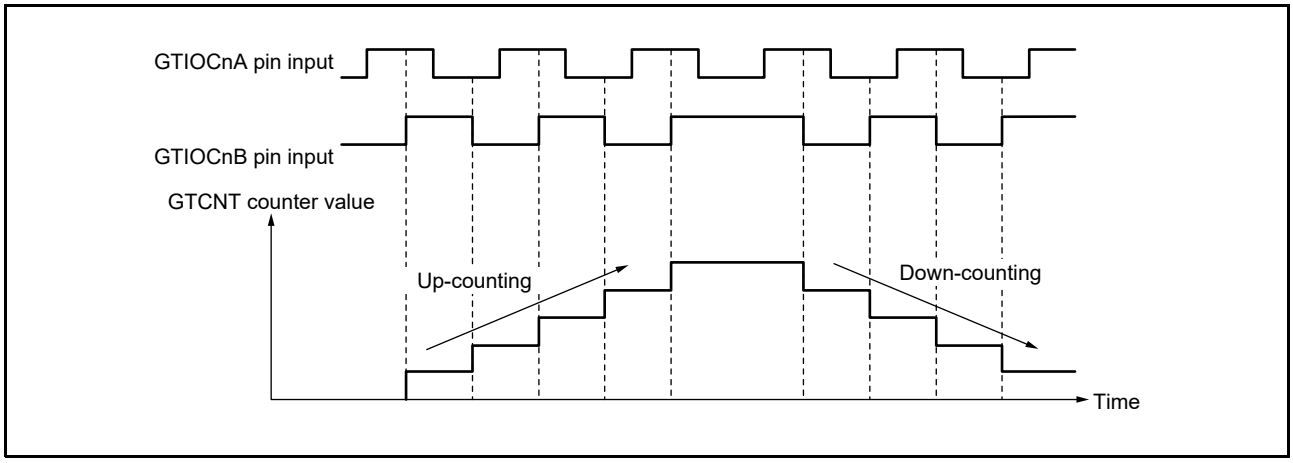


Figure 22.140 Example of setting procedure for the phase counting mode 4 (n = 3 to 7)

Table 22.23 Up-counting and down-counting conditions for the phase counting mode 4 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0000 6000h GTDNSR = 0000 9000h
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

↑: Rising edge

↓: Falling edge

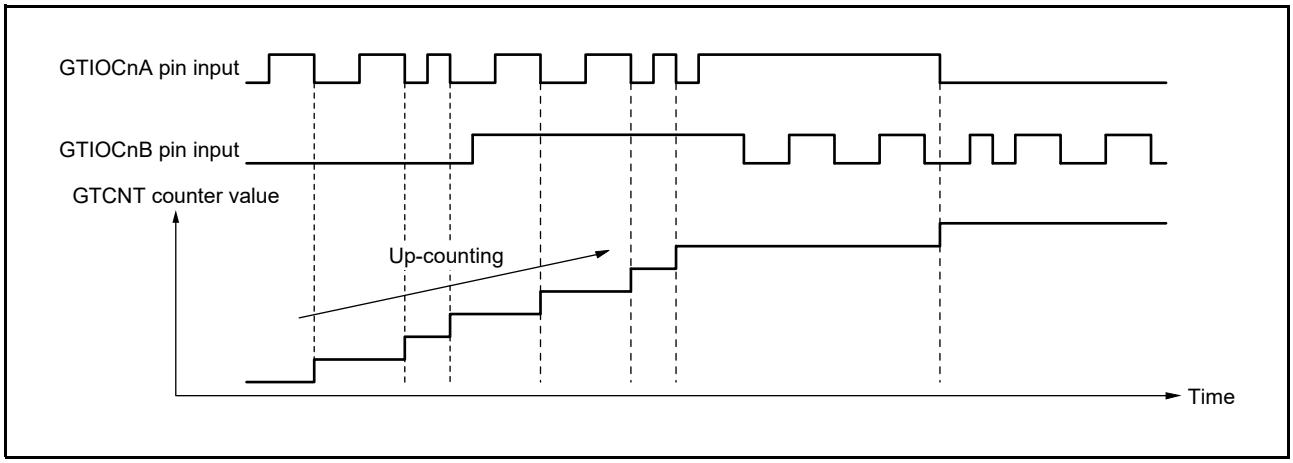


Figure 22.141 Example of setting procedure for the phase counting mode 5 (n = 3 to 7)

Table 22.24 Up-counting and down-counting conditions for the phase counting mode 5 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0C00h GTDNSR = 0000 0000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

↑: Rising edge

↓: Falling edge

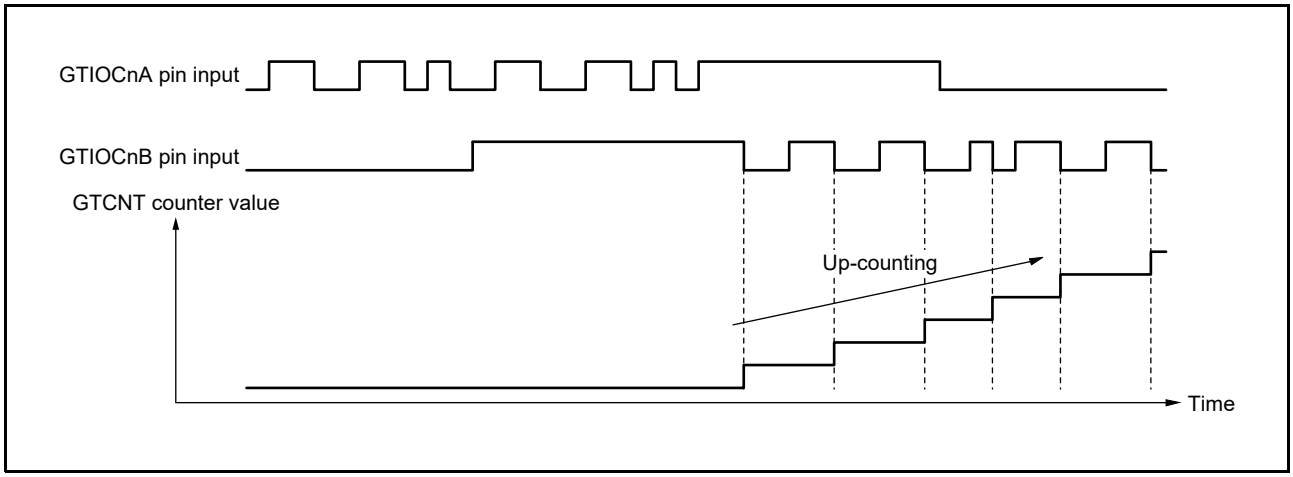


Figure 22.142 Example of setting procedure for the phase counting mode 5 (n = 3 to 7)

Table 22.25 Up-counting and down-counting conditions for the phase counting mode 5 (n = 3 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 C000h GTDNSR = 0000 0000h
Low	↓	Up-counting	
↑	Low	Don't care	
↓	High		
High	↓	Up-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

22.3.11 External Pulse Width Measuring Function

The pulse width of GTIOCnA pin input (n = 3 to 5), GTIOCnB pin input, and GTETRGA/GTETRGB/GTETRGC/GTETRGD pin inputs can be measured.

The setting to enable or disable count-up of the GTCNT counter and the input pin and level which measured pulse width are selected by the USILVL[3:0] bits of the GTUPSR register.

The setting to enable or disable count-down of the GTCNT counter and the input pin and level which measured pulse width are selected by the DSILVL[3:0] bits of the GTDNSR register.

The setting to enable both count-up and count-down of the GTCNT counter at the same time is prohibited.

The counting operation performs cycle counting with the period of the GTPR register.

If the phase counting function and the pulse width measuring function are enabled at the same time, the pulse width measuring function does not work and the phase counting function works.

Figure 22.143 and Figure 22.144 show examples of external pulse width measuring function and Figure 22.145 shows an example for setting external pulse width measuring function.

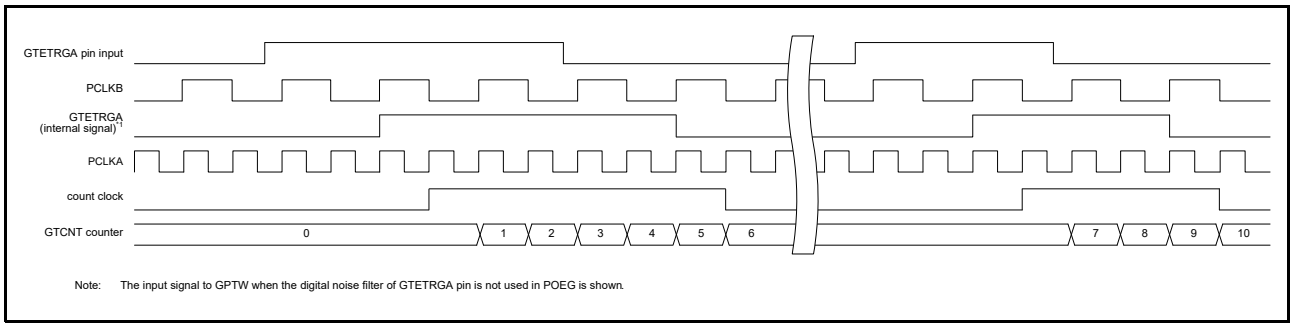


Figure 22.143 Example of External Pulse Width Measuring Function (Up-Counting)

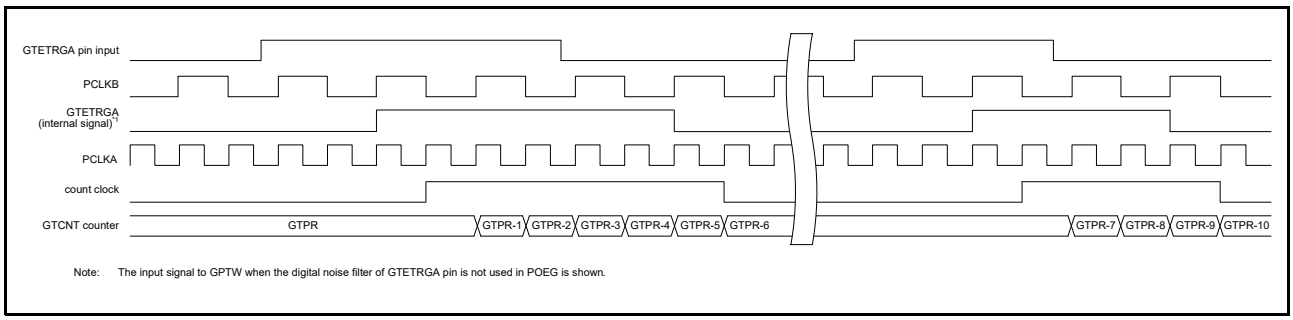


Figure 22.144 Example of External Pulse Width Measuring Function (Down-Counting)

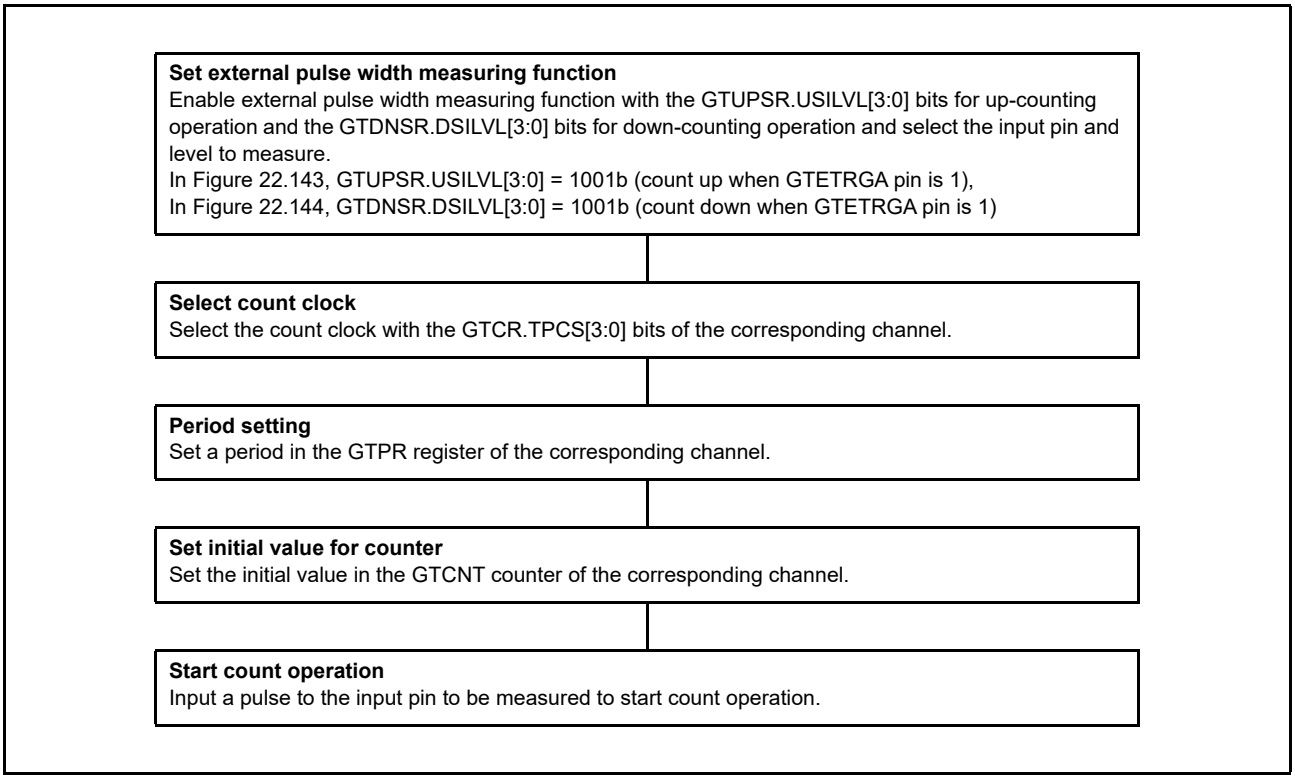


Figure 22.145 Example for Setting External Pulse Width Measuring Function

22.3.12 Output Phase Switching Control Function (OPS)

OPS can easily control Brushless DC motor using Output Phase Switching Control Register (OPSCR).
OPS uses S/W setting value (OPSCR.UF, VF, WF bits) or external signals detected by the Hall element as input signals.
OPS outputs either level signals or chopped signals by GPTW0's PWM as the 6-phase (U-positive phase/negative phase, V-positive phase/negative phase, W-positive phase/negative phase) signals to control motor.
Figure 22.146 shows the block diagram of OPS.
The GPT_UVWEDGE signal is output signal to ELC generated by detecting the edge of input signal.

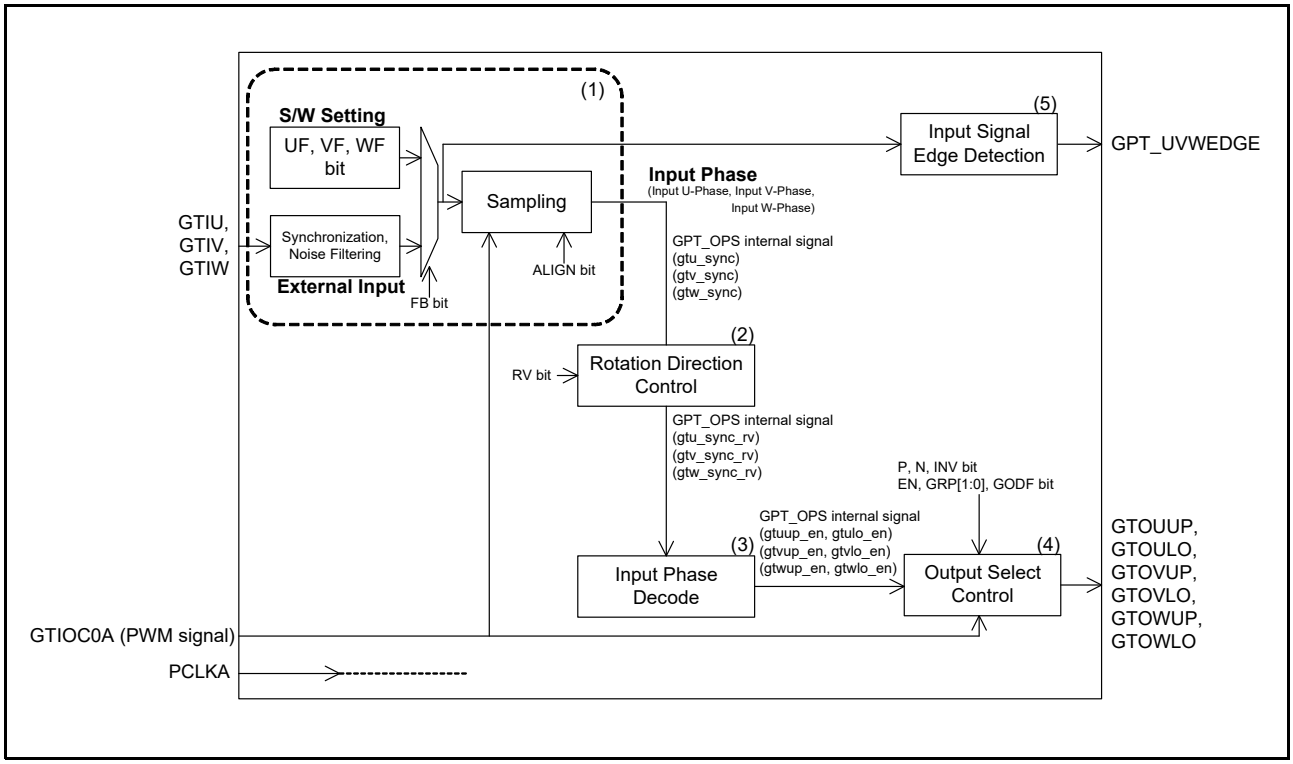


Figure 22.146 OPS Block Diagram

Figure 22.147 and Figure 22.148 show examples of OPS level output operation.

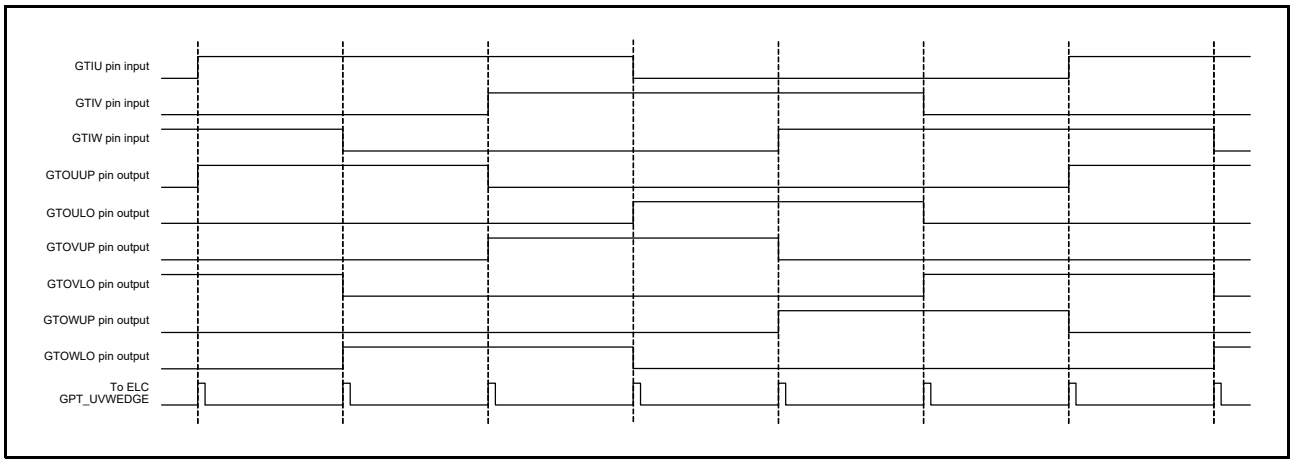


Figure 22.147 Example of OPS Level Output Operation (Forward Rotation) (FB = 0, RV = 0, P = 0, N = 0, INV = 0)

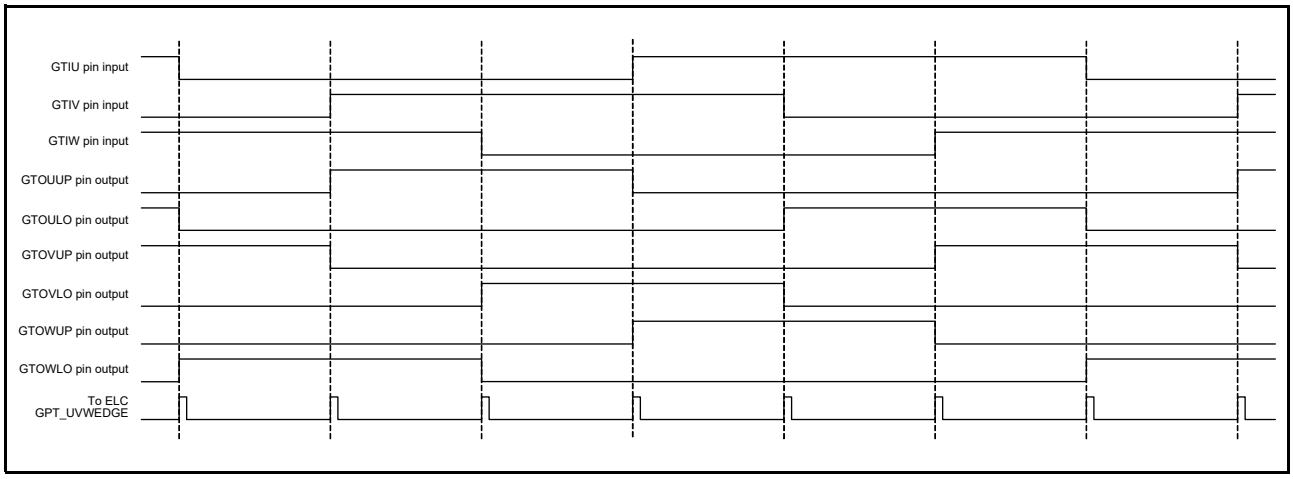


Figure 22.148 Example of OPS Level Output Operation (Reverse Rotation) (FB = 0, RV = 1, P = 0, N = 0, INV = 0)

Figure 22.149 and Figure 22.150 show examples of OPS chopped output operation.

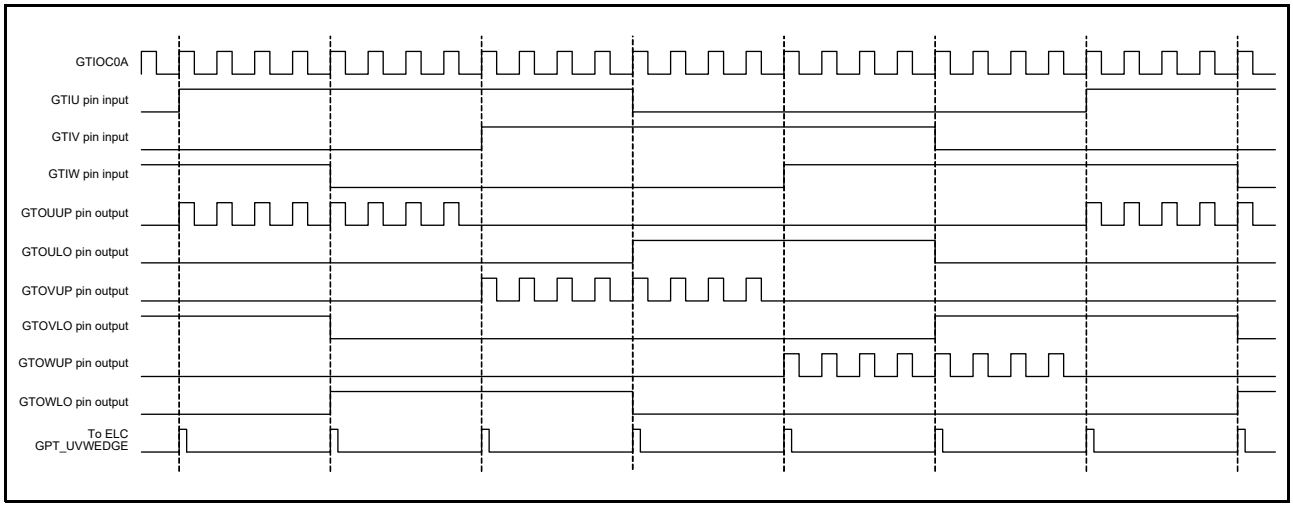


Figure 22.149 Example of OPS Chopped Output Operation (Positive Phase 120-degree) (FB = 0, RV = 0, P = 1, N = 0, INV = 0)

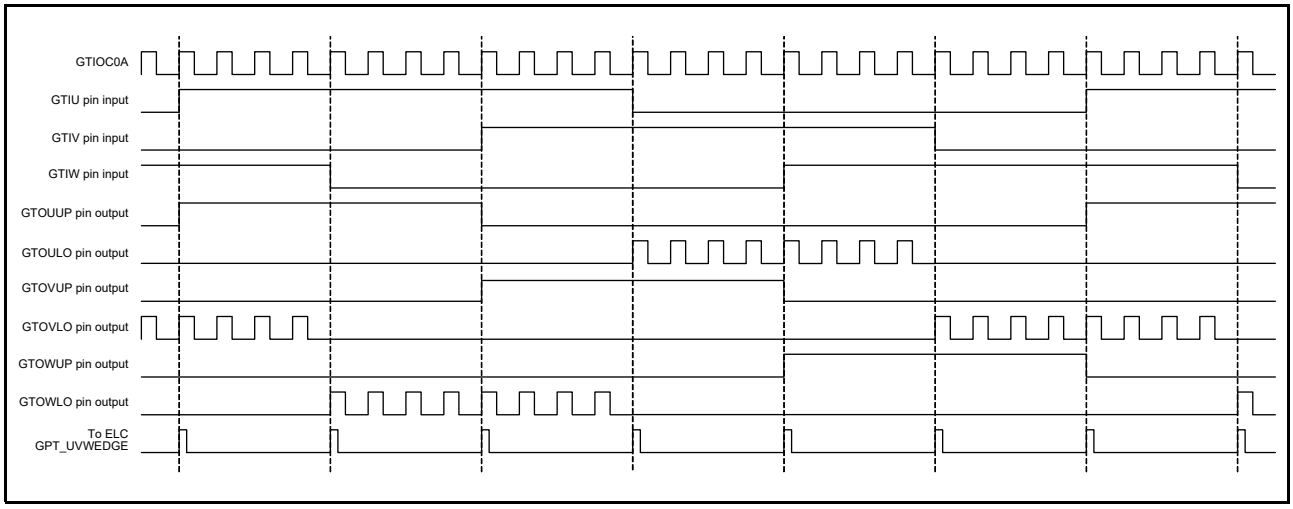


Figure 22.150 Example of OPS Chopped Output Operation (Negative Phase 120-degree) (FB = 0, RV = 0, P = 0, N = 1, INV = 0)

Figure 22.151 shows an example of OPS output disable control operation.

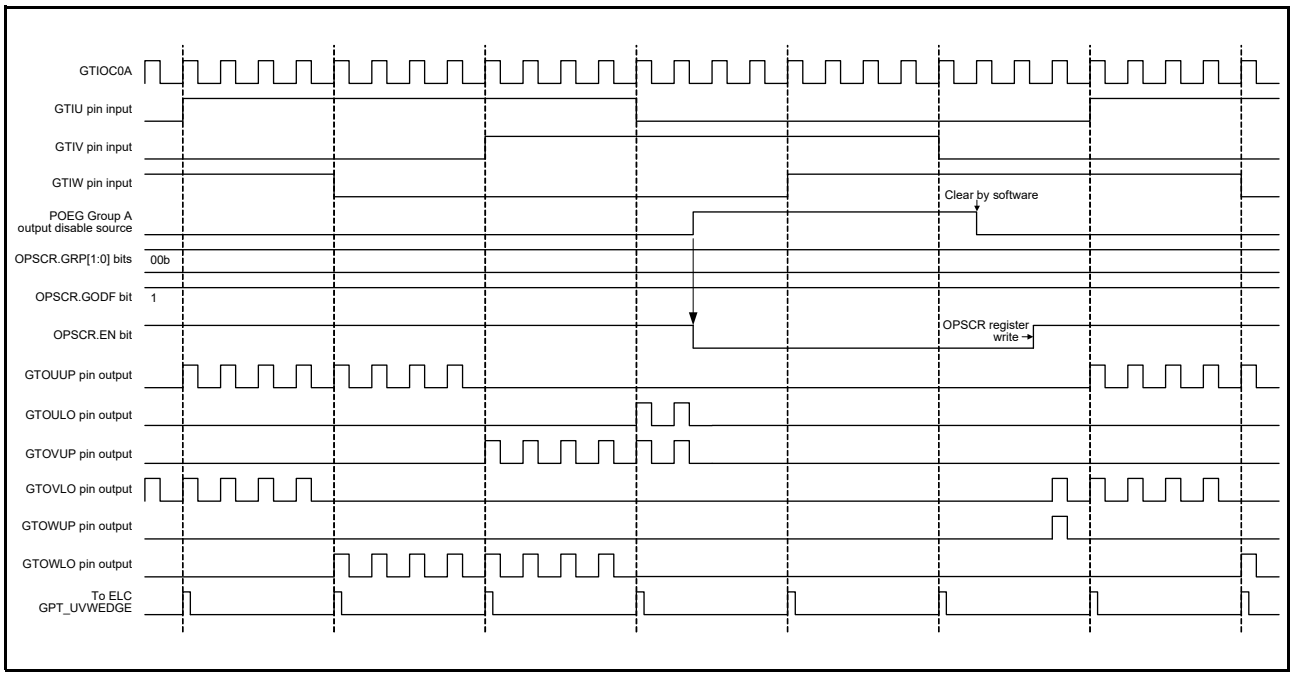


Figure 22.151 Example of OPS Output Disable Control Operation (FB = 0, RV = 0, P = 1, N = 1, INV = 0)

22.3.12.1 Input Selection and Sampling

The FB bit selects either the software setting value or external input for the input signal.
When the FB bit is 0, the GTIU, GTIV, GTIW external input are selected for the input signal to OPS after synchronization with the GPT core clock (PCLKA) and noise filtering.
When the FB bit is 1, the software setting value (UF, VF, WF bits) are selected for the input signal to OPS.
The selected input signals are sampled by the method selected by the ALIGN bit, and they are treated as input phase of OPS.
When the ALIGN bit is 0, the input signals are sampled by PCLKA.
When the ALIGN bit is 1, the input signals are sampled by the falling edge of GTIOC0A pin output.
The signals after sampling can be read by the U, V, W bits.
Table 22.26 shows the input selection by the FB bit and sampling method by the ALIGN bit.

Table 22.26 Input Selection and Sampling Method

OPSCR Register		Input Selection Sampling Method	Interrupt Phase (OPS internal signal)
FB bit	ALIGN bit		
0	0	GTIU, GTIV, GTIW external input PCLKA sampling	Input U-phase (gtu_sync) Input V-phase (gtv_sync) Input W-phase (gtw_sync)
	1	GTIU, GTIV, GTIW external input GTIOC0A falling edge sampling	
1	0	Software setting value UF, VF, WF bit PCLKA sampling	
	1	Software setting value UF, VF, WF bit GTIOC0A falling edge sampling	

22.3.12.2 Rotation Direction Control

When the rotation direction is reverse (RV = 1), the input phase is inverted.

22.3.12.3 Input Phase Decode

The 6-phase signals by decoding input phase after rotation direction control are generated.
Table 22.27 and Table 22.28 show the decode tables of input phase to rotate motor in forward (RV = 0) and reverse (RV = 1).

Table 22.27 The Decode Table of Input Phase (Forward Rotation)

Input Phase			**Input Phase after Rotation Direction Control**			**6-Phase Signals**					
U-phase	V-phase	W-phase	U-phase	V-phase	W-phase	Positive U-phase	Negative U-phase	Positive V-phase	Negative V-phase	Positive W-phase	Negative W-phase
gtu_sync	**gtv_sync**	**gtw_sync**	**gtu_sync_rv**	**gtv_sync_rv**	**gtw_sync_rv**	**gtuup_en**	**gtulo_en**	**gtvup_en**	**gtvlo_en**	**gtwup_en**	**gtwlo_en**
1	0	1	1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0	0	1
1	1	0	1	1	0	0	0	1	0	0	1
0	1	0	0	1	0	0	1	1	0	0	0
0	1	1	0	1	1	0	1	0	0	1	0
0	0	1	0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0

Table 22.28 The Decode Table of Input Phase (Reverse Rotation)

Input Phase			**Input Phase after Rotation Direction Control**			**6-Phase Signals**					
U-phase	V-phase	W-phase	U-phase	V-phase	W-phase	Positive U-phase	Negative U-phase	Positive V-phase	Negative V-phase	Positive W-phase	Negative W-phase
gtu_sync	**gtv_sync**	**gtw_sync**	**gtu_sync_rv**	**gtv_sync_rv**	**gtw_sync_rv**	**gtuup_en**	**gtulo_en**	**gtvup_en**	**gtvlo_en**	**gtwup_en**	**gtwlo_en**
1	0	1	0	1	0	0	1	1	0	0	0
1	0	0	0	1	1	0	1	0	0	1	0
1	1	0	0	0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	0	0	1	0	0
0	1	1	1	0	0	1	0	0	0	0	1
0	0	1	1	1	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0

22.3.12.4 Output Selection Control

The EN, P, N, INV bits select the output wave.
The EN bit enables output of the 6-phase output. When the EN bit is 1, output of the 6-phase output is enabled. When the EN bit is 0, the external pin output is Hi-Z.
The P, N bits select whether chopping positive and negative phase are performed or not. When P, N bits are 1, chopping is performed by GTIOC0A pin output.
When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.
The INV bit selects the polarity (either positive logic or negative logic) of phase output.
Table 22.29 and Table 22.30 show the output selection control method for positive and negative phase output.

Table 22.29 Output Selection Control Method (Positive Phase)

EN bit	P bit	INV bit	GTOnUP
0	—	—	0 (External pin output Hi-z)
1	0	0	Positive logic level output (gtmup_en)
1	0	1	Negative logic level output (~gtmup_en)
1	1	0	Positive logic chopped output (GTIOC0A & gumup_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gumup_en))

n = U, V, W
m = u, v, w

Table 22.30 Output Selection Control Method (Negative Phase)

EN bit	P bit	INV bit	GTOnLO
0	—	—	0 (External pin output Hi-z)
1	0	0	Positive logic level output (gtmlo_en)
1	0	1	Negative logic level output (~gtmlo_en)
1	1	0	Positive logic chopped output (GTIOC0A & gumlo_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gumlo_en))

n = U, V, W
m = u, v, w

22.3.12.5 Output Selection Control (Group Output Disable Function)

When the GODF bit is 1 and signal value selected by the GRP[1:0] bits is High (Output Disable Request), OPS's output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is cleared by the output disable request signal synchronized with PCLKA.
For the return, after clearing the Output Disable Request by software, set the EN bit to 1.
The timing of EN bit cleared is 3 PCLKA cycles after generating the output disable request. In order to perform the output disable control surely, the output disable request flag in POEG should be cleared in the timing that terminating the output disable request is at least 4 PCLKA cycles after generating the output disable request.
The example of the operation of the group output disable control, see the above-mentioned Figure 22.151.

22.3.12.6 Event Link Controller (ELC) Output

The logical sum of the pulse detected by rising and falling edge of U, V, W phase input is output to the event link controller (ELC). When the high level period of input phase is short, there are cases that the detected edge is not transmitted to the ELC correctly because of the logical sum.

22.3.12.7 OPS Start Operation Setting Flow

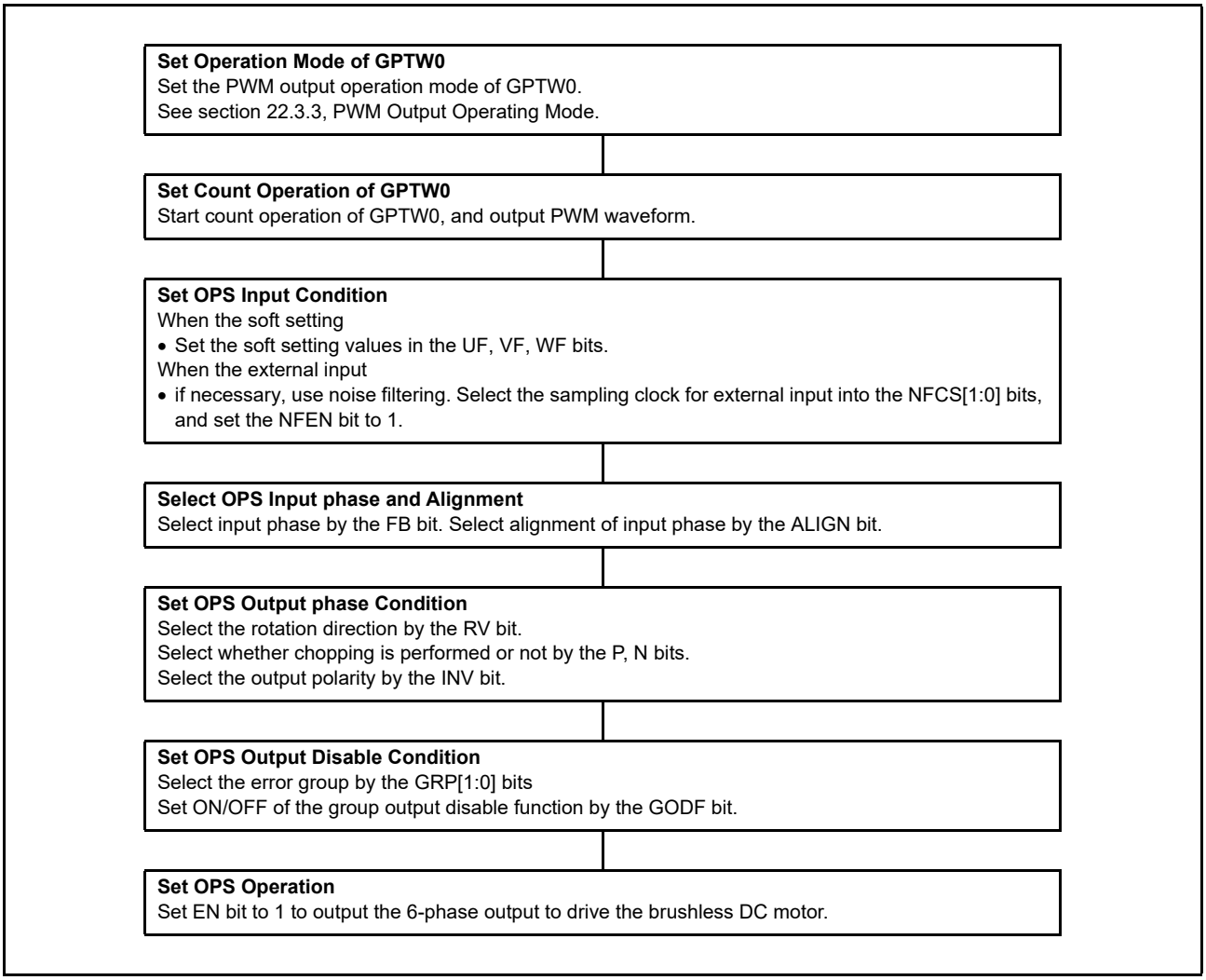


Figure 22.152 Example for Setting OPS Start Operation

22.4 Interrupt Sources

22.4.1 Interrupt Sources and Priorities

There are two kinds of interrupt sources; GTCCRm register (m = A to F) input capture/compare match and GTCNT counter overflow (GTPR register compare match)/underflow.
Each interrupt source has its own status flag and control bit for generating interrupt request signals, where the generation of interrupt request signals can be enabled or disabled individually.
When an interrupt source condition is satisfied, the corresponding status flag in GTST is set to 1, and an interrupt request is generated if the corresponding interrupt request enable or disable bit in the GTINTAD register is 1.
The corresponding status flag in the GTST register is writable to clear. If flag set and flag clear happens at the same time, flag clear have a priority to flag set.
For details, refer to section 14, Interrupt Controller (ICUb). Table 22.31 shows a list of GPTW interrupt sources.

Table 22.31 GPTW Interrupt Sources (n = 0 to 7)

Channel	Name	Interrupt Source	Interrupt Flag
GPTWn	GTCIAn	GTCCRA register input capture/compare match	GPTWn.GTST.TCFA
	GTCIBn	GTCCRB register input capture/compare match	GPTWn.GTST.TCFB
	GTCICn	GTCCRC register compare match	GPTWn.GTST.TCFC
	GTCIDn	GTCCRD register compare match	GPTWn.GTST.TCFD
	GTCIEn	GTCCRE register compare match	GPTWn.GTST.TCFE
	GTCIFn	GTCCRF register compare match	GPTWn.GTST.TCFF
	GTCIVn	GTCNT counter overflow (GTPR register compare match)	GPTWn.GTST.TCFPO
	GTCIUn	GTCNT counter underflow	GPTWn.GTST.TCFPU

(1) GTCIAn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTA bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRA register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRA register.
- When the GTCCRA register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRA register. In complementary PWM mode, GTCCRA register does not function as an input capture register.

(2) GTCIBn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTB bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as a compare match register.
- When the GTCCRB register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as an input capture register.

(3) GTCICn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTC bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRC register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRC register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRC register)

(4) GTCIDn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTD bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRD register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRD register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 10b, 11b (buffer operation with the GTCCRD register)

(5) GTCIEn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTE bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRE register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRE register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRE register)

(6) GTCIFn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTF bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRF register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRF register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 10b, 11b (buffer operation with the GTCCRF register)

(7) GTCIVn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTPR[0] bit is 1, an interrupt request is generated under the following conditions.

- In sawtooth-wave mode, interrupt requests are enabled at overflows (the GTCNT counter value changes from the GTPR register value to 0 during up-counting).
- In sawtooth-wave PWM mode 1 and sawtooth-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from the GTPR register value to 0 during up-counting).
- In sawtooth-wave PWM mode 2, interrupt requests are enabled at overflows (the GTCNT counter value changes from GTCCRm (m = A to F) register value selected with GTCSR.CSCMSC[2:0] bits to 0000 0000h) or the time when GTCNT counter value matches GTPR register value.
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT counter value changes from the GTPR register value to the GTPR register value minus 1).

- In complementary PWM mode, interrupt requests are enabled at crests (the GTCNT counter value of master channel changes from the GTPR register value to the GTPR register value minus 1).
- In counting driven by a hardware source, including counting for the external pulse width measuring function, interrupt requests are enabled on overflows (that is, when the GTCNT counter value changes from the GTPR register value to 0 during up-counting).

(8) GTCIUn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTPR[1] bit is 1, an interrupt request is generated under the following conditions.

- In sawtooth-wave mode, interrupt requests are enabled at underflows (the GTCNT counter value changes from 0 to the GTPR register value during down-counting).
- In sawtooth-wave PWM mode 1 and sawtooth-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from the GTPR register to 0 during down-counting).
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT counter value changes from 0 to 1).
- In complementary PWM mode, interrupt requests are enabled at troughs (the GTCNT counter value of master channel changes from 0 to 1).
- In count operation by a hardware source (include external pulse width measuring function), interrupt requests are enabled at underflows (the GTCNT counter value changes from 0 to the GTPR register value during down-counting).

Table 22.32 Relationship between Interrupt Signals and Interrupt Enable Bits (n = 0 to 7)

Interrupt Signal	Interrupt Enable Bit	Status Flag
GTCIAn	GTINTAD.GTINTA bit	GTST.TCFA Flag
GTCIBn	GTINTAD.GTINTB bit	GTST.TCFB Flag
GTCICn	GTINTAD.GTINTC bit	GTST.TCFC Flag
GTCIDn	GTINTAD.GTINTD bit	GTST.TCFD Flag
GTCIEn	GTINTAD.GTINTE bit	GTST.TCFE Flag
GTCIFn	GTINTAD.GTINTF bit	GTST.TCFF Flag
GTCIVn	GTINTAD.GTINTPR[1:0] bits	GTST.TCFPO Flag
GTCIUn		GTST.TCFPU Flag

22.4.2 DMAC/DTC Activation

The DMAC and DTC can be triggered by the interrupt request in each channel. For details, see section 14, Interrupt Controller (ICUb), section 17, DMA Controller (DMACA), section 18, Data Transfer Controller (DTCb).

22.4.3 Interrupt and A/D Conversion Start Request Skipping Function

22.4.3.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GTCIV) and underflow interrupt (GTCIU) can be skipped. Other interrupts and A/D conversion start request signals can be skipped in coordination with the GTCIV/GTCIU skipping function.
When the interrupt is skipped, the updating of relevant status flag is also skipped. The interrupt skipping is continued even if the status flag is set to 1.
The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit. The interrupt enable bit is used only to control the output of the interrupt signal after thinning out.
When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GTCIV/GTCIU interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GTCIV/GTCIU interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.
Similarly, in sawtooth-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GTCIV/GTCIU interrupt requests cannot be generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GTCIV/GTCIU interrupts at overflows only or underflows only in sawtooth-wave mode, the skipping state should be carefully checked before using.
When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).
Figure 22.153 to Figure 22.158 show examples of skipping function operation.

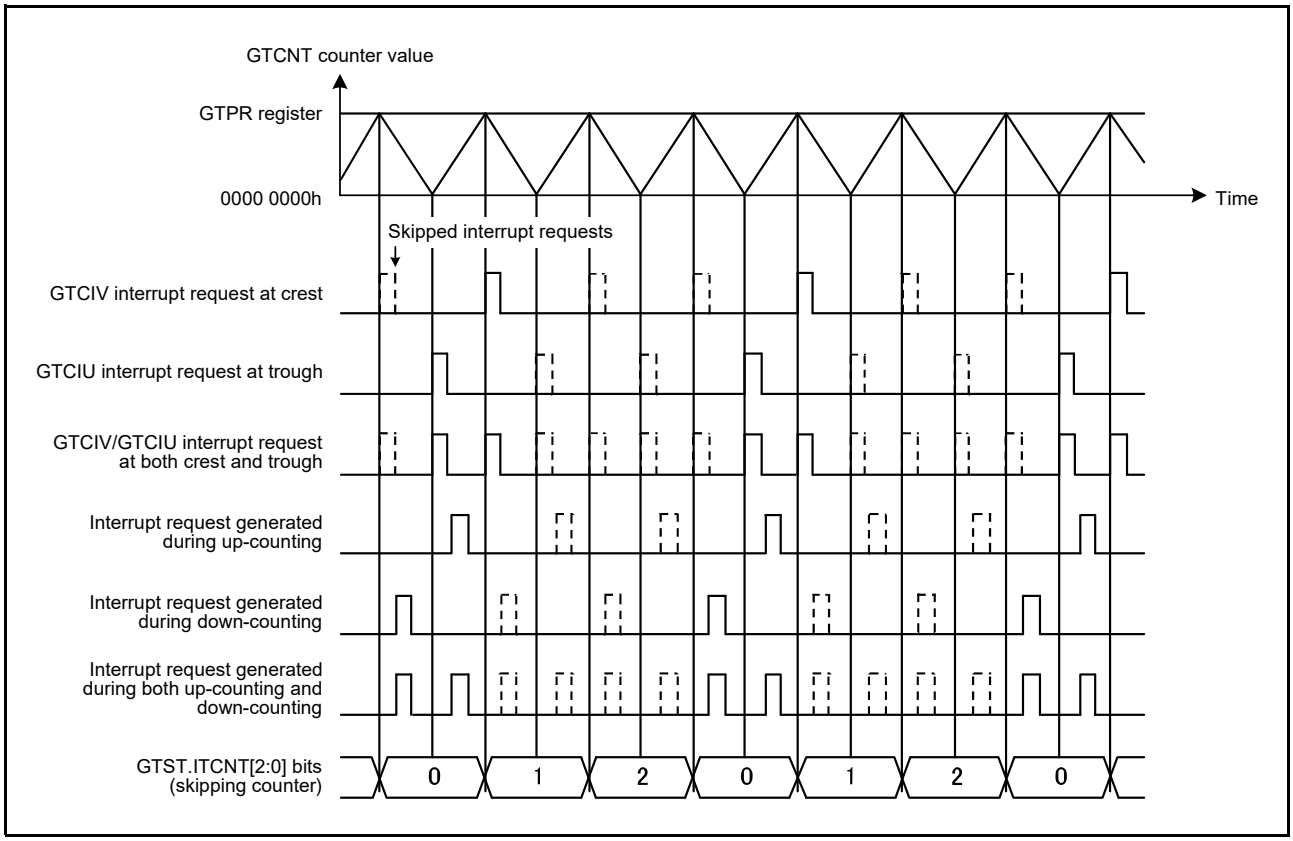


Figure 22.153 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

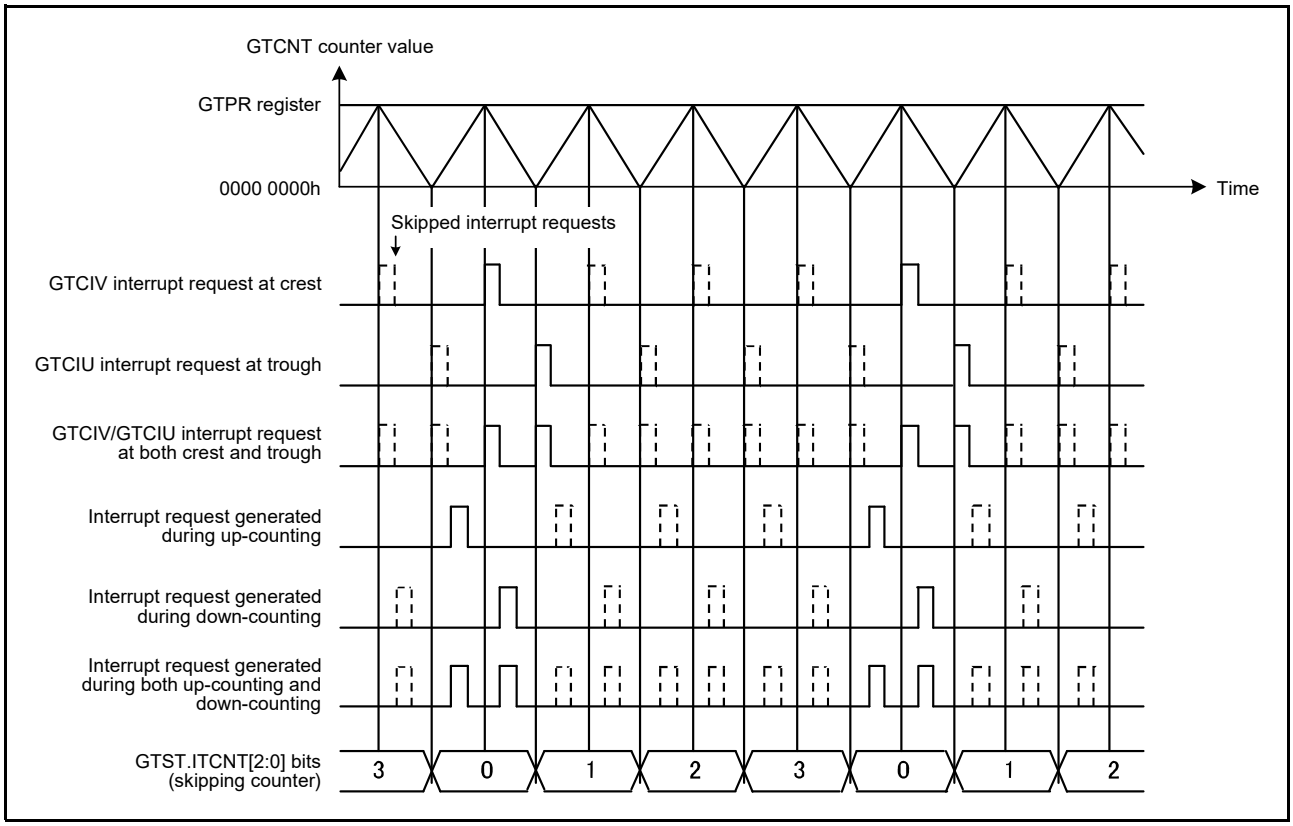


Figure 22.154 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

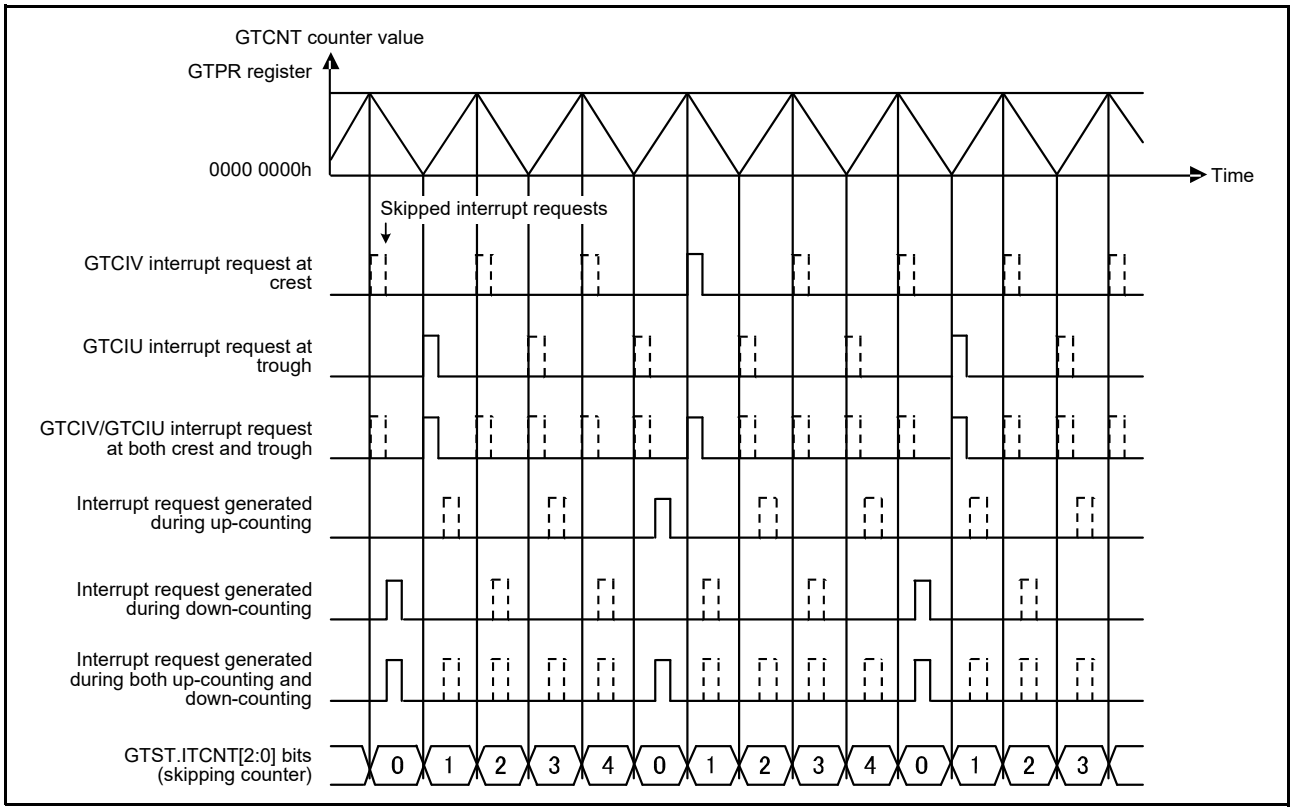


Figure 22.155 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

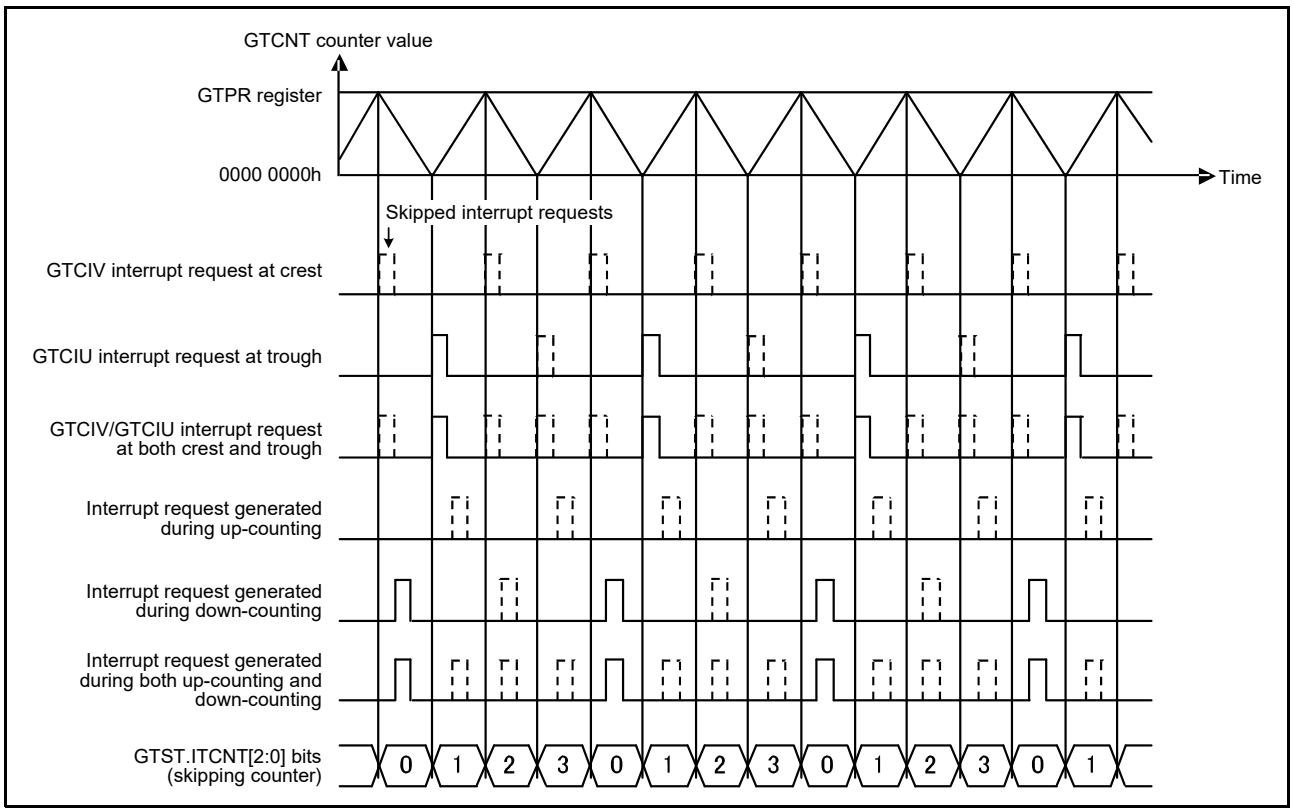


Figure 22.156 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

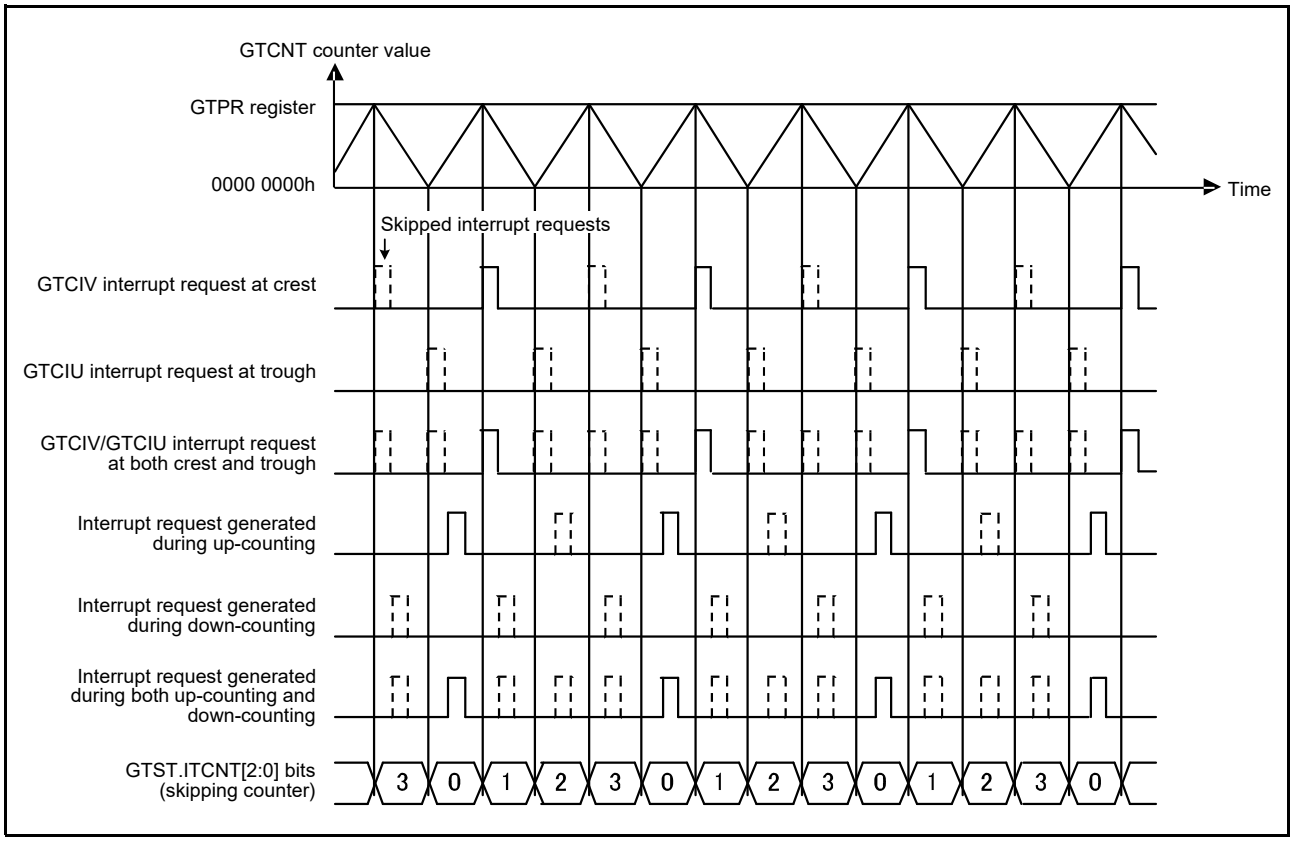


Figure 22.157 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

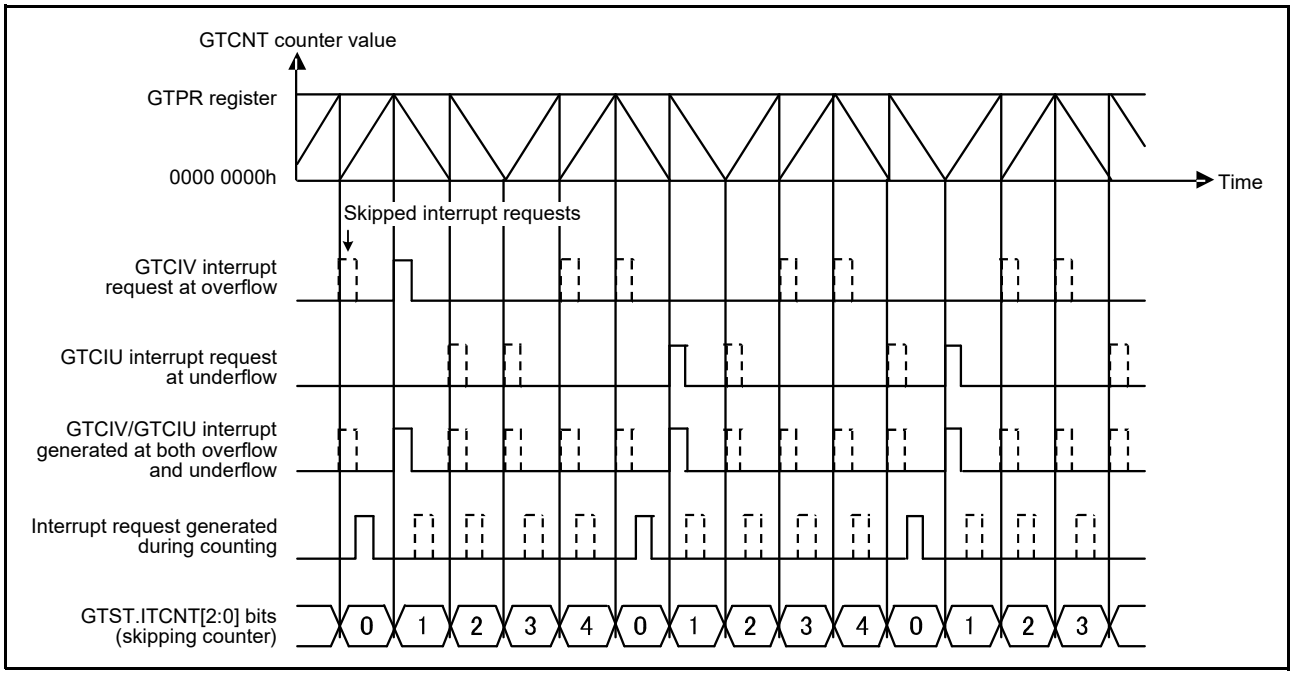


Figure 22.158 Example of Interrupt Skipping Function Operation (Sawtooth Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

22.5 A/D Conversion Start Request

An A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register, up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

In complementary PWM mode, the A/D conversion start request can be issued at a compare match with the GTCNT counter of master channel.

During event count operation, the A/D conversion start request cannot be generated.

The A/D conversion start request is not directly output to the A/D converter, but an interrupt, and an event signal to the ELC are output.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

Figure 22.159 shows an example of A/D conversion start request operation, Figure 22.160 shows an example for setting A/D conversion start request operation, and Figure 22.161 shows an example for A/D conversion start request timing operation.

Figure 22.161 shows an example of the output of A/D conversion start request A by the ELC as start source 0 for the A/D converter. The A/D conversion start request A signal is output by the ELC in response to a match in comparison with the GTADTRA register. The A/D conversion start request A signal is synchronized with PCLKA. The ELC receives the signal in synchronization with PCLKB, and then outputs the A/D conversion start source 0 signal after 1 cycle of the PCLKB has elapsed.

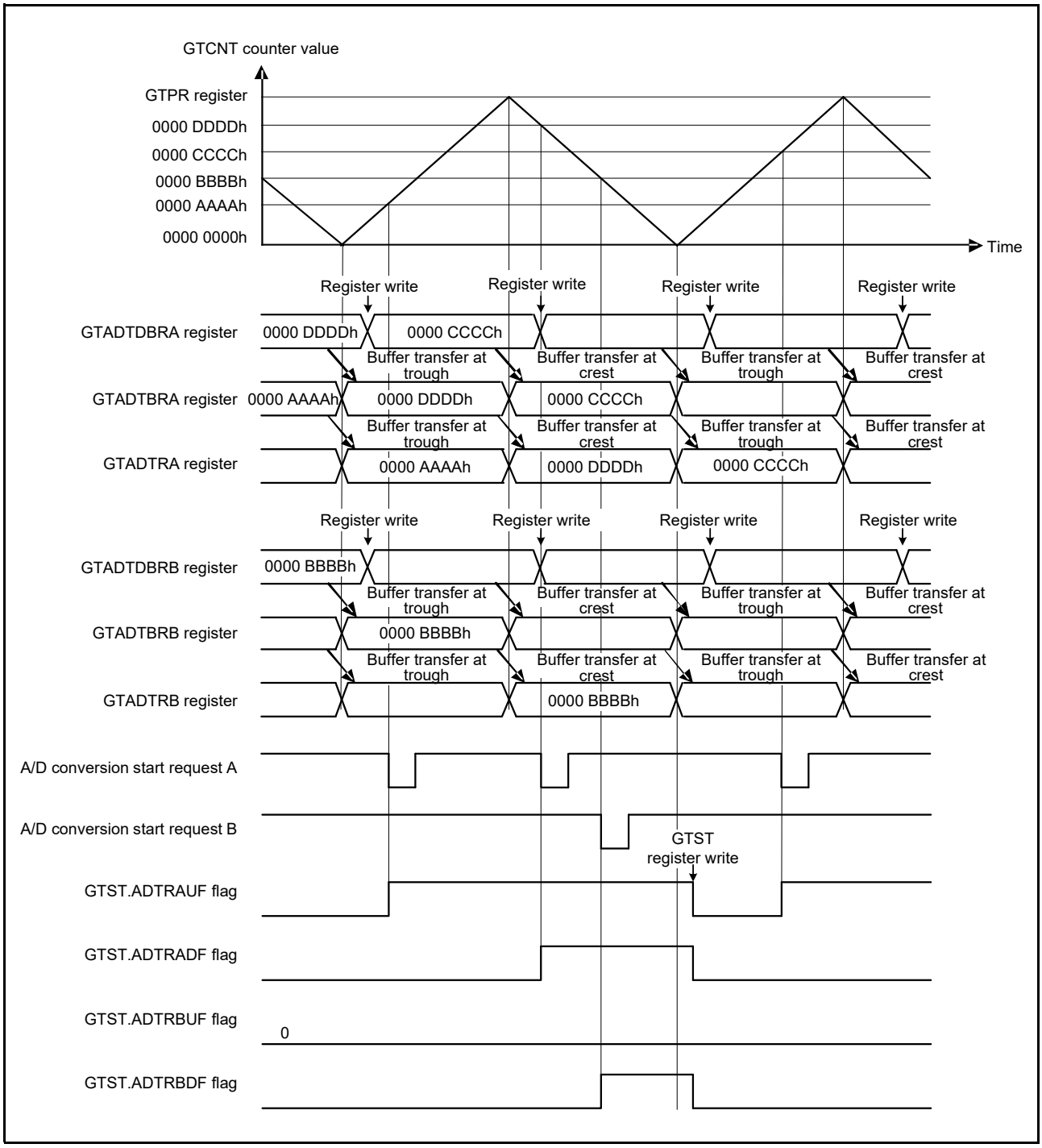


Figure 22.159 Example of A/D Conversion Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Requested by GTADTRA Register at Both Up-Counting and Down-Counting, A/D Conversion Start Requested by GTADTRB Register at Down-Counting)

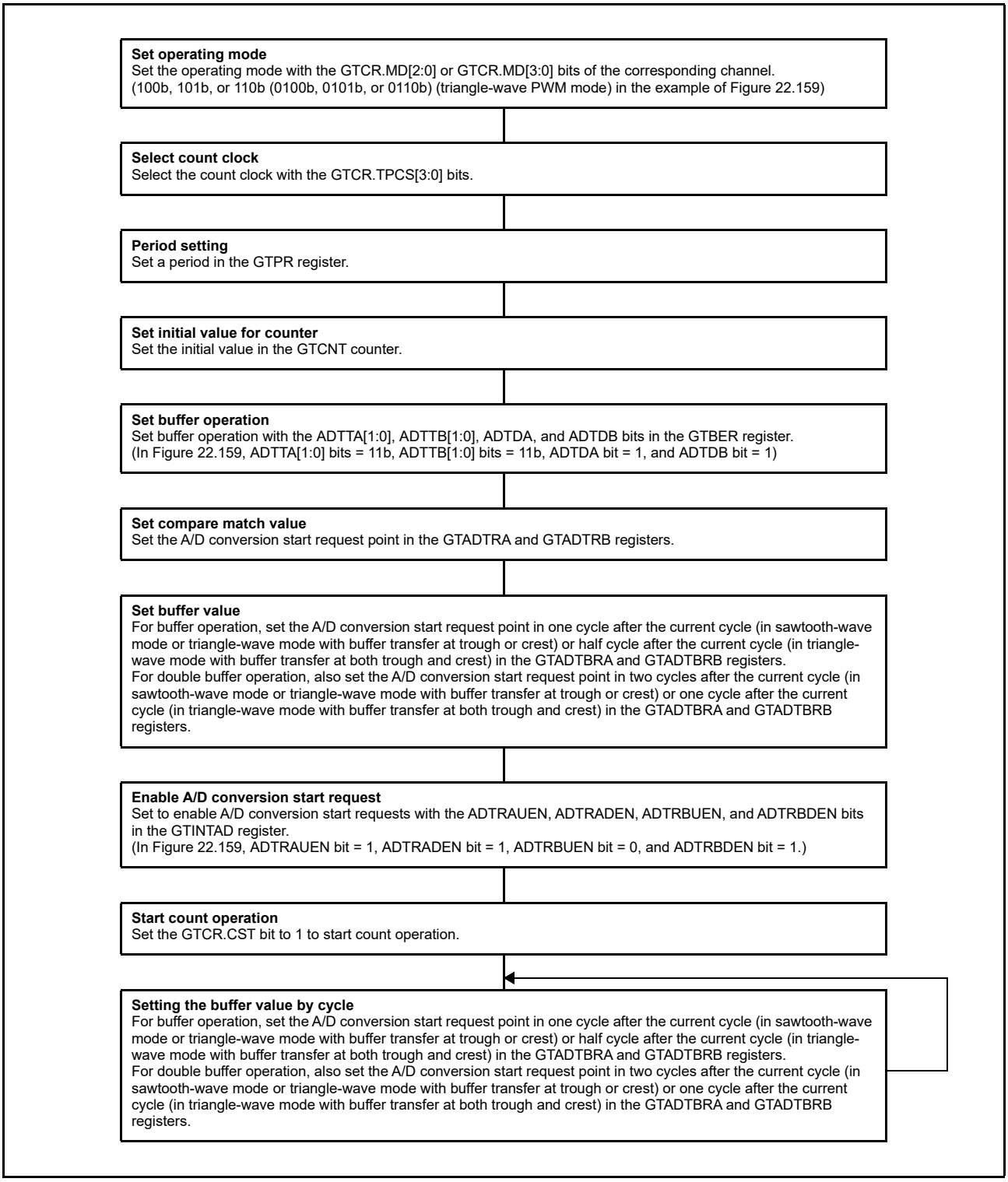


Figure 22.160 Example for Setting A/D Conversion Start Request Operation

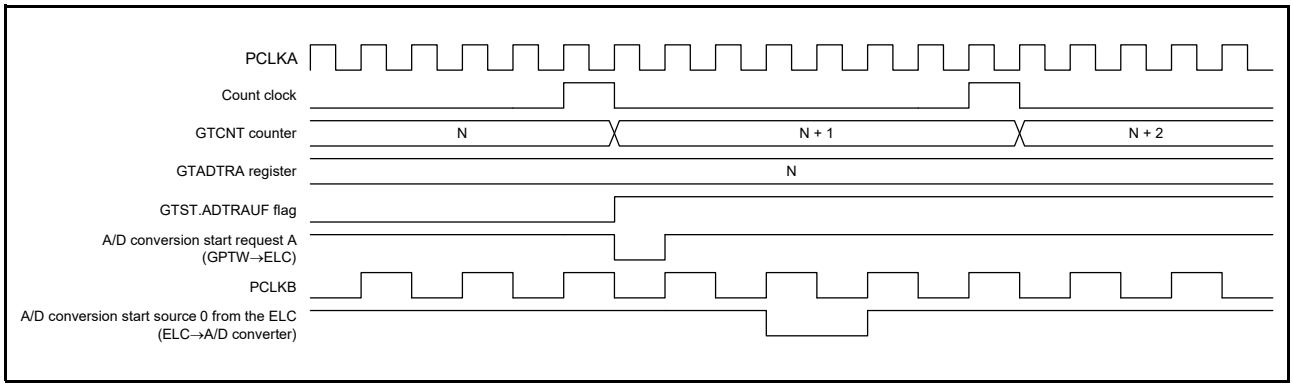


Figure 22.161 Example of A/D Conversion Start Request Timing Operation

22.6 Operations Linked by the ELC

22.6.1 Event Signal Output to the ELC

GPTW 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 event signal can be output regardless of the setting of the corresponding interrupt request enable bits except the A/D conversion start request.
The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output to the ELC.
The following is a list of the event signals output by the GPTW to the ELC. Each GPTW channel has the first 10 event signals. The last of the listed event signals is common to all channels.

- Generation of compare match A interrupt
- Generation of compare match B interrupt
- Generation of compare match C interrupt
- Generation of compare match D interrupt
- Generation of compare match E interrupt
- Generation of compare match F interrupt
- Generation of overflow interrupt
- Generation of underflow interrupt
- Generation of A/D conversion start request A
- Generation of A/D conversion start request B
- GPTW (OPS) U-/V-/W-phase input edge detected

22.6.2 GPTW Operations in Response to Receiving Event Signals from the ELC

The GPTW can perform the following operations by the signals for sources A to H which are output from the ELC. Each signal is provided to all channels. Select an event source for a desired operation by a relevant source select register in a channel.

- Operation of count start/count stop/counter clearing
- Operation of up-counting/down-counting
- Operation of input capture A and B

Refer to operation by hardware sources in section 22.3.1.1, Counter Operation for respective operations.

22.7 Noise Filter Function

Each pin for use in input capture to the GPTW is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling periods.
The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel.
Figure 22.162 shows the timing of noise filtering.

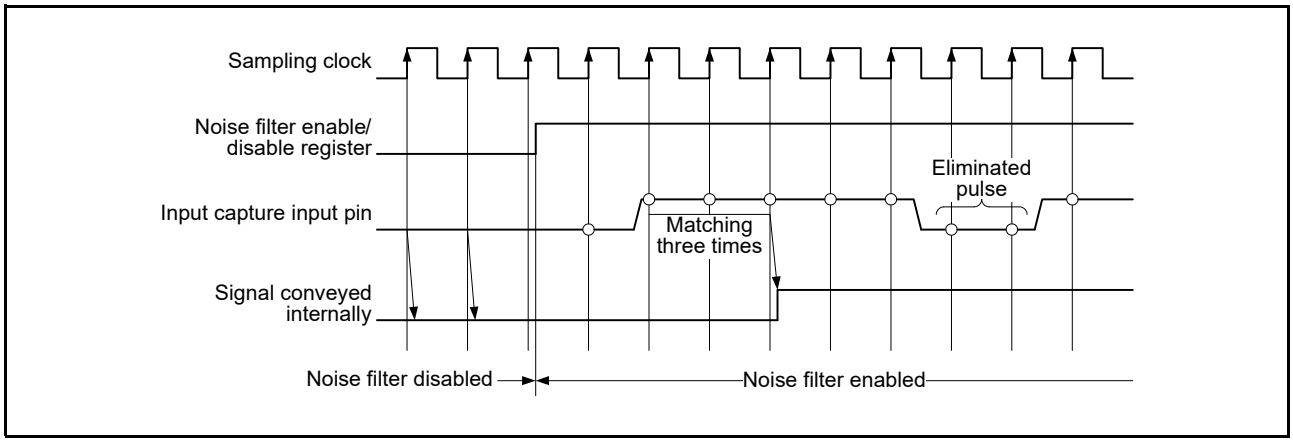


Figure 22.162 Timing of Noise Filtering

If noise filtering is enabled, input capture operation is performed on the edges of noise-filtered signal after a delay of (minimum sampling interval × 2 + PCLKA) due to noise filtering for the input capture input.

22.8 Protection Function

22.8.1 Write-Protection for Registers

In order to avoid incorrect writing to registers, write access to registers can be enabled or disabled per channel by setting the GTWP.WP bit for the given channel.
The WP bits enable or disable writing to the registered are listed below.
GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTICCR
Every bit in registers GTSTR, GTSTP, and GTCLR, which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.
Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can by enabled or disabled by the setting of the GTWP.CMNWP bit.
Protection by the GTWP register only targets writing operation by the CPU. Updating the register generated in related to the CPU writing is not protected.

22.8.2 Disabling of Buffer Operation

If the timing of buffer register write is too late for the buffer transfer timing, buffer operation can be suspended with setting the BD[0], BD[1], and BD[2] bits in the GTBER resister.
Specifically, buffer transfer can be temporarily disabled, even though a buffer transfer condition is generated during buffer register write, by setting the BD[0], BD[1], and BD[2] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.
The BD[0], BD[1], and BD[2] bits can be set on channel basis by writing directly to the GTBER register or can be set simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.
Figure 22.163 shows an example of operation for disabling buffer operation.

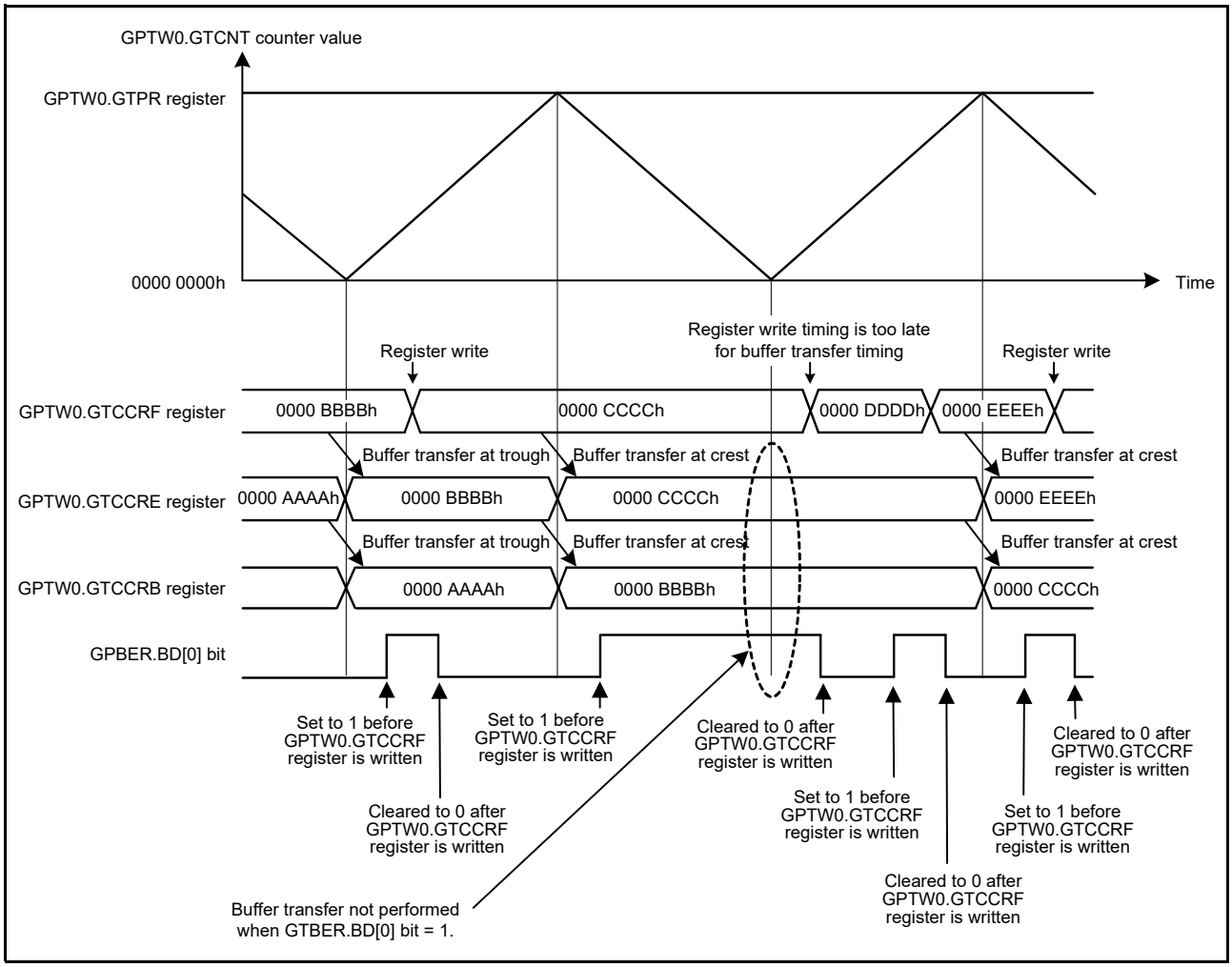


Figure 22.163 Example of Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests)

22.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.
Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

(1) Select the channels for simultaneously setting of the GTSECSR register

Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.

(2) Simultaneously set the GTBER.BD bits by updating the GTSECR register

In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register. Figure 22.164 and Figure 22.165 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

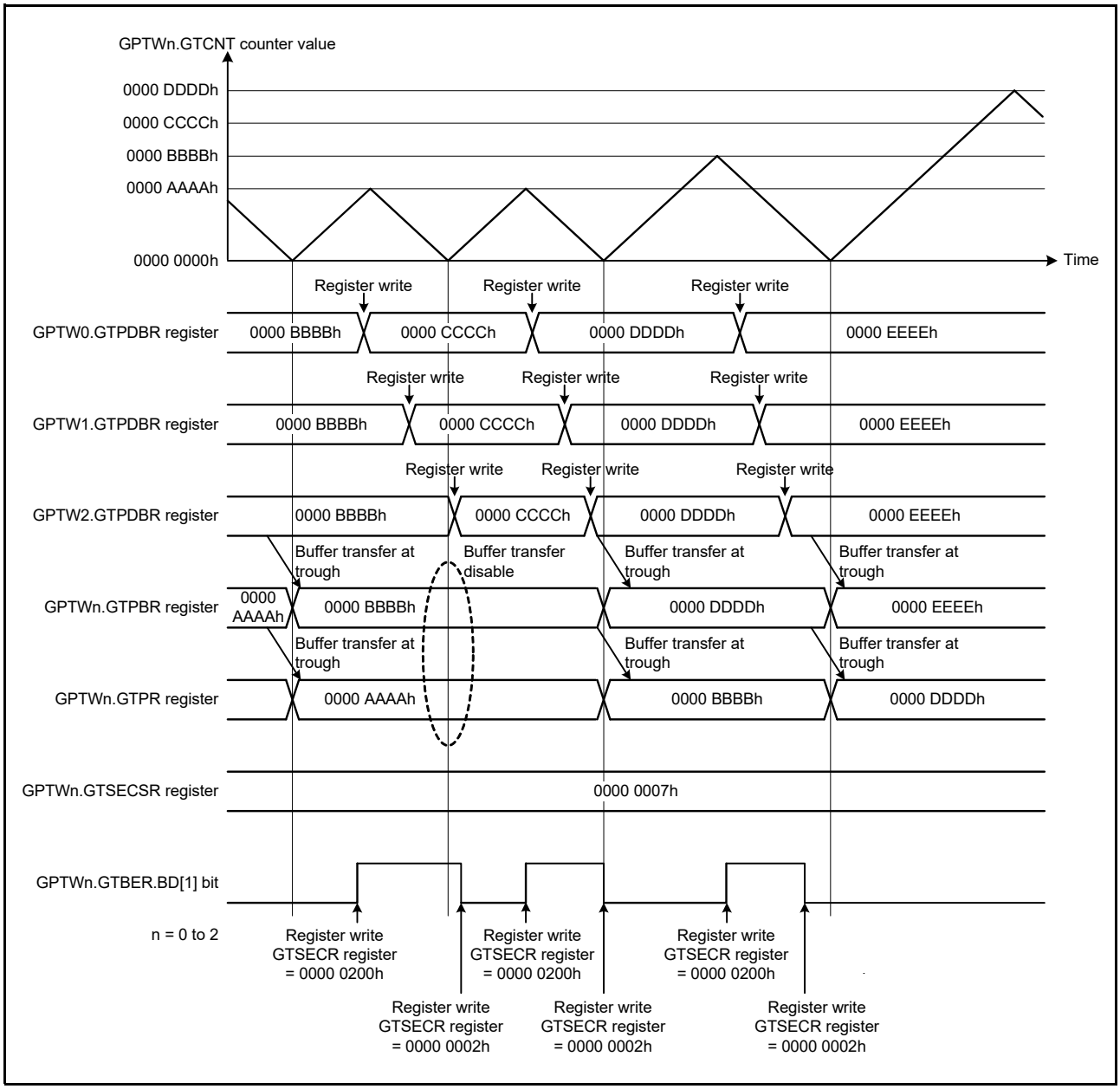


Figure 22.164 Example of Multiple Channel Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation)

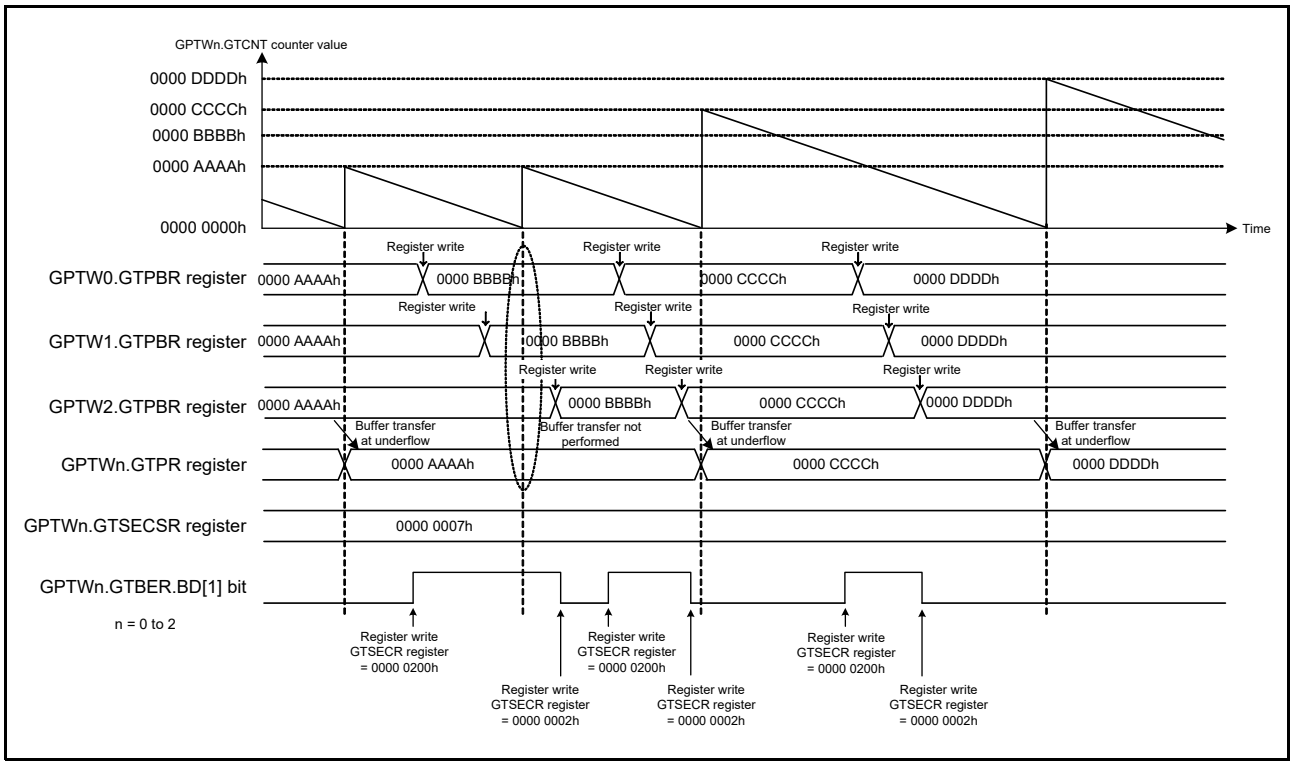


Figure 22.165 Example of Multiple Channel Operation for Disabling Buffer Operation (Sawtooth Waves, Single Buffer Operation)

22.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 7; m = A, B)

For protection from system failure, the output negate control which changes the GTIOCnm pin output forcibly by the output stop request from the POEG is provided.

Output protection is required when the same output level being on the GTIOCnA and GTIOCnB pins is detected. At that time, the detection of disabling of output is output to the group of the POEG set in the GTINTAD.GRP[1:0] bits based on the setting of the output disable detection enable bits, i.e. the GRPABH, and GRPABL bits in the GTINTAD register. The POEG takes the logical OR of detection of the types described above and other forms of detection of the disabling of output, and outputs requests for the disabling of output to the GPTW.

By setting the GTINTAD.GRP[1:0] bits, one output stop request can be selected out of four output stop requests which are input from the POEG as output stop request signal common in the GTIOCnA and GTIOCnB pins. Selected output stop request can be checked by reading the GTST.ODF flag.

The states of outputs at the time of control for their negation can be set with the GTIOR.OADF[1:0] bits for the GTIOCnA pin and with the GTIOR.OBDF[1:0] bits for the GTIOCnB pin.

Transition to the output negate condition by generating the output stop request from the POEG is performed asynchronously, while a release from the output negate condition by making the output stop request no longer satisfied is performed at the end of the cycle. It is after 3 PCLKA at shortest when the output negate condition is released after the output stop request becomes no longer satisfied. To reliably control output negation, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of PCLKA.

To release the output stop condition during event count operation, in sawtooth-wave PWM mode 2, or without waiting the end of the cycle, set the OADF[1:0] bits to 00b for the GTIOCnA pin, and set the OBDF[1:0] bits to 00b for the GTIOCnB pin.

Figure 22.166 shows the output negate control operation for the GTIOCnm pin output.

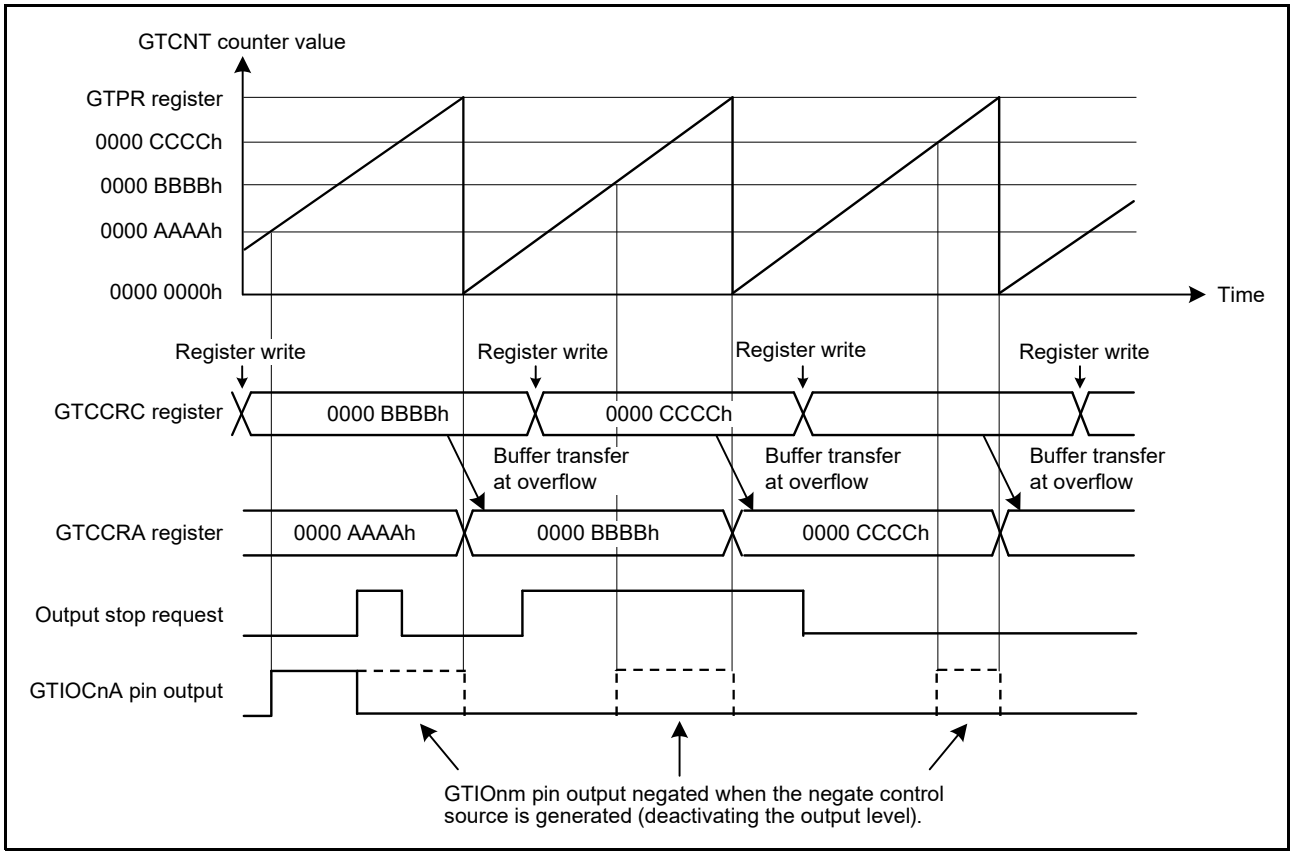


Figure 22.166 Example of Operation for GTIOCnm Pin Output Negate Control (Sawtooth-Wave Up-Counting, Buffer Operation, Active Level: High Output at GTCCRA Register Compare Match, Low Output at the End of the Cycle) and Low Output at Output Negate) (n = 0 to 7; m = A, B)

22.9 Initialization Method of Output Pins

22.9.1 Pin Settings after Reset

The GPTW registers are initialized at a reset. Initialize the GPTW for output to the external pins by making the port mode settings and setting the OAE and OBE bits in the GTIOR register, and then start the counter counting.

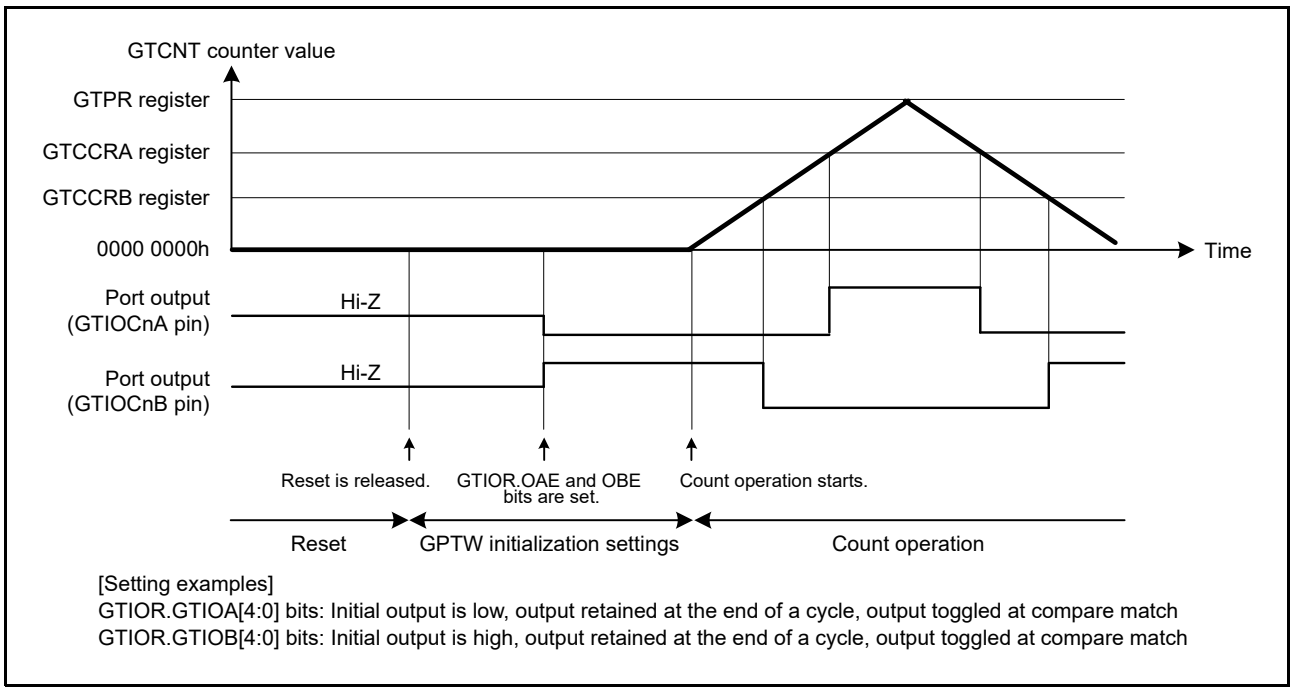


Figure 22.167 Example of Pin Settings after Reset (n = 0 to 7)

22.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPTW operation, the following four types of pin processing can be performed before pin initialization.

(1) Set OAHLD and OBHLD bits in the GTIOR register to 1 and retain the outputs at count stop.
(2) Set OAHLD and OBHLD bits to 0, specify arbitrary output values at OADFLT and OBDFLT bits in the GTIOR register, and output the arbitrary values at count stop.
(3) Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR, and PMR registers of the I/O port in advance. Set the OAE and OBE bits in the GTIOR register to 0 and the control bit in PMR register that corresponds to the pin to 0 to allow the arbitrary values to be output from the pin set as a general output port when an error occurs.
(4) Drive the output to a high impedance state using the POEG function.

When automatic dead time setting has been made, set the GTDTCR.TDE bit to 0 once after counting is stopped.
When counting is stopped, only the values of registers that are changed by a GPTW external source will change. If counting is resumed, operation will carry on from where it was stopped.
If counting was stopped, registers should be initialized before counting is started.

22.10 Usage Notes

22.10.1 Module Stop Function Setting

Operation of the GPTW can be disabled or enabled by the module stop control register. The initial setting is for operation of the GPTW to be halted. Register access is enabled by clearing module stop state. For details, see section 11, Low Power Consumption.

22.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F)

(1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode

The GTCCRA register should satisfy the following conditions:

GTCCRA register > GTDVU register,
GTCCRA register < GTPR register.

(2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode

Set a value greater than 0000 0000h, and less than the setting value of the GTPR register in the GTCCRA register. When 0000 0000h or the same value as that of the GTPR register is set in the GTCCRA register, compare match is generated in one cycle only when [GTCCRA register = 0000 0000h] or [GTCCRA register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRA register, compare match does not occur. Similarly, set a value greater than 0000 0000h, and less than the setting value of the GTPR register in the GTCCRB register. When 0000 0000h or the same value as that of the GTPR register is set in the GTCCRB register, compare match is generated in one cycle only when [GTCCRB register = 0000 0000h] or [GTCCRB register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRB register, compare match does not occur.

(3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC register < GTCCRD register
  GTCCRC register > GTDVU register
  GTCCRD register < GTPR register – GTDVU register
- In down-counting:
  GTCCRC register > GTCCRD register
  GTCCRC register < GTPR register – GTDVU register
  GTCCRD register > GTDVU register

(4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC register < GTCCRD register < GTPR register
- In down-counting: GTPR register > GTCCRC register > GTCCRD register > 0

Similarly, GTCCRE and GTCCRF registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE register < GTCCRF register < GTPR register
- In down-counting: GTPR register > GTCCRE register > GTCCRF register > 0

(5) In Sawtooth-Wave PWM Mode

The GTCCRA register should be set with the range of 0000 0000h < GTCCRA register < GTPR register. If GTCCRA register = 0000 0000h or GTCCRA register = GTPR register is set, a compare match occurs within the cycle only when GTCCRA register = 0000 0000h or GTCCRA register = GTPR register is satisfied. If GTCCRA register > GTPR register is set, no compare match occurs.
Similarly, GTCCRB register should be set with the range of 0000 0000h < GTCCRB register < GTPR register. If GTCCRB register = 0000 0000h or GTCCRB register = GTPR register is set, a compare match occurs within the cycle only when GTCCRB register = 0000 0000h or GTCCRB register = GTPR register is satisfied. If GTCCRB register > GTPR register is set, no compare match occurs.

(6) In Complementary PWM mode 1, 2, 3

The GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

(7) In Complementary PWM mode 4

In single buffer operation, the GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.
In double buffer operation, the GTCCRn register must be set with the range of GTDVU < GTCCRn < GTPR.

22.10.3 Range of Settings for the GTPBR and GTPDBR Registers in the Complementary PWM Mode

When buffer transfer with the use of the GTPR register is to proceed at the end of a crest section in complementary PWM mode 1, 3, or 4, set the GTPBR and GTPDBR registers so that the GTPR register is within the following range after the transfer.
The range consists of values no less than the GTCNT counter value of the master channel at the end of a crest section, that is, the GTPR register before the transfer – the GTDVU register (GTPBR register ≥ GTPR register – GTDVU register, GTPDBR register ≥ GTPR register – GTDVU register)
When buffer transfer with the use of the GTPR register is to proceed at the end of a trough section or on counter clearing, no restriction applies to the ranges of settings for the GTPBR and GTPDBR registers.

22.10.4 Setting Range of the GTCNT Counter

Other than the sawtooth-wave PWM mode 2 and complementary PWM mode, set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.

22.10.5 The GTCNT Counter Start/Stop

The GTCNT counter start/stop control by the GTCR.CST bit is synchronized with the count clock selected by the GTCR.TPCS[3:0] bits. Since the GTCNT counter starts or stops after one count clock cycle selected by the TPCS[3:0] bits following the CST bit updating, events until the GTCNT counter actually starts are ignored, where events may be accepted and the interrupt may be generated after the CST bit becomes 0.

22.10.6 Order of Priority in Events

(1) GTCNT Counter

Order of priority in events to update the GTCNT counter is shown below.

Table 22.33 Order of Priority in Updating GTCNT Counter

The GTCNT counter updating source	Order of priority
CPU writing (GTCNT counter writing/GTCLR register writing)	High
Clearing by a hardware source set by the GTCSR register	↑
Up-counting/down-counting by a hardware source set by the GTUPSR and GTDNSR registers	↑
Count operation	Low

In case of contention between up-counting by the GTUPSR register and down-counting by the GTDNSR register, the counter value is not updated.
When updating of the GTCNT counter conflicts with reading by the CPU, the value before updating is reflected.

(2) The GTCR.CST Bit

In case that start/stop by a hardware source set by the GTSSR and GTPSR registers conflicts with the CPU writing (GTCR register writing/GTSTR register writing /GTSTP register writing), the CPU writing takes priority.
In case of contention between start by a hardware source set by the GTSSR register and stop by a hardware source set by the GTPSR register, the state of the CST bit does not change.
In case of contention between the CST bit updating and the CPU reading (GTCR register reading/GTSTR register reading /GTSTP register reading), data before updating is read.

(3) GTCCRm Register (m = A to F)

In case of contention between the GTCCRm register writing and an input capture/buffer transfer, an input capture/buffer transfer takes priority over the GTCCRm register writing.
In case that an input capture conflicts with the CPU writing or the counter updating by a hardware source, the counter value before updating is captured.
In case of contention between the GTCCRm register updating and the CPU reading, data before updating is read.

(4) GTPR Register

In case of contention between a buffer transfer and the GTPR register writing, the GTPR register writing takes priority over a buffer transfer.
In case of contention between the GTPR register updating and the CPU reading, data before updating is read.

(5) GTADTRm Register (m = A, B)

In case of contention between a buffer transfer and the GTADTRm register writing, the GTADTRm register writing takes priority over a buffer transfer.
In case of contention between the GTADTRm register updating and the CPU reading, data before updating is read.

(6) GTDVU Register

In case of contention between a buffer transfer and the GTDVU register writing, the GTDVU register writing takes priority over a buffer transfer.
In case of contention between the GTDVU register updating and the CPU reading, data before updating is read.

(7) GTIOR.GTIOm bit (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm bit, writing to GTIOR.GTIOm bit has priority over buffer transfer operation.
When there is a conflict between updating the GTIOR.GTIOm bit and reading by the CPU, pre-update data is read.

22.10.7 Note on Counter Clearing in the Complementary PWM Mode

Counter clearing at the end of a trough section or the end of the initial output section must be avoided in the complementary PWM mode. This can be achieved by using synchronous clearing in response to a match in comparison in another channel set as the source of the trigger, as described in section 22.3.8.3, Synchronous Clear Operation by Inter Channel Cooperation. When using counter clearing other than as stated above, adjust the timing of the operations to avoid counter clearing at the end of a trough section or the end of the initial output section.

22.10.8 Note on Disabling PWM Initial Output After Synchronous Clearing in the Complementary PWM Mode

When initial output on the GTIOCnA and GTIOCnB pins after synchronous clearing in a trough section in the complementary PWM mode has been disabled by setting the GTIOR.CPSCIR bit to 1, the respective values set in the compare match registers (the GTCCRA, GTCCRC, GTCCRD,GTCCRE, and GTCCRF registers) must be more than twice that of the GTDVU register.

23. GPTW Port Output Enable (POEGc)

23.1 Overview

The POEG issues requests to disable output from output pins of the general PWM timer (GPTW). The combination of output pins of the POEG to be disabled can be specified from any channel of the GPTW.
Select the method of detection for disabling the output from the list below.

- Input level detection on the GTETRGx pin (x = A to D)
- Output disabling condition detection by the GPTW
- Detection by comparator (edge detection)
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGx pins can also be used as the external trigger input pins for the GPTW.
Table 23.1 shows specifications, Figure 23.1 shows a system block diagram around the POEG, Figure 23.2 shows a block diagram of the POEG, and Table 23.2 shows input pins.

Table 23.1 POEG Specifications (x = A to D)

Item	Description
Output disabling request by input level detection on the GTETRGx pin	• The output disabling request is issued to the GPTW when a POEGGx.PIDF flag is set to 1 by the detection of input of the selected level on the GTETRGx pin.
Output disabling request by output disabling condition detection signal from the GPTW	• The output disabling request is issued to the GPTW when the GPTW detects a simultaneous active level (high or low) on the GTIOCA and GTIOCB pins and the POEGGx.IOCF flag is set to 1.
Output disabling request by detection signal from the comparator	• The output disabling request is issued to the GPTW when the POEGGx.IOCF flag is set to 1 by the COMPn edge detection signals.
Output disabling request in response to detecting the stopping of oscillation	The output disabling request is issued to the GPTW when the oscillation stop detection circuit for the main clock detects the oscillation stop and the POEGGx.OSTPF flag is set to 1.
Output disabling request by software	The output disabling request is issued to the GPTW when the software sets the POEGGx.SSF bit to 1.
Interrupt	• An interrupt is generated by an output disabling request via the POEGGx.PIDF flag. • An interrupt is generated by an output disabling request via the POEGGx.IOCF flag.
External trigger output to the GPTW	The input signal from the GTETRGx pin is output to the GPTW as an external trigger.
Noise removal	Each GTETRGx pin has a digital noise filter. • Four types of sampling clock are selectable for the filter.

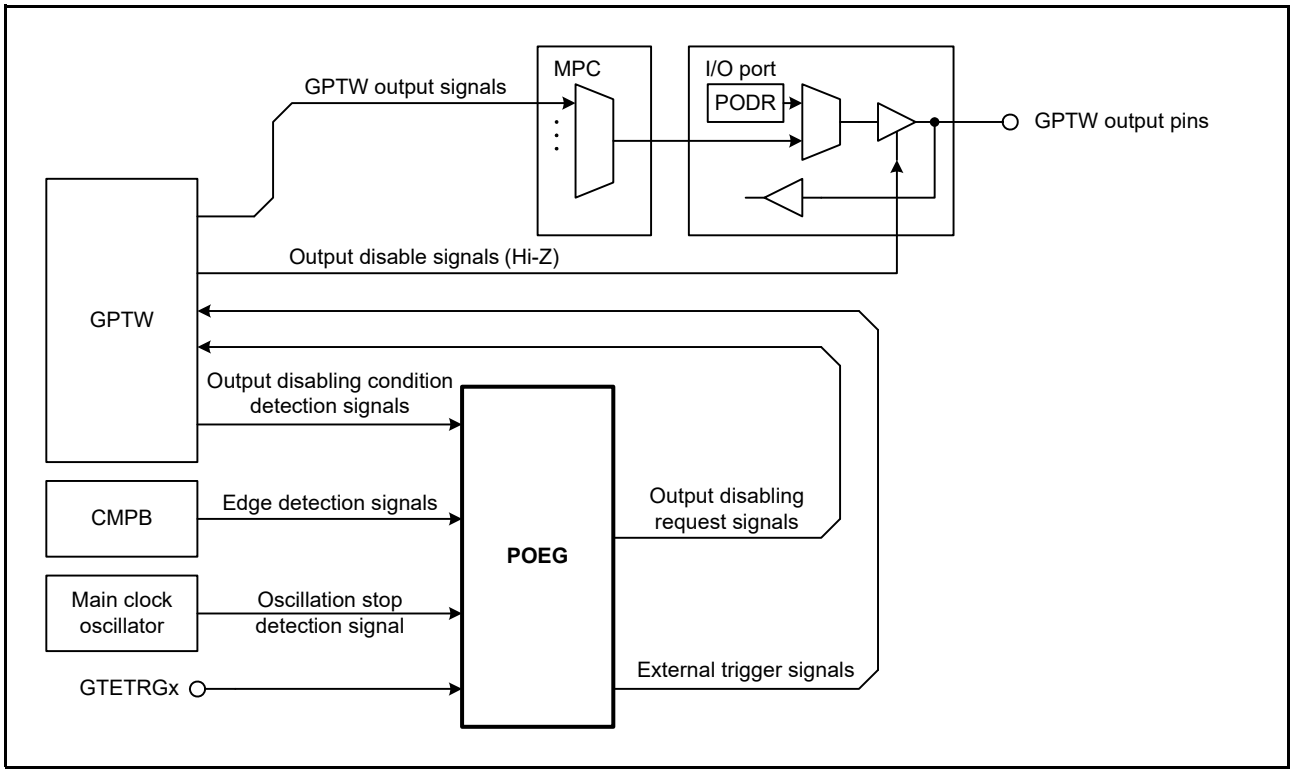


Figure 23.1 POEG System Block Diagram

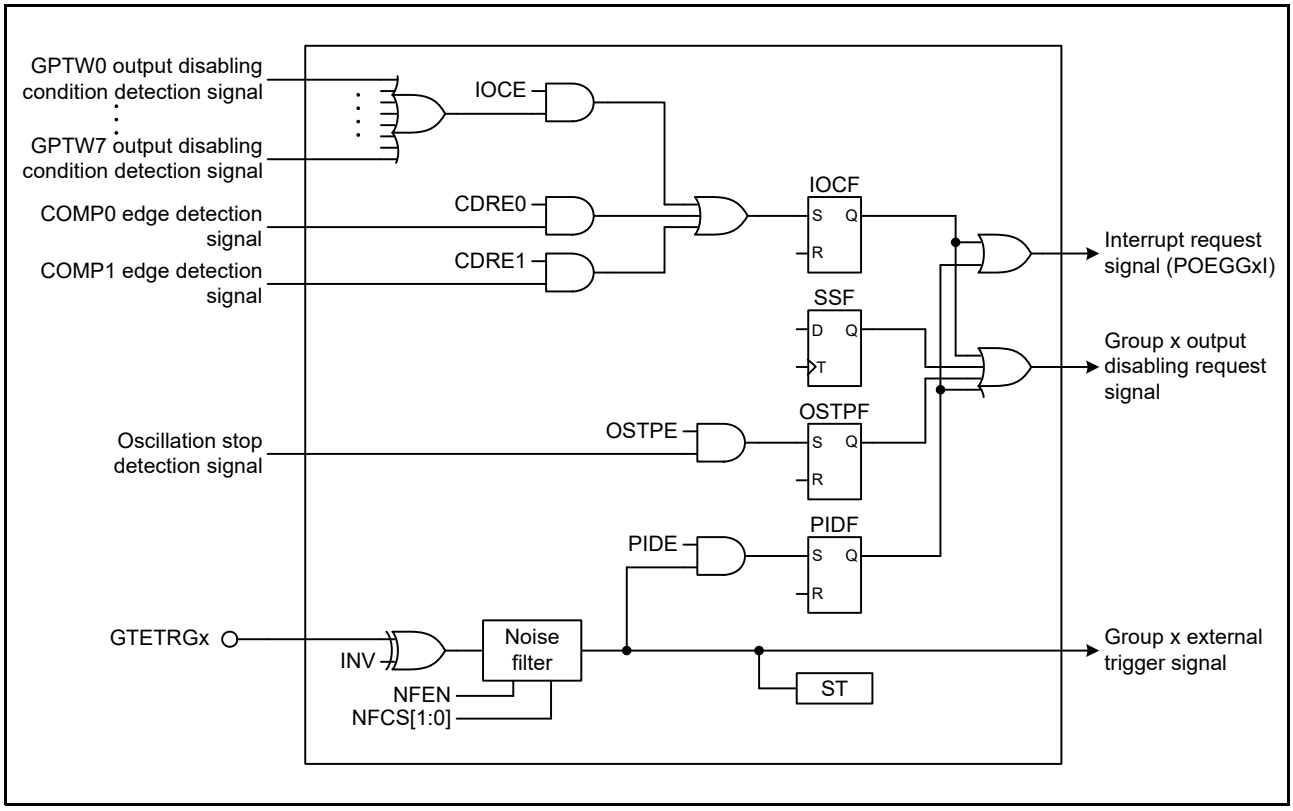


Figure 23.2 POEG Block Diagram (x = A to D)

Table 23.2 POEG I/O Pins

Pin Name	I/O	Function
GTETRGA	Input	Input pin A for output disable request signal for GPTW output pin and GPTW external trigger
GTETRGB	Input	Input pin B for output disable request signal for GPTW output pin and GPTW external trigger
GTETRGC	Input	Input pin C for output disable request signal for GPTW output pin and GPTW external trigger
GTETRGD	Input	Input pin D for output disable request signal for GPTW output pin and GPTW external trigger

23.2 Register Descriptions

23.2.1 POEG Group x Setting Register (POEGGx) (x = A to D)

Address(es): POEG.POEGGA 0009 E000h, POEG.POEGGB 0009 E100h, POEG.POEGGC 0009 E200h, POEG.POEGGD 0009 E300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: Indicates that an output disabling request by the GTETRGx signal has not been issued. 1: Indicates that an output disabling request by the GTETRGx signal has been issued.	R/(W) *1
b1	IOCF	GPTW/CMPB Output Disabling Condition Detection Flag	0: Indicates that an output disabling request by the GPTW output disabling condition detection signal or the COMPn edge detection signal has not been issued. 1: Indicates that an output disabling request by the GPTW output disabling condition detection signal or the COMPn edge detection signal has been issued.	R/(W) *1
b2	OSTPF	Oscillation Stop Detection Flag	0: Indicates that an output disabling request by oscillation stop has not been issued. 1: Indicates that an output disabling request by oscillation stop has been issued.	R/(W) *1
b3	SSF	Software Output Disabling Request	0: Does not issue output disabling request by software. 1: Issue an output disabling request by software.	R/W
b4	PIDE	Port Input Signal Enable	0: Output disabling request by the GTETRGx signal is disabled. 1: Output disabling request by the GTETRGx signal is enabled.	R/W*2
b5	IOCE	GPTW Output Disabling Condition Detection Signal Enable	0: Output disabling request by the GPTW output disabling condition detection signal is disabled. 1: Output disabling request by the GPTW output disabling condition detection signal is enabled.	R/W*2
b6	OSTPE	Oscillation Stop Detection Signal Enable	0: Output disabling request by the oscillation stop detection signal is disabled. 1: Output disabling request by the oscillation stop detection signal is enabled.	R/W*2
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CDRE0	COMP0 Edge Detection Signal Enable	0: Output disabling request by the COMP0 edge detection signal is disabled. 1: Output disabling request by the COMP0 edge detection signal is enabled.	R/W*2
b9	CDRE1	COMP1 Edge Detection Signal Enable	0: Output disabling request by the COMP1 edge detection signal is disabled. 1: Output disabling request by the COMP1 edge detection signal is enabled.	R/W*2
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ST	GTETRGx Signal Status Flag	0: The GTETRGx signal output to the GPTW is low. 1: The GTETRGx signal output to the GPTW is high.	R
b27 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b28	INV	GTETRGx Input Inverting	0: Does not invert the input signal from the GTETRGx pin. 1: Invert the input signal from the GTETRGx pin.	R/W
b29	NFEN	Noise Filter Enable	0: Digital noise filter on the GTETRGx pin is disabled. 1: Digital noise filter on the GTETRGx pin is enabled.	R/W
b31, b30	NFCS[1:0]	Noise Filter Clock Select	b31 b30 0 0: The input signal from the GTETRGx pin is sampled every PCLKB/1 clock 0 1: The input signal from the GTETRGx pin is sampled every PCLKB/8 clock 1 0: The input signal from the GTETRGx pin is sampled every PCLKB/32 clock 1 1: The input signal from the GTETRGx pin is sampled every PCLKB/128 clock	R/W

Note 1. Only 0 can be written to clear the flag.
Note 2. This bit can be written only once after reset.

The POEGGx register controls output disabling requests and an external trigger for the GPTW in response to the various detection signals.

SSF Bit (Software Output Disabling Request)

Writing 1 to this bit leads to an output disabling request being issued to the GPTW, and writing 0 to the bit releases the GPTW from the output disabling request.

23.3 Operation

The POEG has four groups (A to D), all of which operate the same and are independent of each other.
The output signal of each channel of the GPTW is connected to all groups of the POEG, and the output signal of each group of the POEG is connected to all channels of the GPTW. Use the GPTW register to configure which group the GPTW channel is controlled by.

23.3.1 Output Disabling Request by Detection of Input Level on the GTETRGx Pin (x = A to D)

The output disabling requests using the GTETRGx pin are issued via the POEGGx.PIDF flag.
When the input level set by the POEGGx.NFCS[1:0], NFEN, and INV bits is detected while the PIDE bit is 1, the PIDF flag is set to 1 and an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, clear the PIDF flag. For details, refer to section 23.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx Register.

23.3.1.1 Digital Noise Filter

Each GTETRGx pin has a digital noise filter (x = A to D). Figure 23.3 shows an example of the noise filter operation. This example assumes that the POEGGx.INV bit is 0 (no inversion, i.e., high level detection) and NFEN bit is 1 (noise filter enabled).
The signal level on the GTETRGx pin is sampled with the sampling clock selected by the POEGGx.NFCS[1:0] bits, and when last three consecutive samples are high, the POEG determines that high is input and issues an output disabling request to the GPTW.
At this time, if low is detected even once, the POEG does not determine that high is input. Note that changes in the GTETRGx pin level that occur between sampling timings are ignored.
The output of the digital noise filter is used for two purposes: output disabling request via the POEGGx.PIDF flags and an external trigger output to the GPTW.

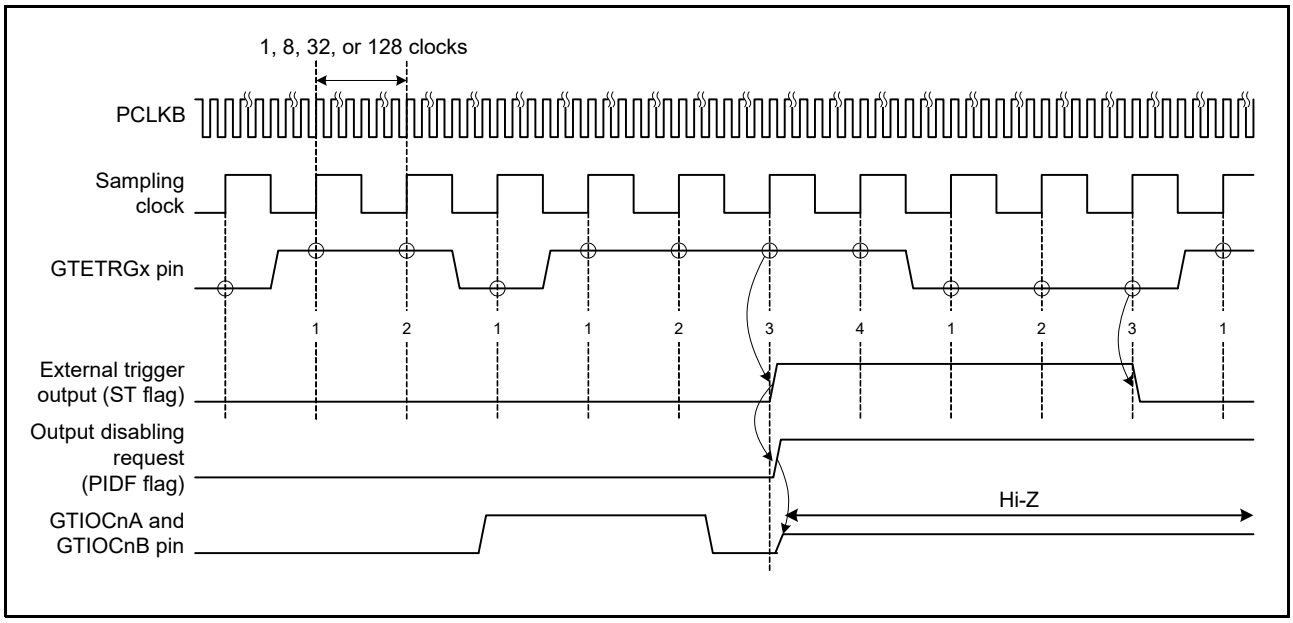


Figure 23.3 Example for Operation of Digital Noise Filter

23.3.2 Output Disabling Request by Output Disabling Condition Detection Signal from GPTW

When the GPTW detects a simultaneous high or low output while the POEGGx.IOCE bit is 1, the IOCF flag is set to 1 and an output disabling request is issued to all GPTW channels. The IOCF flag is also used to indicate the output disabling request by COMPn edge detection signals.
To cancel the output disabling request, clear the IOCF flag. For details, refer to section 23.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx Register.
To detect a simultaneous high output or simultaneous low output in the GPTW, set the GPTWn.GTINTAD.GRPABH or GRPABL bit to 1, respectively. The output disabling condition detection signal generated by each GPTW channel is output only to the POEG group selected by the GPTWn.GTINTAD.GRP[1:0] bits. For details, refer to section 22.2.15, General PWM Timer Interrupt Output Setting Register (GTINTAD).

23.3.3 Output Disabling Request by Comparator Detection Signals

The output disabling requests by the comparator are issued via the POEGGx.IOCF flag using the COMPn edge detection signals
When the COMPn edge detection signal becomes 1 while the POEGGx.CDREn bit is 1, the IOCF flag is set to 1 and an output disabling request is issued to all GPTW channels. The IOCF flag is also used to indicate an output disabling request by GPTW.
To cancel the output disabling request, clear the IOCF flag. For details, refer to section 23.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx Register.

23.3.4 Output Disabling Request by Oscillation Stop Detection Signal

When the oscillation stop detection circuit in the clock generation circuit detects the oscillation stop while the POEGGx.OSTPE bit is 1, the OSTPF flag is set to 1 and an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, clear the OSTPF flag. For details, refer to section 23.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx Register.

23.3.5 Output Disabling Request by a Register

When the POEGGx.SSF bit is set to 1, an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, set the SSF bit to 0. For details, refer to section 23.3.6, Canceling Output Disabling Request.

23.3.6 Canceling Output Disabling Request

There are two ways to cancel an output disabling request as follows.

- Reset (return to the initial state)
- Clearing the flags in the POEGGx register

23.3.6.1 Cancellation of Output Disabling Request by Reset

Any type of reset can cancel an output disabling request. For details on resets, refer to section 6, Resets.

23.3.6.2 Cancellation of Output Disabling Request by Clearing Flags in the POEGGx Register

An output disabling request is canceled by setting all of the following flags and bit that are set to 1 to 0.

- POEGGx.PIDF flag
- POEGGx.IOCF flag
- POEGGx.OSTPF flag
- POEGGx.SSF bit

When the output disabling request is canceled, the GPTW captures this signal at the end of the next PWM cycle and releases the PWM outputs from the disabled state after three PCLKA cycles at the earliest. Figure 23.4 shows the timing for releasing the output disabled state.
To clear the flag, make sure that the detection signals that set the flag to 1 is 0 before writing 0 to the flag. Writing 0 to the flag when the detection signal is 1 does not clear the flag. The COMPn edge detection signals are set to 1 immediately after the cause is detected, and then return to 0. Therefore, writing 0 at any time clears the flag.
The status of the level detection signals can be monitored using the following status flags.

- GTETRGx pin input signal: POEGGx.ST flag (GTETRGx signal status flag)
- Output disabling condition detection signals: GPTWn.GTST.OABLF flag (simultaneous low output flag)
  GPTWn.GTST.OABHF flag (simultaneous high output flag)
- Oscillation stop detection signal: OSTDSR.OSTDF flag (oscillation stop detection flag)

The status of the comparator comparison results before edge detection can be monitored using the following status flags.

- Comparator comparison results: CMPB.CPBFLG.CPB0OUT flag (comparator B0 monitor flag)
  CMPB.CPBFLG.CPB1OUT flag (comparator B1 monitor flag)

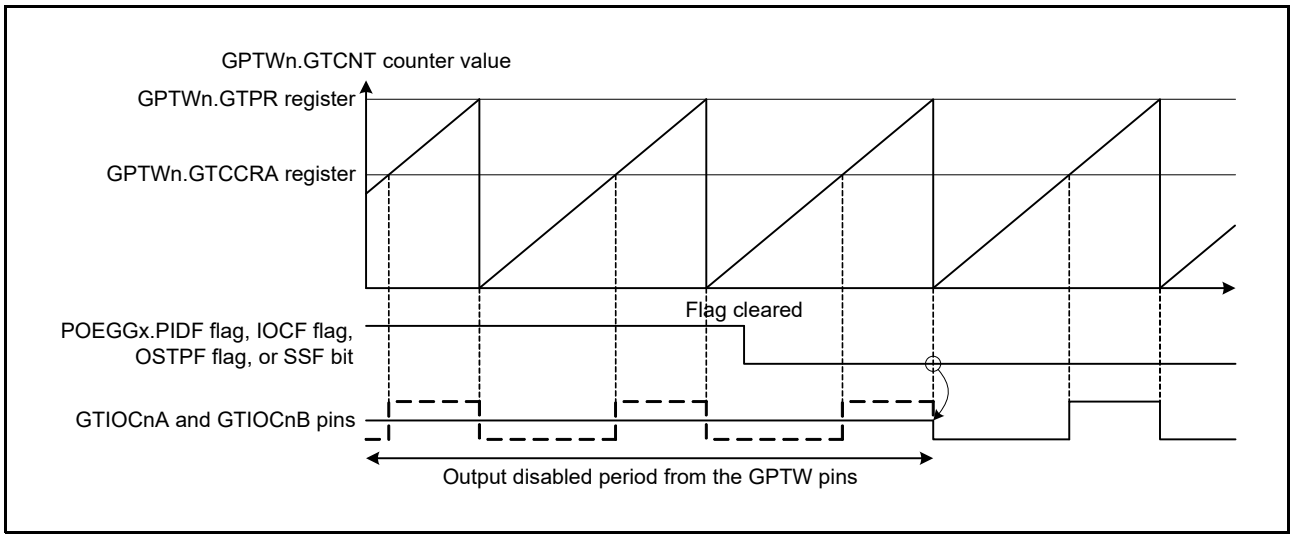


Figure 23.4 Timing for Releasing the Output Disabled State of the GPTW Pins by Canceling the Output Disabling Request

23.4 Interrupt Sources

The POEG has the following interrupt sources for each group.

- Detection of input level on the GTETRGx pin (x = A to D)
- Detection of output disabling condition detection signal from the GPTW
- Detection of COMPn edge detection signal (n = 0 to 1)

Table 23.3 lists the interrupt sources.

Table 23.3 Interrupt sources

Interrupt Name	Symbol	Corresponding Flag	Interrupt Source
Group A Interrupt	POEGGAI	POEGGA.PIDF	Input signal from the GTETRGA pin
		POEGGA.IOCF	Output disabling condition detection signal from the GPTW
			COMPn edge detection signal
Group B Interrupt	POEGGBI	POEGGB.PIDF	Input signal from the GTETRGB pin
		POEGGB.IOCF	Output disabling condition detection signal from the GPTW
			COMPn edge detection signal
Group C Interrupt	POEGGCI	POEGGC.PIDF	Input signal from the GTETRGC pin
		POEGGC.IOCF	Output disabling condition detection signal from the GPTW
			COMPn edge detection signal
Group D Interrupt	POEGGDI	POEGGD.PIDF	Input signal from the GTETRGD pin
		POEGGD.IOCF	Output disabling condition detection signal from the GPTW
			COMPn edge detection signal

23.5 External Trigger Output to GPTW

The POEG outputs the input signals on the GTETRGx pin (x = A to D) to the GPTW as external trigger signals via polarity conversion and a digital noise filter. The external trigger signal can be monitored by the POEGGx.ST flag. The GPTW can perform the following operations with the external trigger signal.

- Count start
- Count stop
- Counter clear
- Count up
- Count down
- Input capture

For details, refer to section 22, General PWM Timer (GPTWa).

23.6 Usage Notes

23.6.1 Transitions to Low-Power Consumption Mode

When using the POEG, do not enter software standby mode. In this mode, the POEG is stopped and output disabling request cannot be issued.

23.6.2 Setting the Function for Stopping the Module

Operation of the POEG can be disabled or enabled using the module stop control register A (MSTPCRA). After reset is released, the POEG function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption. Note that the module stop bit for the POEG is also used for the GPTW.

23.6.3 Duplication of Output Disabling Request

The output disabling request via flags is a logical OR of the POEGGx.PIDF, IOCF, OSTPF flags, and the SSF bit. Therefore, note that the output disabling request will not be canceled if any one of the flags/bit is set to 1.

24. 8-Bit Timer (TMRa)

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, TMR2)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0, TMR2) (2) Event counting operation (TMR0, TMR2) (3) Counting restart operation (TMR0, TMR2)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
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 31, Serial Communications Interface (SCIk, SCIh).
Note 2. For details, refer to section 37, 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	—
SCI base clock generation*1		✓		—	✓		—
Capable of generating operating clock for REMC*2		✓	—	—	—	—	—
ELC output event	Compare match A	✓	—	✓	✓	—	✓
	Compare match B	✓	—	✓	✓	—	✓
	TCNT overflow	✓	—	✓	✓	—	✓
ELC input event	Counting start	✓	—	—	✓	—	—
	Event counting	✓	—	—	✓	—	—
	Counting restart	✓	—	—	✓	—	—
Module stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

✓: Possible

—: Impossible

Note 1. For details, refer to section 31, Serial Communications Interface (SCIk, SCIh).
Note 2. For details, refer to section 37, Remote Control Signal Receiver (REMCa).
Note 3. For details, refer to section 11, Low Power Consumption.

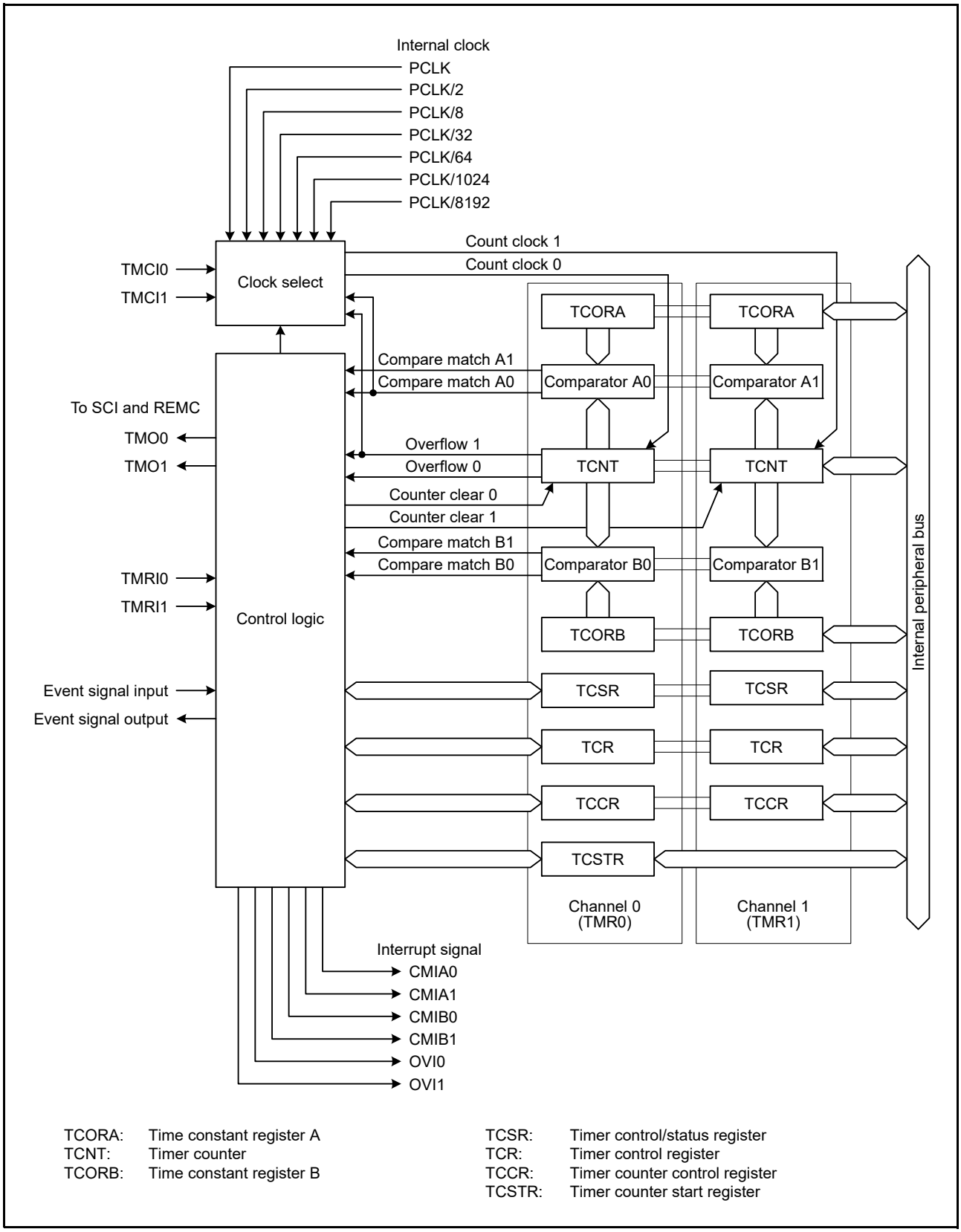


Figure 24.1 Block Diagram of TMR (Unit 0)

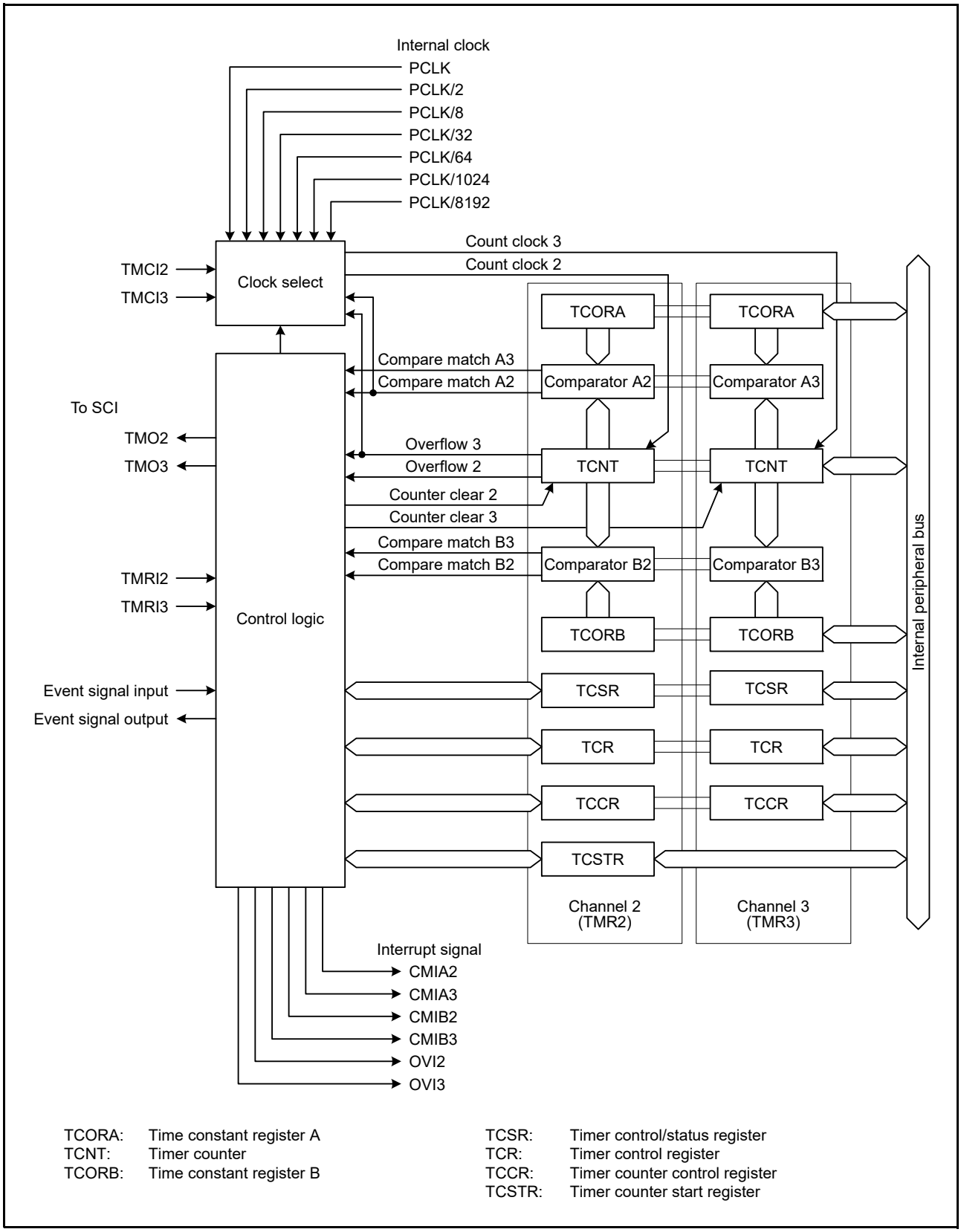


Figure 24.2 Block Diagram of TMR (Unit 1)

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

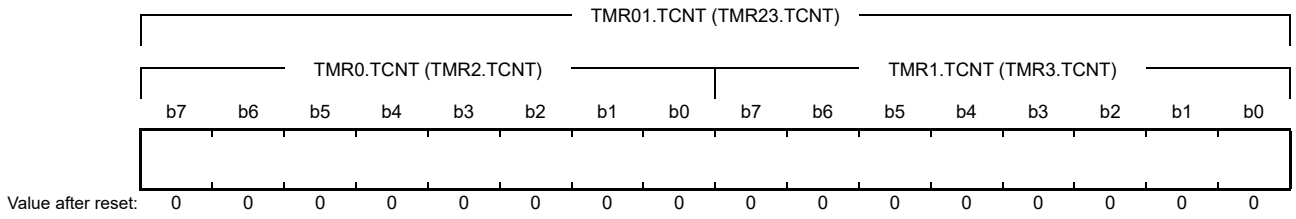
24.2 Register Descriptions

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.

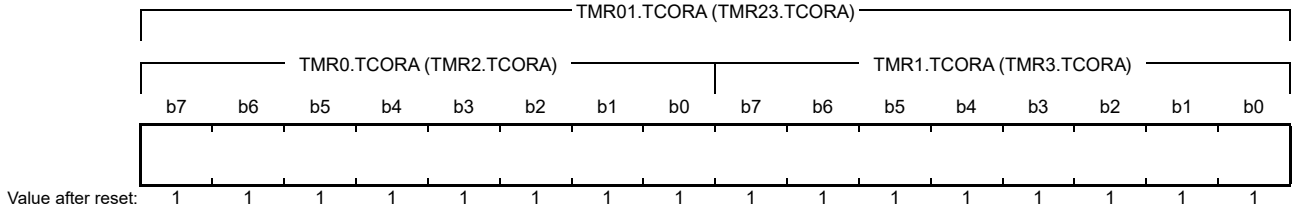
TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

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 (ICUb), and Table 24.6, TMR Interrupt Sources.

24.2.2 Time Constant Register A (TCORA)

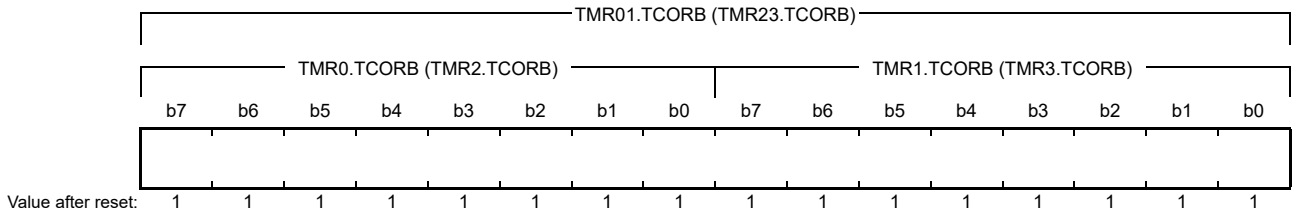
Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h, TMR01.TCORA 0008 8204h, TMR23.TCORA 0008 8214h



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					Description
	CSS[1:0]		CKS[2:0]			
	b4	b3	b2	b1	b0	
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
	—	—	—	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	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, TMR2.TCSTR 0008 821Ch

	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.

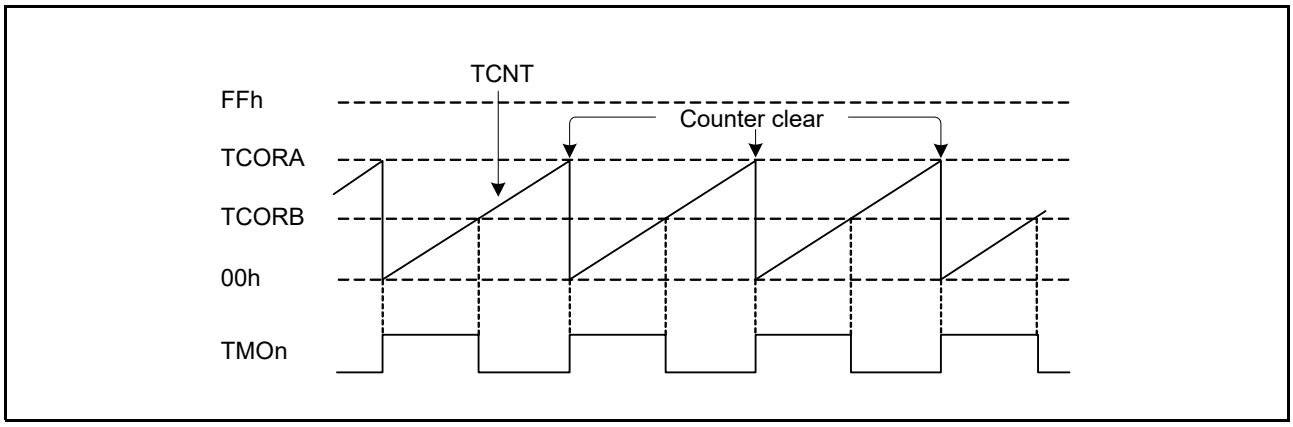


Figure 24.3 Example of Pulse Output (n = 0 to 3)

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.

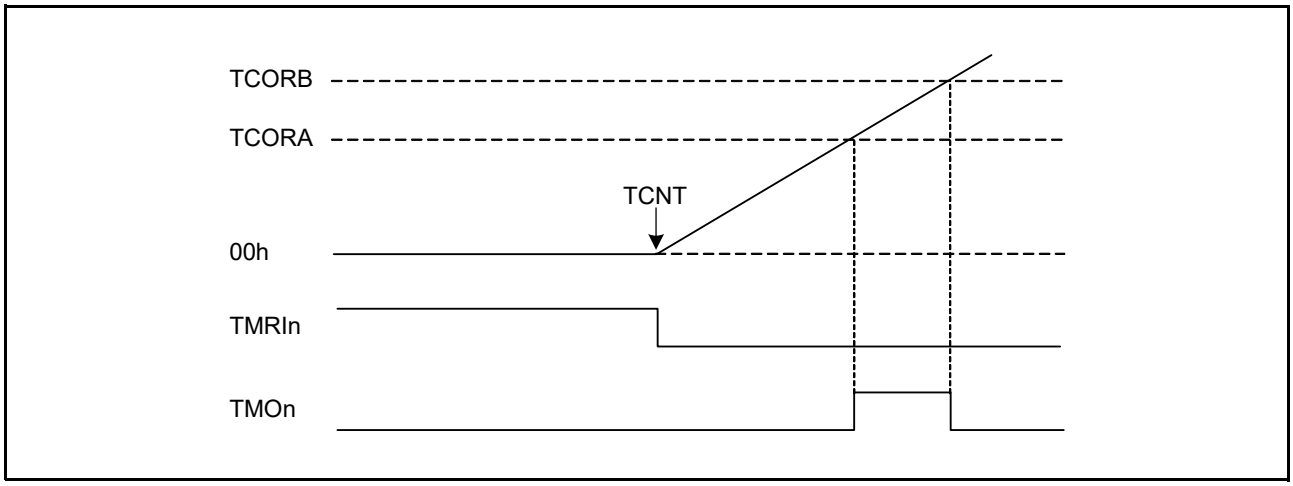


Figure 24.4 Example of External Counter Reset Signal Input (n = 0 to 3)

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.

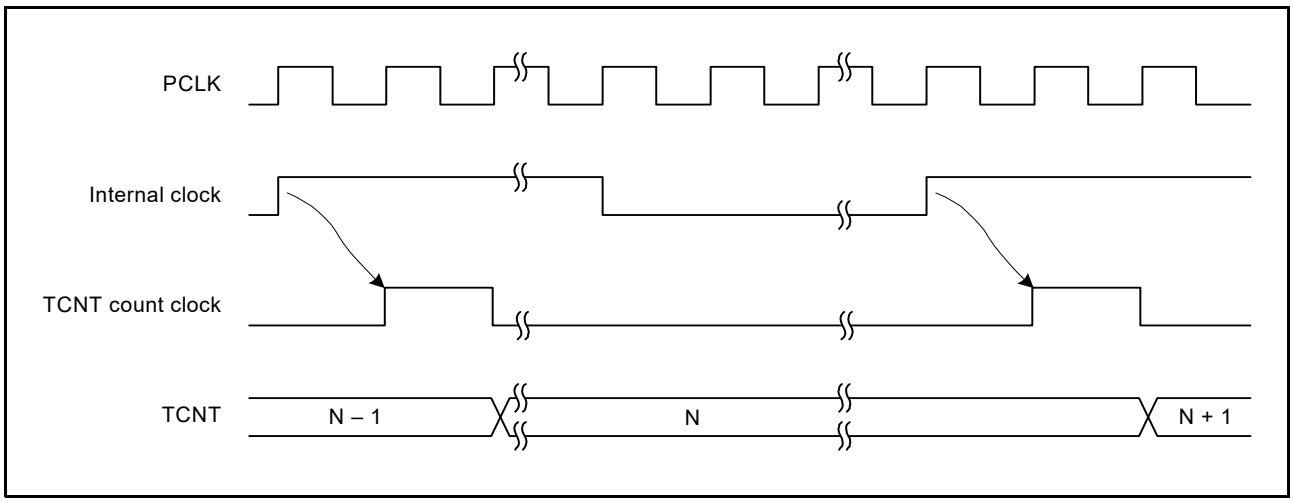


Figure 24.5 Count Timing for Internal Clock

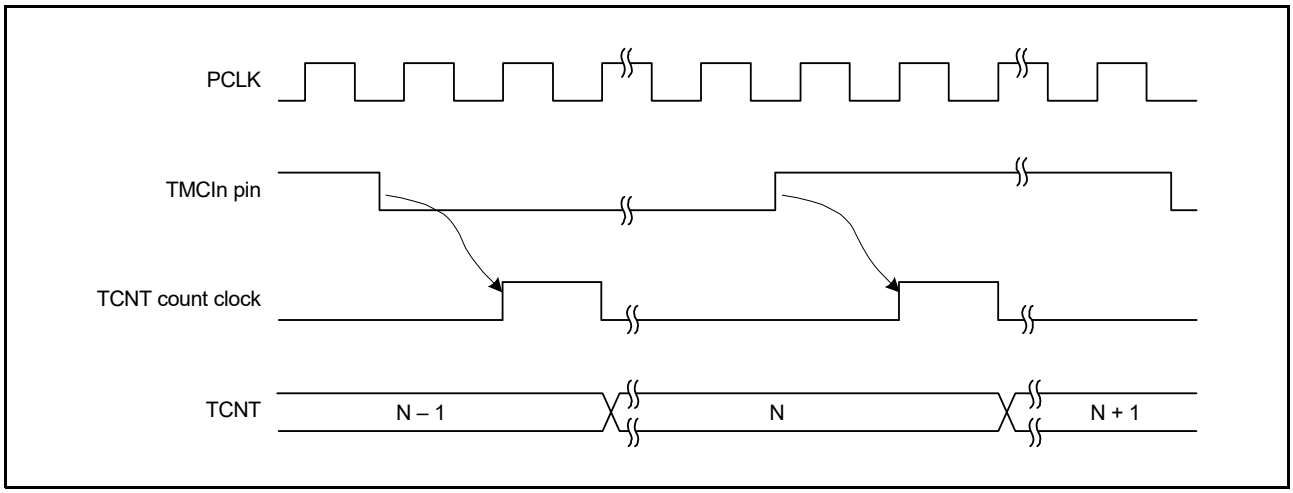


Figure 24.6 Count Timing for External Clock (at Both Edges)

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 (ICUb) and Table 24.6.

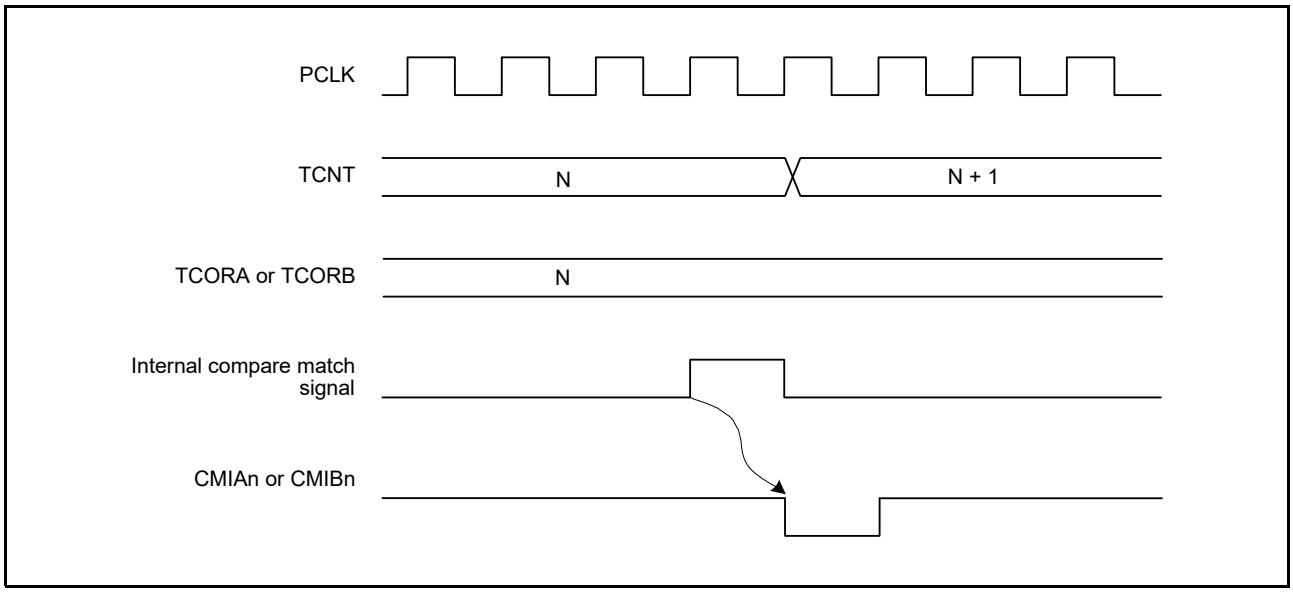


Figure 24.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

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.

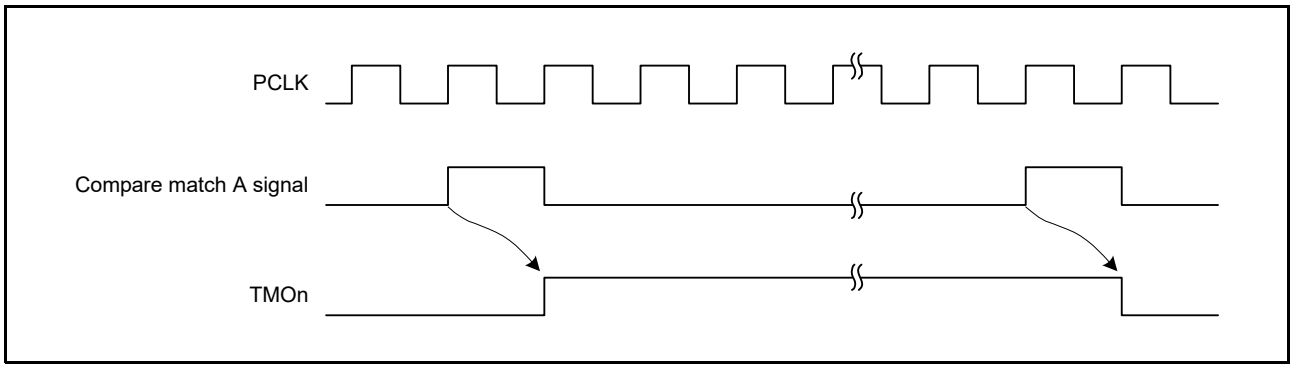


Figure 24.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 3)

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.

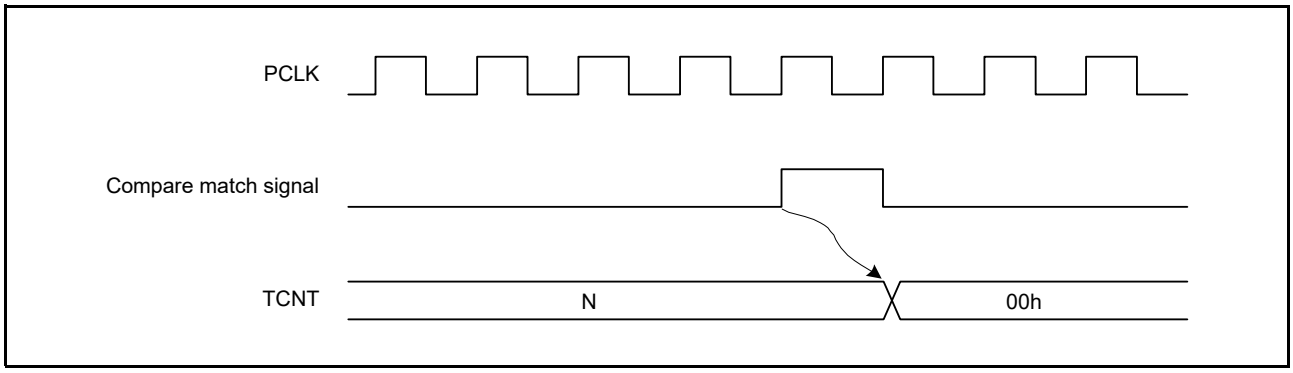


Figure 24.9 Timing of Counter Clear by Compare Match

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.

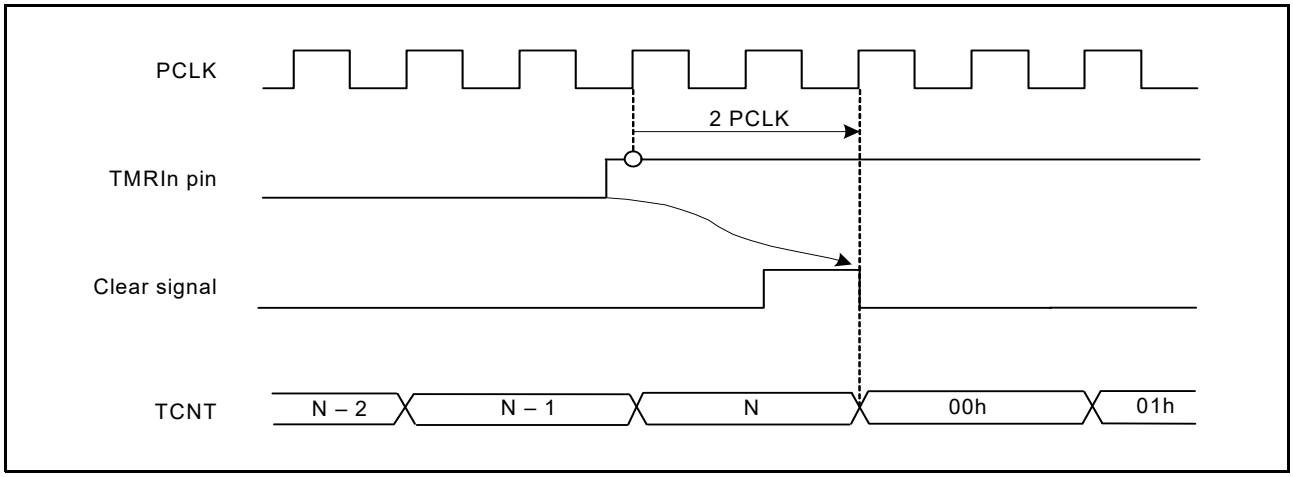


Figure 24.10 Clear Timing by External Counter Reset Signal (Rising Edge)

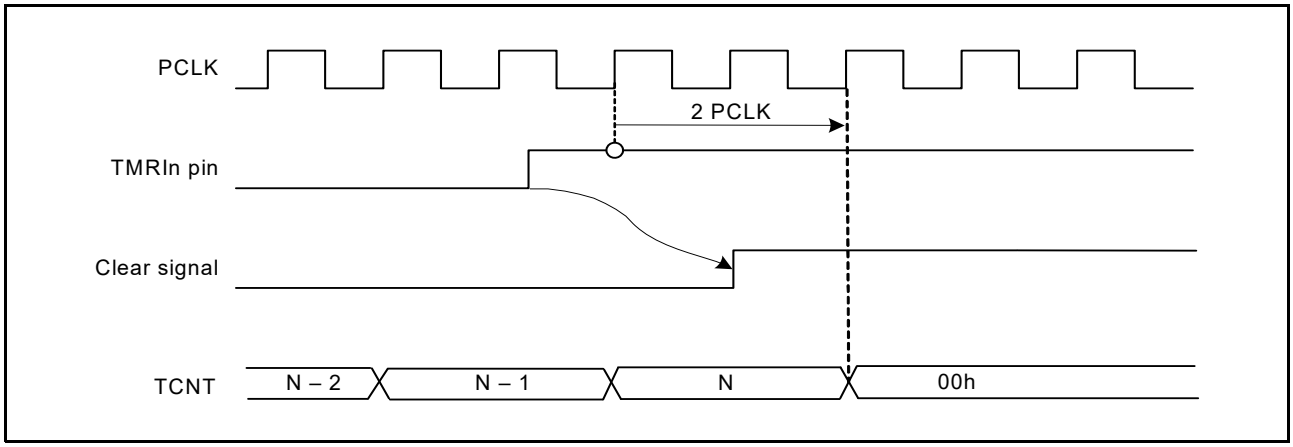


Figure 24.11 Clear Timing by External Counter Reset Signal (High Level)

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 (ICUb) and Table 24.6.

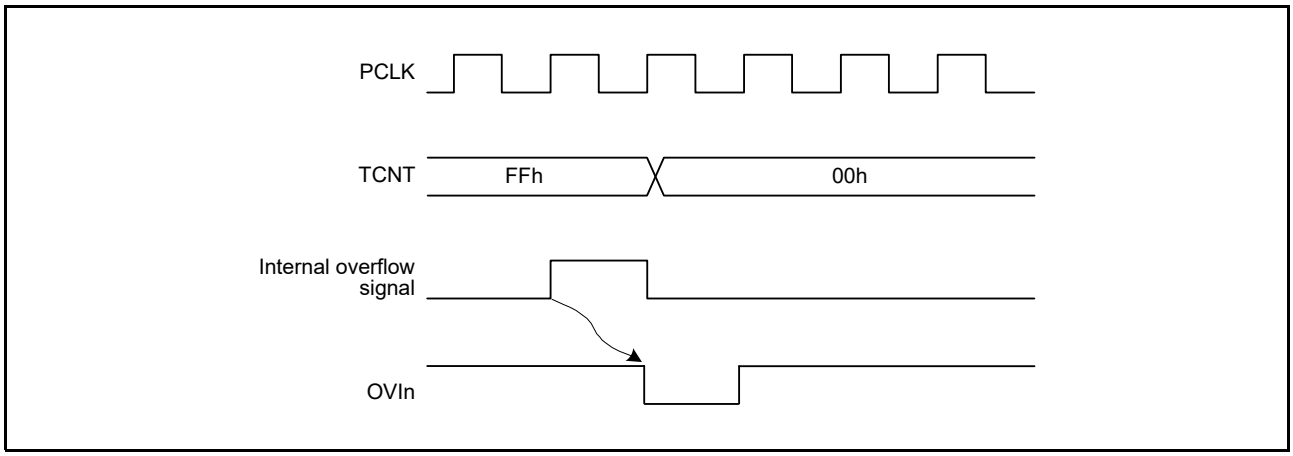


Figure 24.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

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	Priority
CMIA0	TMR0.TCORA compare match	Possible	High
CMIB0	TMR0.TCORB compare match	Possible	
OVI0	TMR0.TCNT overflow	Not possible	
CMIA1	TMR1.TCORA compare match	Possible	
CMIB1	TMR1.TCORB compare match	Possible	
OVI1	TMR1.TCNT overflow	Not possible	
CMIA2	TMR2.TCORA compare match	Possible	
CMIB2	TMR2.TCORB compare match	Possible	
OVI2	TMR2.TCNT overflow	Not possible	
CMIA3	TMR3.TCORA compare match	Possible	
CMIB3	TMR3.TCORB compare match	Possible	
OVI3	TMR3.TCNT overflow	Not possible	Low

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 and TMR2.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMR0.TCR.OVIE or TMR2.TCR.OVIE, TMR0.TCR.CMIEA or TMR2.TCR.CMIEA, and TMR0.TCR.CMIEB or TMR2.TCR.CMIEB). 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] and ELOPD.TMR2MD[1:0] bits of the ELC select the count start operation.

(2) Event Count

When the TMR event count operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the TMR count restart operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

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.

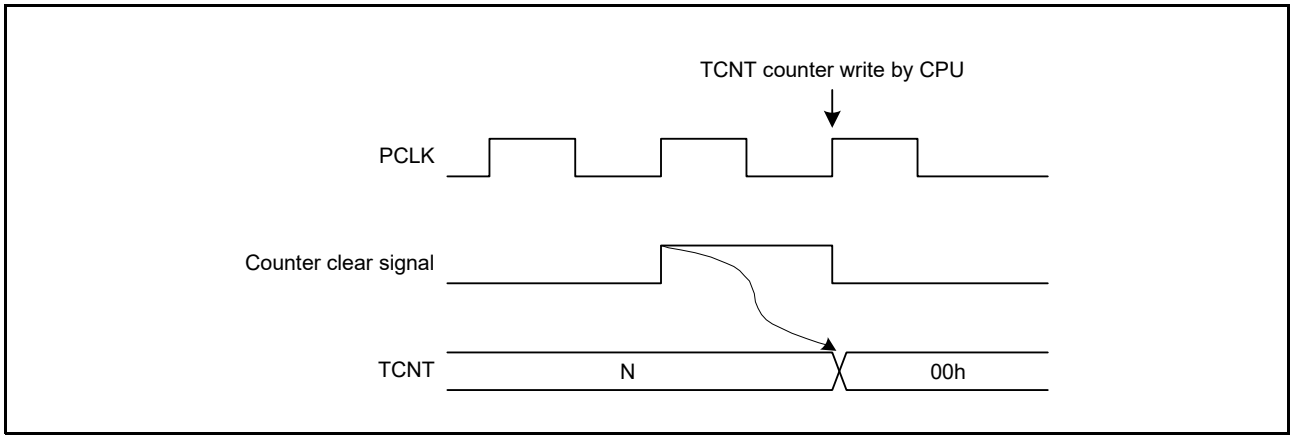


Figure 24.13 Conflict between TCNT Write and Counter Clear

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.

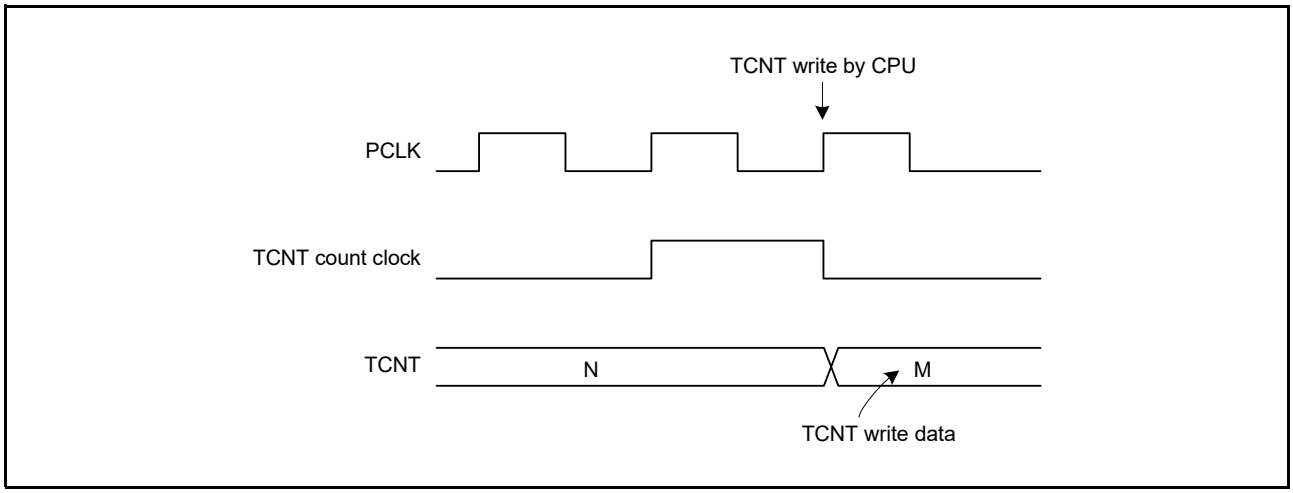


Figure 24.14 Conflict between TCNT Write and Increment

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.

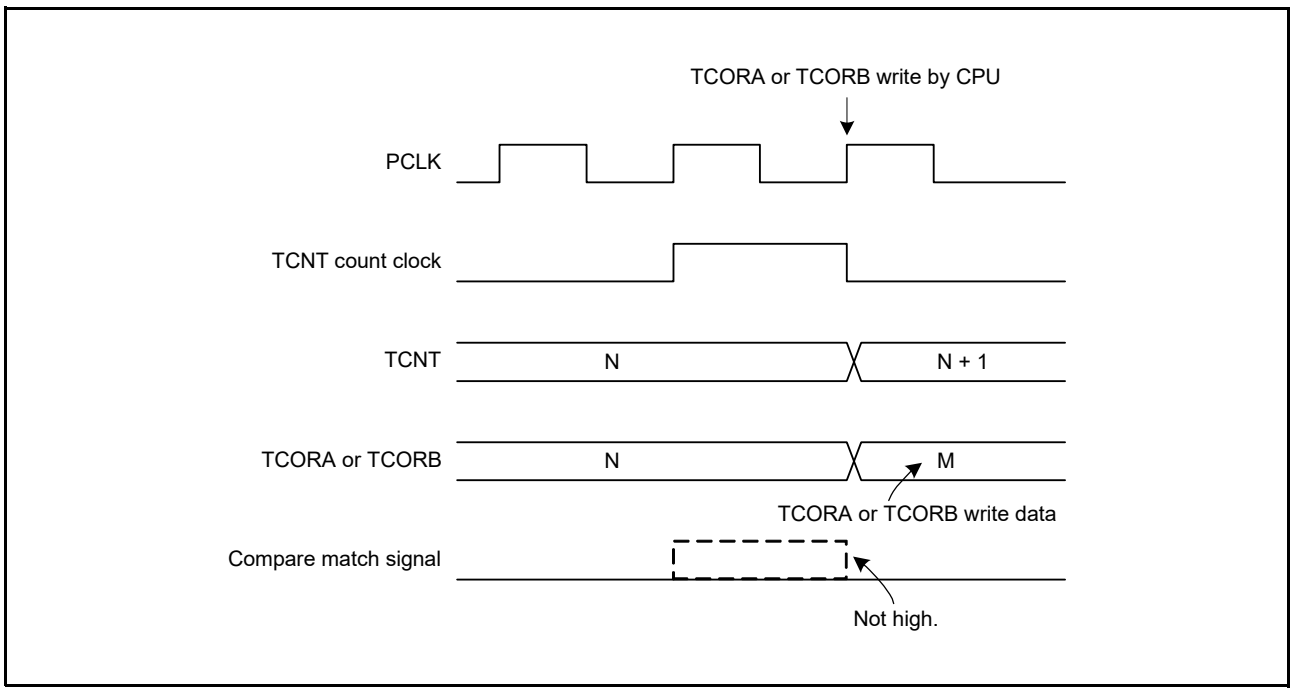


Figure 24.15 Conflict between TCORA or TCORB Write and Compare Match

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 24.7.

Table 24.7 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 24.8 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 24.8, 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 24.8 Switching of Internal Clocks and TCNT Operation (1/2)

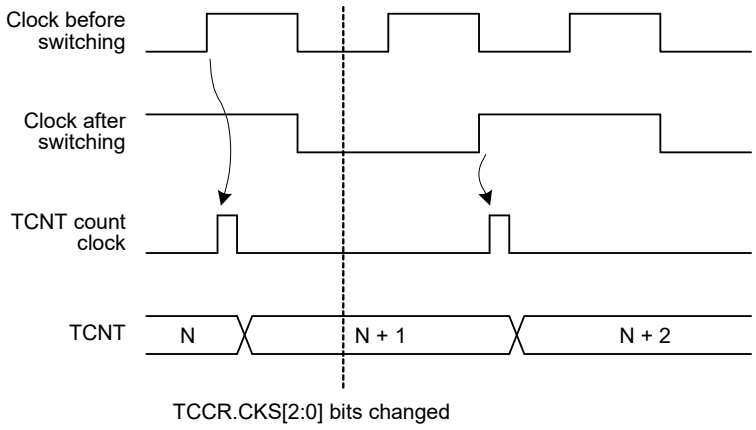
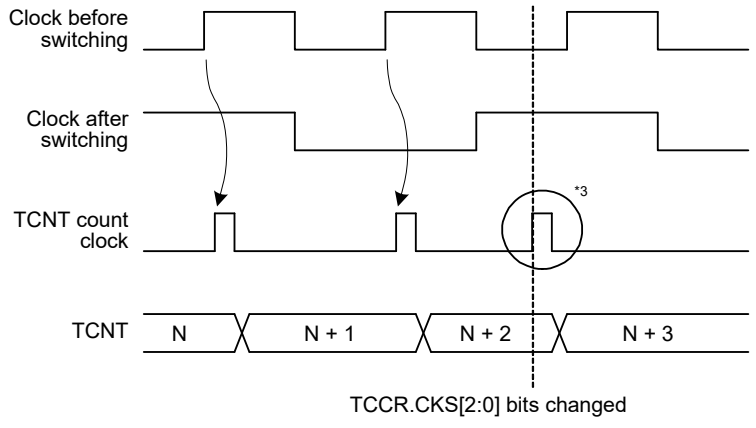
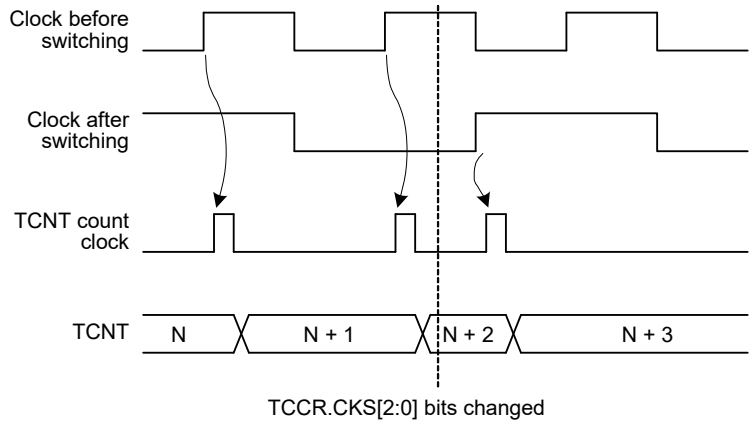
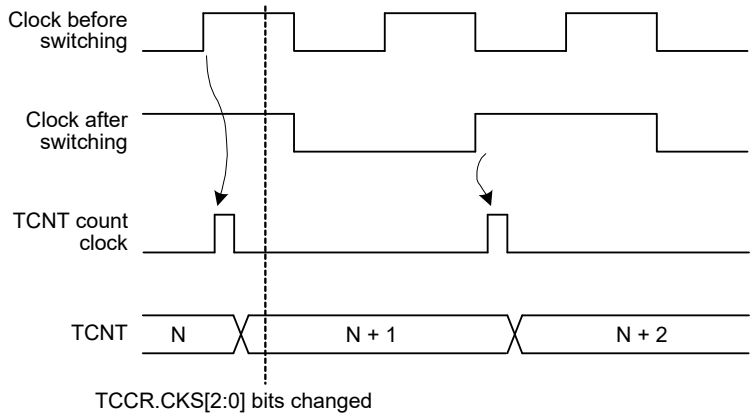
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Table 24.8 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.

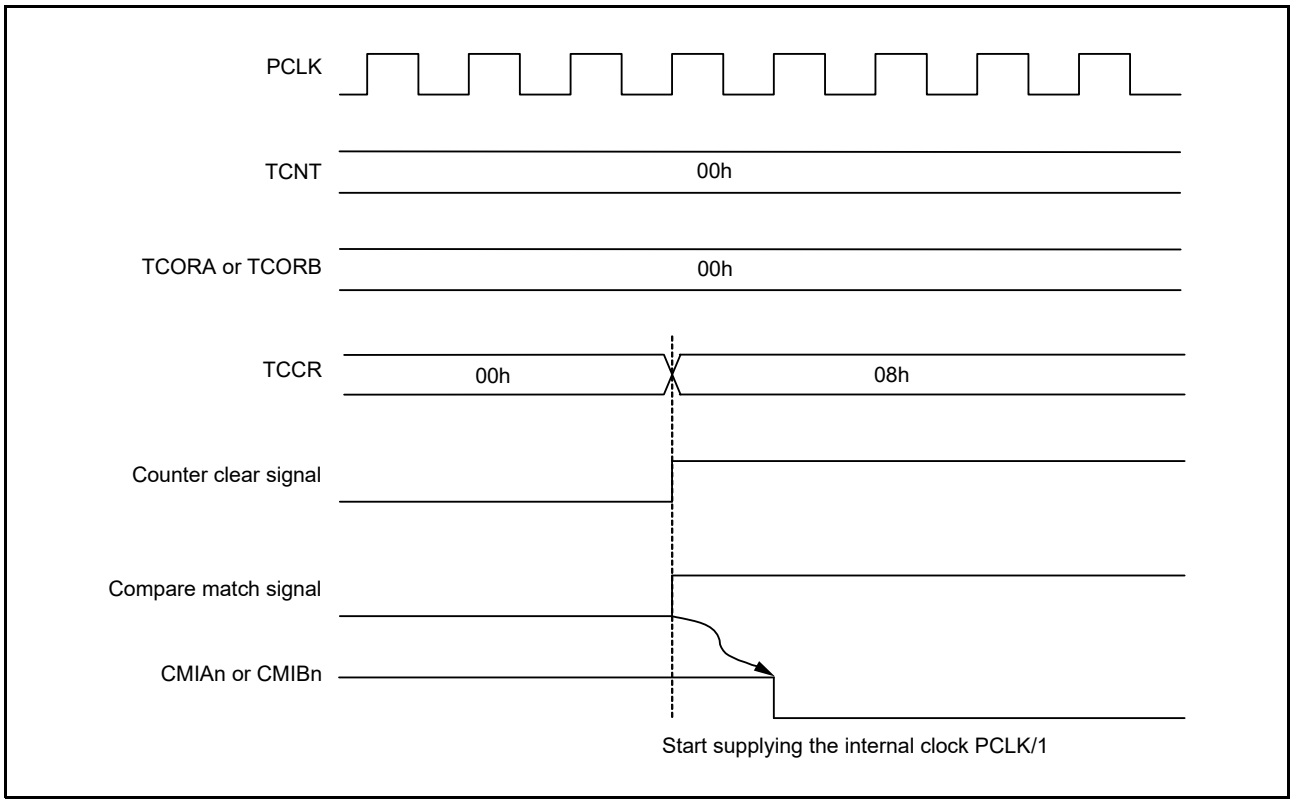


Figure 24.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

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.

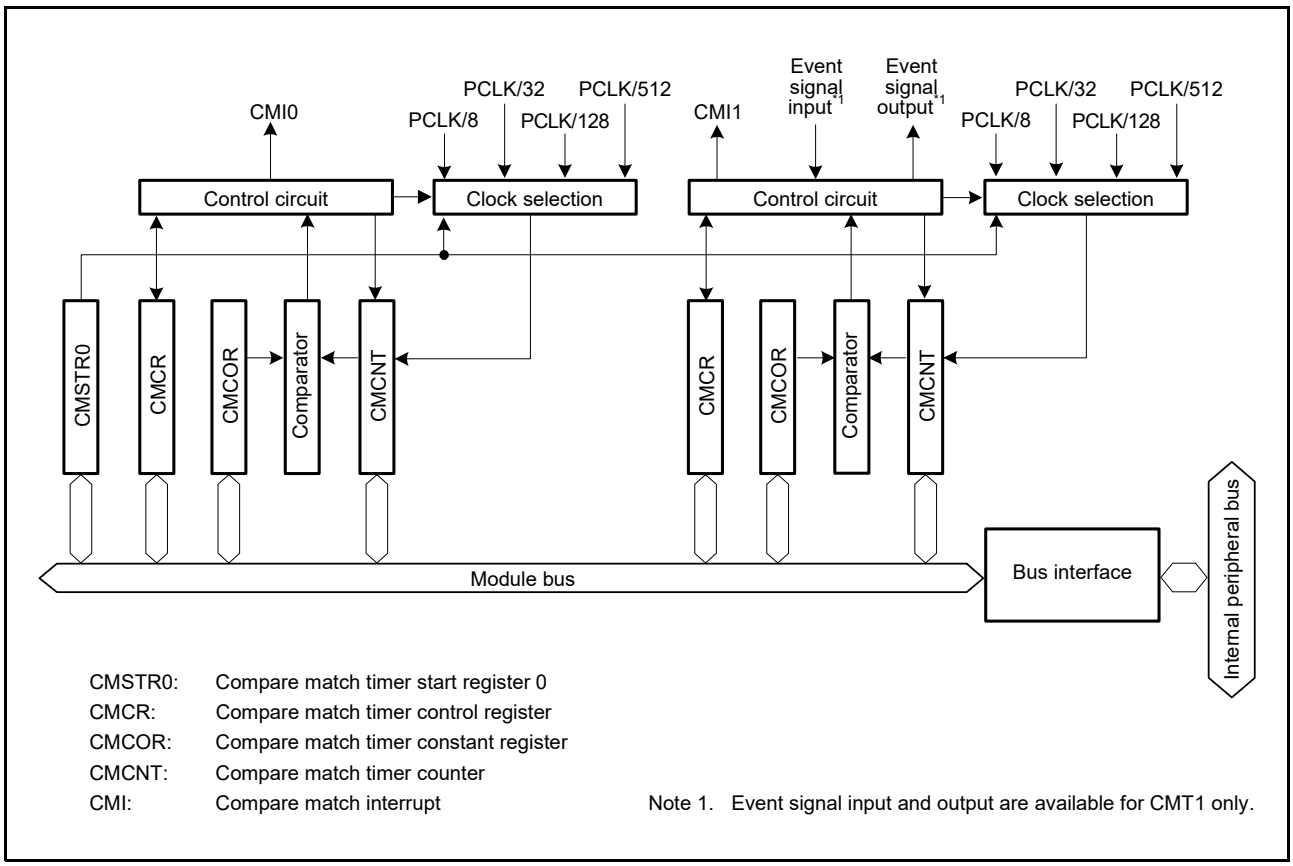


Figure 25.1 CMT (Unit 0) Block Diagram

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.

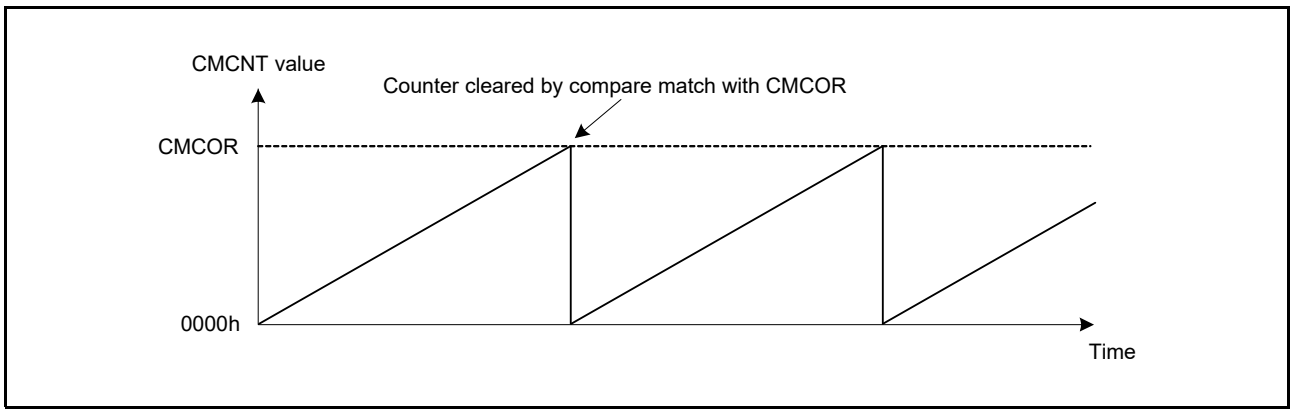


Figure 25.2 CMCNT Counter Operation

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.

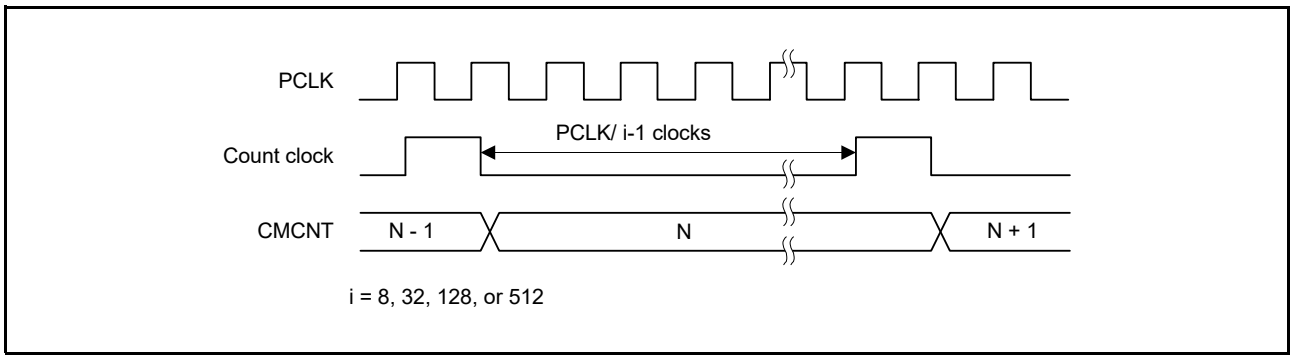


Figure 25.3 CMCNT Count Timing

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 (ICUb).

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.

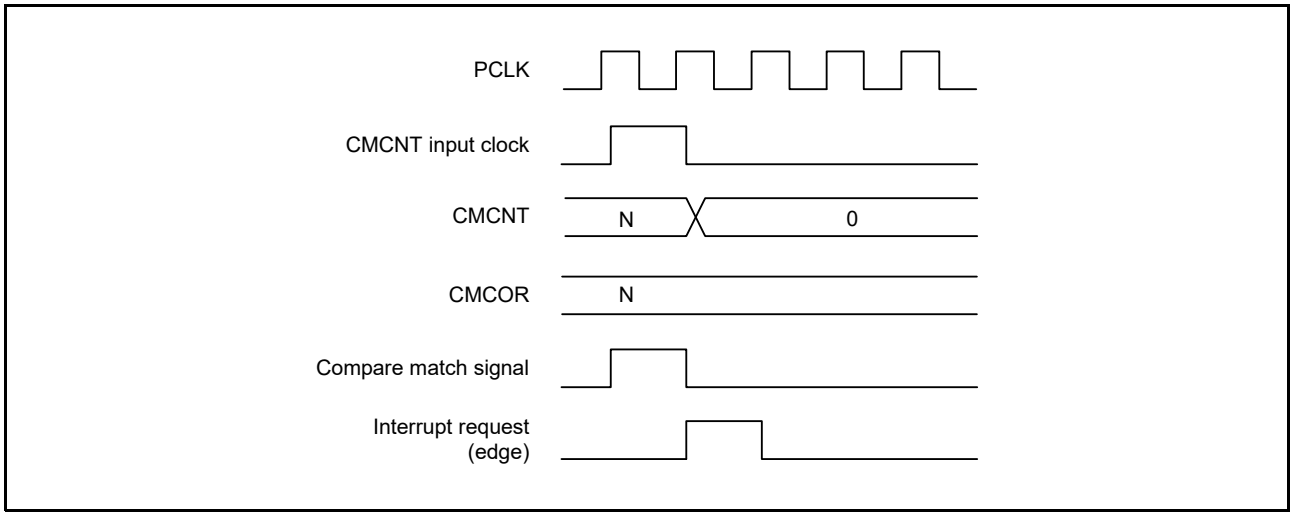


Figure 25.4 Timing of a Compare Match Interrupt

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.

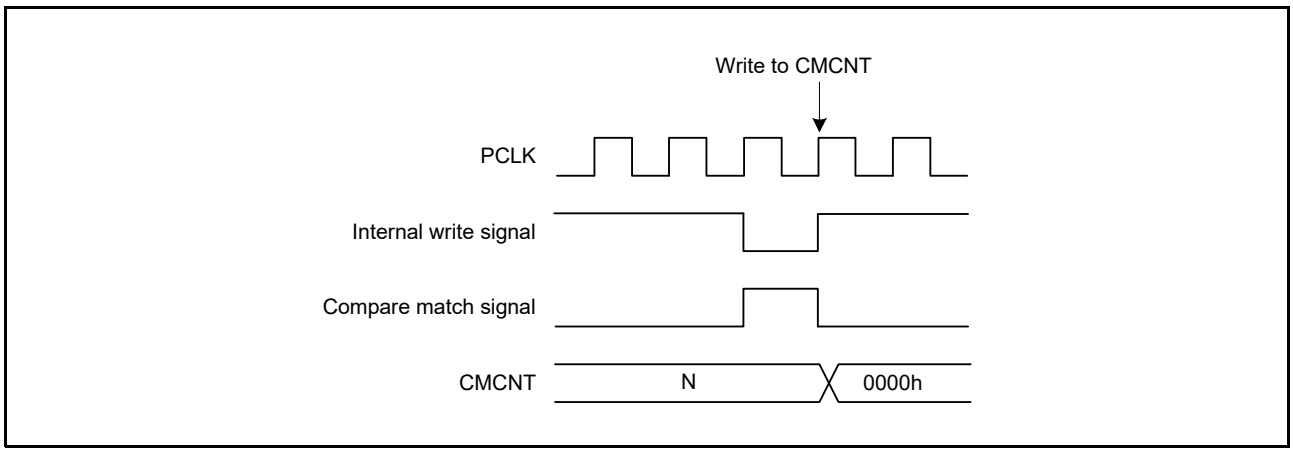


Figure 25.5 Conflict between CMCNT Counter Writing and Compare Match

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.

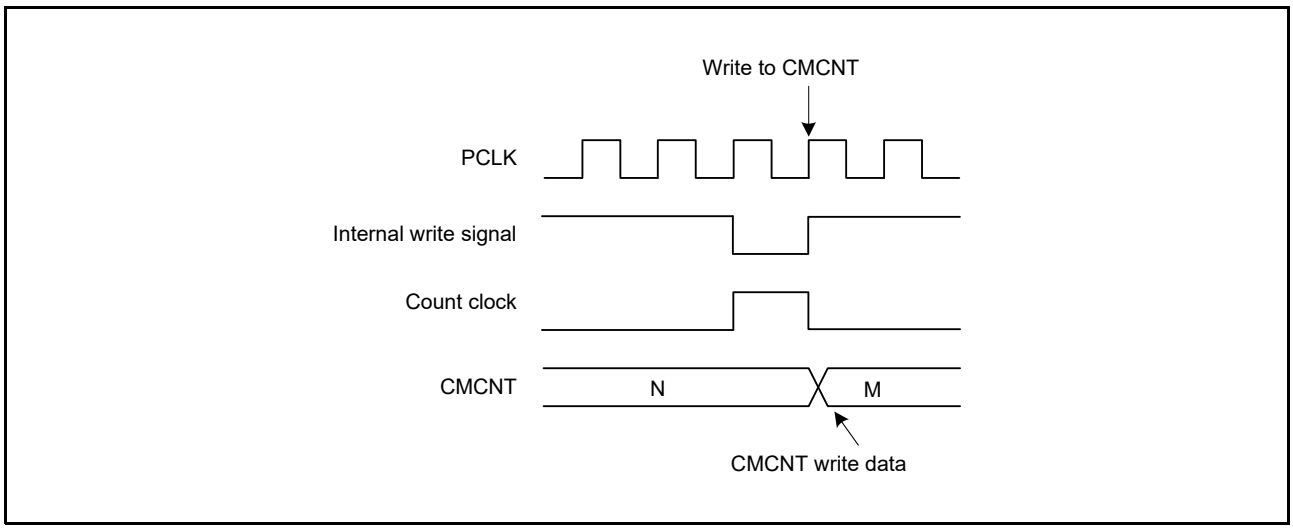


Figure 25.6 Conflict between CMCNT Counter Writing and Incrementing

26. Realtime Clock (RTCBa)

In this section, “PCLK” is used to refer to PCLKB.

26.1 Overview

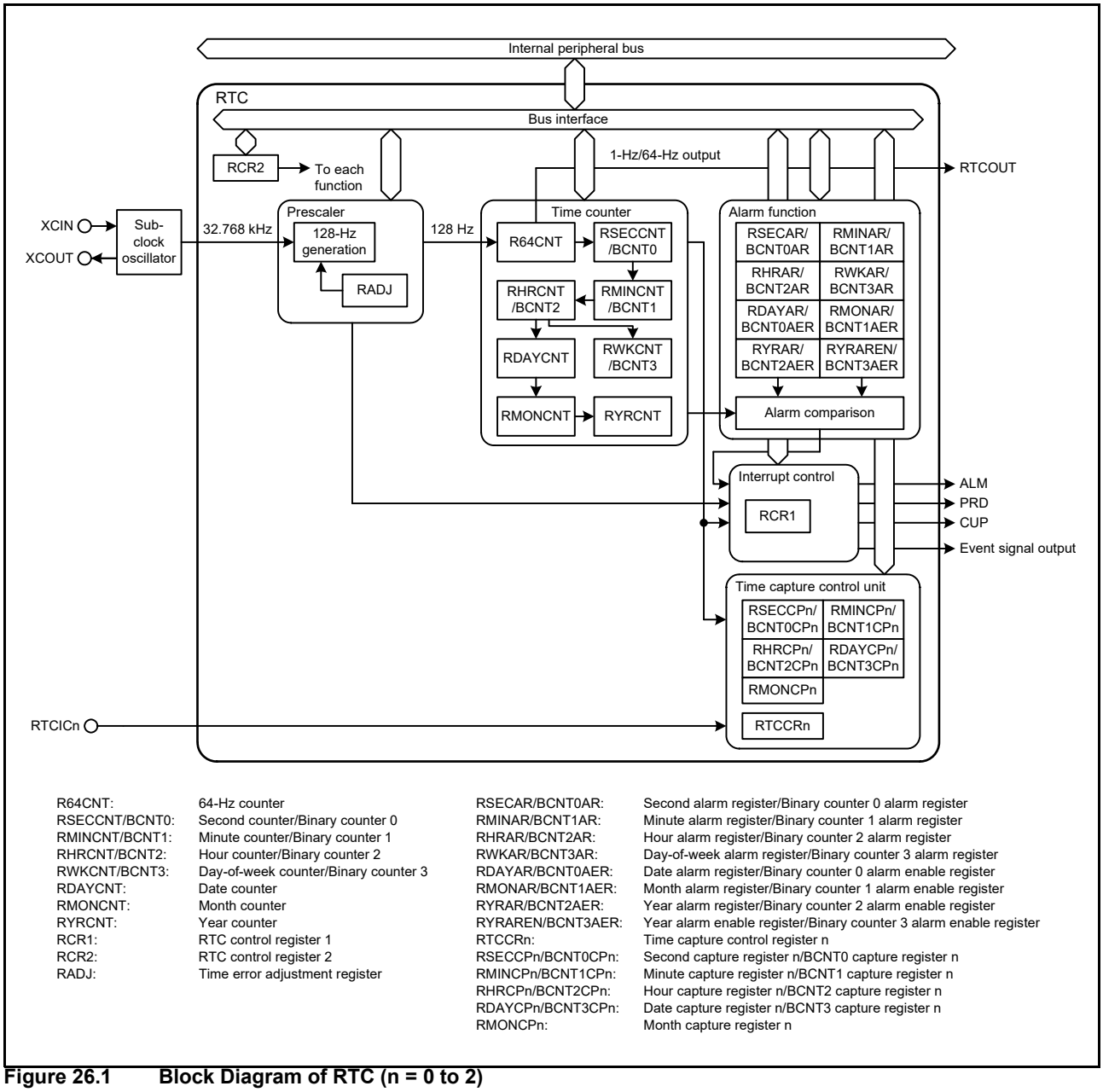
The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.
For binary count mode, the RTC does not count 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 26.1 lists the specifications of the RTC, Figure 26.1 shows a block diagram of the RTC, and Table 26.2 shows the pin configuration of the RTC.

Table 26.1 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 can be performed by an alarm interrupt or periodic interrupt
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or 32-bit binary counter value is captured.
Event link function	Periodic event output

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source.



R64CNT:	64-Hz counter	RSECAR/BCNT0AR:	Second alarm register/Binary counter 0 alarm register
RSECCNT/BCNT0:	Second counter/Binary counter 0	RMINAR/BCNT1AR:	Minute alarm register/Binary counter 1 alarm register
RMINCNT/BCNT1:	Minute counter/Binary counter 1	RHRAR/BCNT2AR:	Hour alarm register/Binary counter 2 alarm register
RHRCNT/BCNT2:	Hour counter/Binary counter 2	RWKAR/BCNT3AR:	Day-of-week alarm register/Binary counter 3 alarm register
RWKCNT/BCNT3:	Day-of-week counter/Binary counter 3	RDAYAR/BCNT0AER:	Date alarm register/Binary counter 0 alarm enable register
RDAYCNT:	Date counter	RMONAR/BCNT1AER:	Month alarm register/Binary counter 1 alarm enable register
RMONCNT:	Month counter	RYRAR/BCNT2AER:	Year alarm register/Binary counter 2 alarm enable register
RYRCNT:	Year counter	RYRAREN/BCNT3AER:	Year alarm enable register/Binary counter 3 alarm enable register
RCR1:	RTC control register 1	RTCCRn:	Time capture control register n
RCR2:	RTC control register 2	RSECCPn/BCNT0CPn:	Second capture register n/BCNT0 capture register n
RADJ:	Time error adjustment register	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 26.1 Block Diagram of RTC (n = 0 to 2)

Table 26.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	These pins are used to connect a 32.768-kHz crystal for the sub-clock.
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform.
RTCIC0	Input	These pins are used to input time capture events
RTCIC1	Input	
RTCIC2	Input	

26.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 26.6.5, 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 immediately after setting any of these registers. For details, refer to section 26.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

26.2.1 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 26.3.5, Reading 64-Hz Counter and Time.

26.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 0008 C402h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT0 0008 C402h

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

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 26.3.5, Reading 64-Hz Counter and Time.

26.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 0008 C404h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT1 0008 C404h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

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 26.3.5, Reading 64-Hz Counter and Time.

26.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 0008 C406h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT2 0008 C406h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

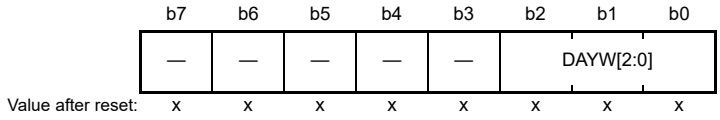
x: Undefined

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 26.3.5, Reading 64-Hz Counter and Time.

26.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): RTC.RWKCNT 0008 C408h



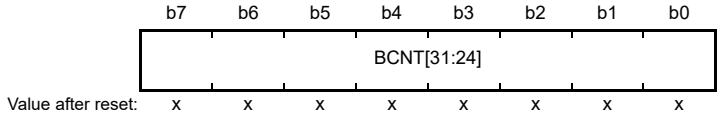
x: Undefined

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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT3 0008 C408h



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 26.3.5, Reading 64-Hz Counter and Time.

26.2.6 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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

26.2.7 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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

26.2.8 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 26.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

26.2.9 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 26.6.5, 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.

26.2.10 Minute Alarm Register (RMINAR)/ Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 0008 C412h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	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 26.6.5, 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

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

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.

26.2.11 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 26.6.5, 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.

26.2.12 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 26.6.5, 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.

26.2.13 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 26.6.5, 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.

26.2.14 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 26.6.5, 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.

26.2.15 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 26.6.5, 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.

26.2.16 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.

26.2.17 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.

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.

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 21.3.1, 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.

26.2.18 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 26.6.5, 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)

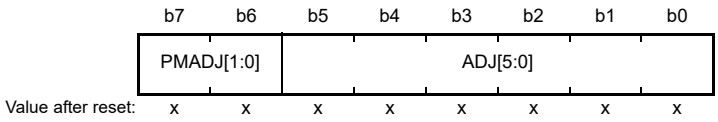
This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.
This bit is reserved in binary counter 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 26.3.1, Outline of Initial Settings of Registers after Power On.

26.2.19 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.

26.2.20 Time Capture Control Register n (RTCCRn) (n = 0 to 2)

Address(es): RTC.RTCCR0 0008 C440h, RTC.RTCCR1 0008 C442h, RTC.RTCCR2 0008 C444h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x *1	0	x	x	0	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected. 0 1: Rising edge is detected. 1 0: Falling edge is detected. 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status Flag	0: No event is detected. 1: An event is detected.	R/(W) *2
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. When a power-on reset occurs, this bit is set to 0.
Note 2. Writing 1 has no effect. This flag is cleared by writing 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 flag 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 RCR2.START and TCEN bits are 1.

TCST Flag (Time Capture Status Flag)

This flag indicates that an event of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) has been detected.
When the TCST flag is 0, no event is detected.
When the TCST flag is 1, this flag 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.
Writing 0 sets the TCST flag to 0. In addition, writing any other value except 0 has no effect.
Set the TCST flag while the TCCT[1:0] bits are 00b (no event is detected).
The TCST flag is set to 0 in synchronization with the count source. When the TCST flag is set to 0, check that the flag 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 TCEN bit to 1, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT[1:0] bits.

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 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.

26.2.21 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.

26.2.22 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

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

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

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPn[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

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.

26.2.23 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.

26.2.24 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.

26.2.25 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:	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 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.

26.3 Operation

26.3.1 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.

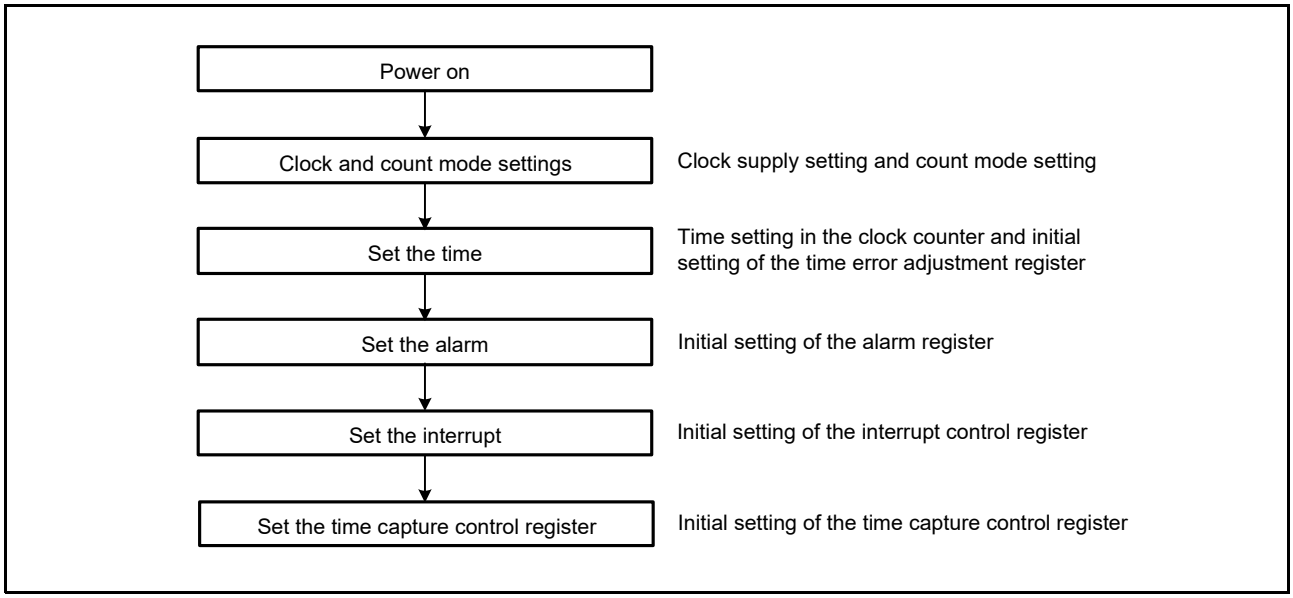
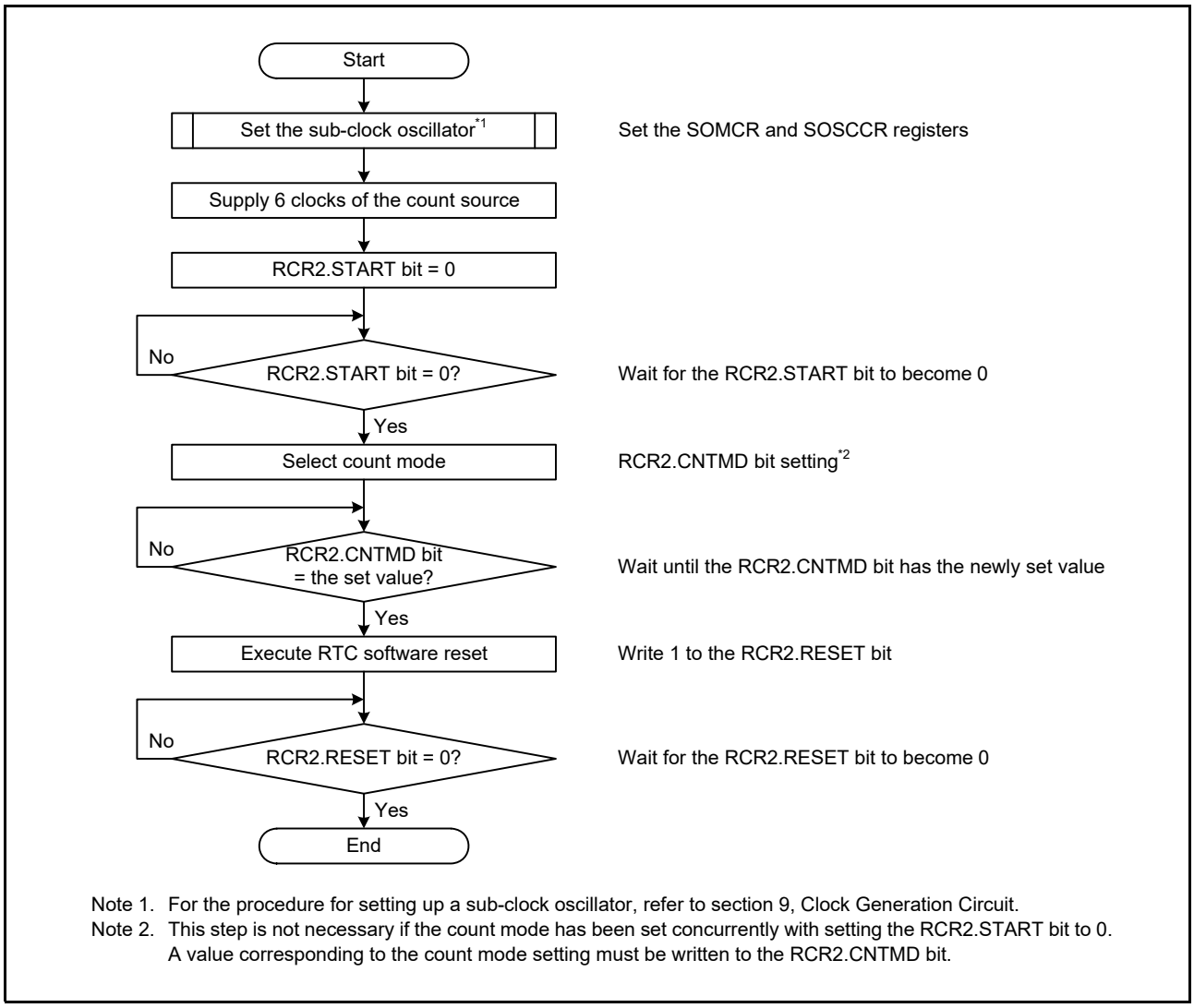


Figure 26.2 Outline of Initial Settings after Power On

26.3.2 Clock and Count Mode Setting Procedure

Figure 26.3 shows how to set the clock and the count mode.



Note 1. For the procedure for setting up a sub-clock oscillator, refer to section 9, Clock Generation Circuit.
Note 2. This step is not necessary if the count mode has been set concurrently with setting the RCR2.START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 26.3 Clock and Count Mode Setting Procedure

26.3.3 Setting the Time

Figure 26.4 shows how to set the time.

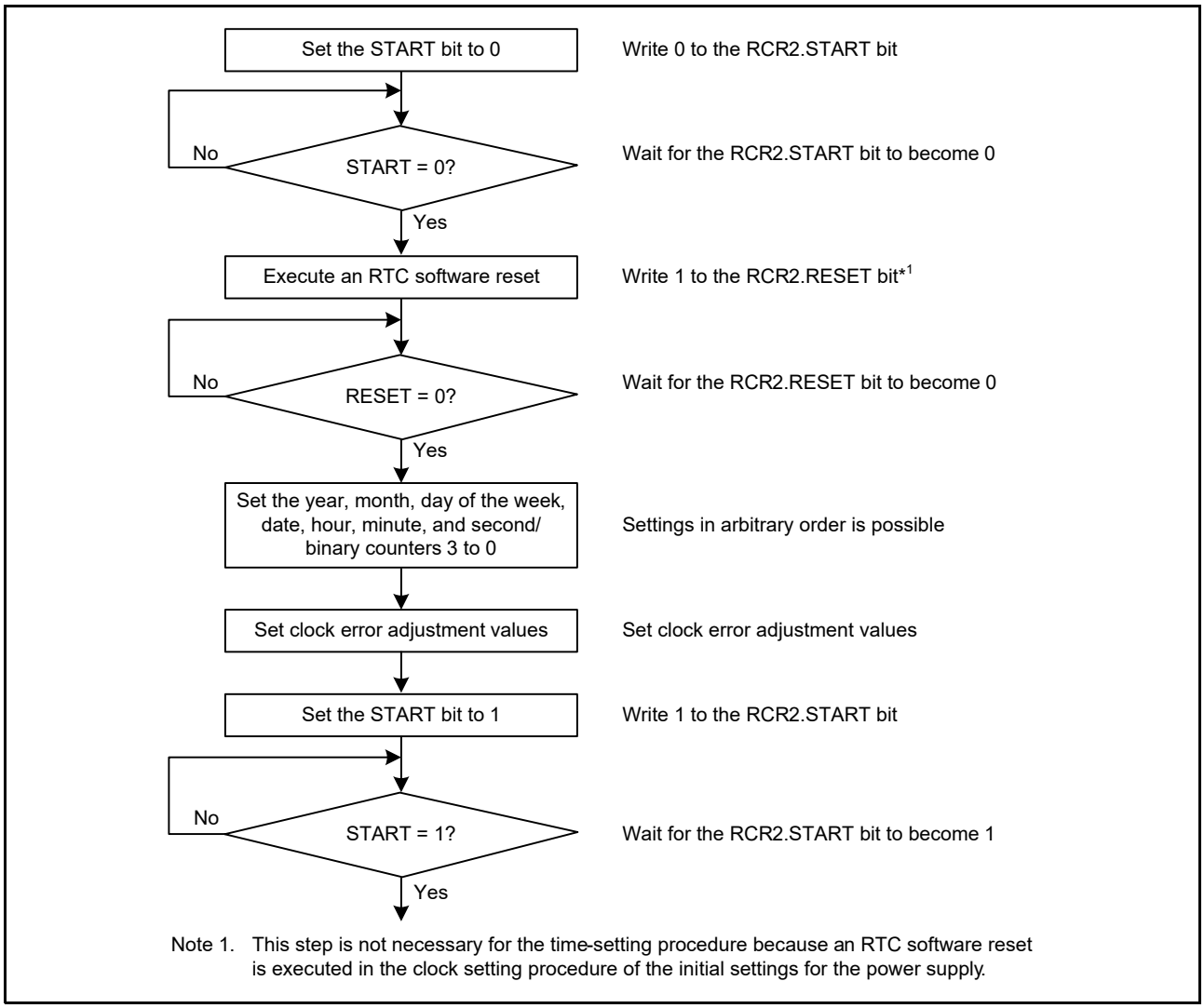


Figure 26.4 Setting the Time

26.3.4 30-Second Adjustment

Figure 26.5 shows how to execute 30-second adjustment.

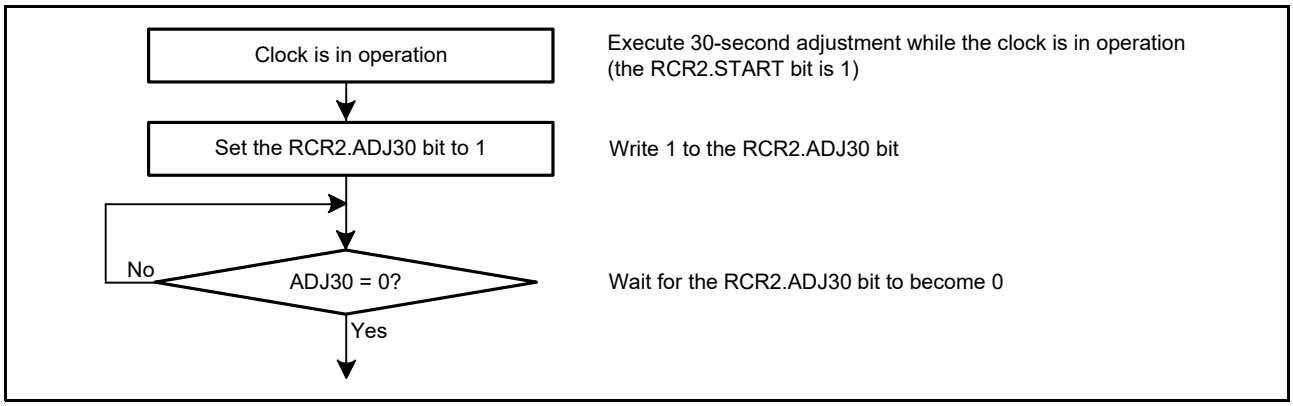


Figure 26.5 30-Second Adjustment

26.3.5 Reading 64-Hz Counter and Time

Figure 26.6 shows how to read the 64-Hz counter and time.

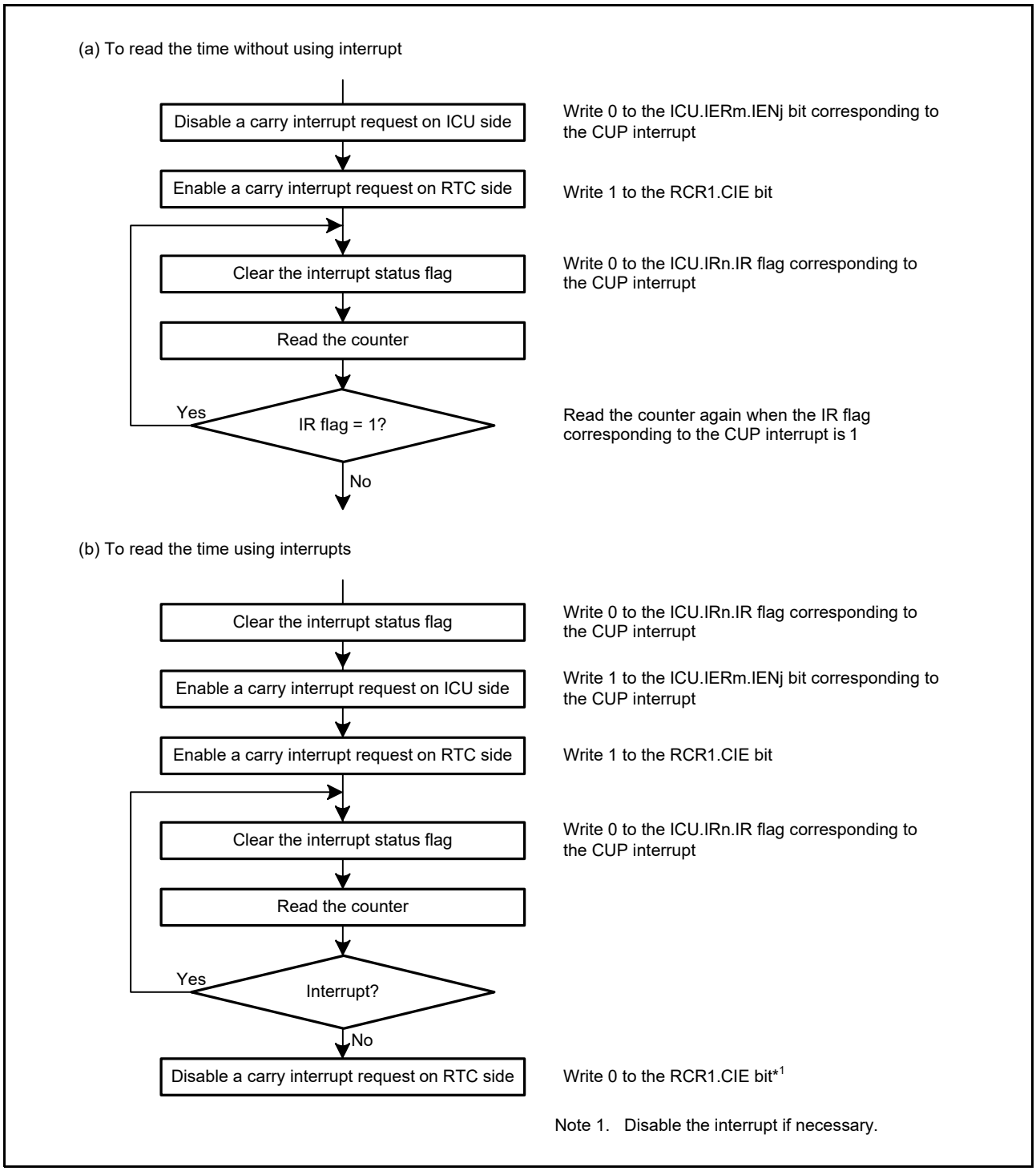


Figure 26.6 Reading Time

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 26.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

26.3.6 Alarm Function

Figure 26.7 shows how to use the alarm function.

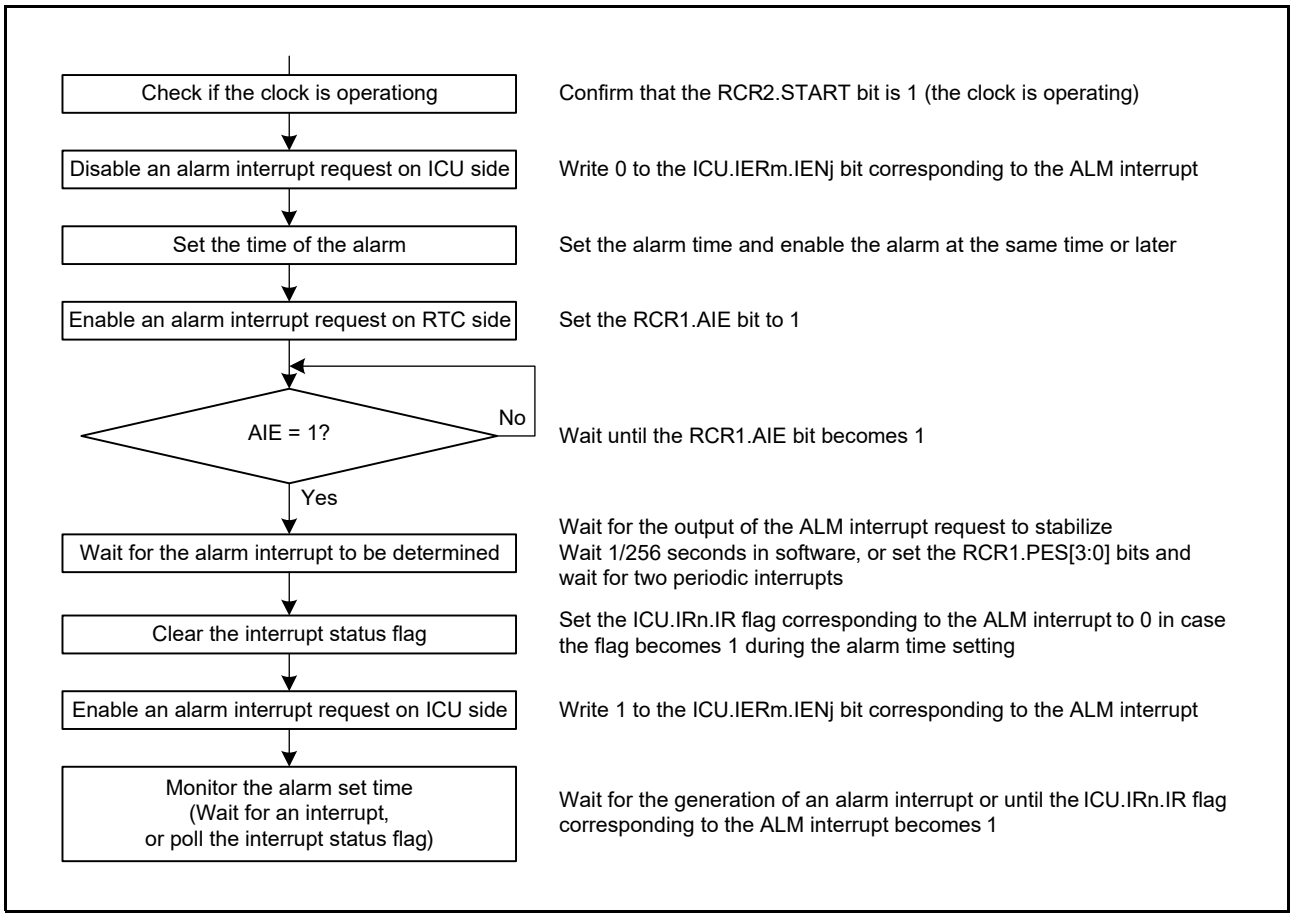


Figure 26.7 Using Alarm Function

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.

26.3.7 Procedure for Disabling Alarm Interrupt

Figure 26.8 shows the procedure for disabling the enabled alarm interrupt request.

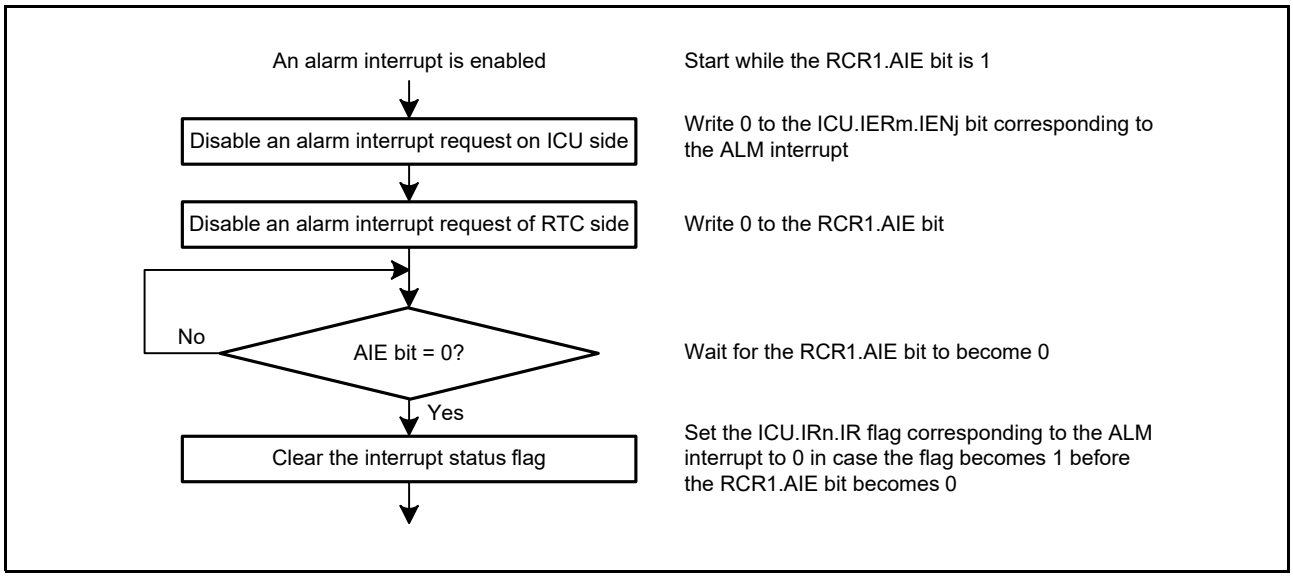


Figure 26.8 Procedure for Disabling Alarm Interrupt Request

26.3.8 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.

26.3.8.1 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)

26.3.8.2 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.

26.3.8.3 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.

26.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

26.3.9 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 flag 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 26.9 and operation when the noise filter is on is shown in Figure 26.10.

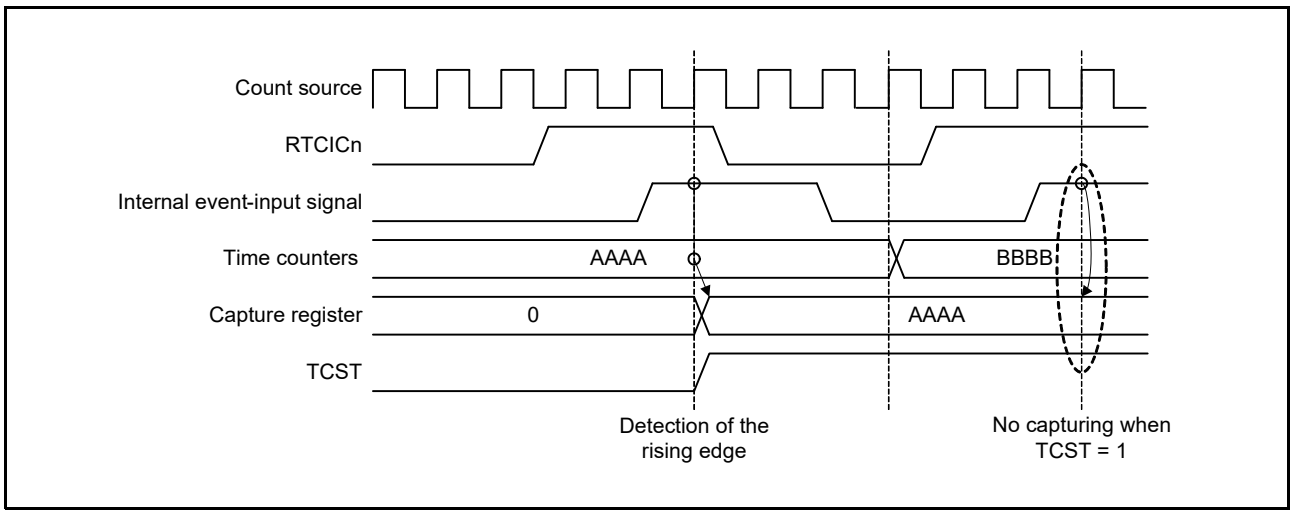


Figure 26.9 Timing of a Time Capture Operation (with the Filter Off) (n = 0 to 2)

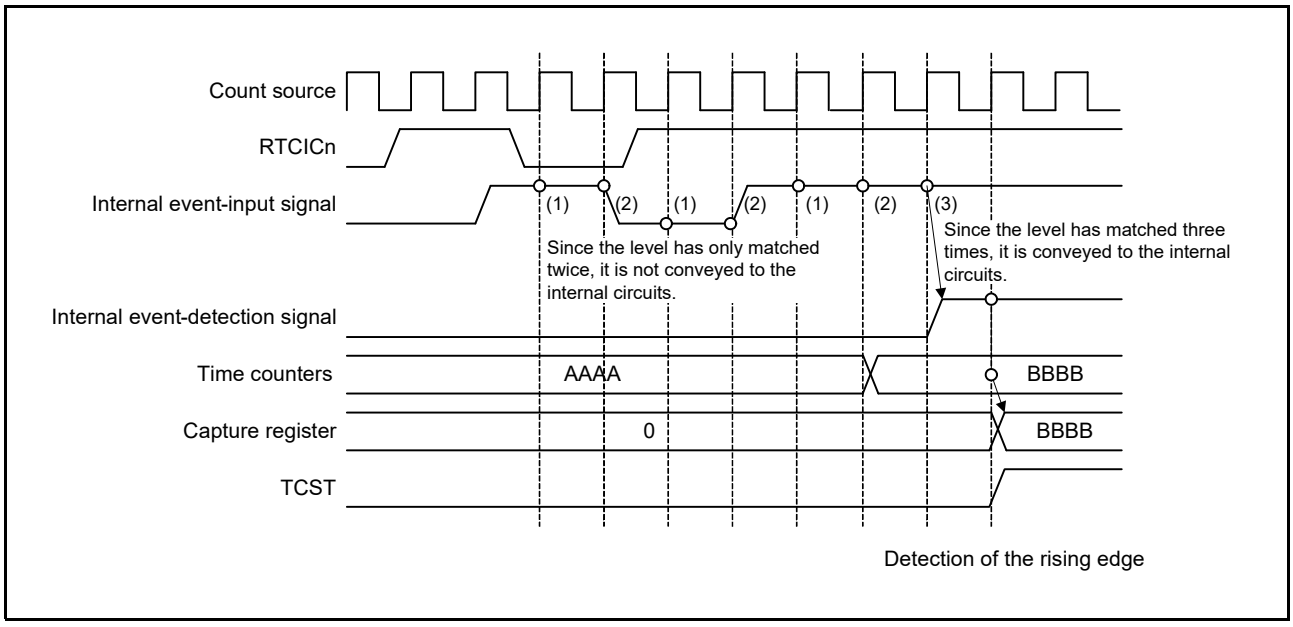


Figure 26.10 Timing of a Time Capture Operation (with the Filter On) (n = 0 to 2)

26.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 26.3 lists interrupt sources for the RTC.

Table 26.3 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 26.3.6, 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.

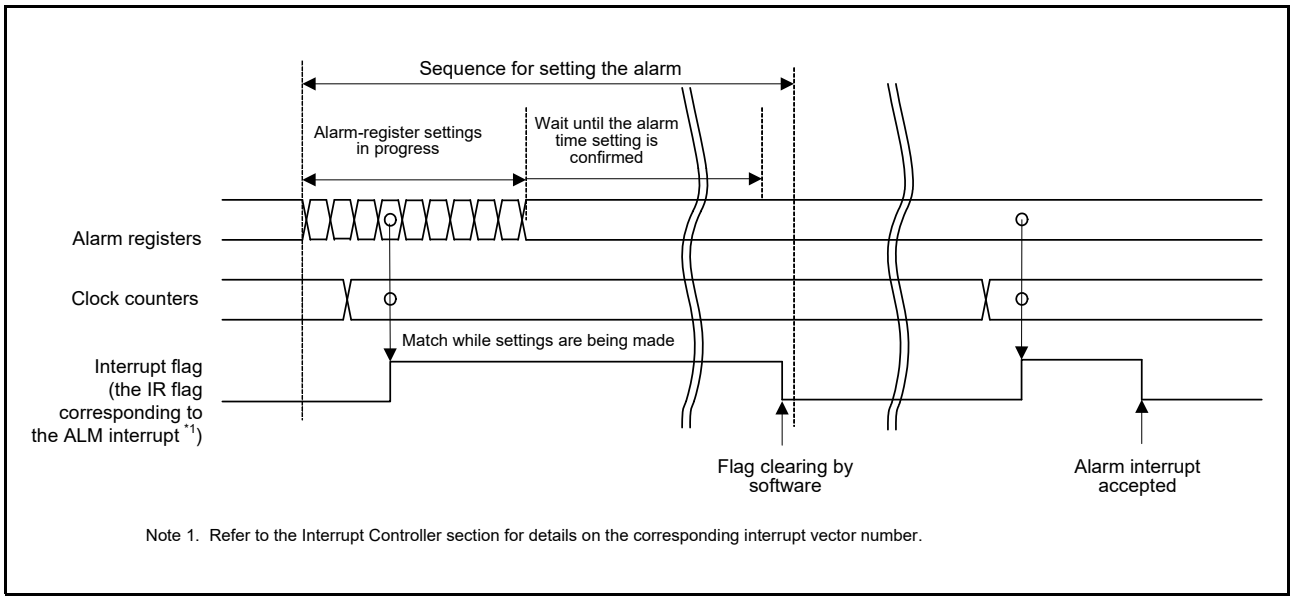


Figure 26.11 Timing Chart for the Alarm Interrupt (ALM)

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

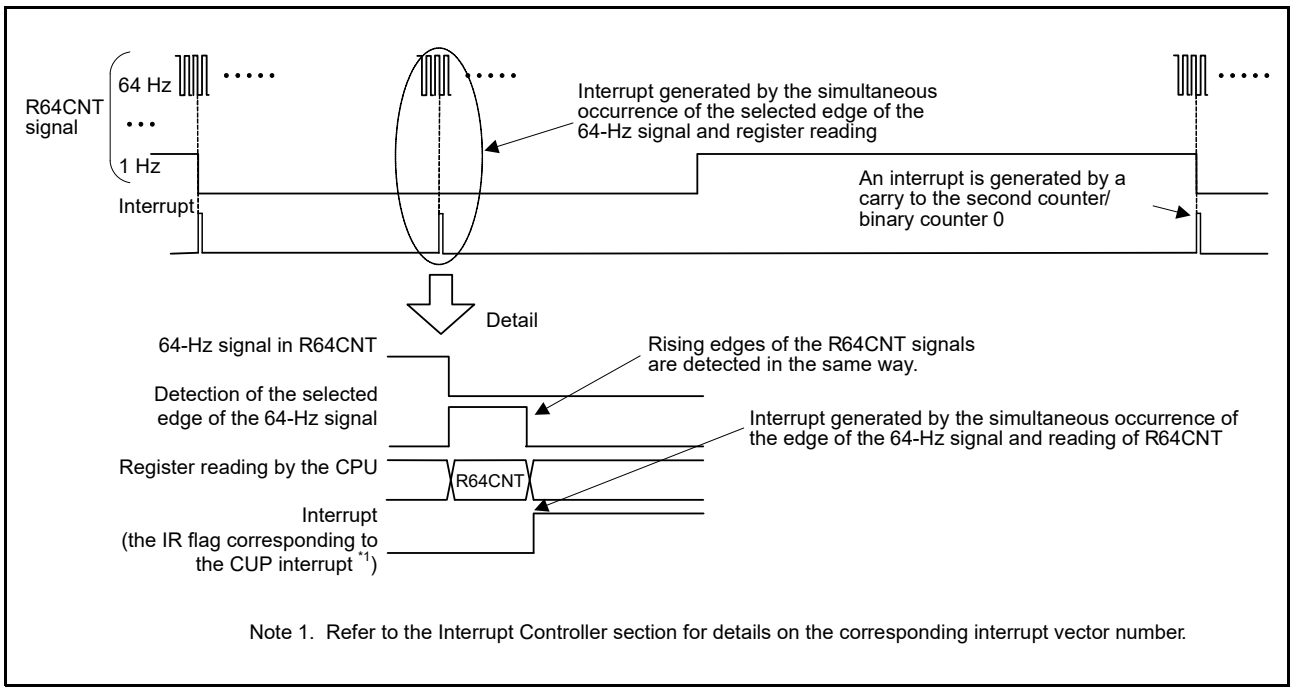


Figure 26.12 Carry Interrupt (CUP) Timing Chart

26.5 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.

26.5.1 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, the periodic event signals for the ELC are not output.

26.6 Usage Notes

26.6.1 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.

26.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 26.13.
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.

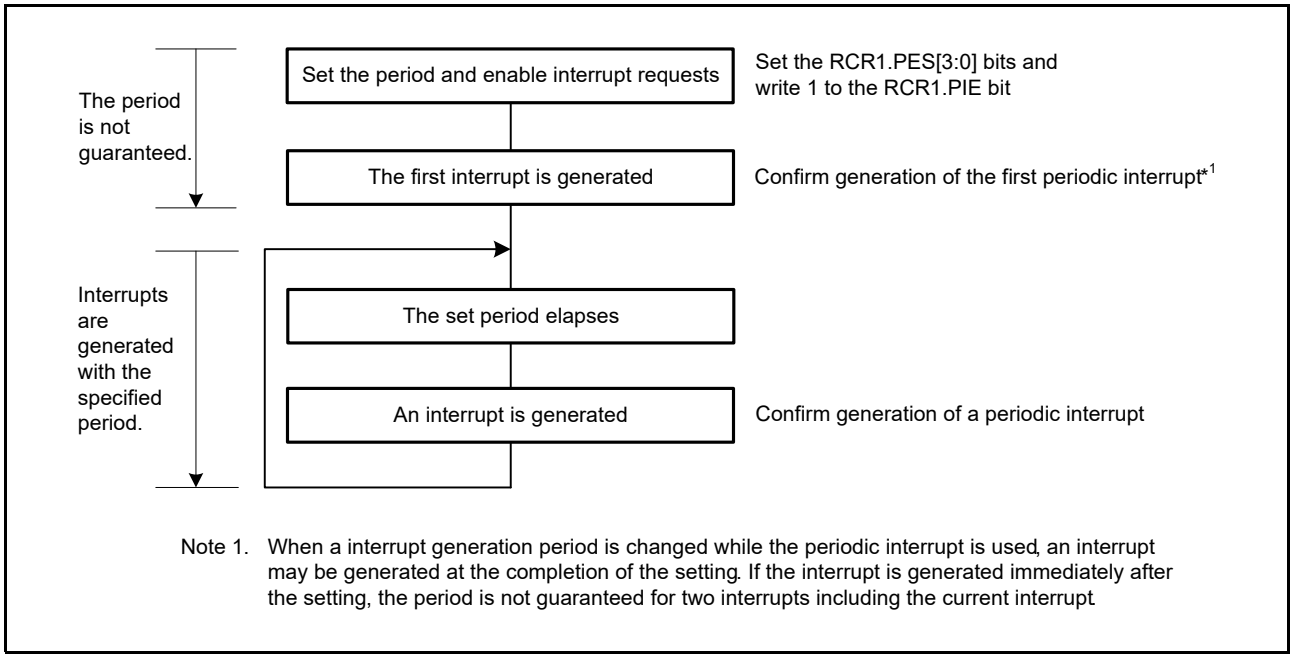


Figure 26.13 Using Periodic Interrupt Function

26.6.3 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.

26.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (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.

26.6.5 Notes on Writing to and Reading from Registers

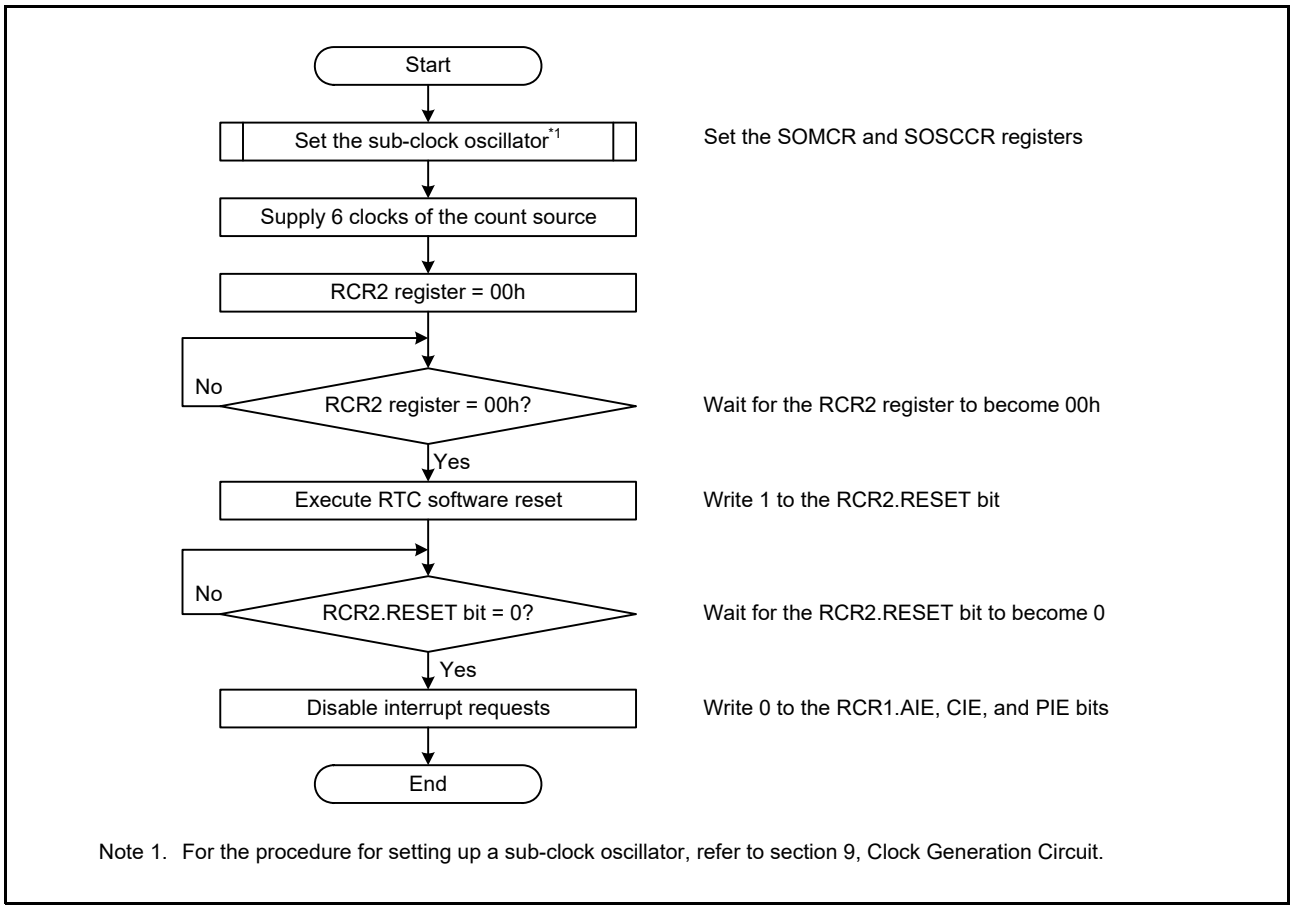
- When reading a counter register such as the second counter/binary counter 0 after having written to the counter register, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, or bits RCR2.AADJE, AADJP, and HR24 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 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.

26.6.6 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 26.3.1, Outline of Initial Settings of Registers after Power On.

26.6.7 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 26.14.



Note 1. For the procedure for setting up a sub-clock oscillator, refer to section 9, Clock Generation Circuit.

Figure 26.14 Initialization Procedure

27. Low-Power Timer (LPTa)

27.1 Overview

This MCU integrates a low-power timer (LPT) that consists of a single-channel 16-bit timer. The LPT uses a sub-clock, LOCO clock, or IWDT-dedicated clock as the clock source, and can continue counting operation even in software standby mode. A compare match signal can be used to return from software standby mode to normal operating mode. The LPT channel 0 can generate a PWM waveform.

Table 27.1 lists the specifications of the LPT and Figure 27.1 shows a block diagram of the LPT.

Table 27.1 LPT Specifications

Item	Description
Clock source	Sub-clock, LOCO clock divided by 4, or IWDT-dedicated clock
Clock division ratio	Divided by 1, 2, 4, 8, 16, or 32
Counting operation	• Count up using the 16-bit up-counter • Counting operation can be continued even in software standby mode
Compare match	Compare match 0 (a compare match signal is generated only in software standby mode) Compare match 1
PWM waveform generation	PWM waveform can be output from the LPTO pin.
Interrupt	Compare match 1
Event link function (output)	Compare match 0 (a compare match signal is generated only in software standby mode) Compare match 1

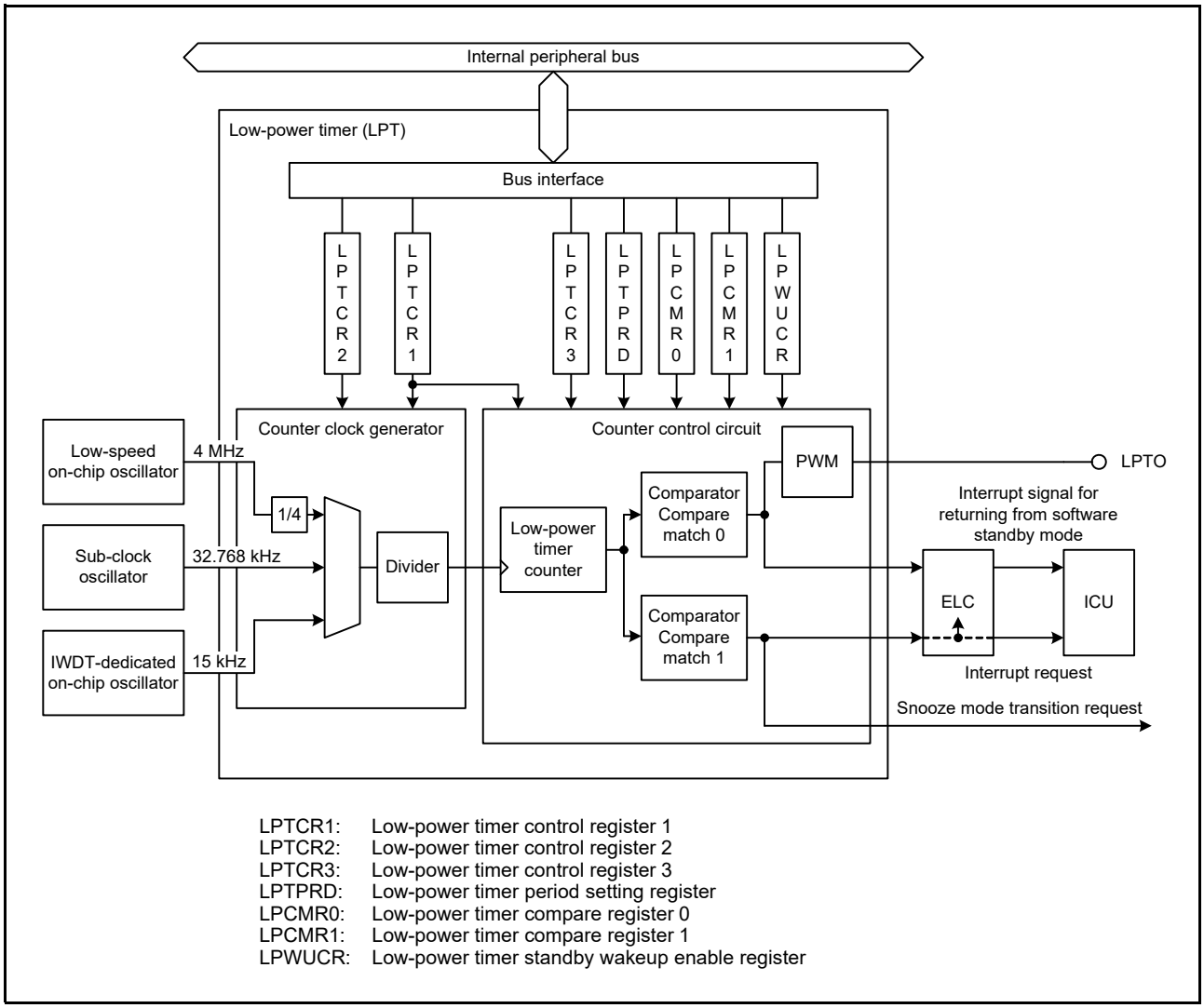


Figure 27.1 LPT Block Diagram

Table 27.2 LPT Pin Configuration

Pin Name	I/O	Function
LPTO	Output	PWM waveform output pin

27.2 Register Descriptions

27.2.1 Low-Power Timer Control Register 1 (LPTCR1)

Address(es): LPT.LPTCR1 0008 00B0h

	b7	b6	b5	b4	b3	b2	b1	b0
	LPCMRE1	LPCMRE0	—	LPCNTCKSEL	LPCNTCKSEL2	LPCNTPSSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	LPCNTPSSEL[2:0]	Clock Division Ratio Select*1	b2 b0 0 0 0: Divided by 1 (no division) 0 0 1: Divided by 2 0 1 0: Divided by 4 0 1 1: Divided by 8 1 0 0: Divided by 16 1 0 1: Divided by 32 Settings other than above are prohibited.	R/W
b3	LPCNTCKSEL2	Clock Source Select 2*1, *2	b4 b3 0 0: Sub-clock 0 1: LOCO clock divided by 4*3	R/W
b4	LPCNTCKSEL	Clock Source Select*1, *2	1 0: IWDT-dedicated clock (IWDTCLK)*4 1 1: LOCO clock divided by 4*3	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	LPCMRE0	Compare Match 0 Enable*5	0: Compare match 0 is disabled 1: Compare match 0 is enabled	R/W
b7	LPCMRE1	Compare Match 1 Enable*5	0: Compare match 1 is disabled 1: Compare match 1 is enabled	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note 1. Rewrite these bits while the LPTCR2.LPCNTSTP bit is 1 (supply of clock to the low-power timer is stopped).
Note 2. Satisfy that the frequency of the system clock (ICLK) and peripheral module clock (PCLKB) ≥ 4 × (the frequency of the clock source).
Note 3. A clock (LOCO clock) generated by the low-speed on-chip oscillator (LOCO) divided by 4 is supplied to the low-power timer. When the LOCO clock is used as the clock source for the low-power timer and the oscillation is to be continued even in software standby mode, set the LOFCR.LOFXIN bit to 1.
Note 4. A clock generated by the IWDT-dedicated on-chip oscillator (IWDTCLK) is supplied to the low-power timer. When modifying this bit, make sure that the IWDT-dedicated on-chip oscillator is oscillating stably.
When the IWDTCLK is used as the clock source for the low-power timer, set the OFS0.IWDTSLCSTP bit to 0 (counting stop is disabled) in IWDT auto-start mode operation, and set the IWDTCSTPR.SLCSTP bit to 0 (count stop is disabled) in other modes. Without this setting, the IWDT-dedicated on-chip oscillator is stopped in software standby mode.
Note 5. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTCR1 register is used to control the low-power timer.

LPCNTCKSEL2 Bit (Clock Source Select 2)

This bit is used to select the LOCO clock as the clock source for the low-power timer.

LPCNTCKSEL Bit (Clock Source Select)

This bit is used to select the sub-clock or IWDT-dedicated clock as the clock source for the low-power timer. This bit is effective when the LPCNTCKSEL2 bit is 0.

LPCMRE0 Bit (Compare Match 0 Enable)

This bit enables or disables low-power timer compare match 0.
When the low-power timer is put into operation and the MCU makes a transition to software standby mode while this bit and the LPWUCR.LPWKUPEN bit are set to 1 (wakeup from software standby mode using low-power timer is enabled), the MCU returns from software standby mode to normal operating mode through the event link controller (ELC) when the value of the low-power timer counter matches the value of the LPCMR0 register.
Settings for the interrupt and ELC are necessary to use a compare match 0 as a trigger source to return from software standby mode.
Refer to section 19, Event Link Controller (ELC) for details on the ELC settings, and refer to section 14, Interrupt Controller (ICUb) for details on the interrupt settings.

An interrupt request at compare match 0 is generated only in software standby mode. It is not generated in normal operating mode, sleep mode, and deep sleep mode.

LPCMRE1 Bit (Compare Match 1 Enable)

This bit enables or disables low-power timer compare match 1.
When the value of the low-power timer counter matches the value of the LPCMR1 register while this bit is set to 1, an event signal is output to the event link controller (ELC) and the same signal is sent as an interrupt request signal to the interrupt controller (ICU) via ELC. Therefore, it is necessary to release the ELC module stop to generate an interrupt.

27.2.2 Low-Power Timer Control Register 2 (LPTCR2)

Address(es): LPT.LPTCR2 0008 00B1h

	b7	b6	b5	b4	b3	b2	b1	b0
	PWME	OLVL	OPOL	—	—	—	—	LPCNTSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	LPCNTSTP	Clock Supply Control	0: Clock is supplied to the low-power timer. 1: Supply of clock to the low-power timer is stopped.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	OPOL	Output Polarity Select*1	0: Initial output is low, and the subsequent output is as the OLVL bit setting. 1: Initial output is high, and the subsequent output is the inverse of the OLVL bit setting.	R/W
b6	OLVL	Output Level Select*1	0: High is output at compare match 0 and low is output when counter is cleared. 1: Low is output at compare match 0 and high is output when counter is cleared.	R/W
b7	PWME	PWM Mode Enable*1	0: PWM mode is disabled. 1: PWM mode is enabled.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTCR2 register is used to control supply of the clock to be used for the low-power timer and settings of the PWM mode.

LPCNTSTP Bit (Clock Supply Control)

This bit is used to supply or stop the clock to be used for the low-power timer. When this bit is set to 0, the clock signal is supplied to the low-power timer counter and divider.

OPOL Bit (Output Polarity Select)

This bit is used to select the polarity of the output waveform in the PWM mode.

OLVL Bit (Output Level Select)

This bit is used to select the level of the output waveform at compare match 0.
The following waveforms can be generated in combination with the OPOL bit.

Table 27.3 Combination of the OPOL and OLVL Bits and the Level of the Output Waveform

OPOL bit	OLVL bit	Initial Level	Level at Compare Match 0	Level at Counter Clear
0	0	Low	High	Low
0	1	Low	Low	High
1	0	High	Low	High
1	1	High	High	Low

PWME Bit (PWM Mode Enable)

This bit is used to control enable or disable of the PWM mode. When this bit is set to 0, the output from the LPTO pin is fixed to the initial level. When 1, PWM waveform can be output from the LPTO pin.

27.2.3 Low-Power Timer Control Register 3 (LPTCR3)

Address(es): LPT.LPTCR3 0008 00B2h

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

Bit	Symbol	Bit Name	Description	R/W
b0	LPCNTEN	Low-Power Timer Counter Operation Control	0: Low-power timer counter stops 1: Low-power timer counter operates	R/W
b1	LPCNTRST	Low-Power Timer Counter Clear*1, *2	• When writing 0: Has no effect 1: Clears divider and counter • When reading 0: Clearing is completed 1: Clearing is in progress	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: Rewrite this register while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer).
Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).
Note 2. When clearing the low-power timer counter successively, confirm that the LPCNTRST bit becomes 0, wait for at least one cycle of the clock selected by the LPTCR1.LPCNTCKSEL and LPCNTCKSEL2 bits, and then write 1 to the LPCNTRST bit again.

The LPTCR3 register controls operations of and clears the low-power timer counter and divider.

LPCNTEN Bit (Low-Power Timer Counter Operation Control)

This bit is used to operate or stop the low-power timer counter and divider.
When this bit is set to 1 while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer), the low-power timer counter and divider start operating.
Do not write 1 to the LPCNTRST bit while this bit is 1.

LPCNTRST Bit (Low-Power Timer Counter Clear)

This bit is used to clear the low-power timer counter and divider.
When this bit is set to 1 while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer), the low-power timer counter and divider are cleared in synchronization with the clock used for the low-power timer. Once clearing is complete, this bit automatically becomes 0.
When 1 is written to this bit, confirm that its value becomes 0 before executing the next processing.

27.2.4 Low-Power Timer Period Setting Register (LPTPRD)

Address(es): LPT.LPTPRD 0008 00B4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LPCNTPRD[15:0]															
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
b15 to b0	LPCNTPRD[15:0]	Low-Power Timer Period Setting	Set the period of the low-power timer. Setting range: 0001h to FFFFh	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: Rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTPRD register is used to set the period of the low-power timer.
The period of the low-power timer is proportional to “LPTPRD + 1” and calculated by the following formula:

Period of low-power timer = period of clock source × division ratio × (LPTPRD + 1)

When the value of the low-power timer counter matches the set value, the counter is cleared to 0000h and continues counting.
Do not set this register to 0000h.

Table 27.4, Table 27.5, and Table 27.6 lists examples of setting the periods of the low-power timer. These examples show values most approximate to the target periods.

Table 27.4 Example of Low-Power Timer Period Settings for IWDTCLK

Division ratio	Divided by 1			Divided by 2			Divided by 4		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms	Error (%)
1	000Eh	1.00	0.00	0006h	0.93	–6.67	0003h	1.07	6.67
2	001Dh	2.00	0.00	000Dh	1.87	–6.67	0006h	1.87	–6.67
5	004Ah	5.00	0.00	0024h	4.93	–1.33	0011h	4.80	–4.00
10	0095h	10.00	0.00	004Ah	10.00	0.00	0024h	9.87	–1.33
20	012Bh	20.00	0.00	0095h	20.00	0.00	004Ah	20.00	0.00
50	02EDh	50.00	0.00	0176h	50.00	0.00	00BAh	49.87	–0.27
100	05DBh	100.00	0.00	02EDh	100.00	0.00	0176h	100.00	0.00
200	0BB7h	200.00	0.00	05DBh	200.00	0.00	02EDh	200.00	0.00
500	1D4Bh	500.00	0.00	0EA4h	499.87	–0.03	0751h	499.73	–0.05
1000	3A97h	1000.00	0.00	1D4Ah	999.87	–0.01	0EA4h	999.73	–0.03
2000	752Fh	2000.00	0.00	3A96h	1999.87	–0.01	1D4Ah	1999.73	–0.01
5000	—	—	—	927Bh	5000.00	0.00	493Dh	5000.00	0.00
10000	—	—	—	—	—	—	—	—	—
20000	—	—	—	—	—	—	—	—	—
50000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 8			Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	0001h	1.07	6.67	—	—	—	—	—	—
2	0003h	2.13	6.67	0001h	2.13	6.67	—	—	—
5	0008h	4.80	–4.00	0004h	5.33	6.67	0001h	4.27	–14.67
10	0011h	9.60	–4.00	0008h	9.60	–4.00	0004h	10.67	6.67
20	0024h	19.73	–1.33	0011h	19.20	–4.00	0008h	19.20	–4.00
50	005Ch	49.60	–0.80	002Dh	49.07	–1.87	0016h	49.07	–1.87
100	00BAh	99.73	–0.27	005Ch	99.20	–0.80	002Dh	98.13	–1.87
200	0176h	200.00	0.00	00BAh	199.47	–0.27	005Ch	198.40	–0.80
500	03A8h	499.73	–0.05	01D3h	499.20	–0.16	00E9h	499.20	–0.16
1000	0751h	999.47	–0.05	03A8h	999.47	–0.05	01D3h	998.40	–0.16
2000	0EA4h	1999.47	–0.03	0751h	1998.93	–0.05	03A8h	1998.93	–0.05
5000	249Eh	5000.00	0.00	124Eh	4999.47	–0.01	0926h	4998.40	–0.03
10000	493Dh	10000.00	0.00	249Eh	10000.00	0.00	124Eh	9998.93	–0.01
20000	927Bh	20000.00	0.00	493Dh	20000.00	0.00	249Eh	20000.00	0.00
50000	—	—	—	B71Ah	50000.00	0.00	5B8Ch	49998.93	0.00

Table 27.5 Example of Low-Power Timer Period Settings for Sub-Clock

Division ratio	Divided by 1			Divided by 2			Divided by 4		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms	Error (%)
1	0020h	1.01	0.71	000Fh	0.98	–2.34	0007h	0.98	–2.34
2	0041h	2.01	0.71	001Fh	1.95	–2.34	000Fh	1.95	–2.34
5	00A3h	5.00	0.10	0050h	4.94	–1.12	0027h	4.88	–2.34
10	0147h	10.01	0.10	00A2h	9.95	–0.51	0050h	9.89	–1.12
20	028Eh	19.99	–0.05	0146h	19.96	–0.21	00A2h	19.90	–0.51
50	0665h	49.99	–0.02	0332h	49.99	–0.02	0198h	49.93	–0.15
100	0CCCh	100.01	0.01	0665h	99.98	–0.02	0332h	99.98	–0.02
200	1999h	200.01	0.01	0CCBh	199.95	–0.02	0665h	199.95	–0.02
500	3FFFh	500.00	0.00	1FFFh	500.00	0.00	0FFFh	500.00	0.00
1000	7FFFh	1000.00	0.00	3FFFh	1000.00	0.00	1FFFh	1000.00	0.00
2000	FFFFh	2000.00	0.00	7FFFh	2000.00	0.00	3FFFh	2000.00	0.00
5000	—	—	—	—	—	—	9FFFh	5000.00	0.00
10000	—	—	—	—	—	—	—	—	—
20000	—	—	—	—	—	—	—	—	—
50000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 8			Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	0003h	0.98	–2.34	0001h	0.98	–2.34	—	—	—
2	0007h	1.95	–2.34	0003h	1.95	–2.34	0001h	1.95	–2.34
5	0013h	4.88	–2.34	0009h	4.88	–2.34	0004h	4.88	–2.34
10	0027h	9.77	–2.34	0013h	9.77	–2.34	0009h	9.77	–2.34
20	0050h	19.78	–1.12	0027h	19.53	–2.34	0013h	19.53	–2.34
50	00CBh	49.80	–0.39	0065h	49.80	–0.39	0032h	49.80	–0.39
100	0198h	99.85	–0.15	00CBh	99.61	–0.39	0065h	99.61	–0.39
200	0332h	199.95	–0.02	0198h	199.71	–0.15	00CBh	199.22	–0.39
500	07FFh	500.00	0.00	03FFh	500.00	0.00	01FFh	500.00	0.00
1000	0FFFh	1000.00	0.00	07FFh	1000.00	0.00	03FFh	1000.00	0.00
2000	1FFFh	2000.00	0.00	0FFFh	2000.00	0.00	07FFh	2000.00	0.00
5000	4FFFh	5000.00	0.00	27FFh	5000.00	0.00	13FFh	5000.00	0.00
10000	9FFFh	10000.00	0.00	4FFFh	10000.00	0.00	27FFh	10000.00	0.00
20000	—	—	—	9FFFh	20000.00	0.00	4FFFh	20000.00	0.00
50000	—	—	—	—	—	—	C7FFh	50000.00	0.00

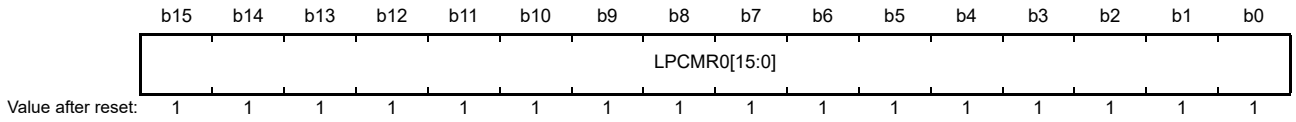
Table 27.6 Example of Low-Power Timer Period Settings for LOCO Clock

Division ratio	Divided by 1			Divided by 2			Divided by 4		
Target period (ms)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	03E7h	1.00	0.00	01F3h	1.00	0.00	00F9h	1.00	0.00
2	07CFh	2.00	0.00	03E7h	2.00	0.00	01F3h	2.00	0.00
5	1387h	5.00	0.00	09C3h	5.00	0.00	04E1h	5.00	0.00
10	270Fh	10.00	0.00	1387h	10.00	0.00	09C3h	10.00	0.00
20	4E1Fh	20.00	0.00	270Fh	20.00	0.00	1387h	20.00	0.00
50	C34Fh	50.00	0.00	61A7h	50.00	0.00	30D3h	50.00	0.00
100	—	—	—	C34Fh	100.00	0.00	61A7h	100.00	0.00
200	—	—	—	—	—	—	C34Fh	200.00	0.00
500	—	—	—	—	—	—	—	—	—
1000	—	—	—	—	—	—	—	—	—
2000	—	—	—	—	—	—	—	—	—
5000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 8			Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)	Set Value	Actual period (ms	Error (%)
1	007Ch	1.00	0.00	003Eh	1.01	0.80	001Eh	0.99	–0.80
2	00F9h	2.00	0.00	007Ch	2.00	0.00	003Eh	2.02	0.80
5	0270h	5.00	0.00	0138h	5.01	0.16	009Bh	4.99	–0.16
10	04E1h	10.00	0.00	0270h	10.00	0.00	0138h	10.02	0.16
20	09C3h	20.00	0.00	04E1h	20.00	0.00	0270h	20.00	0.00
50	1869h	50.00	0.00	0C34h	50.00	0.00	061Ah	50.02	0.03
100	30D3h	100.00	0.00	1869h	100.00	0.00	0C34h	100.00	0.00
200	61A7h	200.00	0.00	30D3h	200.00	0.00	1869h	200.00	0.00
500	F423h	500.00	0.00	7A11h	500.00	0.00	3D08h	500.00	0.00
1000	—	—	—	F423h	1000.00	0.00	7A11h	1000.00	0.00
2000	—	—	—	—	—	—	F423h	2000.00	0.00
5000	—	—	—	—	—	—	—	—	—

27.2.5 Low-Power Timer Compare Register 0 (LPCMR0)

Address(es): LPT.LPCMR0 0008 00B8h



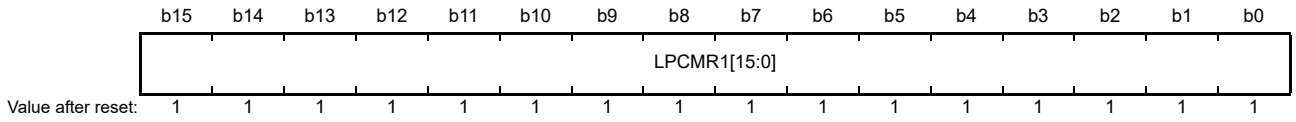
Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LPCMR0[15:0]	Low-Power Timer Compare 0	Set the value of compare match 0 for comparison with the low-power timer counter.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: When the LPTCR2.PWME bit is 0 (PWM mode is disabled), rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPCMR0 register is used to set the value of compare match 0 for comparison with the low-power timer counter. Set the LPCMR0 register to a value smaller than or equal to the value of the LPTPRD register.

27.2.6 Low-Power Timer Compare Register 1 (LPCMR1)

Address(es): LPT.LPCMR1 0008 00BAh



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LPCMR1[15:0]	Low-Power Timer Compare 1	Set the value of compare match 1 for comparison with the low-power timer counter.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note: Rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPCMR1 register is used to set the value of compare match 1 for comparison with the low-power timer counter.
When the LPTCR2.PWME bit is 0 (PWM mode is disabled), set the LPCMR1 register to a value smaller than or equal to the value of the LPTPRD register.
When the LPTCR2.PWME bit is 1 (PWM mode is enabled), set the LPCMR1 register to a value equal to the value of the LPTPRD register.

27.2.7 Low-Power Timer Standby Wakeup Enable Register (LPWUCR)

Address(es): LPT.LPWUCR 0008 00BCh

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

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	LPWKUPEN	Low-Power Timer Standby Wakeup Enable*1	0: Wakeup from software standby mode using low-power timer is disabled 1: Wakeup from software standby mode using low-power timer is enabled	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPWUCR register is used to enable the function that allows to return from software standby mode to normal operating mode when compare match 0 occurs in the low-power timer.

LPWKUPEN Bit (Low-Power Timer Standby Wakeup Enable)

This bit enables or disables the function that allows to return from software standby mode to normal operating mode when compare match 0 occurs in the low-power timer.
When the LPWKUPEN bit is to be set to 1, set the SNZCR register in section 11, Low Power Consumption to 0000h.
When any of the bits in the SNZCR register is to be set to 1, do not set the LPWKUPEN bit to 1.

27.3 Operation

27.3.1 Periodic Counting Operation

The low-power timer is a 16-bit up-counter that operates regardless of the MCU operating mode*1.
When the LPTCR3.LPCNTEN bit is set to 1 (low-power timer counter operates) after setting the LPTCR1.LPCNTPSSEL[2:0] bits to select the division ratio, the LPTCR1.LPCNTCKSEL and LPCNTCKSEL2 bits to select the clock source, and the LPTCR2.LPCNTSTP bit to 0 (clock is supplied to the low-power timer), the low-power timer counter starts counting with the selected clock.
When the value of the low-power timer counter matches the value of the LPTPRD register, the counter restarts counting from 0000h.
When the value of the low-power timer counter matches the value of the LPCMR0 register in software standby mode while the LPTCR1.LPCMRE0 bit is set to 1 (Compare match 0 is enabled) and the LPWUCR.LPWKUPEN bit is set to 1 (wakeup from software standby mode using low-power timer is enabled), the MCU returns from software standby mode to normal operating mode by the function of the event link controller (ELC).
When the value of the low-power timer counter matches the value of the LPCMR1 register while the LPTCR1.LPCMRE1 bit is set to 1, an event signal is output to the event link controller (ELC) and an interrupt is requested to the interrupt controller (ICU).
Figure 27.2 shows operation of the low-power timer and Figure 27.3 shows an example of procedure for the initial settings.

Note 1. When the LPTCR1.LPCNTCKSEL and LPCNTCKSEL2 bits are set to 10b (IWDT-dedicated clock), the counter stops because the IWDTCLK stops in the low-power consumption mode under the following settings:
The OFS0.IWDTSLCSTP bit is set to 1 (counting stop is enabled) while IWDT is activated in auto-start mode, or the IWDTCSTPR.SLCSTP bit is set to 1 (counting stop is enabled) while IWDT is activated in register start mode.

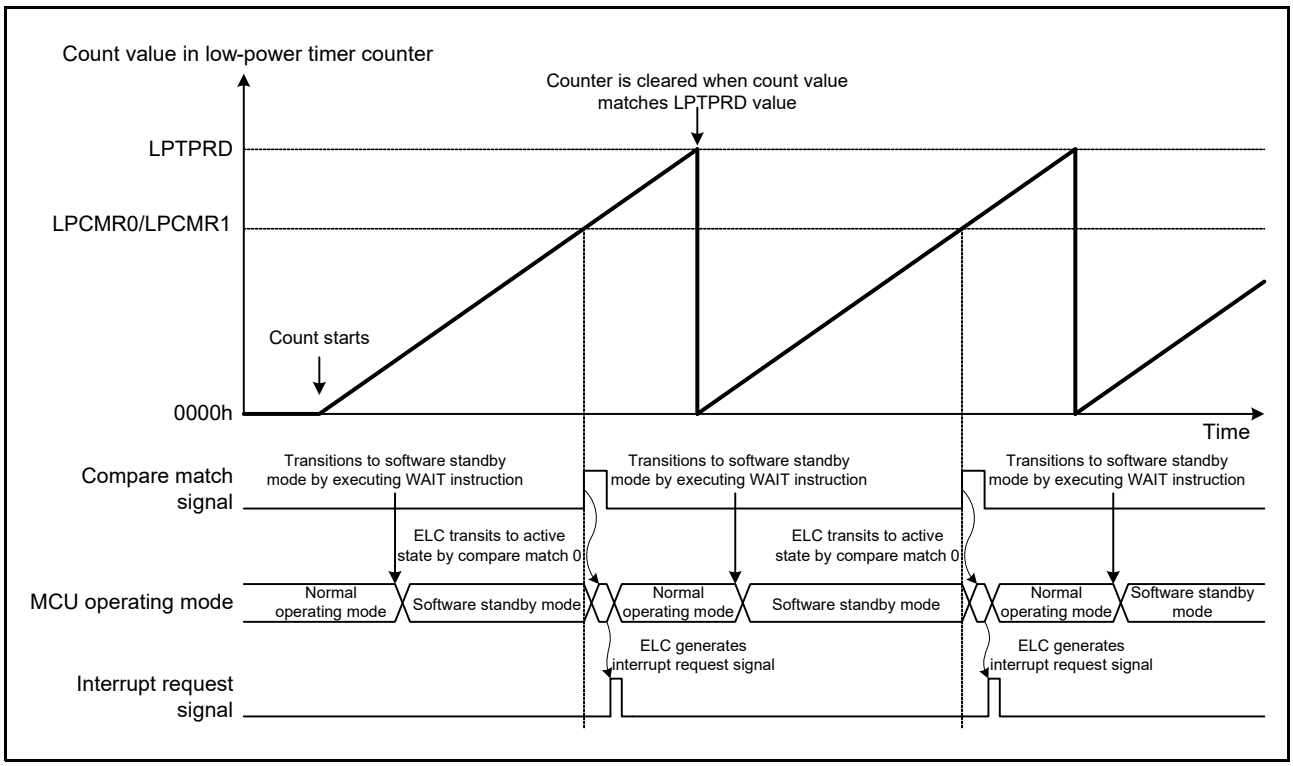


Figure 27.2 Operation of Low-Power Timer

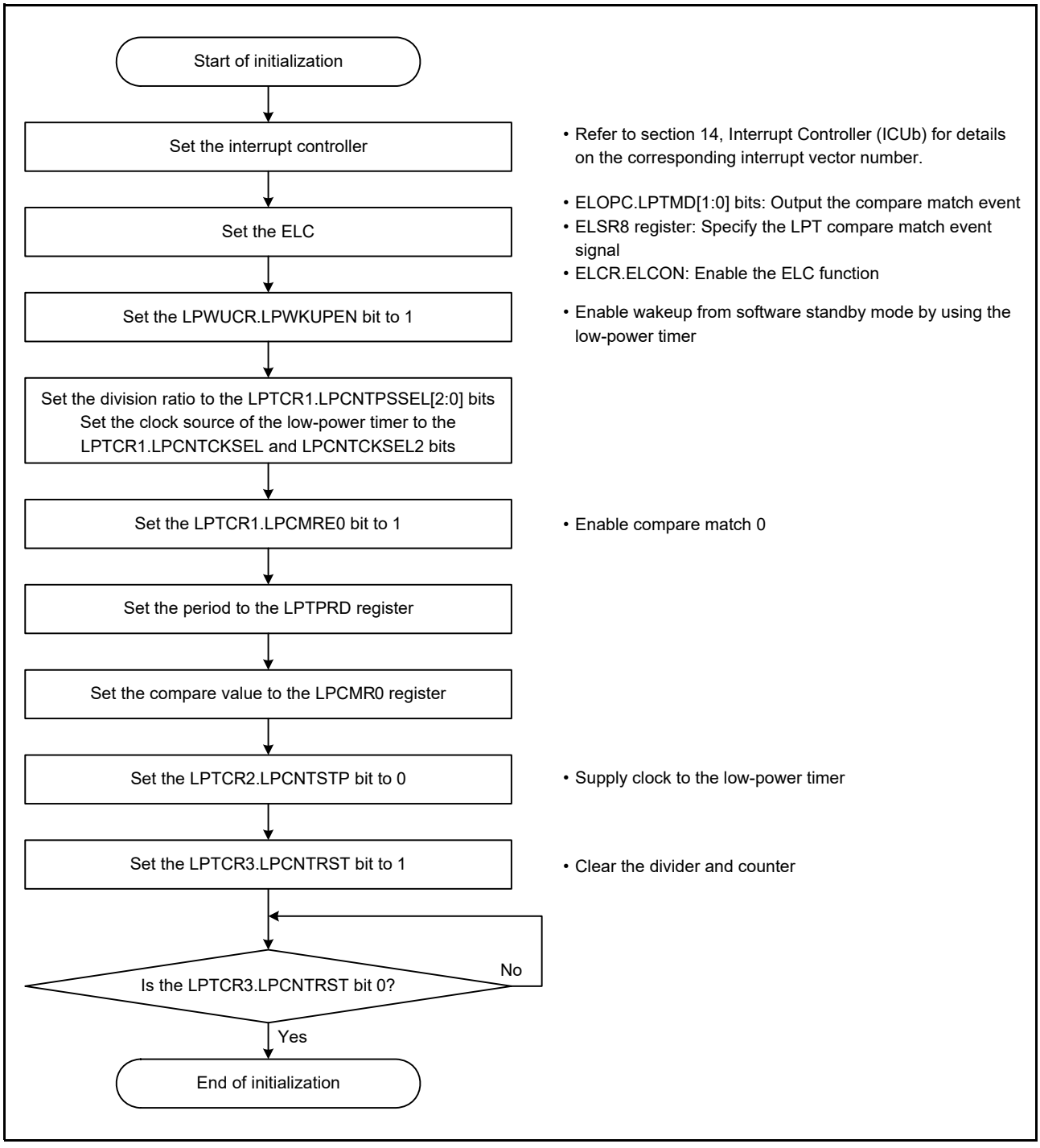


Figure 27.3 Example of Initial Settings

27.3.2 PWM Operation

When the LPTCR2.PWME bit is set to 1, the low-power timer operates in PWM mode. Set the LPCMR1 register to the same value as the LPTPRD register in PWM mode.

Any PWM waveform can be output from the PWM waveform output pin (LPTO) in combination of the LPTCR2.OPOL and OLVL bits. Figure 27.4 shows the waveforms that LPT can output.

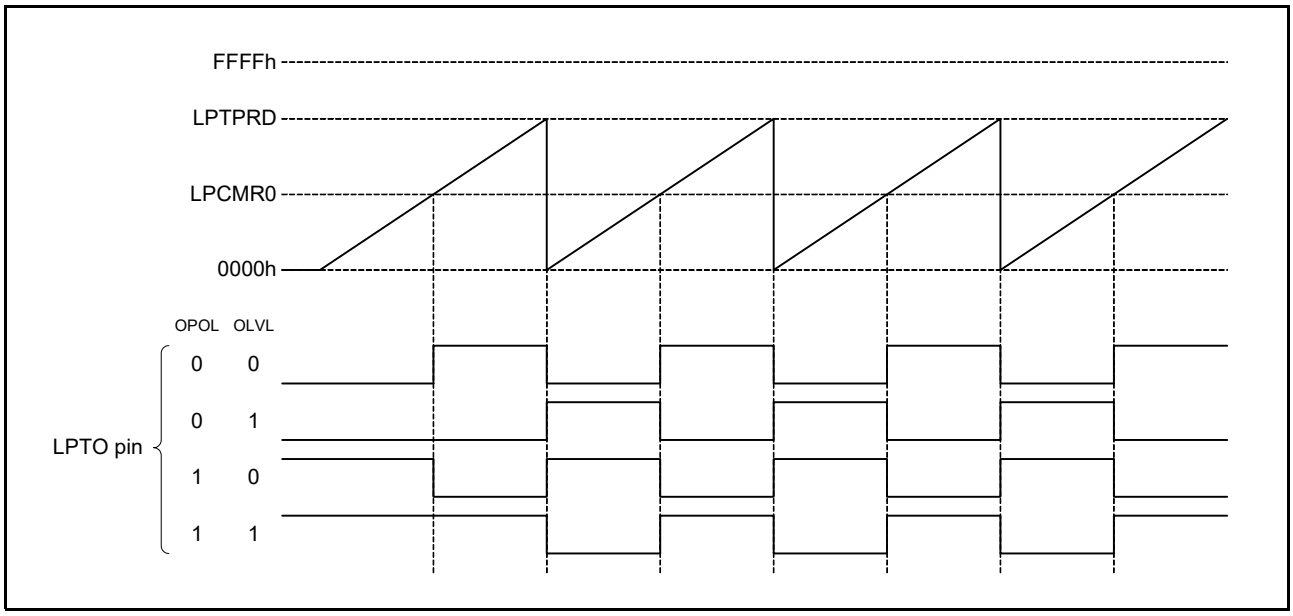


Figure 27.4 PWM Waveform

Figure 27.5 shows an example of PWM mode operation when both the LPTCR2.OPOL and OLVL bits are 0. At this time, the high width of the LPTO pin output is expressed as “LPTPRD – LPCMR0”, and the low width is expressed as “LPCMR0 + 1”.

The counting operation of the low-power timer is the same as the periodic counting operation. When the initial setting is made and the LPTCR3.LPCNTEN bit is set to 1, the low-power timer starts counting. When the value of the counter matches the value of the LPCMR0 register, the LPTO pin goes high and a compare match 0 event occurs. When the value of the counter matches the value of the LPTPRD register (= the value of the LPCMR1 register), the counter is reset, the LPTO pin goes low, and the compare match 1 event occurs.

The duty of the PWM output can be changed by rewriting the value of the LPCMR0 register in the compare match 1 interrupt processing routine. It is also possible to rewrite the LPCMR0 register by DTC or DMA transfer without using the CPU.

In addition, when rewriting the LPCMR0 register by DTC transfer in snooze mode, the PWM waveform can be changed while maintaining the low power consumption mode. For details on snooze mode, refer to section 11, Low Power Consumption.

If the LPCMR0 register is rewritten before compare match 0 occurs and the rewritten value is smaller than the counter value, compare match 0 does not occur in that cycle, so the LPTO pin level does not change.

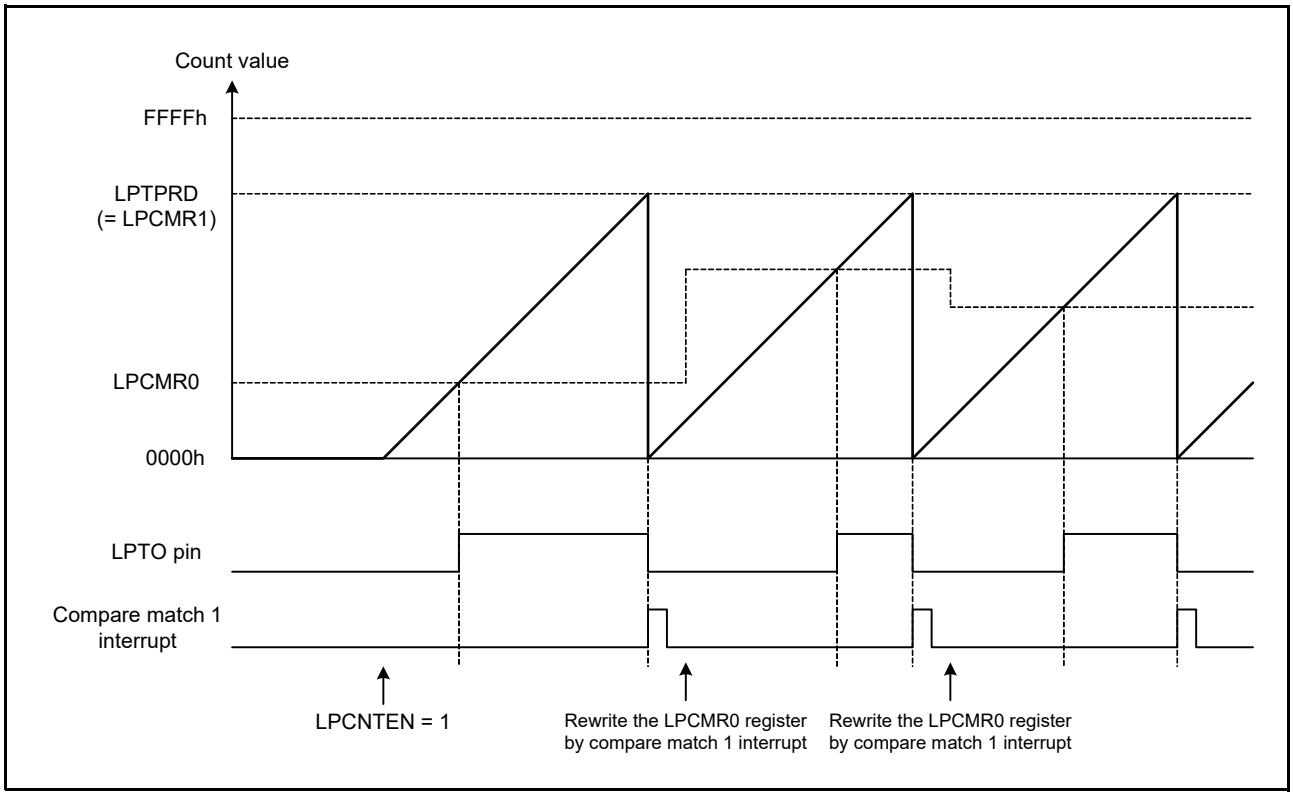


Figure 27.5 Operation Example of PWM Mode (OPOL = 0, OLVL = 0)

27.3.3 Count Timing of Low-Power Timer Counter

The LPTCR1.LPCNTPSSEL[2:0] bits are used to select the counter clock to be input to the low-power timer counter from among six clocks derived from dividing the clock source selected by the LPTCR1.LPCNTCKSEL and LPCNTCKSEL2 bits by 1 to 32, respectively.
Figure 27.6 shows the count timing of the low-power timer counter in this case.

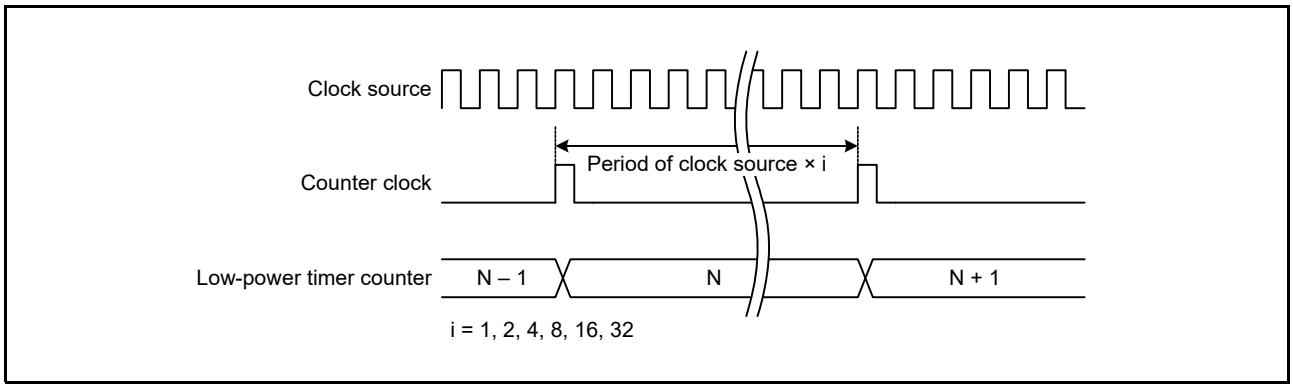


Figure 27.6 Count Timing of Low-Power Timer Counter

27.3.4 Clearing Timing of Low-Power Timer Counter

Writing 1 to the LPTCR3.LPCNTRST bit*1 clears the low-power timer counter.
This bit automatically becomes 0 when the clearing of the counter is completed.
Figure 27.7 shows the clearing timing of the low-power timer counter in this case.

Note 1. Write to the LPTCR3.LPCNTRST bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

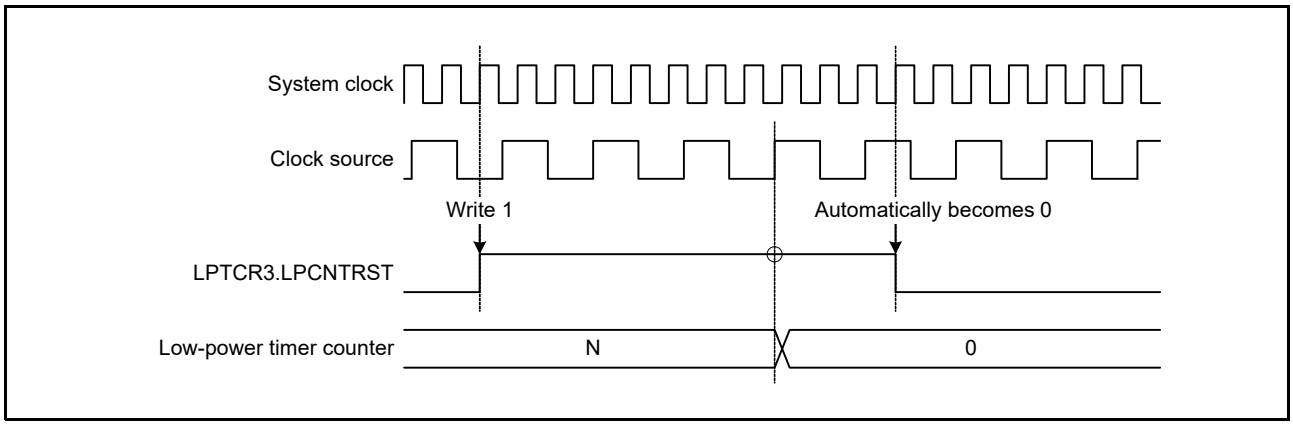


Figure 27.7 Clearing Timing of Low-Power Timer Counter

27.4 Interrupt Sources

The low-power timer has a interrupt source of compare match 1 (LPTCM1). The compare match 1 interrupt can start DTC or DMA transfer. When using the compare match 1 interrupt, release the module stop of the event link controller (ELC).

27.5 Event Link Function (Output)

The low-power timer can generate an event signal upon compare match 1 to the event link controller (ELC) and can make a previously specified module operate. Refer to section 19, Event Link Controller (ELC) for details.

27.6 Requesting Transition to Snooze Mode

The compare match 1 event of the low-power timer can be used as a request signal to transit to snooze mode. Refer to section 11, Low Power Consumption for details.

27.7 Wakeup from Software Standby Mode by an Interrupt through the Event Link Controller (ELC)

The low-power timer generates an event signal upon compare match 0 to the event link controller (ELC) only in software standby mode.
Setting the ELOPC.LPTMD[1:0] bits to 00b (output the compare match 0 event to ICU as an interrupt request) and the ELSR8 register to 32h (LPT compare match 0) leads to the generation of an interrupt from the event signal, and the MCU returns from software standby mode to normal operating mode.

27.8 Usage Notes

27.8.1 Notes on Transition to Software Standby Mode

When the MCU is to re-enter to software standby mode after returning from software standby mode to normal operating mode, wait for at least one cycle of the clock selected by the LPTCR1.LPCNTCKSEL and LPCNTCKSEL2 bits before executing the WAIT instruction.

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 on 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.

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	Description
Count source	Peripheral module clock (PCLKB)
Clock divide ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit 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 • In low power consumption modes • 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	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The counter value can be read by the WDTSR register.

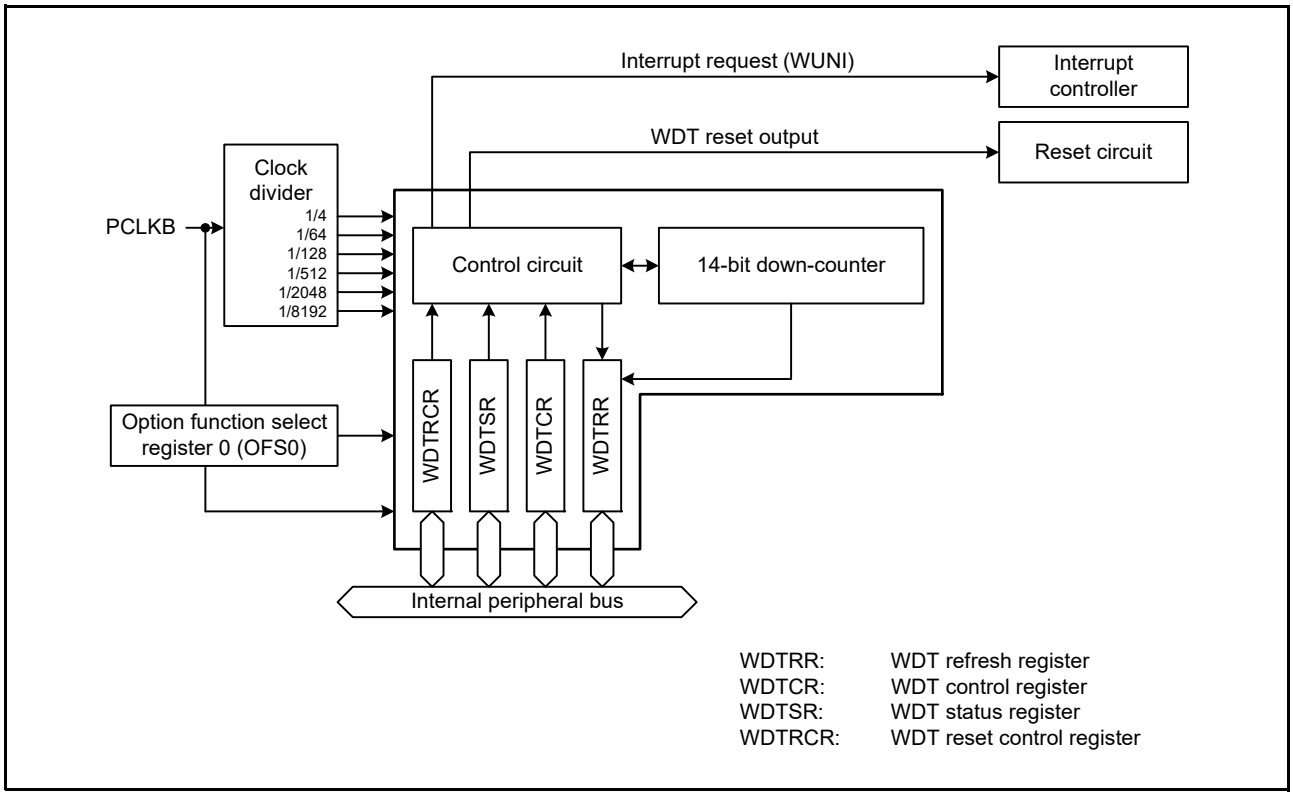
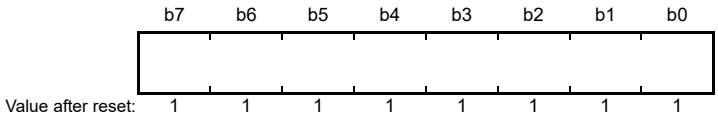


Figure 28.1 WDT Block Diagram

28.2 Register Descriptions

28.2.1 WDT Refresh Register (WDTRR)

Address(es): WDT.WDTRR 0008 8020h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register	R/W

The WDTRR register is used to refresh the counter of the WDT.
The counter of the WDT is refreshed by writing 00h and then writing FFh to the WDTRR register (refresh operation) within the refresh-permitted period.
After the 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 28.3.2, Refresh Operation.

28.2.2 WDT Control Register (WDTCR)

Address(es): WDT.WDTCR 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 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 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/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

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the time until a timeout occurs and the refresh-permitted period when using register start mode.

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 the OFS0 register. For details, refer to section 28.3.6, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the 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 counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLKB cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles are listed in Table 28.2.

Table 28.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of PCLKB
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 Divide Ratio Select)

These bits specify the divide ratio of the clock used for the counter. The divide ratio can be selected from among the PCLKB 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 PCLKB can be selected for the WDT.

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

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 Select)

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 the window end setting is greater than or equal to the window start setting, the window end position is set to 0%.

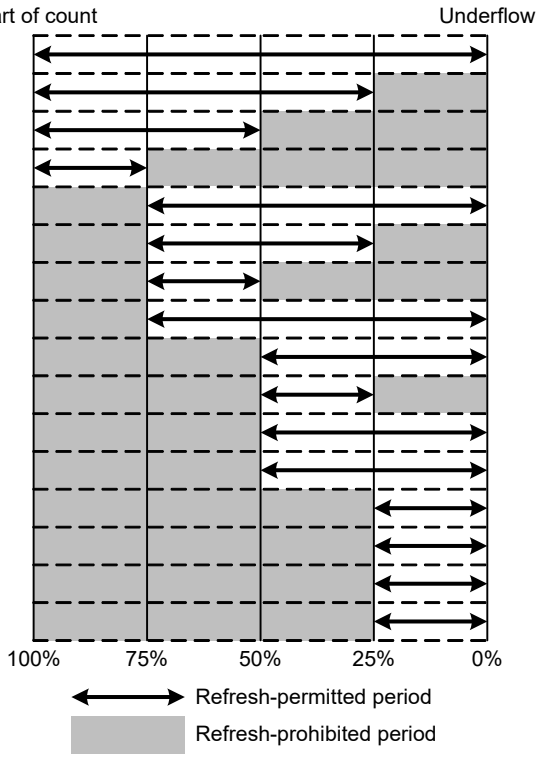


Figure 28.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

28.2.3 WDT Status Register (WDTSR)

Address(es): WDT.WDTSR 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]	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.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the 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 flag 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 flag is used 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): WDT.WDTRCR 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. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request is output. 1: Reset signal is output.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the behavior when a timeout or refresh error occurs when using register start mode. In auto-start mode, the WDTRCR register setting is disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTRCR register can also be made in the OFS0 register. For details, refer to section 28.3.6, 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 28.3.6, 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 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 divide 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 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 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 WDT outputs a reset signal or an interrupt request (WUNI). Reset signal 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)

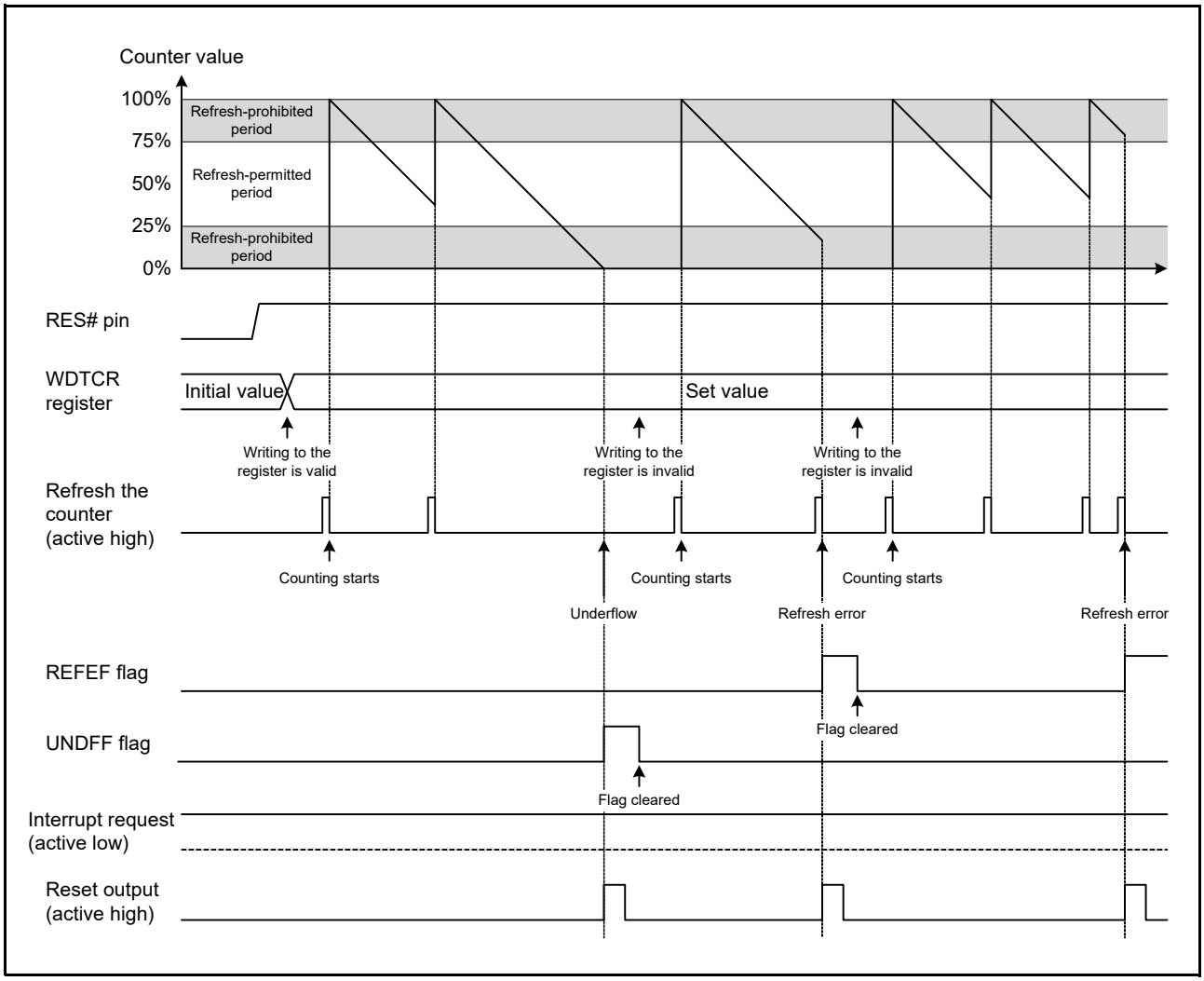


Figure 28.3 Operation Example in Register Start Mode

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.
During the reset state, the setting values (clock divide ratio, window start and end positions, timeout period, and reset output or interrupt request output) of the OFS0 register are set in the WDT registers. When the reset is released, the 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 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 WDT outputs the reset signal or interrupt request (WUNI).
After the reset signal or interrupt request is output of for one cycle of counting, the value of the timeout period is set in the counter counting is restarted.
Reset signal output or interrupt request output can be selected by setting the OFS0.WDTRSTIRQS bit.

Figure 28.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.WDTSTRT = 0)
- 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)

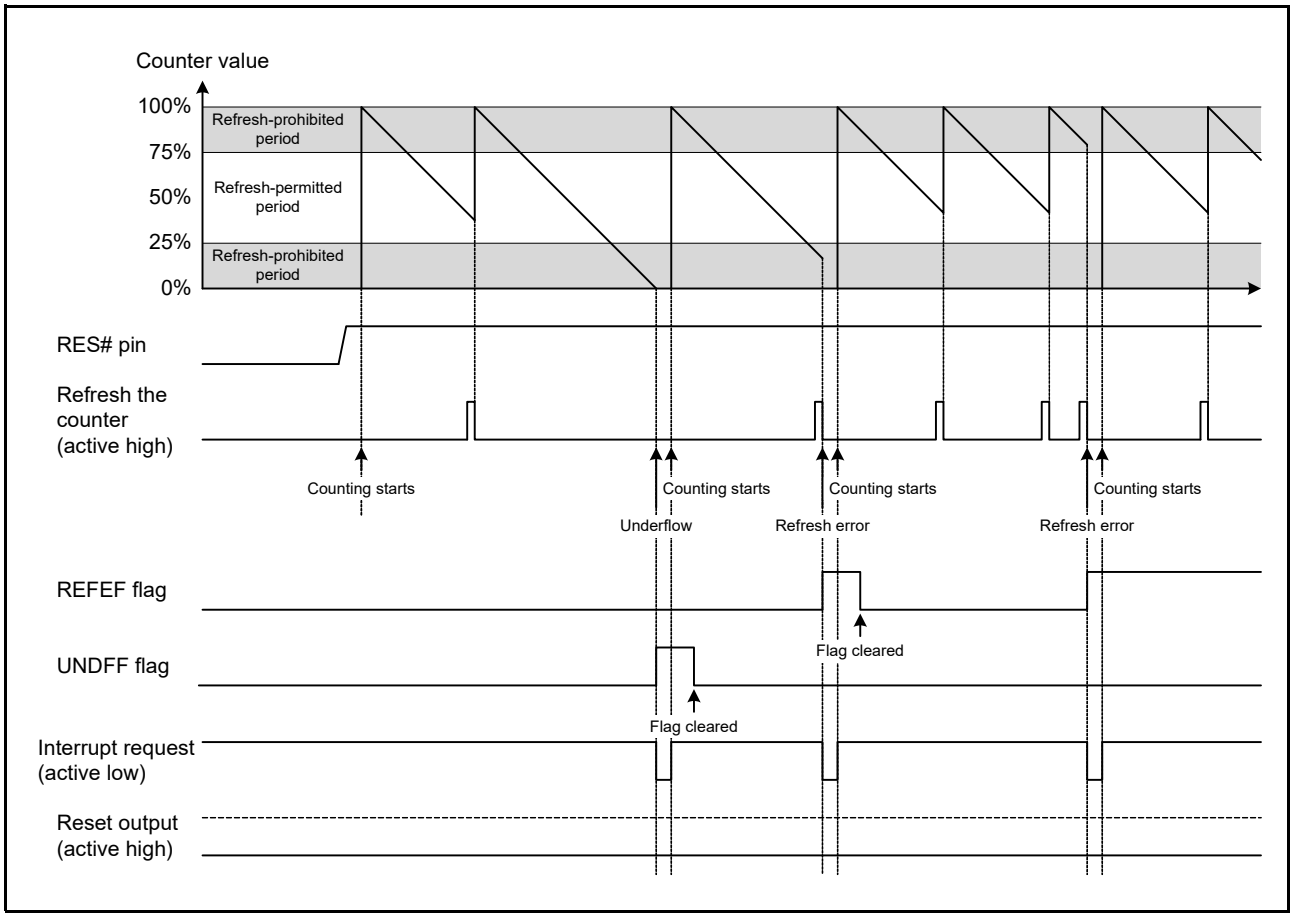


Figure 28.4 Operation Example in Auto-Start Mode

28.3.2 Refresh Operation

The 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 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 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 counter underflows.

Figure 28.5 shows the WDT refresh operation waveforms when the count clock is PCLKB/64.

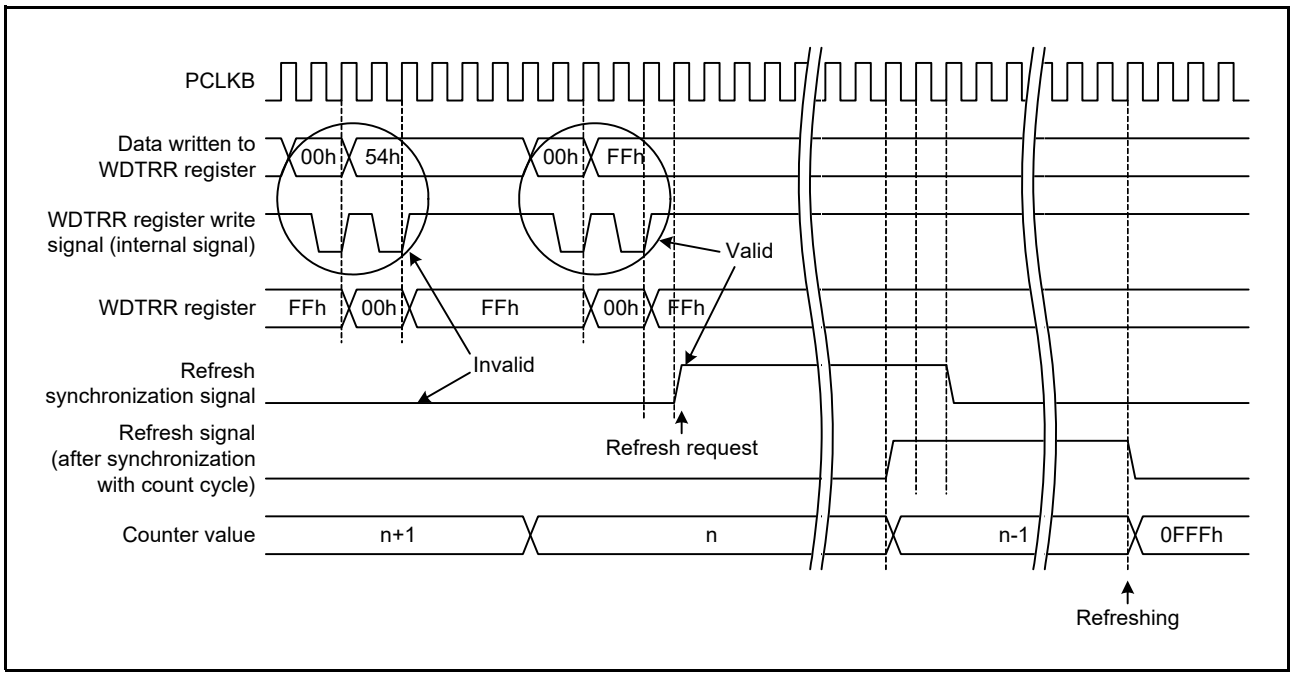


Figure 28.5 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

28.3.3 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 when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (0000h) and stopped in that state after output of the reset signal. After the reset is released, the counter value is restored and down-counting is started by performing a refresh operation.
In auto-start mode, counting down automatically starts after the reset output.

28.3.4 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. For details, refer to section 14, Interrupt Controller (ICUb).

Table 28.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Counter underflow Refresh error	Not possible	Not possible

28.3.5 Reading the 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 28.6 shows the processing for reading the WDT counter value when the count clock is PCLKB/64.

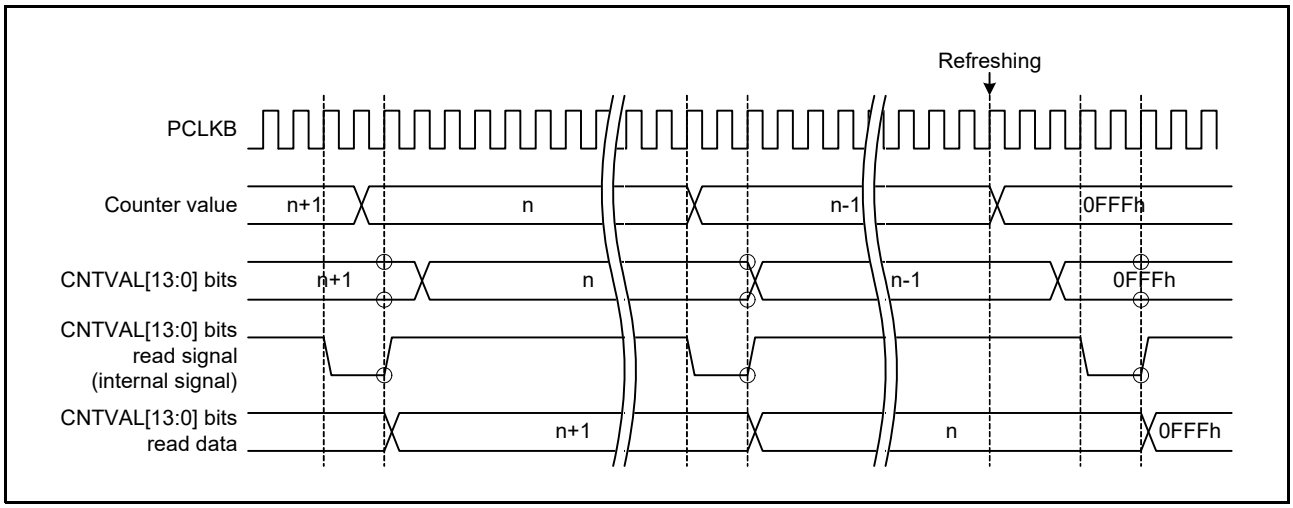


Figure 28.6 Processing for Reading WDT Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

28.3.6 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.
For details on the OFS0 register, refer to section 7.2.1, 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
Counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock divide 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)

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 peripheral module clock).
- The IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select not to stop the counter in low power consumption mode.

29.1 Overview

Table 29.1 lists the specifications 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 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 • In low power consumption modes (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	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The counter value can be read by the IWDTSR register.
Event link function (output)	• Counter underflow event output • Refresh error event 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 counter operation/stop in low power consumption 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 counter operation/stop in low power consumption mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 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.

Figure 29.1 is a block diagram of the IWDT.
In IWDT, the bus interface and registers operate with the peripheral module clock (PCLKB), while the 14-bit down-counter and control circuits operate with IWDTCLK.

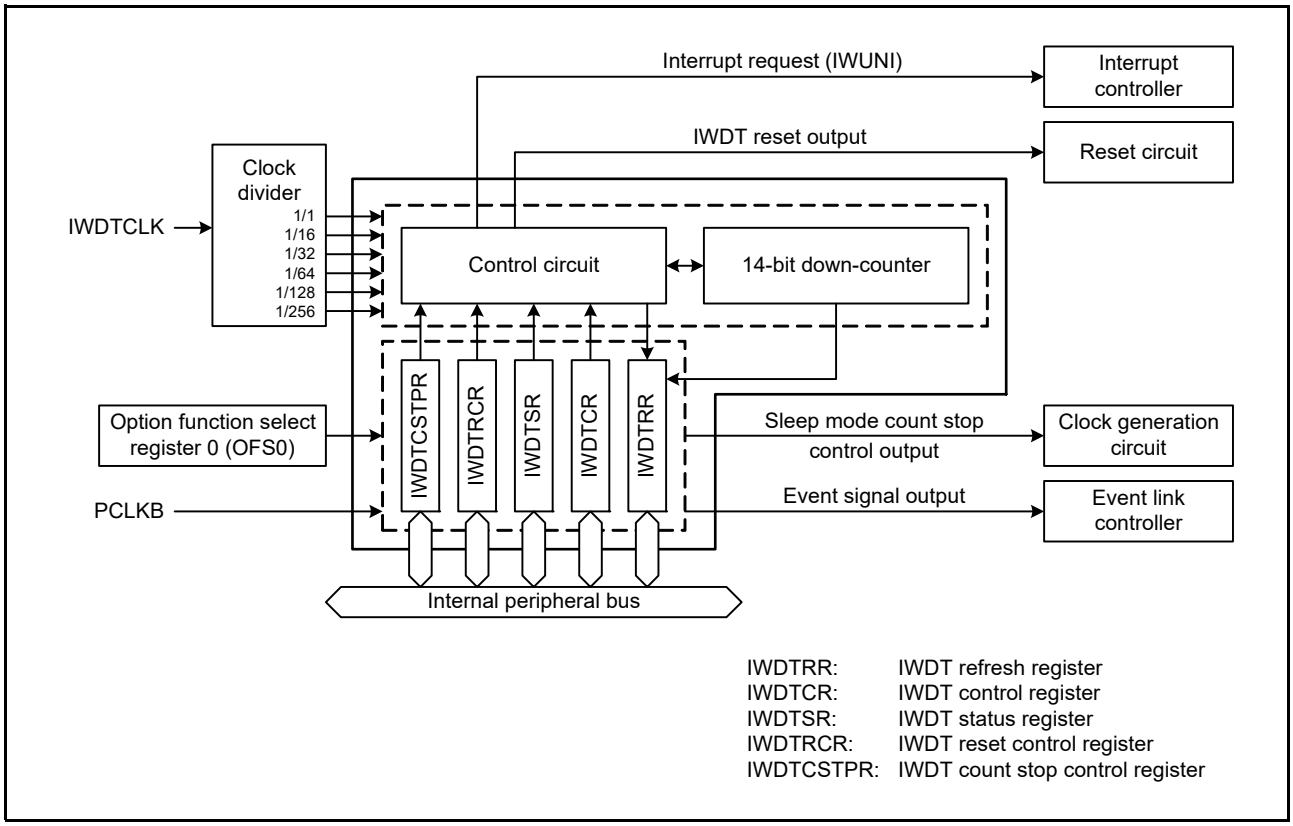
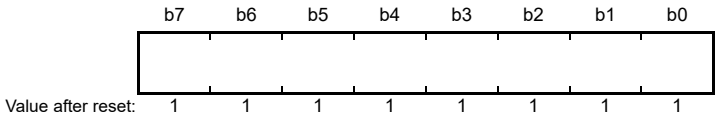


Figure 29.1 IWDT Block Diagram

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 is used to refresh 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 29.3.2, 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: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	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 Settings other than above 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

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the time until a timeout occurs and the refresh-permitted period when using register start mode.

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 29.3.7, 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 128, 512, 1024, or 2048 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	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	Divide-by-16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	Divide-by-32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	Divide-by-64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	Divide-by-128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	Divide-by-256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK 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 128 and 524288 cycles of the IWDTCLK 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	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

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 the window end setting is greater than or equal to the window start setting, the window end position is set to 0%.

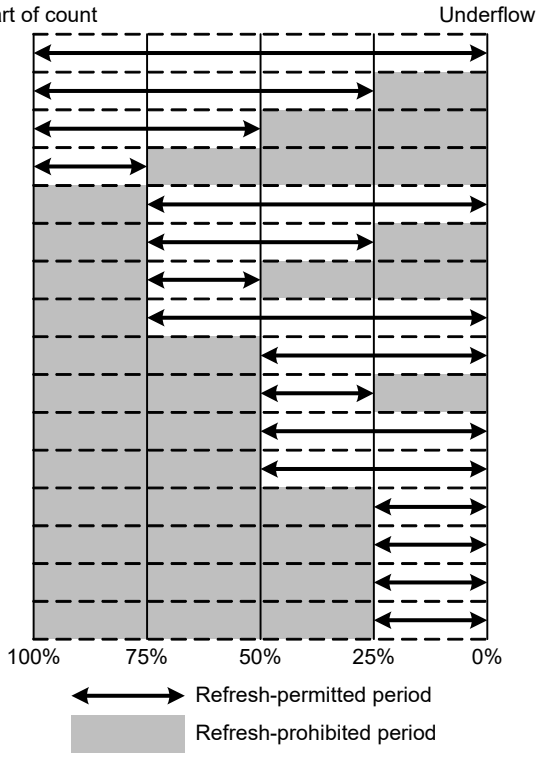


Figure 29.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

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.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the 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 flag 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 flag is used 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.

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 is output. 1: Reset signal is output.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the behavior when a timeout or refresh error occurs when using register start mode. In auto-start mode, the IWDTRCR register setting is disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 29.3.7, 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: Counter continues counting even in low power consumption mode.*1 1: Counter stops in low power consumption mode.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.
Note 1. The counter continues counting in sleep mode, deep sleep mode, or software standby mode.

This register is used to specify whether the counter operates in low power consumption mode when using register start mode.
In auto-start mode, the IWDTCSTPR register setting is 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 29.3.7, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether the counter continues or stops counting in low power consumption mode.

29.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 29.3.7, 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 counter operation/stop in low power consumption mode 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 an interrupt request (IWUNI). Set the IWDTRCR.RSTIRQS bit to select either reset signal 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)

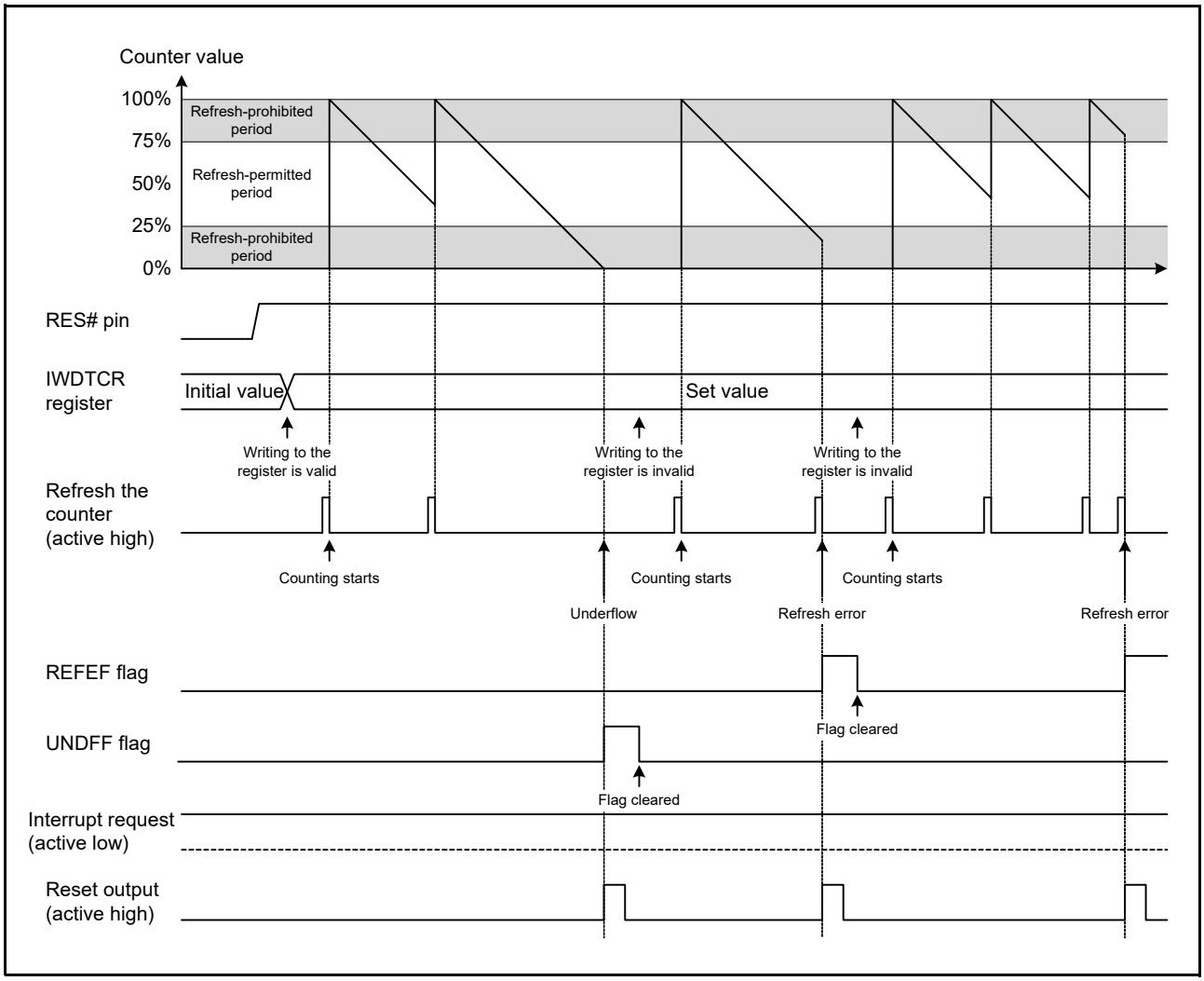


Figure 29.3 Operation Example in Register Start Mode

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.
During the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter operation/stop in low power consumption mode 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 an interrupt request (IWUNI). After the reset signal or 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 signal output or interrupt request output.

Figure 29.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- 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)

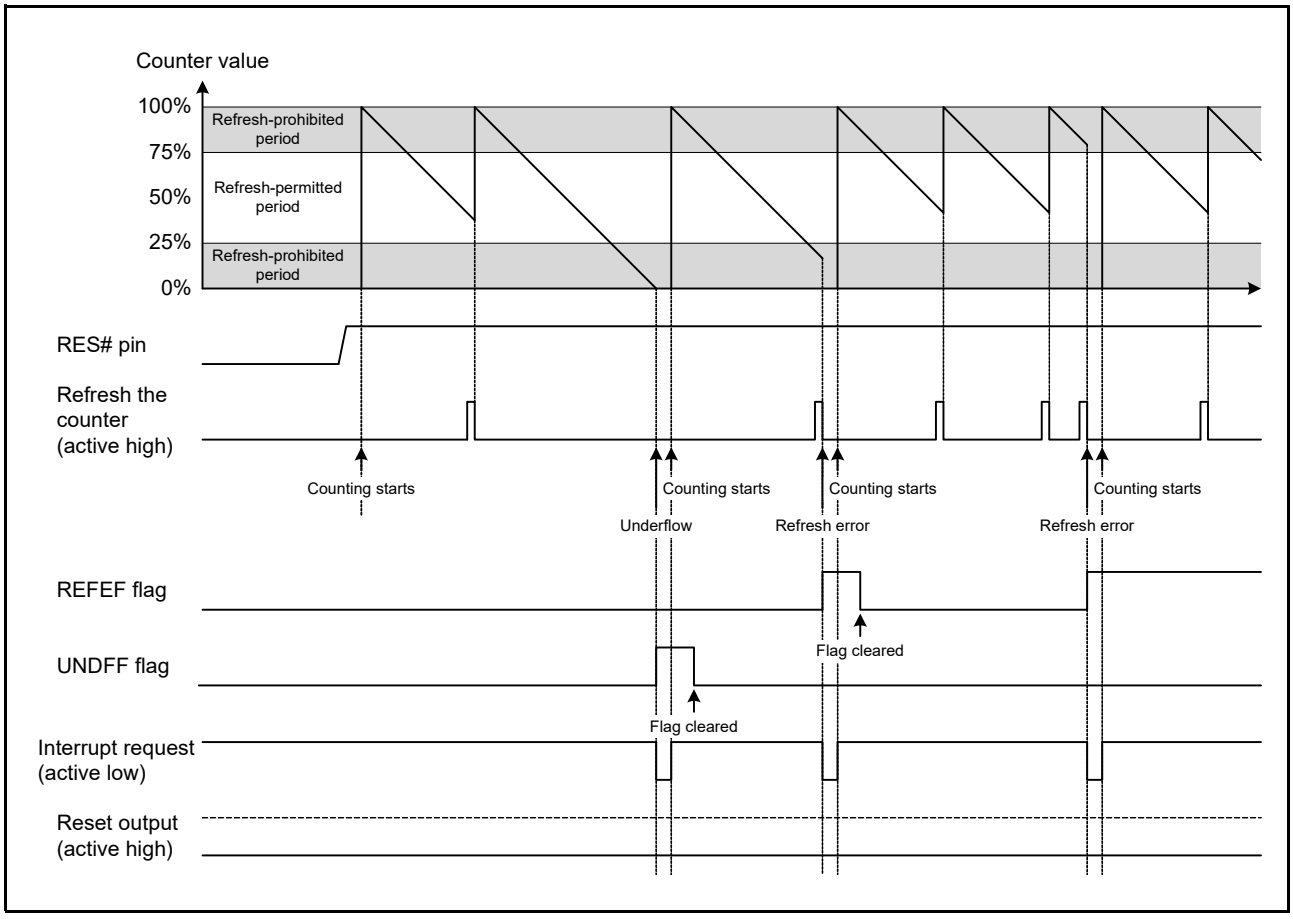


Figure 29.4 Operation Example in Auto-Start Mode

29.3.2 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 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 01FFh, even if 00h is written to the IWDTRR register before 01FFh is reached (0202h, 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 01FFh.
- When the window end position is set to 01FFh, refreshing is done if 0203h (four-count cycles before 01FFh) 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 29.5 shows the IWDT refresh operation waveforms when PCLKB > IWDTCLK and the count clock is IWDTCLK/1.

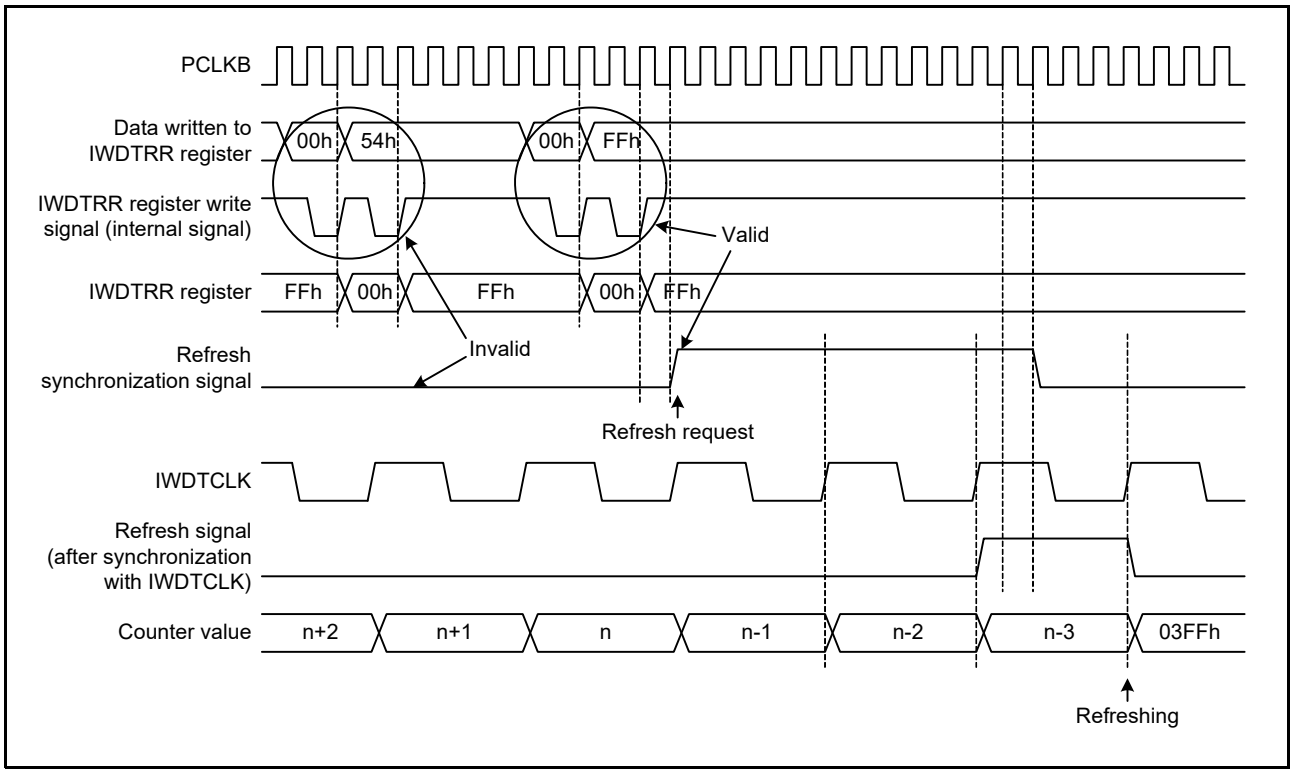


Figure 29.5 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 10b)

29.3.3 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 PCLKB cycles are required before the value is reflected.

29.3.4 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 stopped in that state after output of the reset signal. After the reset is released, the counter value is restored and down-counting is started by performing a refresh operation.
In auto-start mode, counting down automatically starts after the reset output.

29.3.5 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. For details, refer to section 14, Interrupt Controller (ICUb).

Table 29.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
IWUNI	Counter underflow Refresh error	Not possible	Not possible

29.3.6 Reading the Counter Value

As the counter in IWDTCLK, the counter value cannot be read directly. The IWDT synchronizes the counter value with the PCLKB 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 PCLKB 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 29.6 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the count clock is IWDTCLK/1.

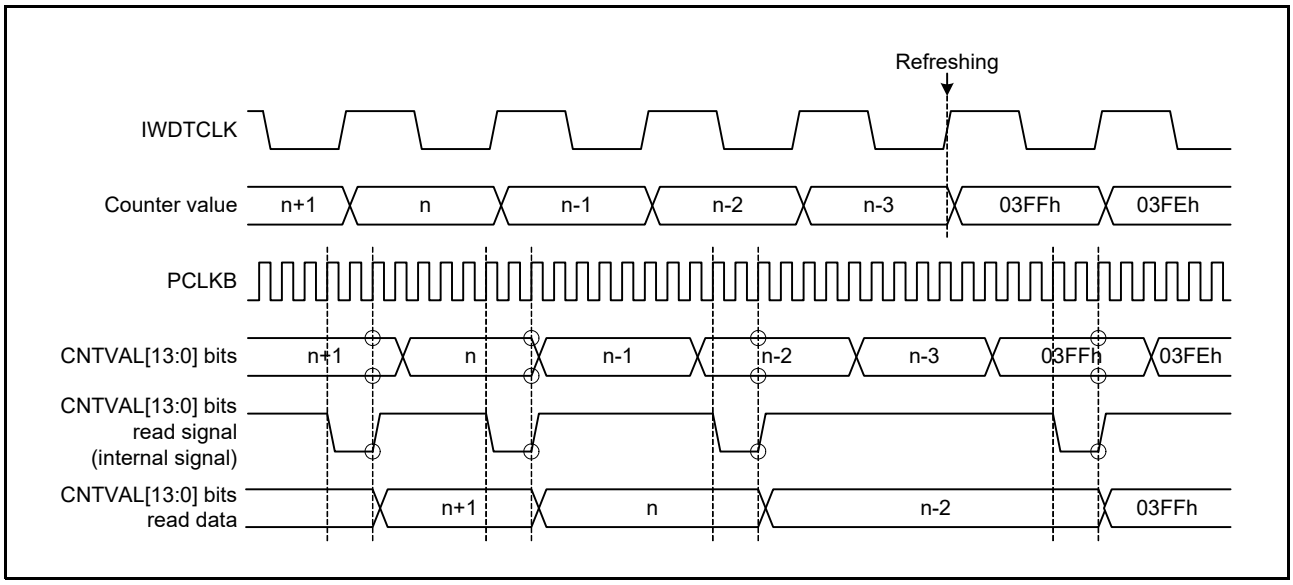


Figure 29.6 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 10b)

29.3.7 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.
For details on the OFS0 register, refer to section 7.2.1, 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 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 PCLKB and IWDTCLK and set values which ensure that refreshing is possible.

29.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the PCLKB $\geq 4 \times$ (the frequency of the count source after divide).

30. USB 2.0 FS Host/Function Module (USBe)

30.1 Overview

This MCU incorporates a USB 2.0 FS host/function module (USB0).
The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed (only for the host controller) transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0.
The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.
Table 30.1 shows the specifications of the USB.

Table 30.1 USB Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (one channel) • The host controller and the function controller can be switched by software. • Self-power mode or bus power mode can be selected.
	When the host controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Multiple peripheral devices can be connected for communication via a one-stage hub.
	When the function controller is selected: • Full-speed transfer (12 Mbps) is supported*1 • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• Buffer memory for USB communication is provided. • Up to 10 pipes can be selected (including the default control pipe). • Pipes 1 to 9 can be assigned any endpoint number.
	Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer, 64-byte single buffer • Pipes 1 and 2: 64-byte double buffer can be specified for bulk transfer and 256-byte double buffer for isochronous transfer • Pipes 3 to 5: Bulk transfer, 64-byte double buffer • Pipes 6 to 9: Interrupt transfer, 64-byte single buffer
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK) • On-chip pull-up and pull-down resistors of D+/D–
Low power consumption function	Module-stop state can be set.

Note 1. When the function controller is selected, low-speed transfer (1.5 Mbps) is not supported.

Figure 30.1 shows a block diagram of the USB.

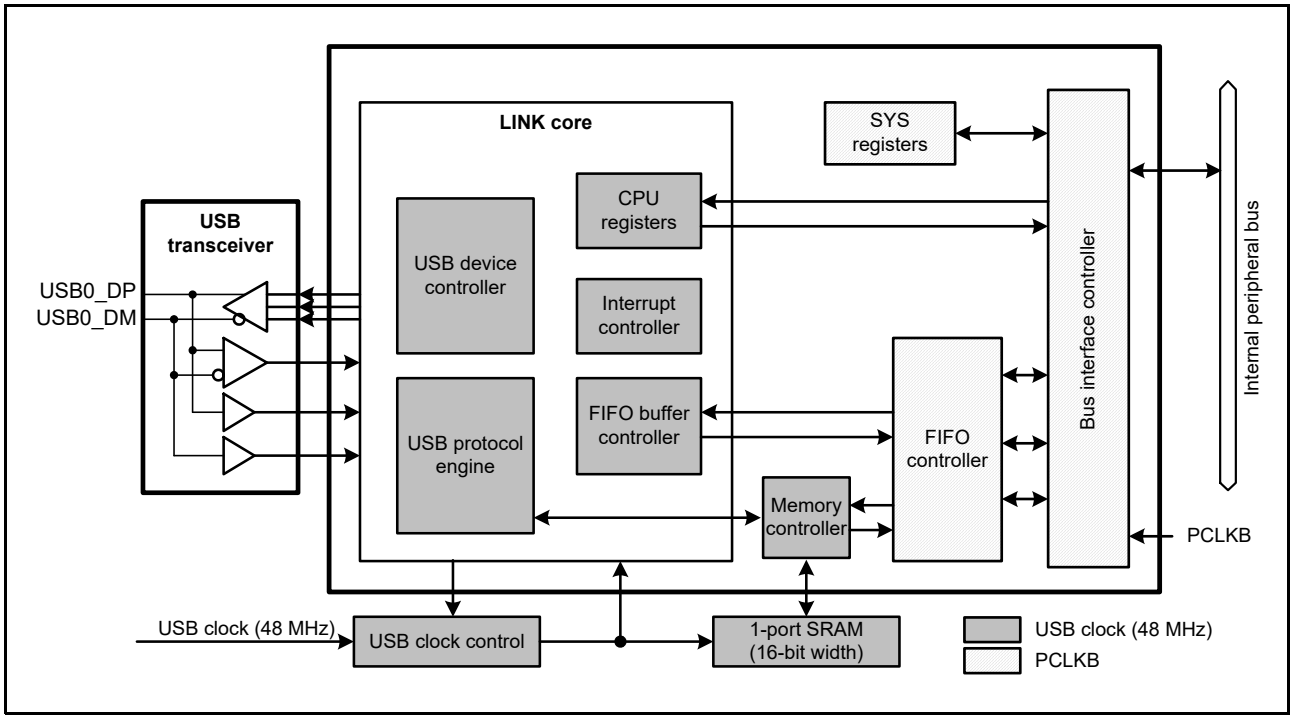


Figure 30.1 USB Block Diagram

Table 30.2 lists the I/O pins of the USB.

Table 30.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB0_DM	I/O	D– I/O pin of the USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.*1
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.

Note 1. P16 is 5 V tolerant.
When using non-5 V tolerant PB5, reduce VBUS to 3.3 V and connect to the pin.

30.2 Register Descriptions

30.2.1 System Configuration Control Register (SYSCFG)

Address(es): USB0.SYSCFG 000A 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*1	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 30.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stabilized.

Table 30.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	—	The value is retained when the host controller is selected.
	USBINDX	—	The value is retained when the host controller is selected.
	USBLENG	—	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.
When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.
This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified when the DPRPU and DRPD bits are all 0.

SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.
When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

30.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USB0.SYSSTS0 000A 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	—	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Refer to Table 30.4.	R
b2	IDMON	ID Input Pin Monitor Flag	0: USB0_ID pin is low 1: USB0_ID pin is high	R
b5 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b6	HTACT	USB Host Sequencer Status Monitor Flag	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	OVRCURA/OVRCURB Input Pin Monitor Flag	The OVCMON[1] flag indicates the status of the USB0_OVRCURA pin. The OVCMON[0] flag indicates the status of the USB0_OVRCURB pin.	R

Note 1. Depends on the status of the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pins.

LNST[1:0] Flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines (D+ and D– lines). Refer to Table 30.4.

The LNST[1:0] flags should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

HTACT Flag (USB Host Sequencer Status Monitor Flag)

The HTACT flag is 0 when the host sequencer of the USB is completely stopped.

OVCMON[1:0] Flags (OVRCURA/OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of overcurrent from an external power supply chip.

Table 30.4 Status of USB Data Bus Lines (D+ Line, D– Line)

LNST[1:0] Flags	During Low-Speed Operation (Only in Host Controller Operation)	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

30.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USB0.DVSTCTR0 000A 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	• When the host controller is selected b2 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection • When the function controller is selected b2 b0 0 0 0: Communication speed not determined 0 0 1: USB bus reset in progress 0 1 0: USB bus reset in progress or full-speed connection	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	VBUSEN Output Pin Control	0: USB0_VBUSEN pin outputs low. 1: USB0_VBUSEN pin outputs high.	R/W
b10	EXICEN	EXICEN Output Pin Control	0: USB0_EXICEN pin outputs low. 1: USB0_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don’t care

RHST[2:0] Flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the status of the USB bus reset.
When the host controller is selected, the RHST[2:0] flags indicate 100b after the USBRST bit has been set to 1 by software.
The USB fixes the value of the RHST[2:0] flags when 0 is written to the USBRST bit by software and the USB completes SE0 driving.
When the function controller is selected, the RHST[2:0] flags indicate 010b (connection while DPRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.
With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.
This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.
With this bit set to 0, the USB enters the idle state after outputting SOF packets.
The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.
Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.
The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.
The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.
This bit should be set to 1 in the suspended state.
Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.
When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.
The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.
Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).
This bit should be set to 0 if the function controller is selected.

RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.

With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 µs) from the downstream port device and performs the resume processing (drives the port to the K-state).

With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.

While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).

This bit should be set to 0 if the function controller is selected.

WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.

The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.

According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).

This bit should be set to 0 if the host controller is selected.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.

After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

30.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

Address(es): USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch

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

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): USB0.CFIFO.L 000A 0014h, USB0.D0FIFO.L 000A 0018h, USB0.D1FIFO.L 000A 001Ch

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).
Each FIFO port has the following precautions.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA/DTC transfer function is used, etc.).
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right for a FIFO buffer can be on the CPU side or on the Serial Interface Engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bit

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY flag in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 30.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the lower byte in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the lower byte in bytes.

Table 30.5 Endian Operation in 16-Bit Access

MDE.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

30.2.5 CFIFO Port Select Register (CFIFOSEL), D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): USB0.CFIFOSEL 000A 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected. 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The CFIFOCTR.DTLN[8:0] flags are set to 000h when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] flag value is set to 000h when all the data has been read from only a single plane.) 1: The CFIFOCTR.DTLN[8:0] flags are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.
Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

- D0FIFOSEL, D1FIFOSEL

Address(es): USB0.D0FIFOSEL 000A 0028h, USB0.D1FIFOSEL 000A 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DnFIFOCTR.DTLN[8:0] flags are set to 000h when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN[8:0] flag value is set to 000h when all the data has been read from only a single plane.) 1: The DnFIFOCTR.DTLN[8:0] flags are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables the DMA/DTC transfer request to be issued.
Before setting the DREQE bit to 1 to enable the DMA/DTC transfer request to be issued, set the CURPIPE[3:0] bits.
When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data in the selected pipe has been read.
With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.
When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the DnFIFOCTR.DTLN[8:0] flags.
When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

30.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USB0.CFIFOCTR 000A 0022h, USB0.D0FIFOCTR 000A 002Ah, USB0.D1FIFOCTR 000A 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length Flag	Indicate the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] flags shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

Registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

DTLN[8:0] Flags (Receive Data Length Flag)

The DTLN[8:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] flags indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1) value as described below.

- RCNT = 0

  The USB sets the DTLN[8:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data from a single FIFO buffer plane.

  While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.

- RCNT = 1

  The USB decrements the value indicated by the DTLN[8:0] flags each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)

  The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

FRDY Flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the following cases, the USB sets the FRDY flag to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only the FIFO buffer on the CPU side even when both planes are read-enabled.
When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL bit and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY flag in the FIFO port control register is 1 (set by the USB).

BVAL Bit (Buffer Memory Valid)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while the FRDY flag is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

30.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USB0.INTENB0 000A 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected. Set these bits to 0 when the host controller is selected.

On detecting the interrupt corresponding to the bit in the INTENB0 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS0 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB0 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS0 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB0 register is modified from 0 to 1 by software.

30.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USB0.INTENB1 000A 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in the INTENB1 register can be set to 1 only when the host controller is selected. Set these bits to 0 when the function controller is selected.

The INTENB1 register specifies the interrupt masks when the host controller is selected, and for the setup transaction. On detecting the interrupt corresponding to the bit in the INTENB1 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS1 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB1 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS1 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB1 register is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

30.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USB0.BRDYENB 000A 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY flag to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in the BRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY flag (n = 0 to 9) and the INTSTS0.BRDY flag. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY flag indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in the BRDYENB register is modified from 0 to 1 by software.

30.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USB0.NRDYENB 000A 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY flag to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in the NRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPEnNRDY flag (n = 0 to 9) and the INTSTS0.NRDY flag. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPEnNRDY flag indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in the NRDYENB register is modified from 0 to 1 by software.

30.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USB0.BEMPENB 000A 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP flag to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in the BEMPENB register to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP flag (n = 0 to 9) and the INTSTS0.BEMP flag. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP flag in the BEMPSTS register indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in the BEMPENB register is modified from 0 to 1 by software.

30.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USB0.SOFCFG 000A 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor Flag*1	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

EDGESTS Flag (Edge Interrupt Output Status Monitor Flag)

The EDGESTS flag indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Confirm that this flag is 0 before stopping the clock supply to the USB.

BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected.
The TRNENSEL bit is valid only when the host controller is selected.
This bit should be set to 0 if the function controller is selected.

30.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USB0.INTSTS0 000A 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
b3	VALID	USB Request Reception Flag	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State Flag	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: USB0_VBUS pin is low. 1: USB0_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status Flag	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status Flag	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status Flag	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*5	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status Flag*5	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: SOF interrupts are not generated. 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status Flag *5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status Flag*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB0_VBUS pin is high and 0 when the USB0_VBUS pin is low.
Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID flag, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 5. The status of the RESM, DVST, and CTRT flags are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.
Note 6. A change in the status indicated by the VBINT and RESM flags can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

CTSQ[2:0] Flags (Control Transfer Stage Flag)

When the host controller is selected, the read value is invalid.

VALID Flag (USB Request Reception Flag)

When the host controller is selected, the read value is invalid.

DVSQ[2:0] Flags (Device State Flag)

The DVSQ[2:0] flags are initialized by a USB bus reset.
When the host controller is selected, the read value is invalid.

BRDY Flag (Buffer Ready Interrupt Status Flag)

Indicates the BRDY interrupt status.
The USB sets the BRDY flag to 1 when at least one PIPEnBRDY flag (n = 0 to 9) is set to 1 among the PIPEnBRDY flags. These bits correspond to the BRDYENB.PIPEnBRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.
For the conditions for PIPEnBRDY status assertion, refer to section 30.3.3.1, BRDY Interrupt.
The USB sets the BRDY flag to 0 when 0 is written by software to all the PIPEnBRDY flags corresponding to the PIPEnBRDYE bits that have been set to 1.
The BRDY flag cannot be set to 0 even if 0 is written to this bit by software.

NRDY Flag (Buffer Not Ready Interrupt Status Flag)

The USB sets the NRDY flag to 1 when at least one PIPEnNRDY flag (n = 0 to 9) is set to 1 among the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPEnNRDY status assertion, refer to section 30.3.3.2, NRDY Interrupt.
The USB sets the NRDY flag to 0 when 0 is written by software to all the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits that have been set to 1.
The NRDY flag cannot be set to 0 even if 0 is written to this bit by software.

BEMP Flag (Buffer Empty Interrupt Status Flag)

The USB sets the BEMP flag to 1 when at least one PIPEnBEMP flag (n = 0 to 9) is set to 1 among the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).
For the conditions for PIPEnBEMP status assertion, refer to section 30.3.3.3, BEMP Interrupt.
The USB sets the BEMP flag to 0 when 0 is written by software to all the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits that have been set to 1.
The BEMP flag cannot be set to 0 even if 0 is written to this bit by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a change in the control transfer stage.
When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.
When the host controller is selected, the read value is invalid.

DVST Flag (Device State Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST flag to 1 on detecting a change in the device state.
When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.
When the host controller is selected, the read value is invalid.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) When the host controller is selected
The USB sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller is selected
The USB sets the SOFR flag to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)
The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Flag (Resume Interrupt Status Flag)

When the function controller is selected, the USB sets the RESM flag to 1 on detecting the falling edge of the signal on the USB0_DP pin in the suspended state (DVSQ[2:0] = 1xxb).
When the host controller is selected, the read value is invalid.

VBINT Flag (VBUS Interrupt Status Flag)

The USB sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB0_VBUS pin input value. The USB sets the VBSTS flag to indicate the USB0_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS flag until the same value is read three or more times, and eliminate chattering.

30.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USB0.INTSTS1 000A 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status Flag	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag *2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status Flag*2	0: OVRCR interrupts are not generated. 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. A change in the status indicated by the OVRCR or BCHG flag can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.
No interrupts other than those indicated by the BCHG and OVRCR flags can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

The INTSTS1 register is used to confirm the status of each interrupt when the host controller is selected.
The various status change interrupts indicated by the flags in the INTSTS1 register should be enabled only when the host controller is selected.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.
The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.
When the function controller is selected, the read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

Indicates the status of the setup transaction error interrupt when the host controller is selected.
The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.
Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

EOFERR Flag (EOF Error Detection Interrupt Status Flag)

Indicates the status of the EOFERR interrupt when the host controller is selected.
The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

ATTCH Flag (ATTCH Interrupt Status Flag)

Indicates the status of the ATTCH interrupt when the host controller is selected.
The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 µs, and sets the ATTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

When the function controller is selected, the read value is invalid.

DTCH Flag (USB Disconnection Detection Interrupt Status Flag)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.
The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB detects bus disconnection based on USB Specification 2.0.
After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

BCHG Flag (USB Bus Change Interrupt Status Flag)

Indicates the status of the USB bus change interrupt.
The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB sets the LNST[1:0] flags to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] flags until the same value is read three or more times, and eliminate chattering.
A change in the USB bus state can be detected even while the internal clock supply is stopped.
When the function controller is selected, the read value is invalid.

OVRCR Flag (Overcurrent Input Change Interrupt Status Flag)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.
The USB detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0_OVRCURA and USB0_OVRCURB pins, and sets the OVRCR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

30.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USB0.BRDYSTS 000A 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status Flag for Pipe 0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status Flag for Pipe 1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status Flag for Pipe 2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status Flag for Pipe 3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status Flag for Pipe 4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status Flag for Pipe 5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status Flag for Pipe 6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status Flag for Pipe 7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status Flag for Pipe 8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status Flag for Pipe 9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the flags in the BRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

30.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USB0.NRDYSTS 000A 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the NRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

30.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USB0.BEMPSTS 000A 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the BEMPSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

30.2.18 Frame Number Register (FRMNUM)

Address(es): USB0.FRMNUM 000A 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error Flag	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status Flag	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

FRNM[10:0] Flags (Frame Number Flag)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] flags at the SOF packet reception.

CRCE Flag (Receive Data Error Flag)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
The CRCE flag can be set to 0 by writing 0 to the CRCE flag by software. Here, 1 should be written to the other bits in FRMNUM.
On detecting a CRC error, the USB generates the internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
The OVRN flag can be set to 0 by writing 0 to the OVRN flag by software. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

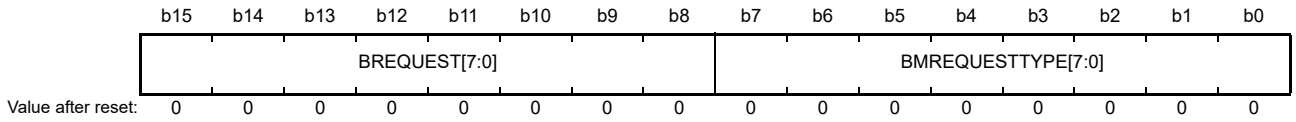
(2) When the function controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

30.2.19 USB Request Type Register (USBREQ)

Address(es): USB0.USBREQ 000A 0054h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBREQ register stores setup requests for control transfers.
When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.
When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.
The USBREQ register is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

30.2.20 USB Request Value Register (USBVAL)

Address(es): USB0.USBVAL 000A 0056h

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

When the function controller is selected, the value of wValue that has been received is stored in the USBVAL register.
When the host controller is selected, the value of wValue to be transmitted is set.
The USBVAL register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not modify the value of the USBVAL register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the USBVAL register has no effect.

30.2.21 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 000A 0058h

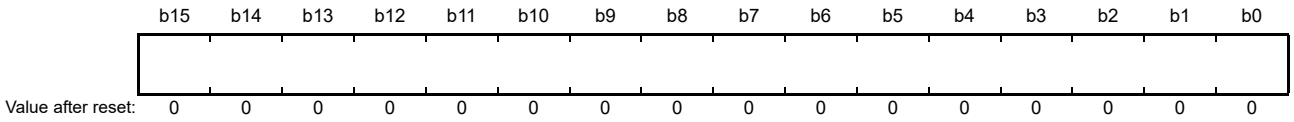
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The USBINDX register stores setup requests for control transfers.
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.
The USBINDX register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not modify the value of the USBINDX register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the USBINDX register has no effect.

30.2.22 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 000A 005Ah



The USBLENG register stores setup requests for control transfers.
When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.
The USBLENG register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not modify the value of the USBLENG register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the USBLENG register has no effect.

30.2.23 DCP Configuration Register (DCPCFG)

Address(es): USB0.DCPCFG 000A 005Ch

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while DCPCTR.PID[1:0] bits are 00b (NAK). Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller is selected, the DIR bit should be set to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe in the receiving direction.
When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

30.2.24 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USB0.DCPMAXP 000A 005Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID[1:0] bits are 00b (NAK) and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID[1:0] = 01b (BUF) while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

30.2.25 DCP Control Register (DCPCTR)

Address(es): USB0.DCPCTR 000A 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.
Note 2. Write 1 to the SQSET and SQCLR bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from 00b (NAK) to 01b (BUF) using the following procedure.

- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then write 01b (BUF response). After PID[1:0] have been set to 01b (BUF), the USB executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then set PID[1:0] bits to 01b (BUF). After PID[1:0] bits have been set to 01b (BUF), the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID[1:0] bits to 11b (STALL) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to 00b (NAK) on receiving the setup packet. Here, the USB sets the INTSTS0.VALID flag to 1. The setting of the PID[1:0] bits cannot be modified until the VALID flag is set to 0 by software.
- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 1xb (STALL) on detecting the control transfer sequence error.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.
The PID[1:0] bits are initialized by a USB bus reset.

CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to 01b (BUF), the USB completes the control transfer status stage.
During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
1 cannot be written to the CCPL bit by software while the INTSTS0.VALID flag is 1.
The CCPL bit is initialized by a USB bus reset.
When the host controller is selected, be sure to write 0 to the CCPL bit.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from 01b (BUF) to 00b (NAK).
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 30.3.4.1, Pipe Control Register Switching Procedures.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.
When the function controller is selected, the USB sets the SQMON flag to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.
When the function controller is selected, the USB does not reference the SQMON flag during the IN/OUT transaction of the status stage, and does not allow the SQMON flag to toggle upon normal completion.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.
Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.
Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.
After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.
The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.
Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to 00b (NAK). After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).
Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.
When the function controller is selected, be sure to write 0 to the SUREQ bit.

BSTS Flag (Buffer Status Flag)

Indicates whether DCP FIFO buffer access is enabled or disabled.
The meaning of the BSTS flag depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS flag indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS flag indicates whether the data to be transmitted can be written to the buffer.

30.2.26 Pipe Window Select Register (PIPESEL)

Address(es): USB0.PIPESEL 000A 0064h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Pipes 1 to 9 should be set using registers PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using registers PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to registers PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from registers PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in registers PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

30.2.27 Pipe Configuration Register (PIPECFG)

Address(es): USB0.PIPECFG 000A 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000b means an unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe assignment continued at the end of transfer 1: Pipe assignment disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 2. Modify the BFRE, DBLB, and DIR bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), checking the PIPEnCTR.PBUSY flag through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID[1:0] bits and CURPIPE[3:0] bits are in the state described in the above note 2.

The PIPECFG register specifies the transfer type, buffer memory access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.
Setting 0000b means an unused pipe.
Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 0000b can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.
When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify the PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe is pipe 1 to 5 in the receiving direction.
When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.
The DBLB bit is valid when the selected pipe is pipe 1 to 5.

BFRE Bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.
When the BRDY interrupt is generated with the above conditions, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 30.3.3.1, BRDY Interrupt.

TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).
Before setting the PID[1:0] bits to 01b (BUF) for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

30.2.28 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USB0.PIPEMAXP 000A 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• Pipes 1 and 2: 1 byte (001h) to 256 bytes (100h) • Pipes 3 to 5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) • Pipes 6 to 9: 1 byte (001h) to 64 bytes (040h)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

Note 2. Modify the MXPS[8:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEMAXP register specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] Bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of pipes 1 and 2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

30.2.29 Pipe Cycle Control Register (PIPEPERI)

Address(es): USB0.PIPEPERI 000A 006Eh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEPERI register selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set the PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are invalid for pipes 3 to 5; set the IITV[2:0] bits to 000b for pipes 3 to 5.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

30.2.30 Pipe n Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): USB0.PIPE1CTR 000A 0070h, USB0.PIPE2CTR 000A 0072h, USB0.PIPE3CTR 000A 0074h, USB0.PIPE4CTR 000A 0076h, USB0.PIPE5CTR 000A 0078h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Write disabled 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Write disabled 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status Flag	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read.
Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEnCTR register can be set regardless of the pipe selection in the PIPESEL register.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 30.6 and Table 30.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 00b (NAK) on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 30.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.
To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.
Table 30.8 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.
When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.
When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.
(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB transmits a zero-length packet in response to the IN token.
The USB updates (allows toggling of) the sequence toggle bit (data PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).
In this case, the USB does not generate the BRDY or BEMP interrupt.
(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB returns NAK in response to the OUT token and generates the NRDY interrupt.
For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.
When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.
When the host controller is selected, be sure to set the ATREPM bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.
When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM flag to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.
The INBUFM flag indicates the same value as the BSTS flag when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 30.9.

Table 30.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 30.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR bit = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

Table 30.8 Information Cleared by USB by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	When the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	FIFO buffer toggle control	When the PIPECFG.DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 30.9 Operation of BSTS Flag

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Flag Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

- PIPEnCTR (n = 6 to 9)

Address(es): USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh, USB0.PIPE9CTR 000A 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Sequence Toggle Bit Confirmation Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Invalid 1: Specifies DATA0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.
Note 2. Write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while the PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 30.6 and Table 30.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits has been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

SQMON Flag (Sequence Toggle Bit Confirmation Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 30.10 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 30.9.

Table 30.10 Information Cleared by USB by Setting the ACLRM Bit = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

30.2.31 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USB0.PIPE1TRE 000A 0090h, USB0.PIPE2TRE 000A 0094h, USB0.PIPE3TRE 000A 0098h, USB0.PIPE4TRE 000A 009Ch, USB0.PIPE5TRE 000A 00A0h

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Modify each bit in the PIPEnTRE register while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN register through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the PIPEnTRN register.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID[1:0] bits to 00b (NAK) for the corresponding pipe on having received the number of packets equal to the setting of the PIPEnTRN register.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the PIPEnTRN register and then reading the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the PIPEnTRN register before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

30.2.32 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USB0.PIPE1TRN 000A 0092h, USB0.PIPE2TRN 000A 0096h, USB0.PIPE3TRN 000A 009Ah, USB0.PIPE4TRN 000A 009Eh, USB0.PIPE5TRN 000A 00A2h

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

The PIPEnTRN register retains the setting by a USB bus reset.

- When written to:
  This register specifies the total of packets (number of transactions) to be received in corresponding pipe.
- When read from:
  This register indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0.
  This register indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.

The USB increments the value of the PIPEnTRN register by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB sets the value of the PIPEnTRN register to 0000h when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the PIPEnTRN register to 0000h.
When the transaction counter is not used, set the PIPEnTRN register to 0000h.
Setting the number of transactions to be transferred to the PIPEnTRN register is only enabled when the PIPEnTRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPEnTRE.TRENB bit to 1.

30.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): USB0.DEVADD0 000A 00D0h, USB0.DEVADD1 000A 00D2h, USB0.DEVADD2 000A 00D4h, USB0.DEVADD3 000A 00D6h, USB0.DEVADD4 000A 00D8h, USB0.DEVADD5 000A 00DAh

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device which is the communication target for pipes 0 to 9.

When the host controller is selected, the bits in the DEVADDn register should be set before starting communication using each pipe.

The bits in the DEVADDn register should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- The DEVADDn register is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 01b (BUF) for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.

Set these bits to 10b when a full-speed device is connected via the HUB.

When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.

When the function controller is selected, set these bits to 00b.

30.3 Operation

30.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

30.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

30.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pull-up resistor of the D+line and pull-down resistors of the D+ and D– lines are disabled (the SYSCFG.DPRPU bit = 0 and DRPD bit = 0).

30.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected, confirm that connection to the USB host is made, then set the DPRPU bit to 1 and pull up the D+ line (during full-speed).
When the DPRPU bit is set to 0 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.
When the host controller is selected, set the DRPD bit and pull down the D+ and D– lines.

Table 30.11 USB Data Bus Resistor Control

SYSCFG register				
DRPD bit	**DPRPU bit**	**D–**	**D+**	**Function**
0	0	Open	Open	Not in use
0	1	Open	Pull-up	When operating as the function controller (in full-speed)
1	0	Pull-down	Pull-down	When operating as the host controller
1	1	—	—	Setting prohibited

30.3.1.4 Example of USB External Connection Circuit

Figure 30.2 shows an example of connection between OTG device and USB connector in the self-powered state.
The USB controls the signals for enabling a pull-up resistor for the D+ signal and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected and the DPRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

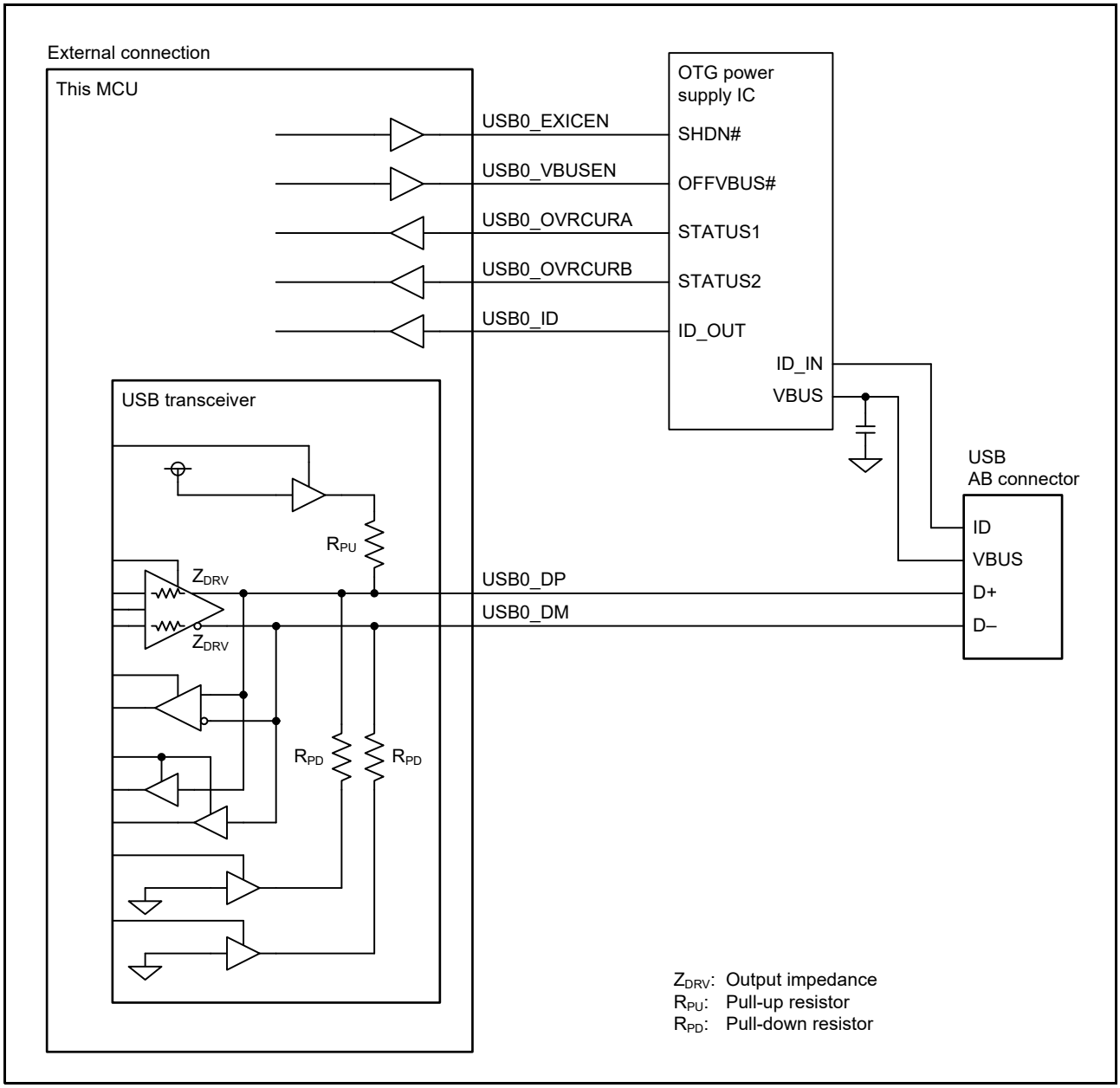


Figure 30.2 Example of Connection between Self-Powered OTG Device and USB Connector

Figure 30.3 shows an example of connection between function controller and USB connector in self-powered state.

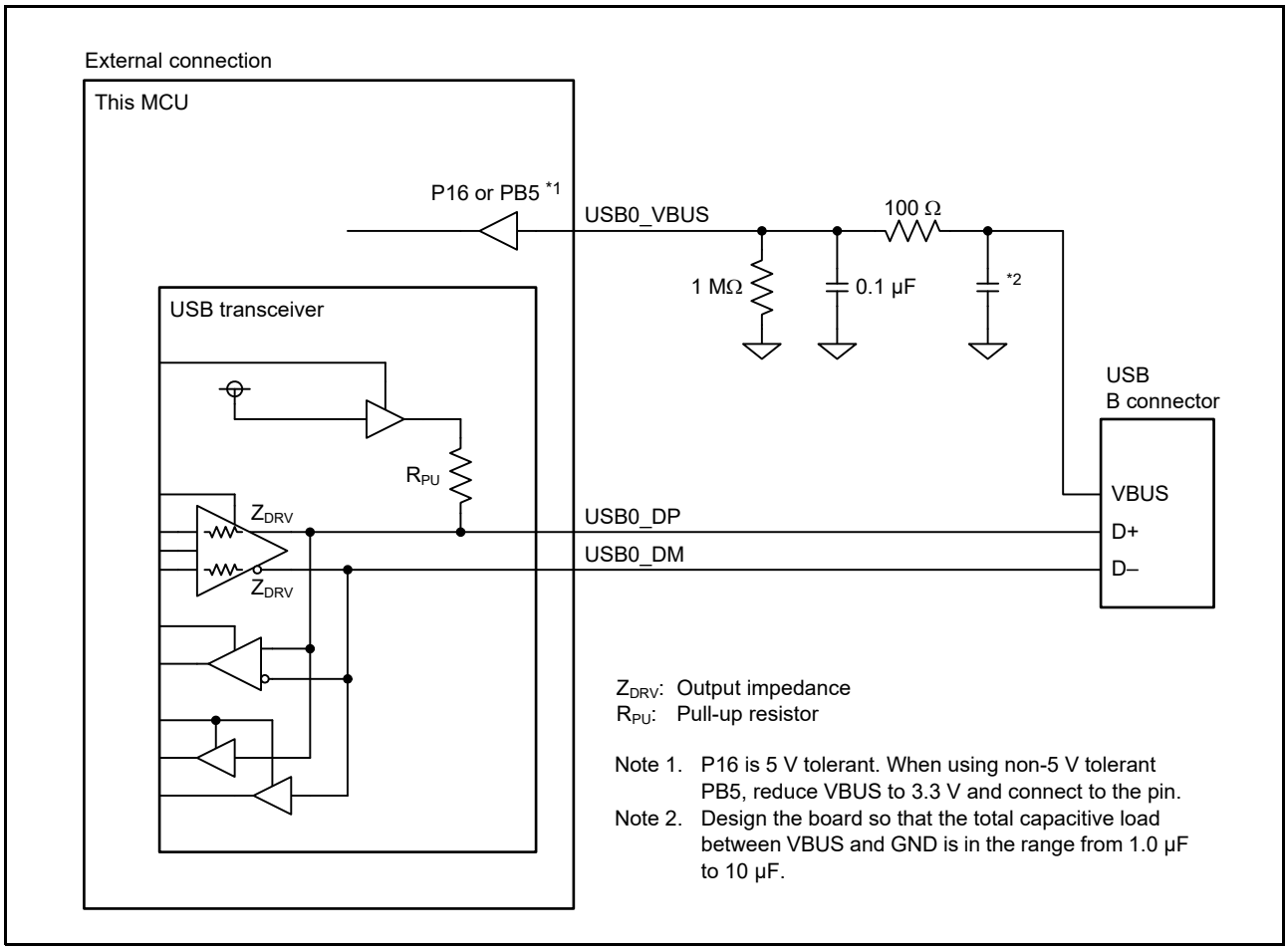


Figure 30.3 Example of Connection between Self-Powered Function and USB Connector

Figure 30.4 shows an example of connection between host controller and USB connector.

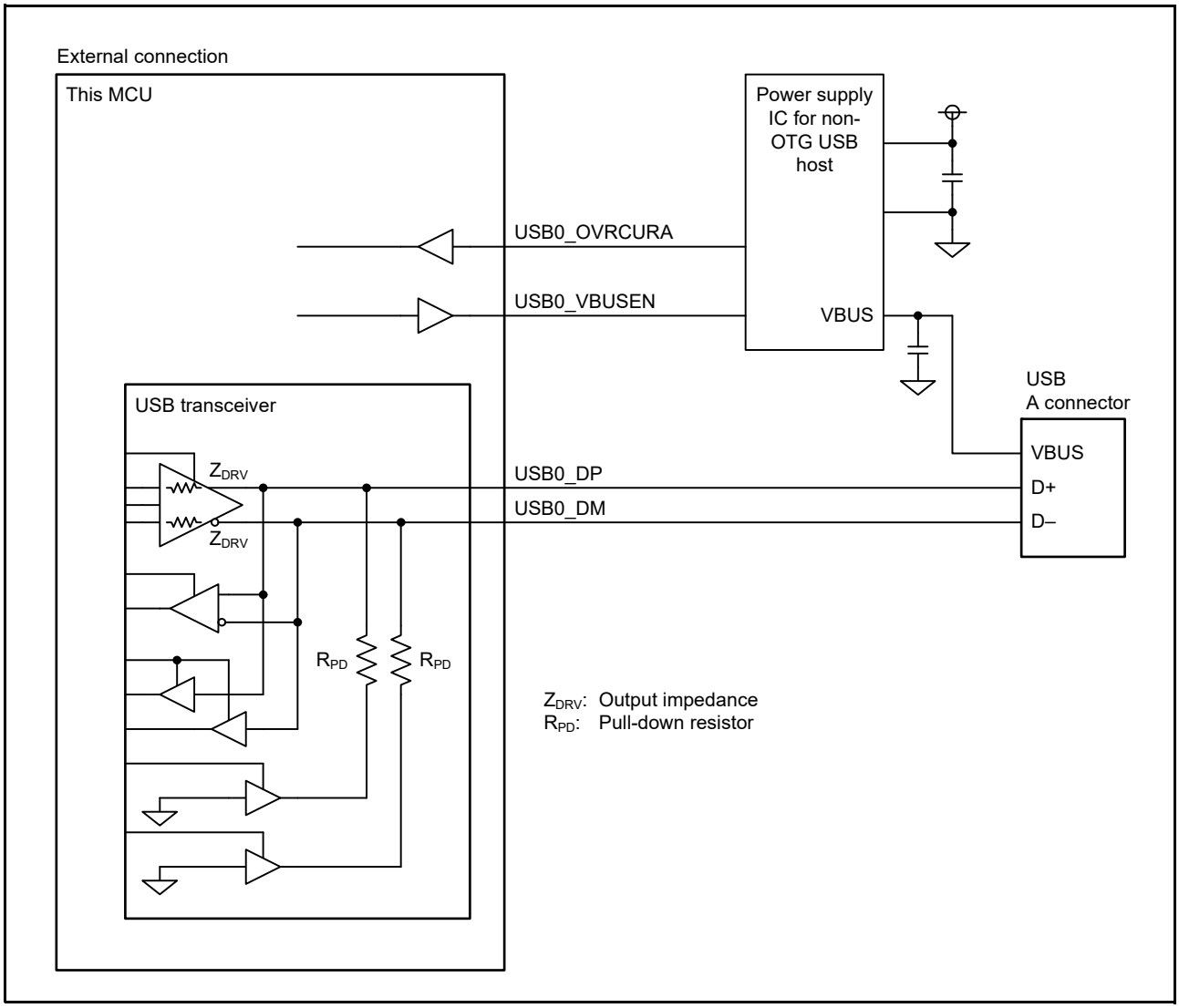


Figure 30.4 Example of Connection between Host and USB Connector

Figure 30.5 shows an example of connection between function controller and USB connector in bus powered state.

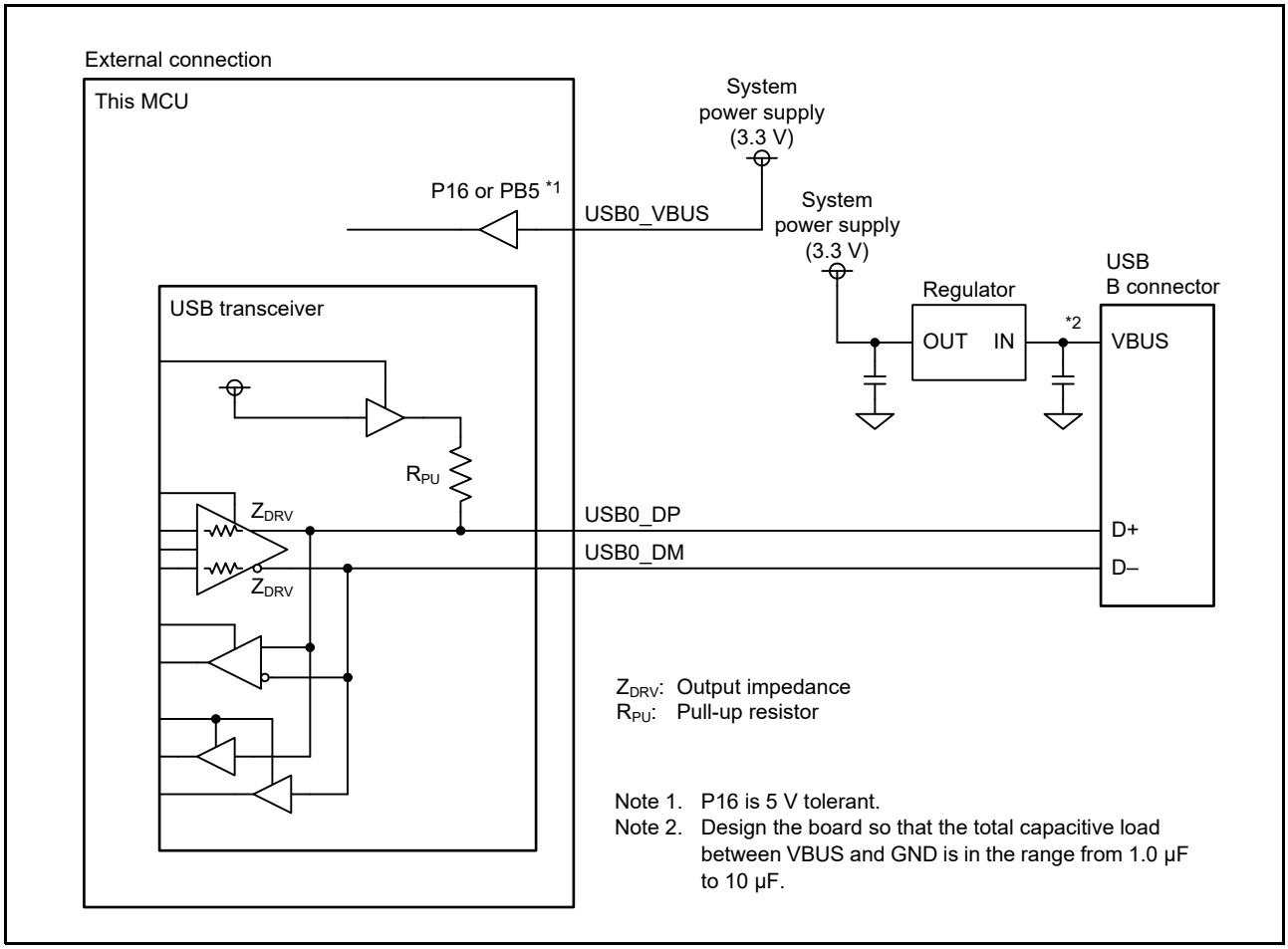


Figure 30.5 Example of Connection between Bus-Powered Function and USB Connector

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

30.3.2 Interrupt Sources

Table 30.12 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

Table 30.12 Interrupt Sources

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
VBINT	VBUS interrupt	• When a change in the state of the USB0_VBUS input pin has been detected (low to high or high to low)	Host/function*1	INTSTS0. VBSTS
RESM	Resume interrupt	• When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	[Host controller is selected] • When an SOF packet with a different frame number has been transmitted [Function controller is selected] • When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	• When a device state transition has been detected (any of the following conditions) A USB bus reset detected Suspended state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer (any of the following conditions) Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	BEMPSTS. PIPEnBEMP
NRDY	Buffer not ready interrupt	[Host controller is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) • When an overrun/underrun occurred during isochronous transfer [Function controller is selected] • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are 01b (BUF) • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an overrun/underrun occurred during data reception in isochronous transfer	Host/function	NRDYSTS. PIPEnNRDY
BRDY	Buffer ready interrupt	• When the buffer has become ready (reading or writing is enabled)	Host/function	BRDYSTS. PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low)	Host	INTSTS1. OVRCR
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	SYSSTS0. LNST[1:0]
DTCH	Disconnection detection during full-speed operation	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0. RHST[2:0]
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 30.6 shows the circuits related to the interrupts in the USB.

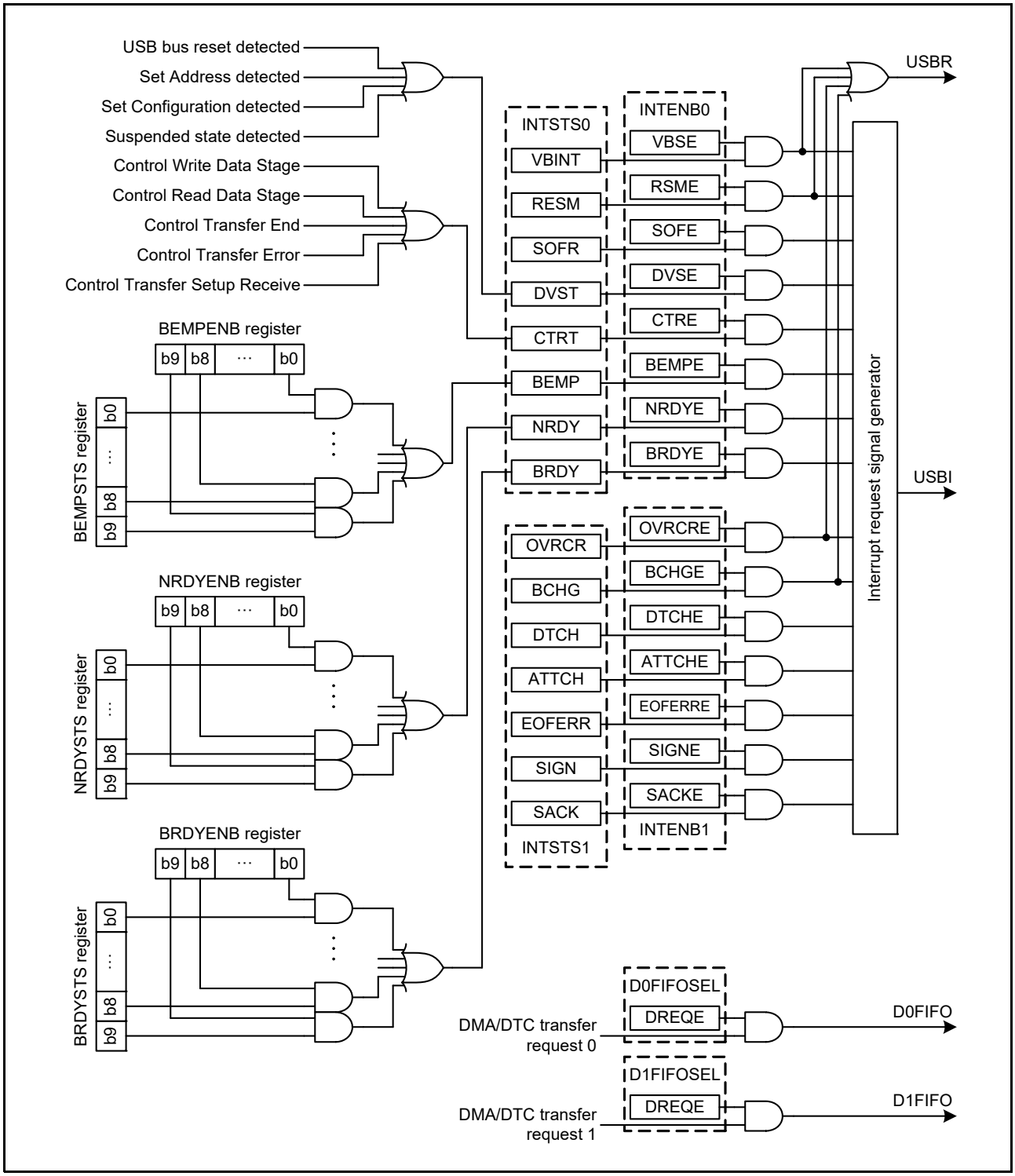


Figure 30.6 Circuits Related to Interrupts in USB

Table 30.13 shows the interrupts generated in the USB0.

Table 30.13 USB Interrupts

Interrupt Name	Interrupt Status Flag	DTC Activation	DMAC Activation
D0FIFO	DMA/DTC transfer request 0	Possible	Possible
D1FIFO	DMA/DTC transfer request 1	Possible	Possible
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, and setup error	Not possible	Not possible
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible

30.3.3 Interrupt Descriptions

30.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY flag corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

(b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
  No request trigger is generated for the transaction in which data PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the pipe n transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN register are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[8:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.
With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status flags (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEnBRDY flag cannot be set to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Figure 30.7 shows the timing of BRDY interrupt generation.

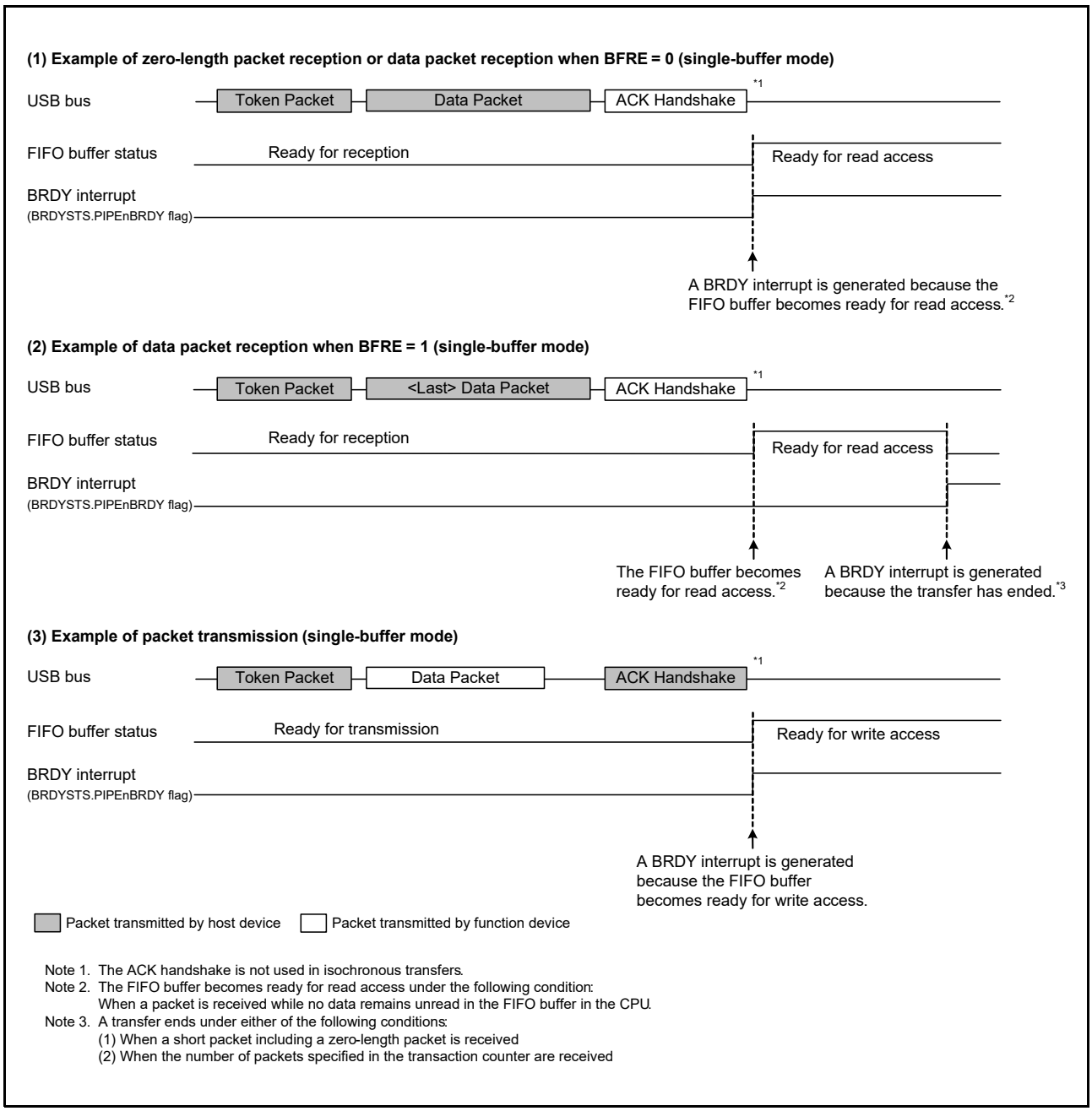


Figure 30.7 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting. Table 30.14 shows the condition for clearing the BRDY flag.

Table 30.14 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing BRDY Flag
0	The USB sets the BRDY flag to 0 when all bits in the BRDYSTS register have been set to 0 by software.
1	The USB sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

30.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding NRDYSTS.PIPEnNRDY flag to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB sets the INTSTS0.NRDY flag to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

(1) When Host Controller is Selected

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY flag corresponding to the pipe and the OVRN flag to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE flag to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(2) When Function Controller is Selected

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY flag to 1.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY flag to 1 and OVRN flag to 1.
  For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY flag to 1.
  However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY flag to 1.

Figure 30.8 shows the timing of NRDY interrupt generation when the function controller is selected.

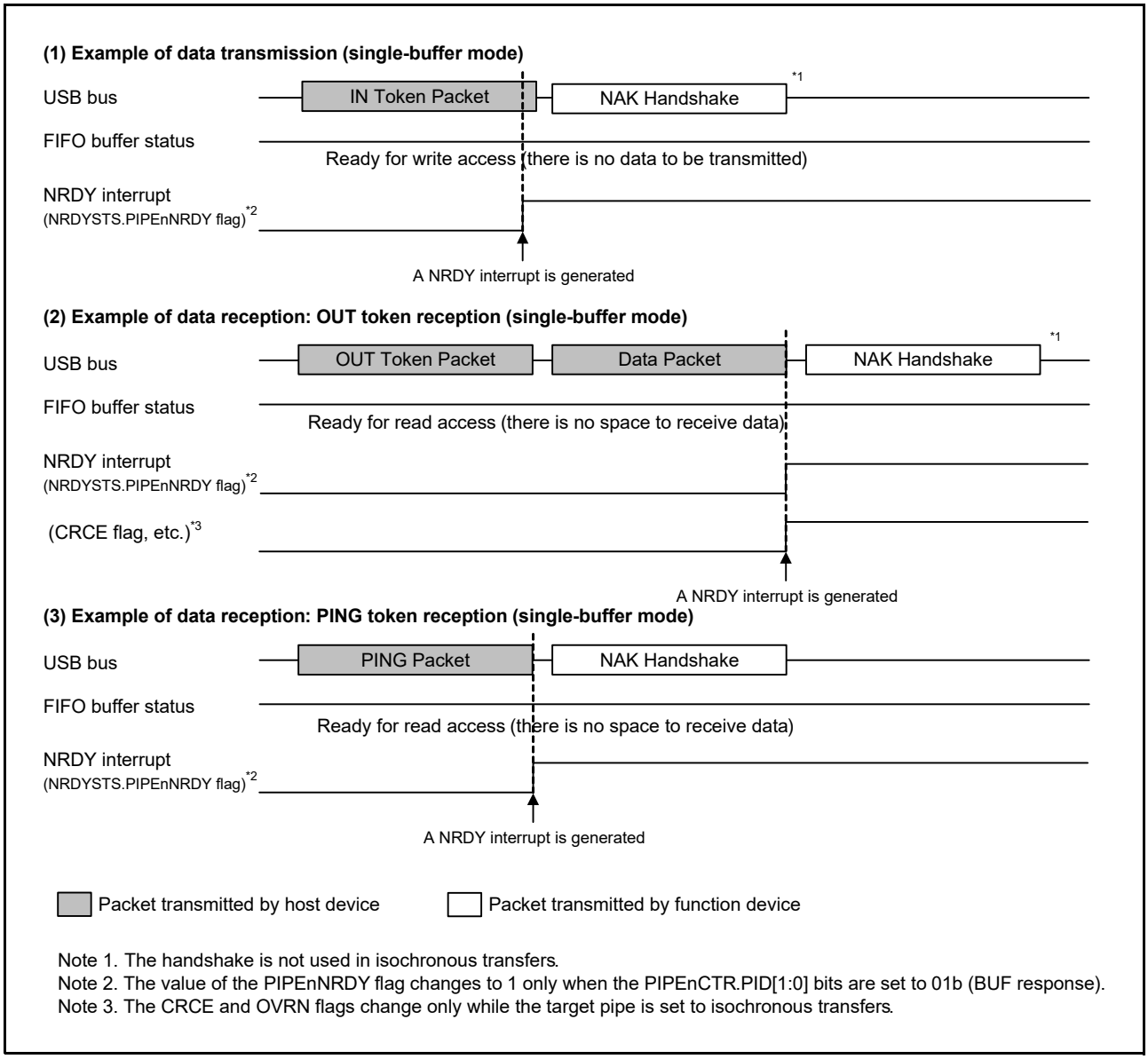


Figure 30.8 Timing of NRDY Interrupt Generation (When Function Controller is Selected)

30.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP flag to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.
In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
  Writing 0 to the BEMPSTS.PIPEnBEMP flag clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP flag has no effect.

Figure 30.9 shows the timing of BEMP interrupt generation when the function controller is selected.

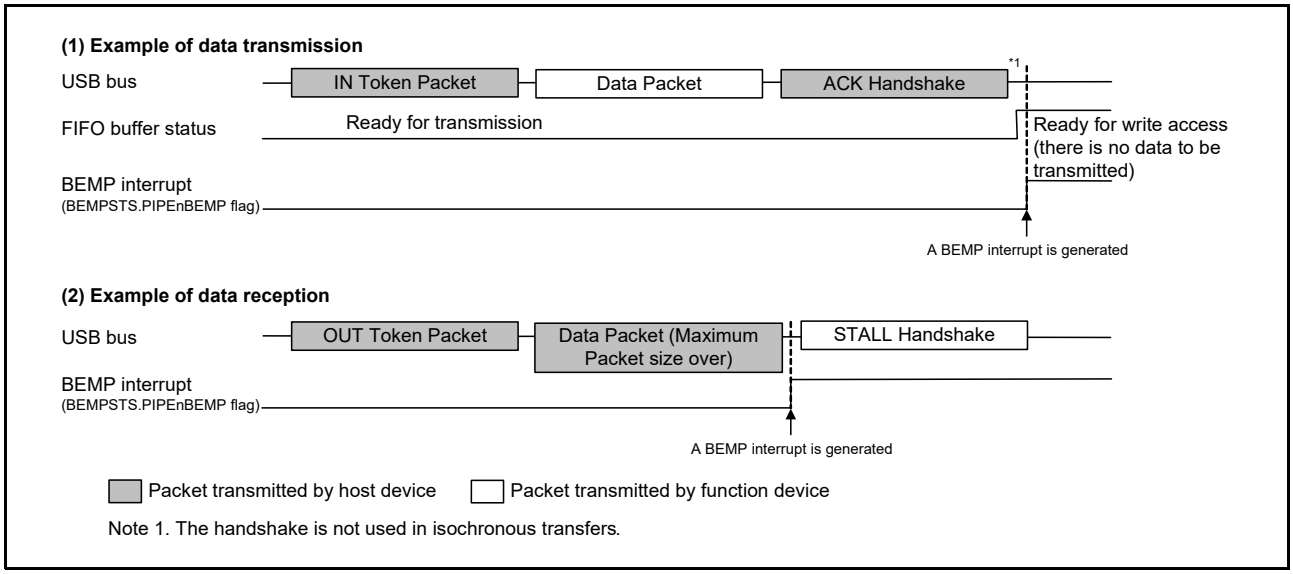


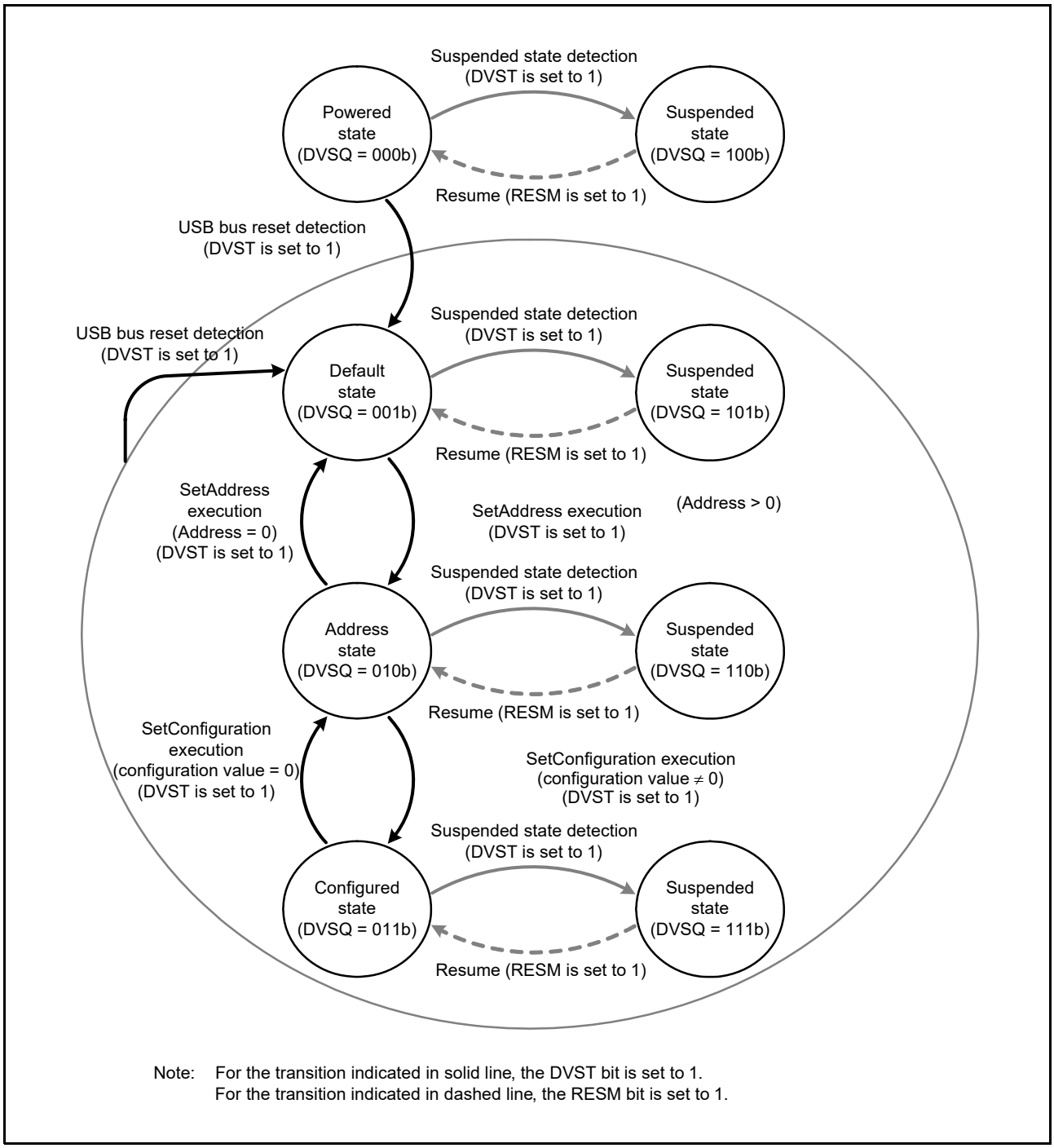
Figure 30.9 Timing of BEMP Interrupt Generation (When Function Controller is Selected)

30.3.3.4 Device State Transition Interrupt

Figure 30.10 is a diagram of device state transitions in the USB. The USB controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.



Note: For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

Figure 30.10 Device State Transitions

30.3.3.5 Control Transfer Stage Transition Interrupt

Figure 30.11 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the INTSTS0.CTSQ[2:0] flags.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with PID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with PID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT flag = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

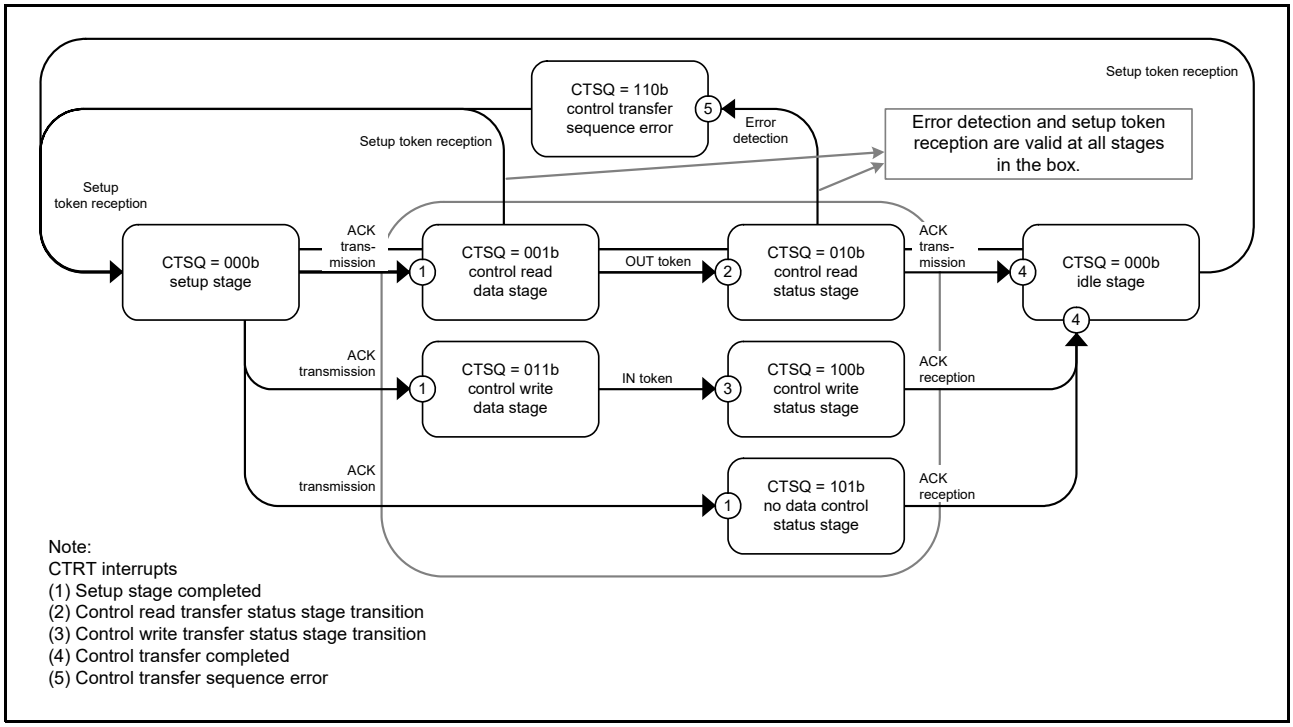


Figure 30.11 Control Transfer Stage Transitions

30.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

30.3.3.7 VBUS Interrupt

When the USB0_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB0_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USB0_VBUS pin level.

30.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.
When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

30.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB0_OVRCURA or USB0_OVRCURB pin level has changed. The levels of the USB0_OVRCURA and USB0_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

30.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

30.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.
After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

30.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

30.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

30.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

30.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.
After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

30.3.4 Pipe Control

Table 30.15 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.
Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 30.15 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE[1:0]	Specifies the transfer type	Pipes 1 to 9: Can be set
	BFRE	Selects the BRDY interrupt mode	Pipes 1 to 5: Can be set
	DBLB	Selects double buffer mode	Pipes 1 to 5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Can be set (only when bulk transfer has been selected) Pipes 3 to 5: Can be set
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selects a device	Referenced only when the host controller is selected.
	MXPS[8:0]	Maximum packet size	Compliant with USB Specification 2.0.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 9: Cannot be set
	IITV[2:0]	Interval counter	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 5: Cannot be set Pipes 6 to 9: Can be set (only when the host controller has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	ATREPM	Auto response mode	Pipes 1 to 5: Can be set Can be set only when the function controller has been selected.
	ACLRM	Auto buffer clear	Pipes 1 to 9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	
	PID[1:0]	Response PID	Refer to section 30.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Can be set
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Can be set
PIPEnTRN	—	Transaction counter	Pipes 1 to 5: Can be set

30.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] = 00b (NAK)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] = 01b (BUF)).

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR register
- Bits in registers PIPECFG, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID[1:0] = 01b (BUF)) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to 00b (NAK).
3. Wait until the corresponding PBUSY flag is set to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in the DCPCFG and DCPMAXP register
- Bits in registers PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

30.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

30.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

30.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined by USB Specification 2.0. For pipes 6 to 9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID[1:0] = 01b (BUF)).

- DCP: Set 8, 16, 32, or 64.
- Pipes 1 to 5: Set 8, 16, 32, or 64 when using bulk transfer.
- Pipes 1 and 2: Set a value between 1 and 256 when using isochronous transfer.
- Pipes 6 to 9: Set a value between 1 and 64.

30.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID[1:0] = 01b (BUF), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

30.3.4.6 Response PID

The PID[1:0] bits in the DCPCTR and PIPEnCTR registers are used to set the response PID for each pipe.
The following shows the USB operation with various response PID settings:

(1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
  For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

(2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and issuing of tokens is automatically stopped:
  When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.
  (For details, refer to section 30.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
  - If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and issuing of tokens is automatically stopped:
  When STALL is received in response to the transmitted token.
  When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and NAK is returned in response to transactions:
  When the SETUP token is received normally (DCP only).
  If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and STALL is returned in response to transactions:
  When a maximum packet size exceeded error is detected in the received data packet.
  When a control transfer sequence error has been detected (DCP only).

30.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON flag in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The DCPCTR.SQCLR bit and the PIPEnCTR.SQSET bit can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

30.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (response PID = NAK) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

30.3.4.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

30.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

30.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

30.3.5 FIFO Buffer Memory

30.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

(1) Buffer Status

Table 30.16 and Table 30.17 show the buffer status in the USB. The buffer memory status can be confirmed using the DCPCTR.BSTS flag and the PIPEnCTR.INBUFM flag. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM flag is valid for pipe 0 to pipe 5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS flag to monitor the buffer memory status on the CPU side and the INBUFM flag to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DMAC/DTC is slow, software can use the INBUFM flag to confirm the end of transmission.

Table 30.16 Buffer Status Indicated by the BSTS Flag

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 30.17 Buffer Status Indicated by the INBUFM Flag

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

30.3.5.2 FIFO Buffer Clearing

Table 30.18 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.
Either a single or double buffer configuration can be selected for pipes 1 to 5, using the PIPECFG.DBLB bit.

Table 30.18 List of Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.
If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.
An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

30.3.5.3 FIFO Port Functions

Table 30.19 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.
In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] flags = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] flags in the port control register.

Table 30.19 FIFO Port Function Settings

Register	Bit	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[8:0] read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA/DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE[3:0]	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN[8:0]	Checks the length of receive data.

(1) FIFO Port Selection

Table 30.20 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY flag in a port control register = 1 is checked.
In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

Table 30.20 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.
If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte.
If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.
To access the FIFO port, the FRDY flag in the port control register = 1 should be checked after selecting a pipe.

30.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.
The unit of transfer to the FIFO port should be selected using the DnFIFOSEL.MBW bit and the pipe targeted for the DMA transfer should be selected using the DnFIFOSEL.CURPIPE[3:0] bits. The selected pipe should not be changed during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USB automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.
Table 30.21 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 30.21, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.
The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 30.21 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary
Zero-length packet reception	Clearing is necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Normal short packet reception	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Transaction count end	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary

30.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

30.3.6.1 Control Transfers When the Host Controller is Selected

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in the DEVADD2 register; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in the DEVADD5 register.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK flag in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON flag.

(2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID[1:0] bits = 01b (BUF). Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.
For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

30.3.6.2 Control Transfers When the Function Controller is Selected

(1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.
When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID flag to 1.
- Set the DCPCTR.PID[1:0] bits to 00b (NAK).
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID flag = 0. In the VALID flag = 1 state, PID[1:0] = 01b (BUF) cannot be set, and the data stage cannot be terminated.
Using the function of the VALID flag, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, refer to Figure 30.11.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.
If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF).
After the above settings have been made, the USB automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
  A zero-length packet is transmitted and an ACK response is received from the USB host.

(4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

30.3.7 Bulk Transfers (Pipes 1 to 5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 30.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1)
- Transaction count function
  (PIPEnTRE.TRENB and TRCLR bits and PIPEnTRN register: refer to section 30.3.4.5, Transaction Counter (For Pipes 1 to 5 in Reading Direction))
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 30.3.4.8, Response PID = NAK Function)
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 30.3.4.9, Auto Response Mode)

30.3.8 Interrupt Transfers (Pipes 6 to 9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

30.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL).
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

30.3.9 Isochronous Transfers (Pipes 1 and 2)

The USB has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

30.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 30.22 and Table 30.23 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

(a) PID errors

- If the PID of the received packet is illegal.

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

(d) Overrun and underrun errors

- When the host controller is selected
  When the buffer memory is full at the token sending timing in the IN (receiving) direction.
  When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
  When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
  When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

(e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

Table 30.22 Error Detection When a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN flag to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected.

Table 30.23 Error Detection When a Data Packet is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to 1xb (STALL) in both cases when the host controller is selected and the function controller is selected.

30.3.9.2 Data PID

When the function controller is selected, the USB operates as follows in response to the received PID.
IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

30.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 30.24 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 30.24 Interval Counter Function When the Function Controller is Selected

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the $2^{IITV[2:0]}$ frames.

(1) Counter Initialization When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token while the PID[1:0] bits are 01b (BUF).
2. An SOF is received after reception of data of an OUT token while the PID[1:0] bits are 01b (BUF).

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL)
  The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
  The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control When the Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF) by software.

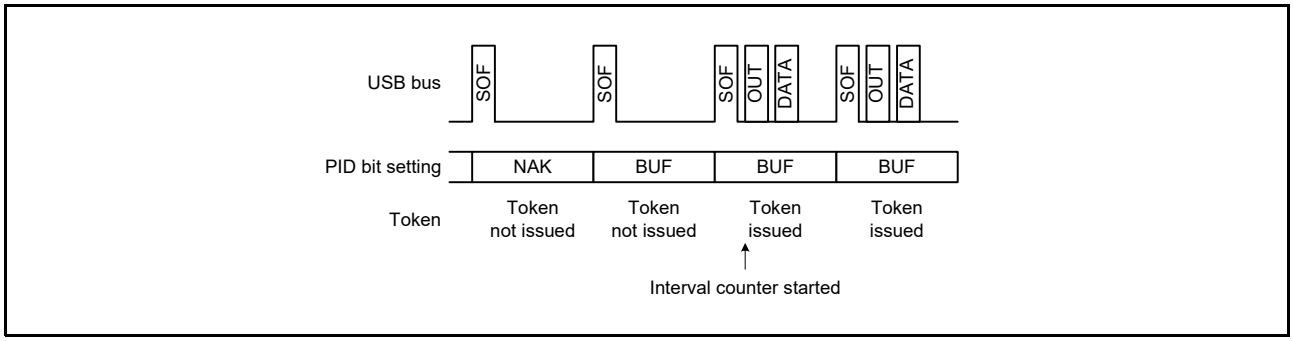


Figure 30.12 Token Issuance When IITV[2:0] = 000b

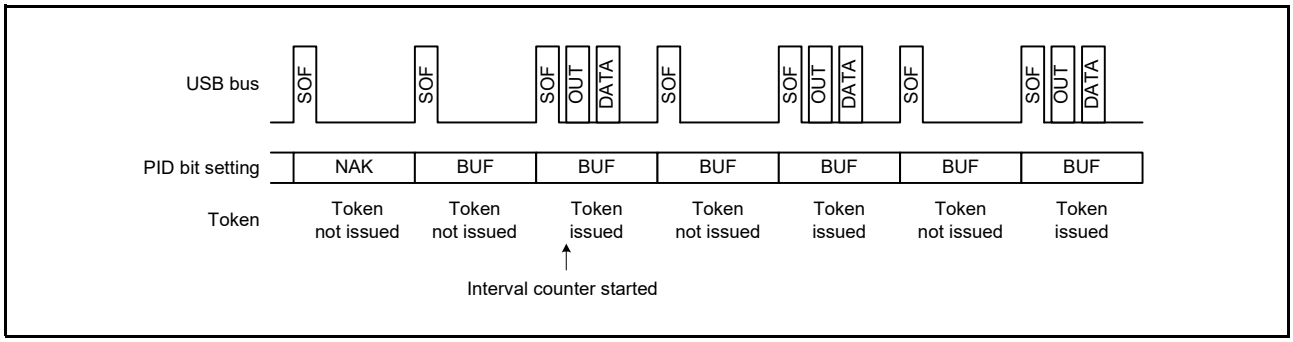


Figure 30.13 Token Issuance When IITV[2:0] = 001b

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USB sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
  (The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.
The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV[2:0] bits are set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK) by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.
The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] = 000b: The interval counting starts when software has set the PID[1:0] bits for the selected pipe to 01b (BUF).

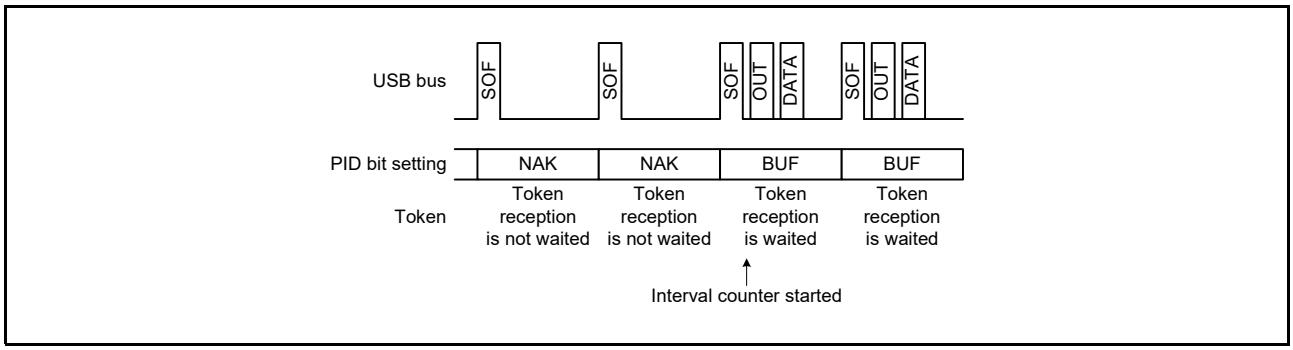


Figure 30.14 Relationship between Frames and Expected Token Reception When IITV[2:0] = 000b

- When the IITV[2:0] ≠ 000b: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF).

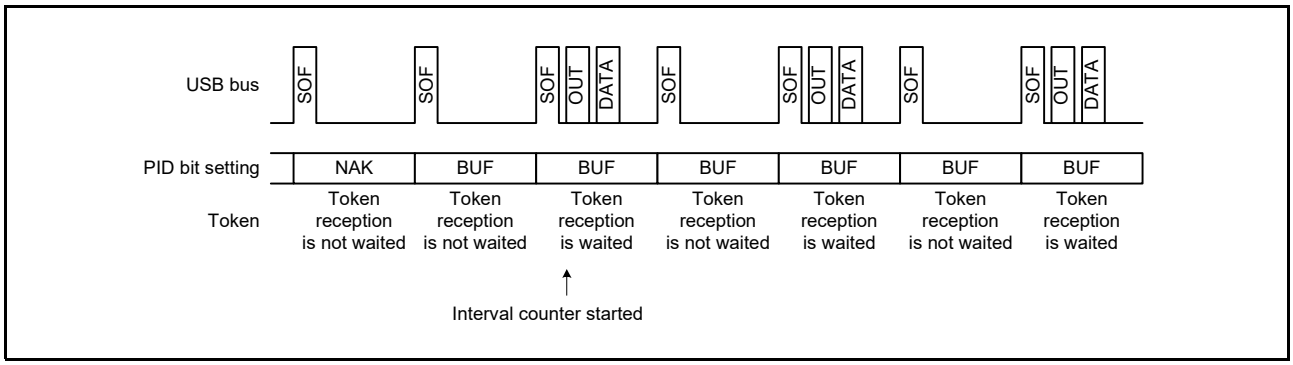


Figure 30.15 Relationship between Frames and Expected Token Reception When IITV[2:0] ≠ 000b

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.
The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 30.16 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV[2:0] = 000b (every frame) has been set.

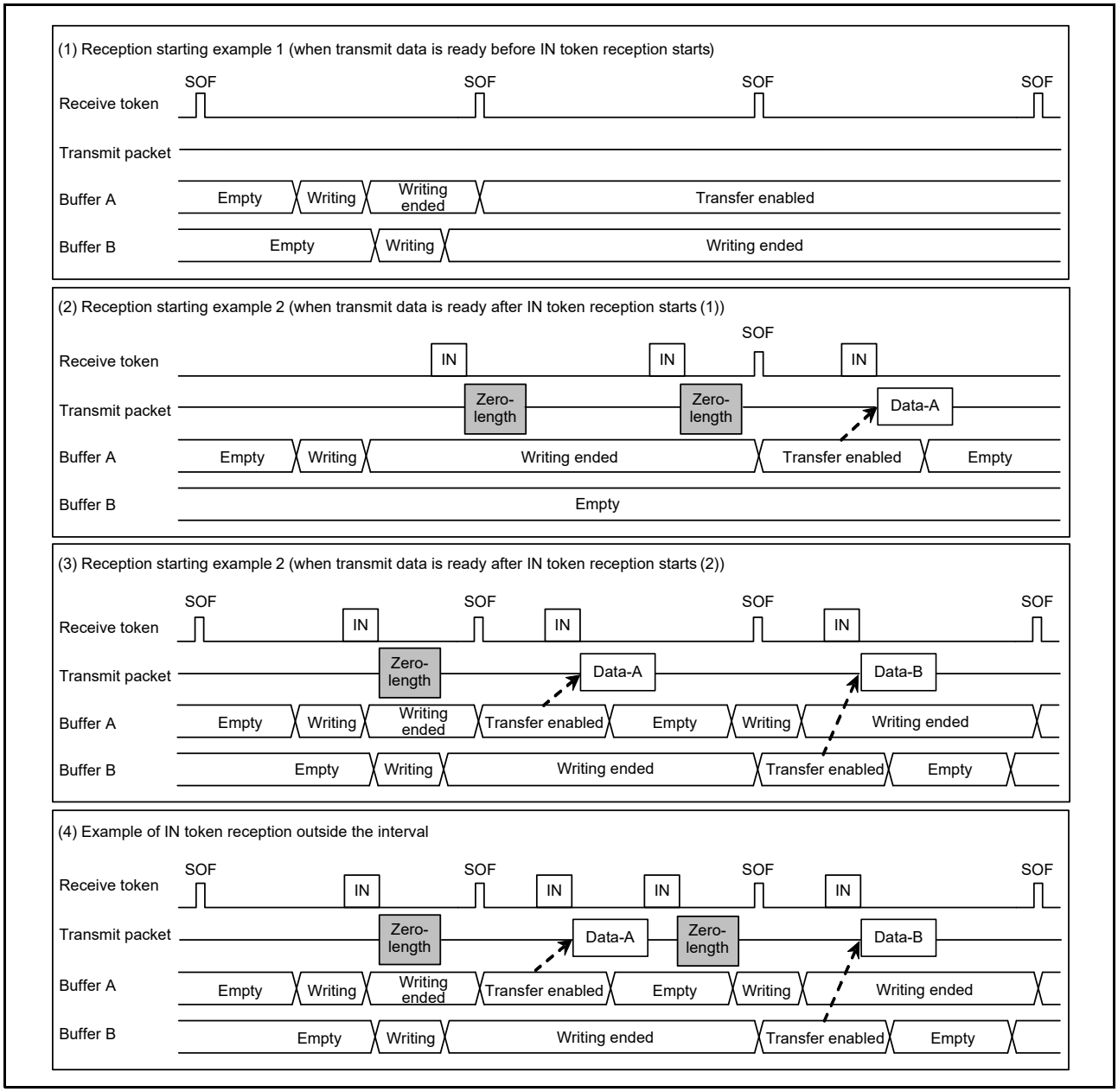


Figure 30.16 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.
If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.
The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV[2:0] = 000b
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV[2:0] ≠ 000b
  The buffer flush operation is carried out after the first successful transaction.

Figure 30.17 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

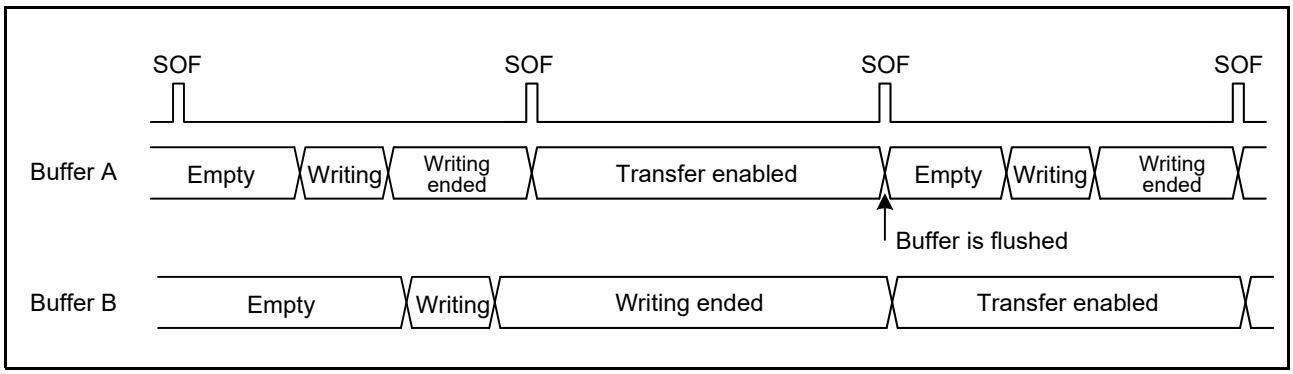


Figure 30.17 Example of Buffer Flush Operation

Figure 30.18 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by ① in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The FRMNUM.OVRN flag should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

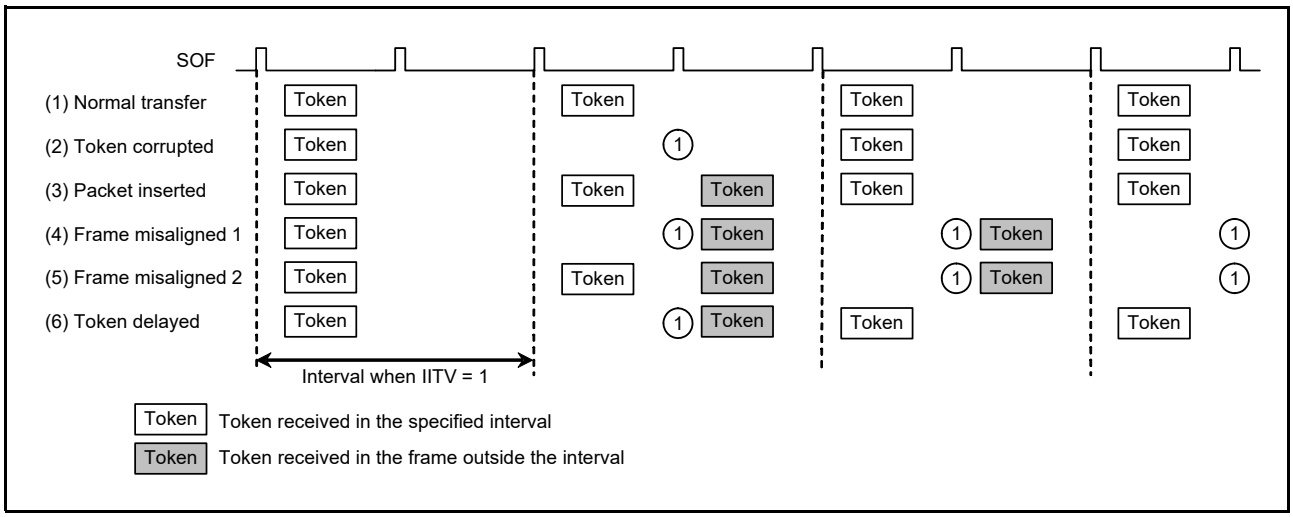


Figure 30.18 Example of Interval Error Occurrence When IITV[2:0] = 001b

30.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] flags are not updated.

30.3.11 Pipe Schedule

30.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 30.25.

Table 30.25 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID[1:0]**	**IITV[0]**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

30.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers
   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

30.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.
Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

30.4 Usage Notes

30.4.1 Setting the Module-Stop Function

Operation of the USB module can be disabled or enabled by setting a bit in the module stop control register B (MSTPCRB). The USB is initially disabled after a reset. Registers in the USB only become accessible after it has been released from the module-stop state. For details, refer to section 11, Low Power Consumption.

31. Serial Communications Interface (SCIk, SCIh)

This MCU has four independent serial communications interface (SCI) channels. The SCI consists of the SCIk module (SCI1, SCI5, and SCI6) and the SCIh module (SCI12).
The SCIk module (SCI1, SCI5, and SCI6) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I^2C-bus interfaces when configured for single-master systems.
The SCIh module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.
In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

Table 31.1 lists the specifications of the SCIk module, Table 31.2 lists the specifications of the SCIh module, and Table 31.3 lists the specifications of the individual SCI channels.
Figure 31.1 shows the block diagram of SCI1, Figure 31.2 shows the block diagram of SCI5 and SCI6, and Figure 31.3 shows the block diagram of SCI12 (SCIh).

Table 31.1 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 31.4 to Table 31.6.
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 31.1 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 31.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function (supported by SCI5 only)		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Note 1. In simple I^2C mode, only MSB first is available.

Table 31.2 SCIh Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I²C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 31.4 to Table 31.7.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I²C mode)
Low power consumption function		Module stop state can be set.
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.
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 31.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.

Table 31.2 SCIh Specifications (2/2)

Item		Description
Extended serial mode	Start Frame transmission	• Output of a low level as the Break Field over a specified width and generation of interrupts on completion • Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	• Detection of the Break Field low width and generation of an interrupt on detection • Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. • A priority interrupt bit can be set in Control Field 1. • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Function for measuring bit rates
	I/O control function	• Selectable polarity for TXDX12 and RXDX12 signals • Selection of a digital filter for the RXDX12 signal • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12
	Timer function	• Usable as a reload timer
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

Table 31.3 Functions of SCI Channels

Item	SCI1	SCI5	SCI6	SCI12
Asynchronous mode	Available	Available	Available	Available
Clock synchronous mode	Available	Available	Available	Available
Smart card interface mode	Available	Available	Available	Available
Simple I^2C mode	Available	Available	Available	Available
Simple SPI mode	Available	Available	Available	Available
Data match detection	Available	Available	Available	Not available
Extended serial mode	Not available	Not available	Not available	Available
TMR clock input	Not available	Available	Available	Available
Event link function	Not available	Available	Not available	Not available

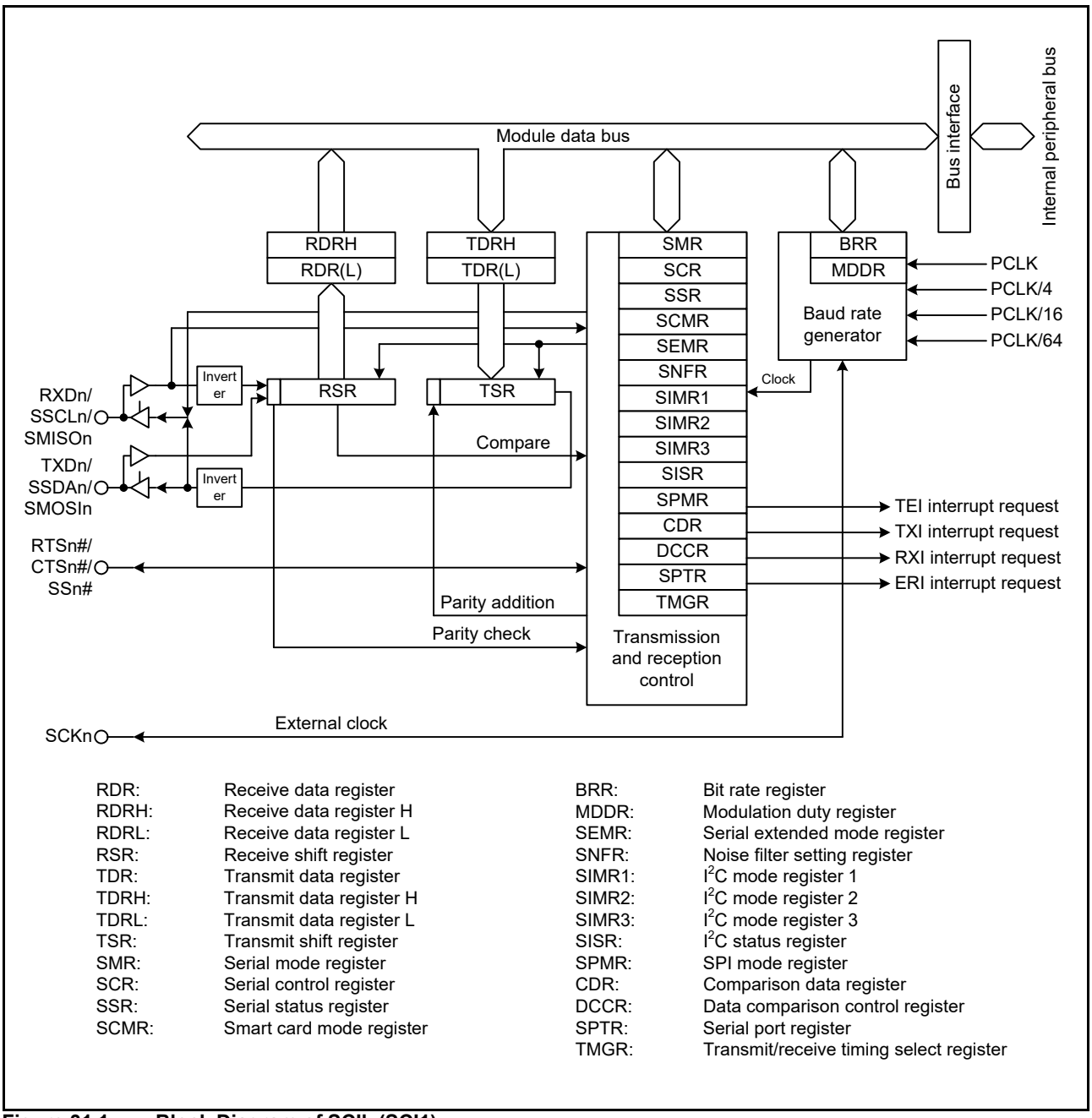
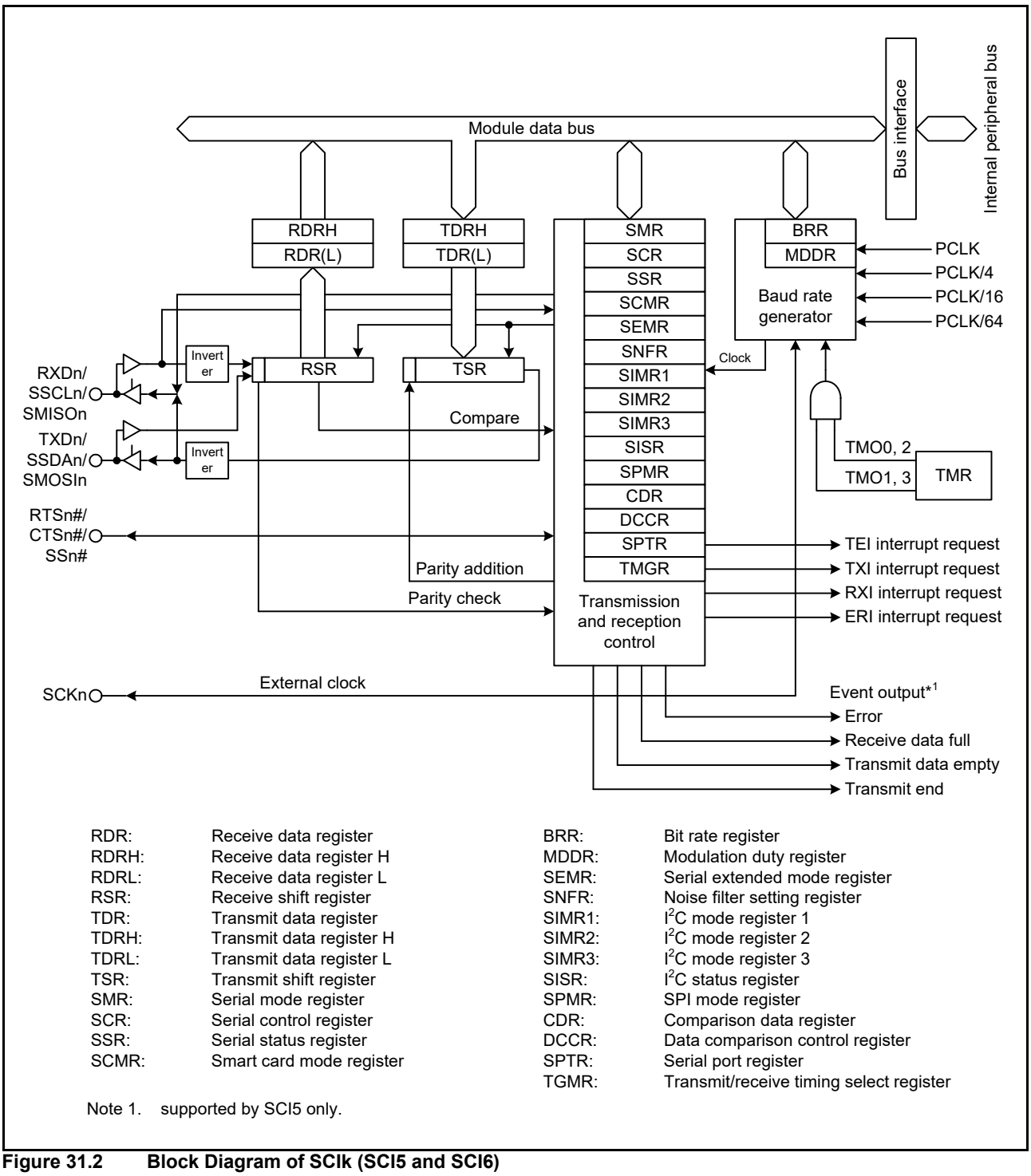


Figure 31.1 Block Diagram of SCIk (SCI1)



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. supported by SCI5 only.

Figure 31.2 Block Diagram of SCIk (SCI5 and SCI6)

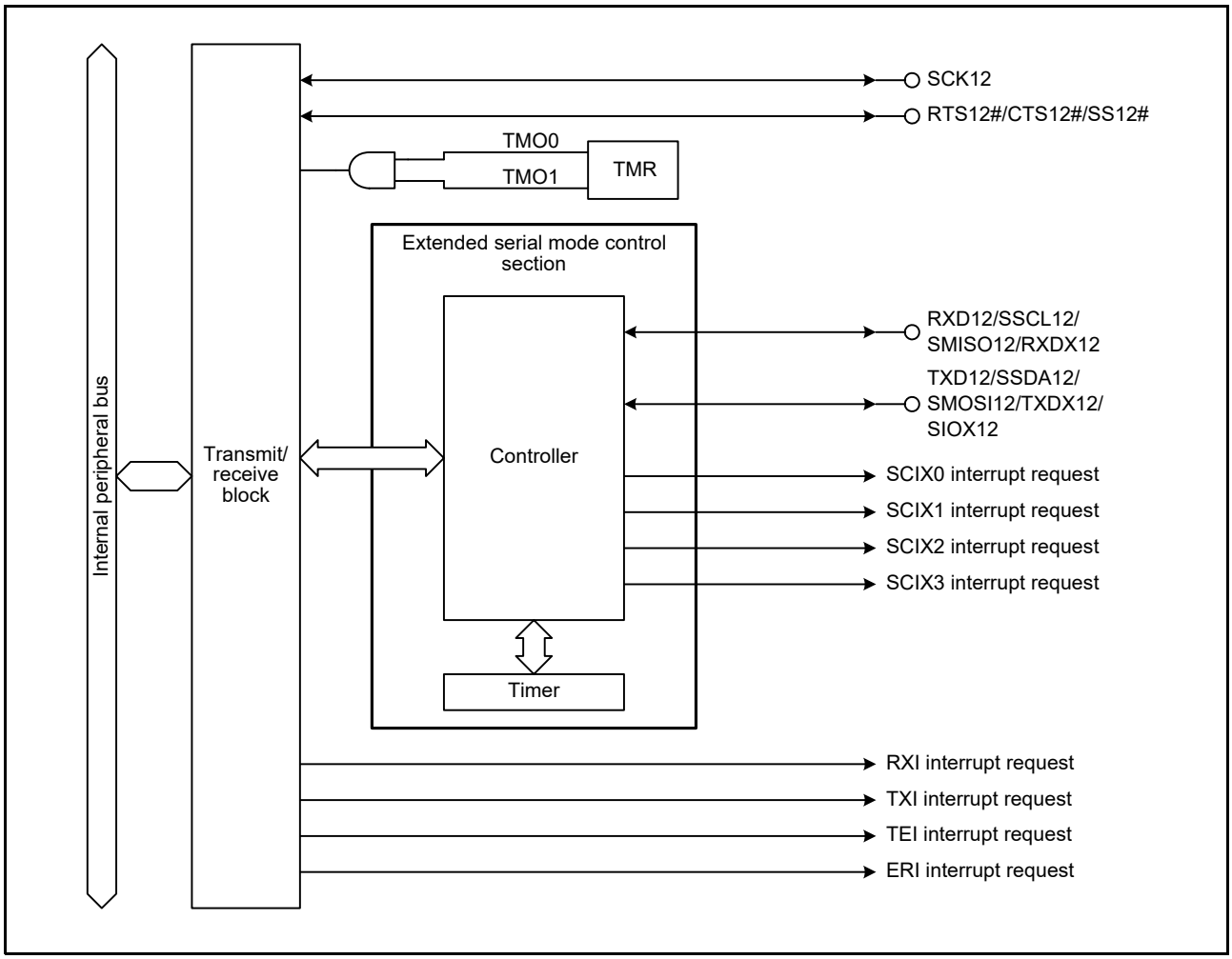


Figure 31.3 Block Diagram of SCIh (SCI12)

Table 31.4 to Table 31.7 list the pin configuration of the SCIs for the individual modes.

Table 31.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
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
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
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 31.5 SCI Pin Configuration in Simple I^2C Mode

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 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
SCI12	SSCL12	I/O	SCI12 I^2C clock input/output
	SSDA12	I/O	SCI12 I^2C data input/output

Table 31.6 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
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
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

Table 31.6 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
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 31.7 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

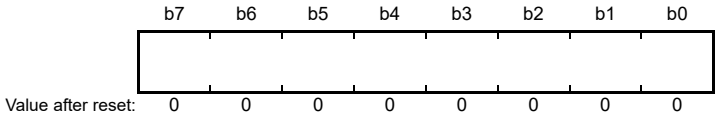
31.2 Register Descriptions

31.2.1 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.

31.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI6.RDR 0008 A0C5h, SCI12.RDR 0008 B305h



The RDR register is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from the RSR register to the RDR register. Then the RSR register can receive the next data.
Since the RSR and RDR registers function as a double buffer in this way, continuous receive operations can be performed.
Read the RDR register only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from the RDR register, an overrun error occurs.
The RDR register cannot be written to by the CPU.

31.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

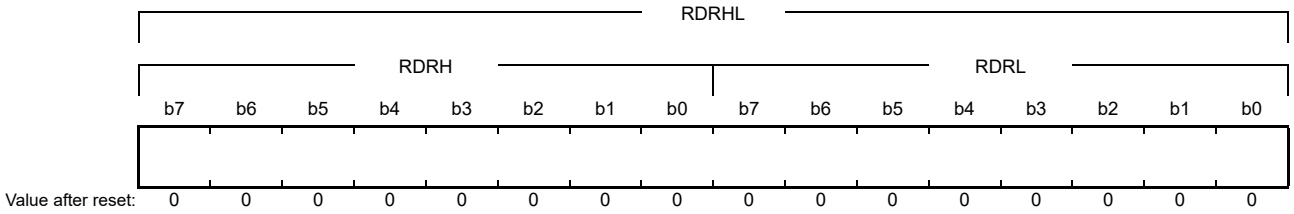
Address(es): SCI1.RDRH 0008 A030h, SCI5.RDRH 0008 A0B0h, SCI6.RDRH 0008 A0D0h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI1.RDRL 0008 A031h, SCI5.RDRL 0008 A0B1h, SCI6.RDRL 0008 A0D1h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI1.RDRHL 0008 A030h, SCI5.RDRHL 0008 A0B0h, SCI6.RDRHL 0008 A0D0h, 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.

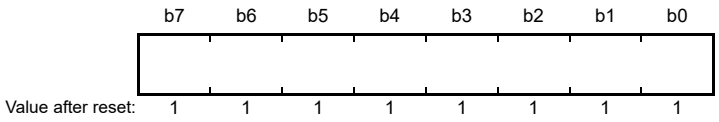
Read the RDRH and RDRL registers should be performed only once in the order from the RDRH register to the RDRL register when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from the 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.

31.2.4 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI6.TDR 0008 A0C3h, 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).

31.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

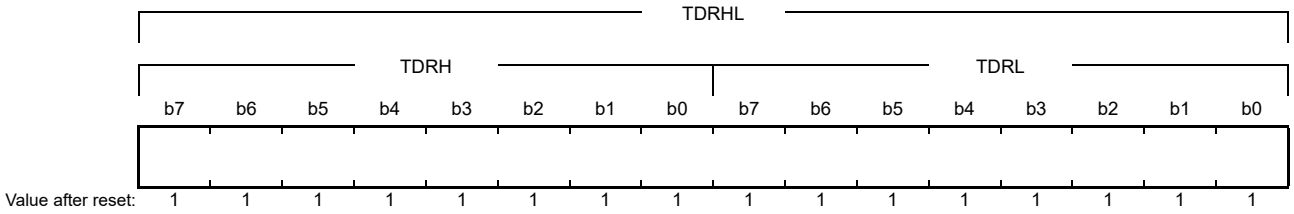
Address(es): SCI1.TDRH 0008 A02Eh, SCI5.TDRH 0008 A0AEh, SCI6.TDRH 0008 A0CEh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI1.TDRL 0008 A02Fh, SCI5.TDRL 0008 A0AFh, SCI6.TDRL 0008 A0CFh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI1.TDRHL 0008 A02Eh, SCI5.TDRHL 0008 A0AEh, SCI6.TDRHL 0008 A0CEh, 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.

31.2.6 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.

31.2.7 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): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, 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 31.2.11, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when 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 31.2.11, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the 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): SMCI1.SMR 0008 A020h, SMCI5.SMR 0008 A0A0h, SMCI6.SMR 0008 A0C0h, 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 31.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: 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 31.2.11, 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 31.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 31.6.4, Receive Data Sampling Timing and Reception Margin.

Table 31.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 31.2.11, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 31.6.2, 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 31.6.3, 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 31.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 31.6.8, Clock Output Control.

31.2.8 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): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, 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 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 31.4, 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 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 to 1 is enabled.
Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to 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): SMCI1.SCR 0008 A022h, SMCI5.SCR 0008 A0A2h, SMCI6.SCR 0008 A0C2h, 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 31.12, 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 31.6.8, 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.

31.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

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 SCI1, SCI5, and SCI6)
- 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 SCI1, SCI5, and SCI6)
- 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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): SMCI1.SSR 0008 A024h, SMCI5.SSR 0008 A0A4h, SMCI6.SSR 0008 A0C4h, 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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

31.2.10 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI6.SCMR 0008 A0C6h, SCI12.SCMR 0008 B306h, SMCI1.SCMR 0008 A026h, SMCI5.SCMR 0008 A0A6h, SMCI6.SCMR 0008 A0C6h, 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 31.9 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 31.9 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 31.2.11, Bit Rate Register (BRR)).

31.2.11 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, 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 31.10 and Table 31.11 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 I²C mode.
The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 31.10 Relationship between N Setting in the BRR Register and Bit Rate B (for SCI1, SCI5, and SCI6)

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 I²C*1				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 31.13 and Table 31.14.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 31.11 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 31.13 and Table 31.14.
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 31.12 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 31.13 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 31.14 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 31.15 lists examples of N settings in the BRR register in normal asynchronous mode. Table 31.17 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 31.20. Examples of BRR (N) settings in smart card interface mode are listed in Table 31.22. Examples of BRR (N) settings in simple I^2C mode are listed in Table 31.24. 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 31.6.4, Receive Data Sampling Timing and Reception Margin. Table 31.18 and Table 31.21 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 31.15. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 31.15 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		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	–0.35
300	2	129	0.16	2	162	–0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35
1200	1	129	0.16	1	162	–0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35
4800	0	129	0.16	0	162	–0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35
31250	0	19	0.00	0	24	0.00	0	29	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73

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.

Table 31.16 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) (for SCI1, SCI5, and SCI6)

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	9.8304	0	0	0	0	0	307200
		1	0	0	0	500000			1	0	0	0	614400
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	1228800
	0 or 1	0 or 1	1	0	0	1333333		0 or 1	0 or 1	1	0	0	1638400
10	0	0	0	0	0	312500	12	0	0	0	0	0	375000
		1	0	0	0	625000			1	0	0	0	750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	1500000
	0 or 1	0 or 1	1	0	0	1666667		0 or 1	0 or 1	1	0	0	2000000
12.288	0	0	0	0	0	384000	14	0	0	0	0	0	437500
		1	0	0	0	768000			1	0	0	0	875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	1750000
	0 or 1	0 or 1	1	0	0	2048000		0 or 1	0 or 1	1	0	0	2333333
16	0	0	0	0	0	500000	17.2032	0	0	0	0	0	537600
		1	0	0	0	1000000			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	2150400
	0 or 1	0 or 1	1	0	0	2666667		0 or 1	0 or 1	1	0	0	2867200
18	0	0	0	0	0	562500	19.6608	0	0	0	0	0	614400
		1	0	0	0	1125000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2250000			1	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	3000000		0 or 1	0 or 1	1	0	0	3276800
20	0	0	0	0	0	625000	25	0	0	0	0	0	781250
		1	0	0	0	1250000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2500000			1	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	3333333		0 or 1	0 or 1	1	0	0	4166667
30	0	0	0	0	0	937500							
		1	0	0	0	1875000							
	1	0	0	0	0								
		1	0	0	0	3750000							
	0 or 1	0 or 1	1	0	0	5000000							

Table 31.17 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	9.8304	0	0	0	0	307200
		1	0	0	500000			1	0	0	614400
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	1228800
10	0	0	0	0	312500	12	0	0	0	0	375000
		1	0	0	625000			1	0	0	750000
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	1500000
12.288	0	0	0	0	384000	14	0	0	0	0	437500
		1	0	0	768000			1	0	0	875000
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	1750000
16	0	0	0	0	500000	17.2032	0	0	0	0	537600
		1	0	0	1000000			1	0	0	1075200
	1	0	0	0			1	0	0	0	
		1	0	0	2000000			1	0	0	2150400
18	0	0	0	0	562500	19.6608	0	0	0	0	614400
		1	0	0	1125000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	2250000			1	0	0	2457600
20	0	0	0	0	625000	25	0	0	0	0	781250
		1	0	0	1250000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	2500000			1	0	0	3125000
30	0	0	0	0	937500						
		1	0	0	1875000						
	1	0	0	0							
		1	0	0	3750000						

Table 31.18 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

Table 31.19 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

Table 31.20 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

	Operating Frequency PCLK (MHz)											
	8		10		16		20		25		30	
Bit Rate (bps)	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
1 k	2	124	2	155	2	249	3	77	3	97	3	116
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187
5 k	1	99	1	124	1	199	1	249	2	77	2	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187
25 k	0	79	0	99	0	159	0	199	0	249	1	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14
1 M	0	1			0	3	0	4	—	—		
2 M	0	0*1			0	1			—	—		
2.5 M			0	0*1			0	1			0	2
4 M					0	0*1						
5 M							0	0*1				
6.25 M									0	0*1		
7.5 M											0	0*1

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. 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 31.21 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

Table 31.22 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

Table 31.23 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

Table 31.24 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	30	0.8	0	49	0.0	0	62	–0.8	0	77	0.2
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	0	24	0.0	0	30	0.8
50 k	0	4	0.0	0	5	4.2	0	9	0.0	0	12	–3.8	2	0	–2.3
100 k	0	2	–16.7	1	0	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	0	25.0	0	1	0.0	0	2	–16.7	0	2	4.2
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	30		
	n	N	Error (%)
10 k	0	93	–0.3
25 k	0	37	–1.3
50 k	0	18	–1.3
100 k	0	9	–6.3
250 k	1	0	–6.3
350 k	0	2	–10.7

Table 31.25 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	30	43.40/49.60	0	49	43.75/50.00	0	62	44.10/50.40
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	0	24	17.50/20.00
50 k	0	4	8.75/10.00	0	5	8.40/9.60	0	9	8.75/10.00	0	12	9.10/10.40
100 k	0	2	5.25/6.00	1	0	5.60/6.40	0	4	4.38/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	0	1.40/1.60	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		
	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	77	43.68/49.92	0	93	43.87/50.13
25 k	0	30	17.36/19.84	0	37	17.73/20.27
50 k	2	0	8.96/10.24	0	18	8.87/10.13
100 k	1	1	4.48/5.12	0	9	4.67/5.33
250 k	0	2	1.68/1.92	1	0	1.87/2.13
350 k	0	2	1.68/1.92	0	2	1.40/1.60

31.2.12 Modulation Duty Register (MDDR)

Address(es): SCI1.MDDR 0008 A032h, SCI5.MDDR 0008 A0B2h, SCI6.MDDR 0008 A0D2h, 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 31.26 and Table 31.27.

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 31.26 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used (for SCI1, SCI5, and SCI6)

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 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 31.13 and Table 31.14, section 31.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] =

00b, SCR.CKE[1] = 0, and BRR = 0).

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 31.27 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 31.13 and Table 31.14, section 31.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple 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.

31.2.13 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, 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 31.28 for details.

The ACS0 bit for SCI1 is reserved. The write value to this bit for SCI1 should be 0.

Table 31.28 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 31.4 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

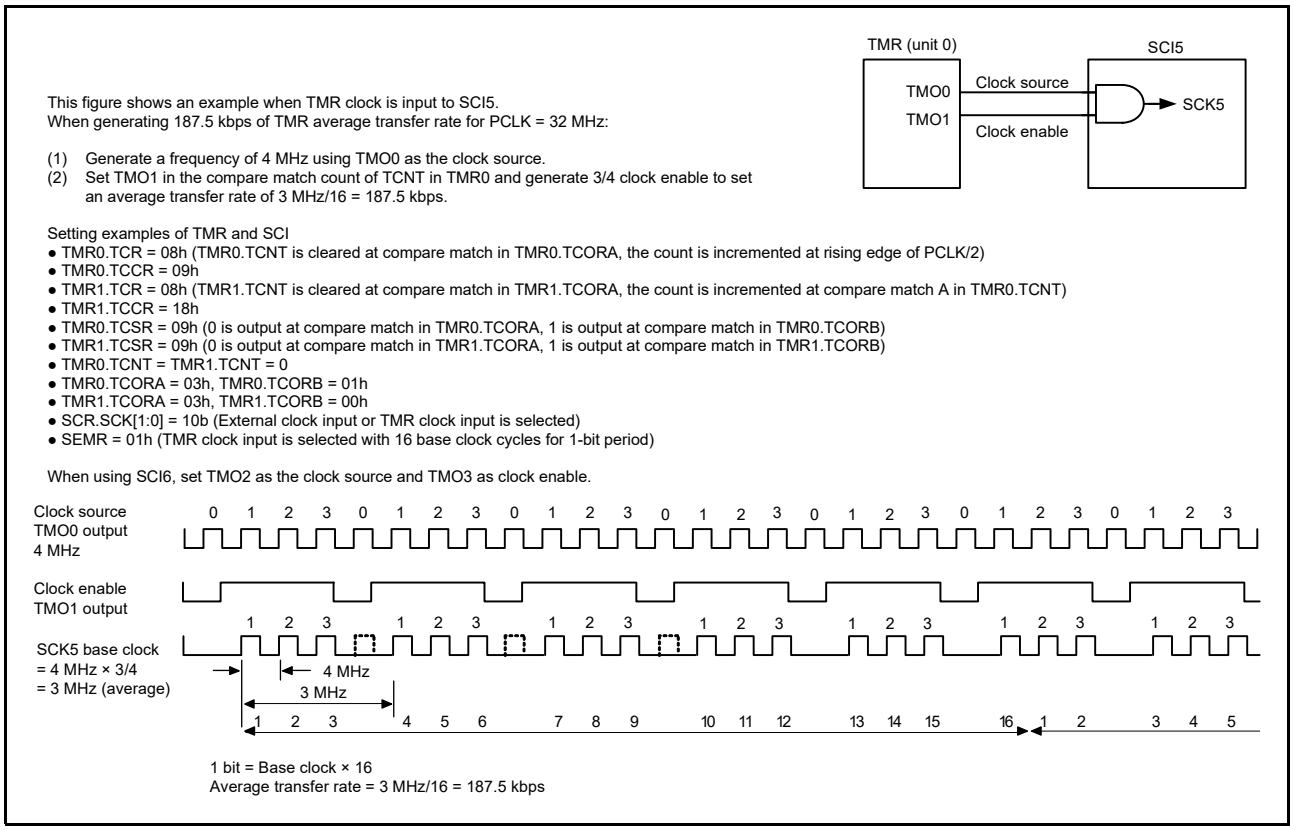


Figure 31.4 Example of Average Transfer Rate Setting When TMR Clock is Input

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.

31.2.14 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h, SCI6.SNFR 0008 A0C8h, SCI12.SNFR 0008 B308h

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple I2C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR 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.

31.2.15 I²C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, 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.

31.2.16 I2C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh, SCI6.SIMR2 0008 A0CAh, 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.

31.2.17 I2C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, 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.

31.2.18 I²C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, 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.

31.2.19 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh, SCI6.SPMR 0008 A0CDh, 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)

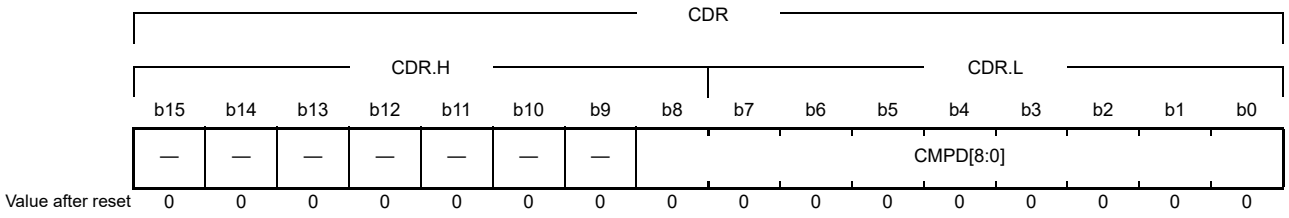
This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 31.62 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 31.62 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

31.2.20 Comparison Data Register (CDR)

Address(es): SCI1.CDR 0008 A03Ah, SCI5.CDR 0008 A0BAh, SCI6.CDR 0008 A0DAh,
SCI1.CDR.H 0008 A03Ah, SCI5.CDR.H 0008 A0BAh, SCI6.CDR.H 0008 A0DAh,
SCI1.CDR.L 0008 A03Bh, SCI5.CDR.L 0008 A0BBh, SCI6.CDR.L 0008 A0DBh



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.

31.2.21 Data Comparison Control Register (DCCR)

Address(es): SCI1.DCCR 0008 A033h, SCI5.DCCR 0008 A0B3h, SCI6.DCCR 0008 A0D3h

	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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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.

31.2.22 Serial Port Register (SPTR)

Address(es): SCI1.SPTR 0008 A03Ch, SCI5.SPTR 0008 A0BCh, SCI6.SPTR 0008 A0DCh

	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 31.29 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 31.29 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 31.29 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.

31.2.23 Transmit/Receive Timing Select Register (TMGR)

Address(es): SCI1.TMGR 0008 A03Dh, SCI5.TMGR 0008 A0BDh, SCI6.TMGR 0008 A0DDh

	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 31.30 for the range of the value that can be set in the RTMG[2:0] bits.

Table 31.30 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 31.31 for the range of the value that can be set in the TTMG[2:0] bits.

Table 31.31 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)

31.2.24 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 31.32 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.

31.2.25 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

31.2.26 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

31.2.27 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.

31.2.28 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.

31.2.29 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.

31.2.30 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

31.2.31 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

31.2.32 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

31.2.33 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.

31.2.34 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

31.2.35 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.

31.2.36 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.

31.2.37 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.

31.2.38 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

31.2.39 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.

31.2.40 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

31.2.41 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.

31.2.42 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.

31.2.43 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.

31.3 Operation in Asynchronous Mode

Figure 31.5 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

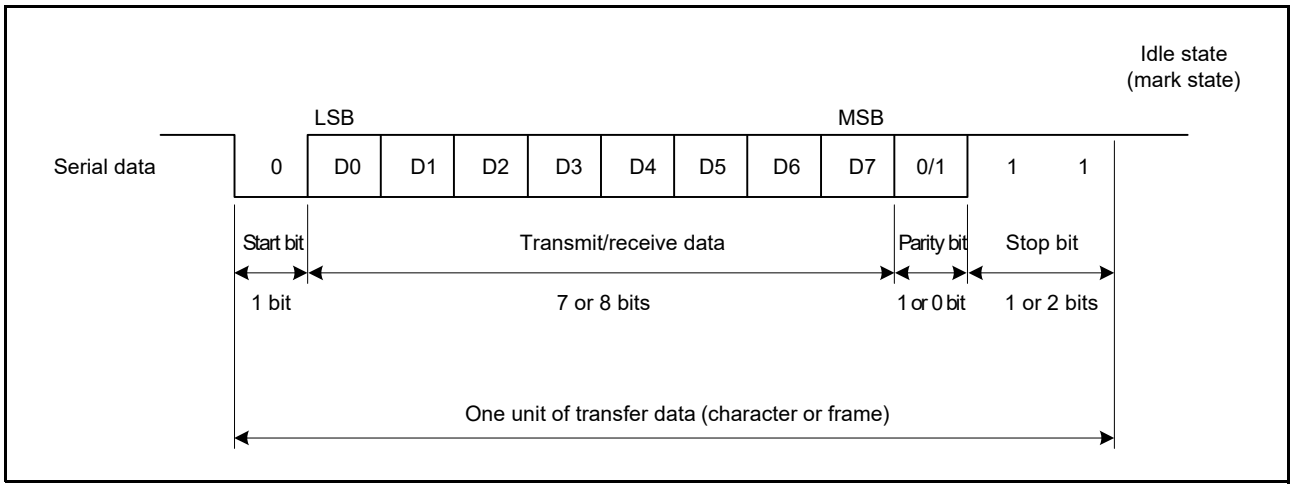


Figure 31.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

31.3.1 Serial Data Transfer Format

Table 31.33 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 31.4, Multi-Processor Communications Function.

Table 31.33 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

31.3.2 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 31.6. 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.

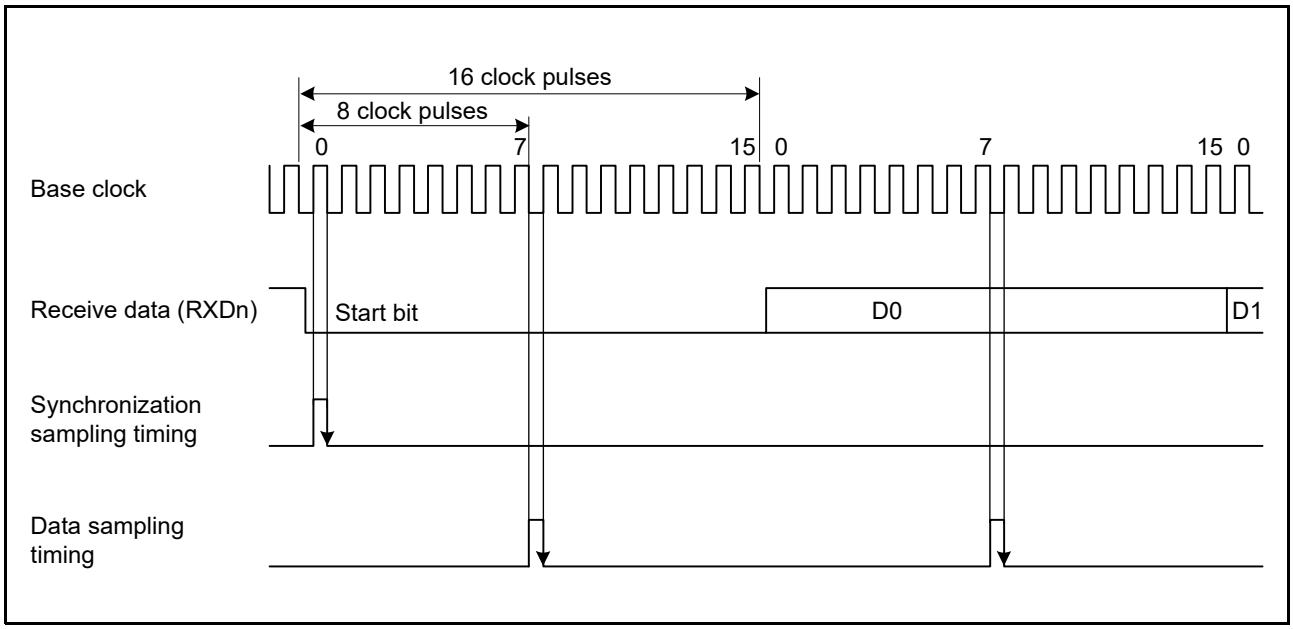


Figure 31.6 Receive Data Sampling Timing in Asynchronous Mode

In SCI1, SCI5, and SCI6, 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.

31.3.2.1 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 31.7 shows an example of the sampling timing adjustment.

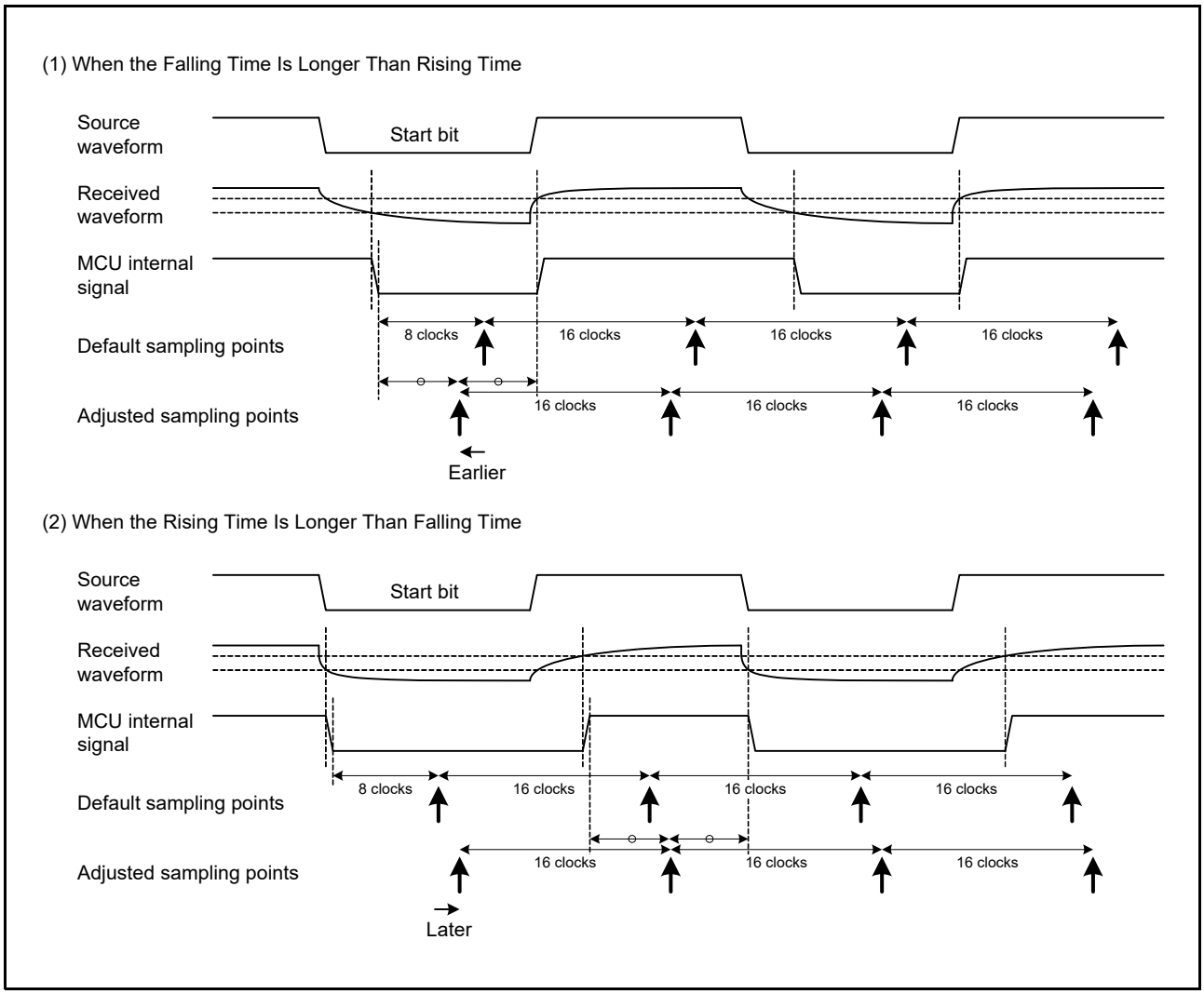


Figure 31.7 Example of Sampling Timing Adjustment (SEMR.ABCSE bit = 0 and SEMR.ABCS bit = 0)

31.3.2.2 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 31.8 shows an example of the transition timing adjustment.

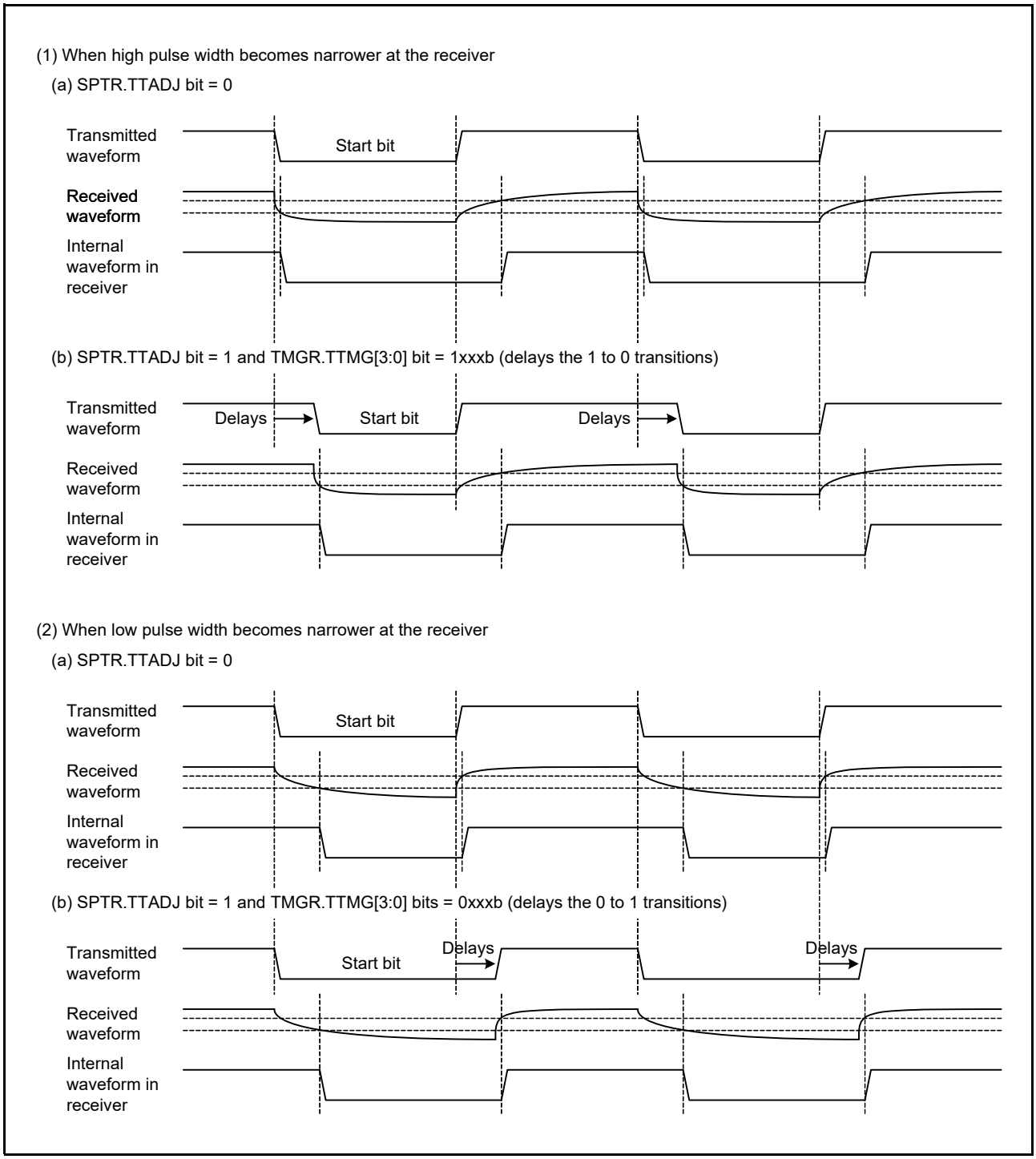


Figure 31.8 Example of Transition Timing Adjustment

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 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 31.9.

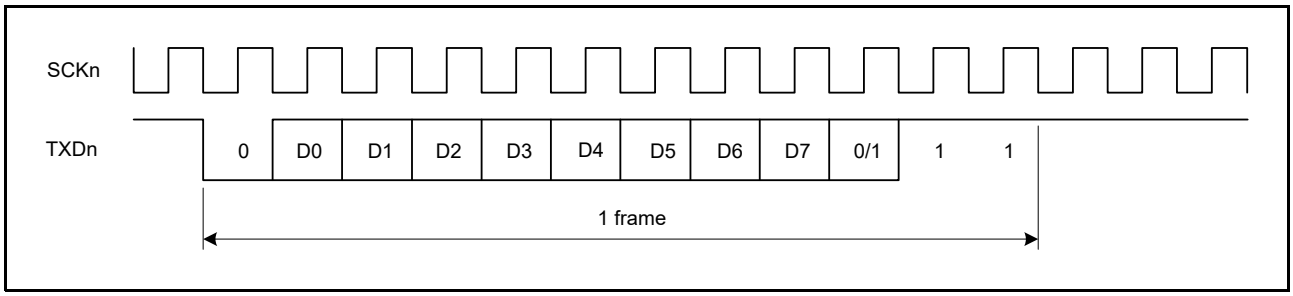


Figure 31.9 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

31.3.4 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 31.3.2, 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.

31.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE bit is 1.
- Reception is not in progress.
- There are no received data yet to be read.
- The ORER, FER, and PER flags in the SSR register are all 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

Note that either one of CTS and RTS can be selected.

31.3.6 Data Match Detection

The data match detection function is available in asynchronous mode for SCI1, SCI5, and SCI6.
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 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 31.10 and Figure 31.11 show examples of data match detection.

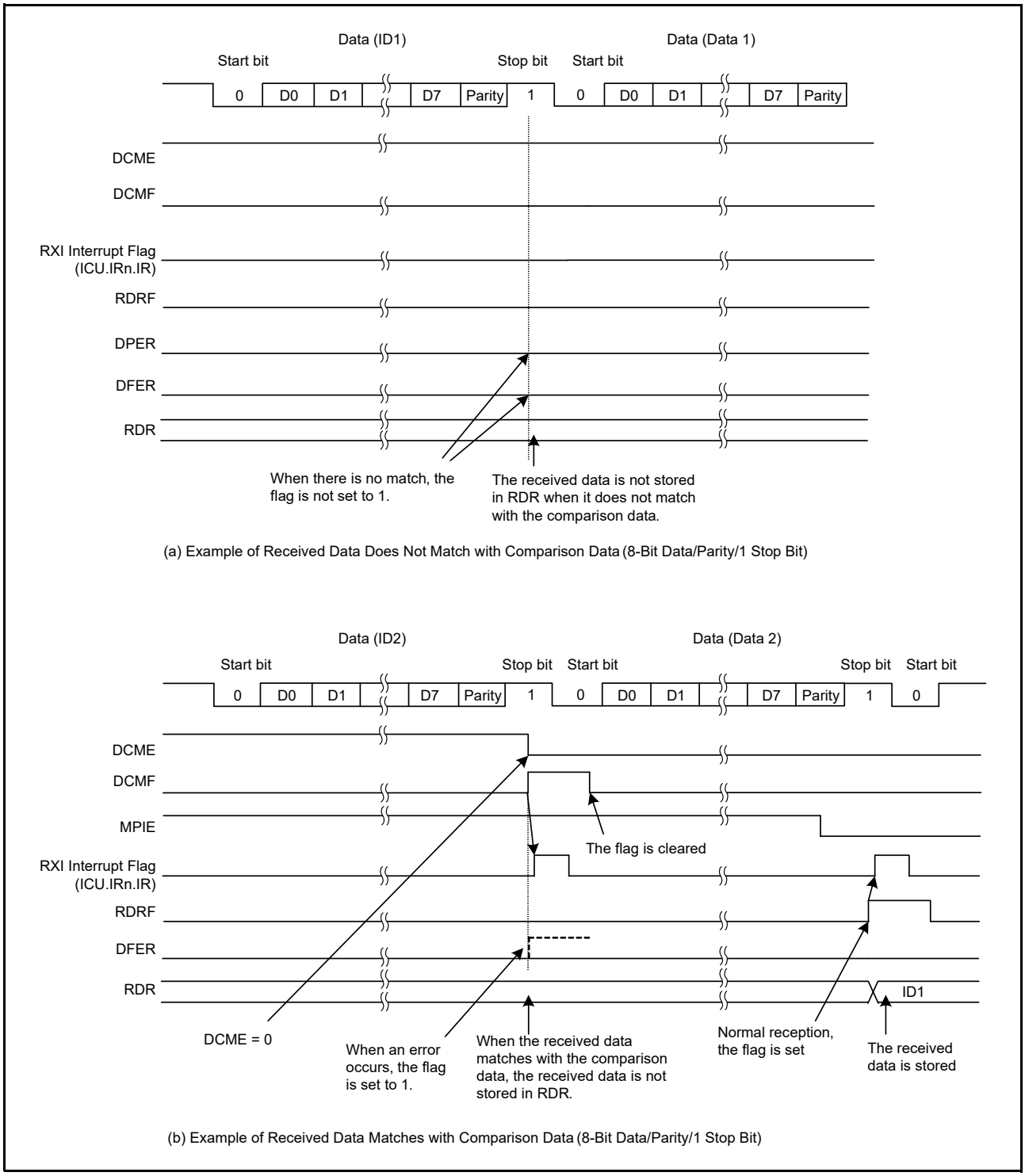


Figure 31.10 Example of Data Match Detection (1) Non Multi-Processor Mode

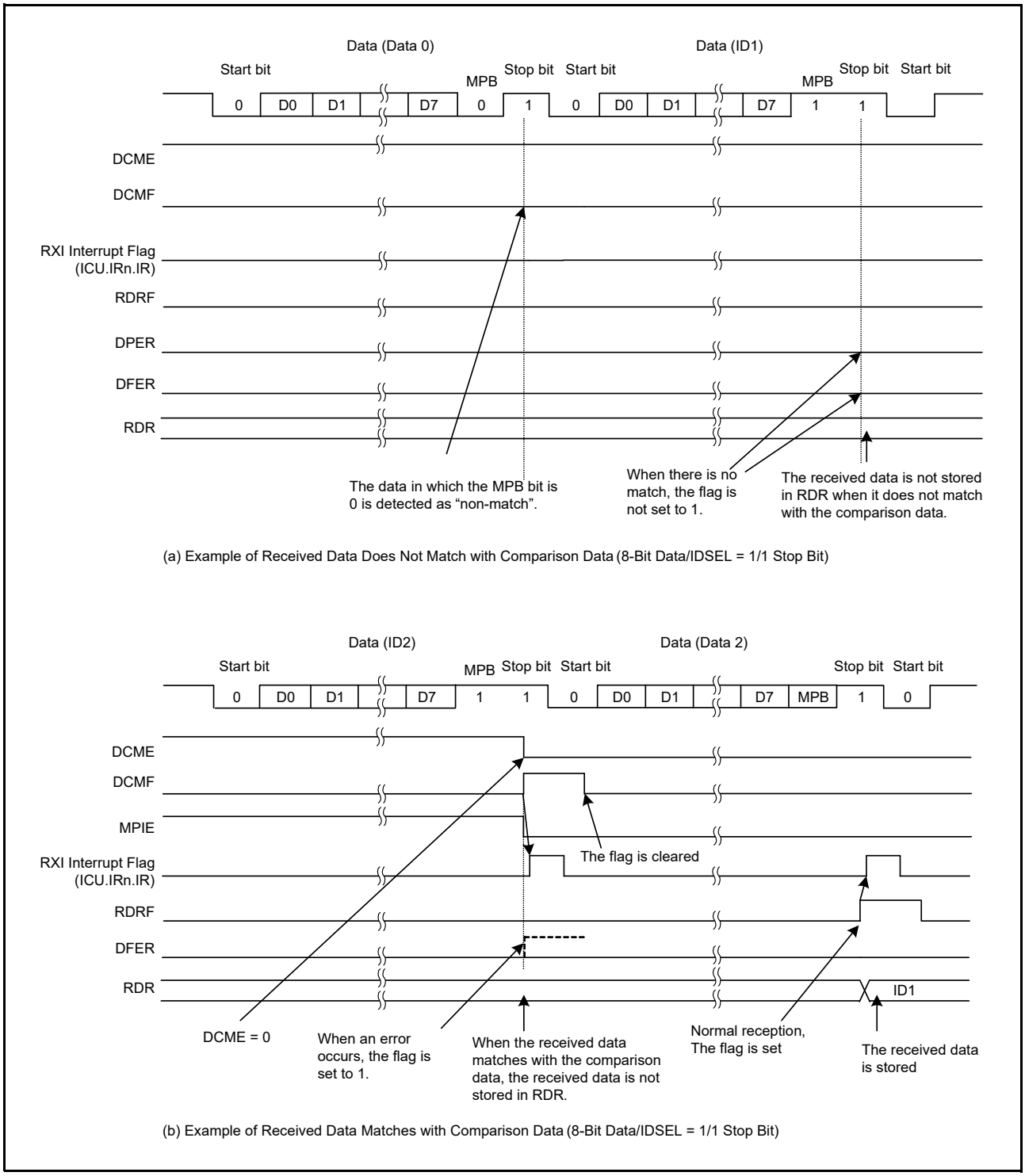


Figure 31.11 Example of Data Match Detection (2) Non Multi-Processor Mode

31.3.7 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 31.12. 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.

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.

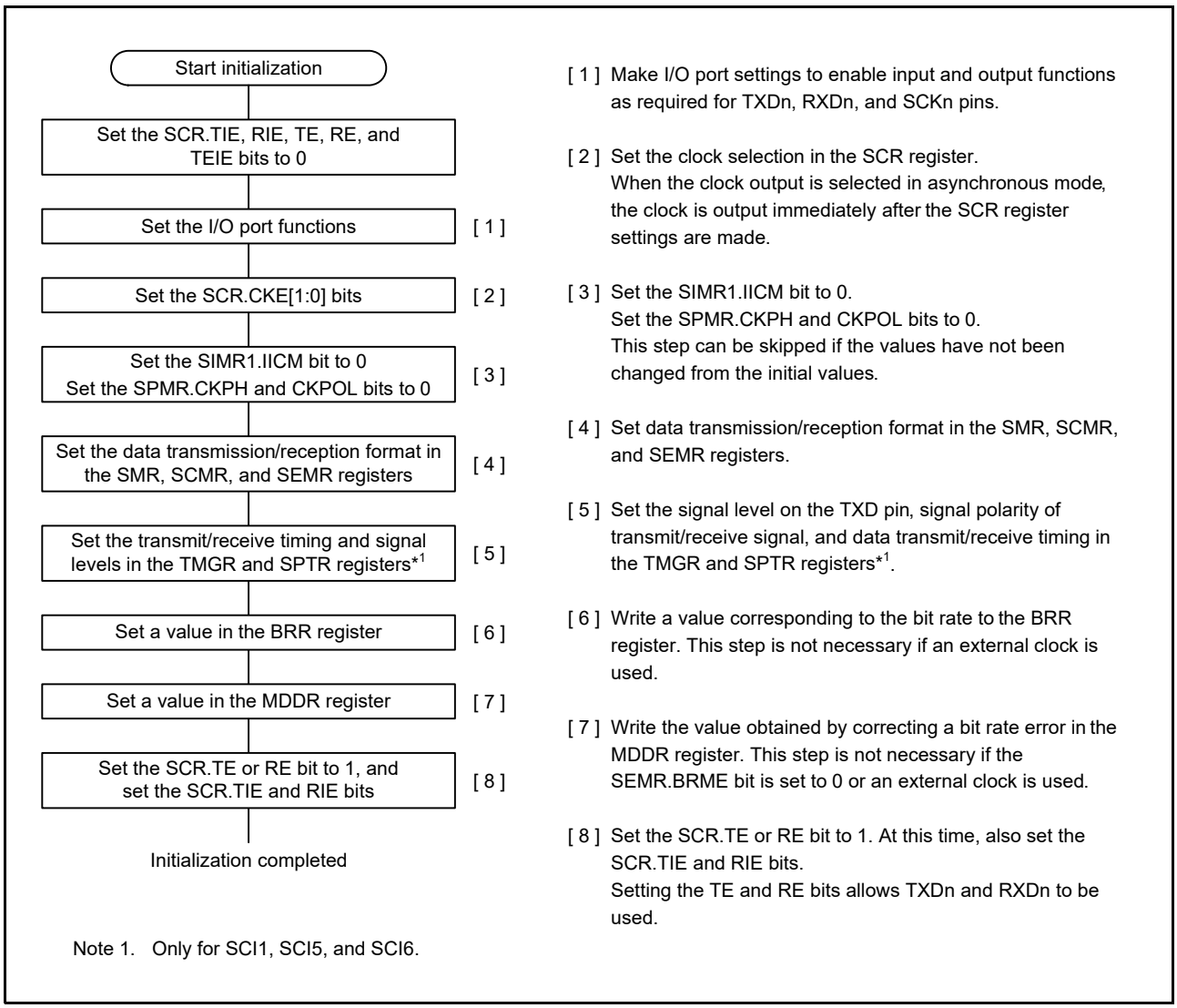


Figure 31.12 Sample SCI Initialization Flowchart (Asynchronous Mode)

Figure 31.13 shows an example of data transmission when the SCI is set to asynchronous mode according to the flow described in Figure 31.12 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.

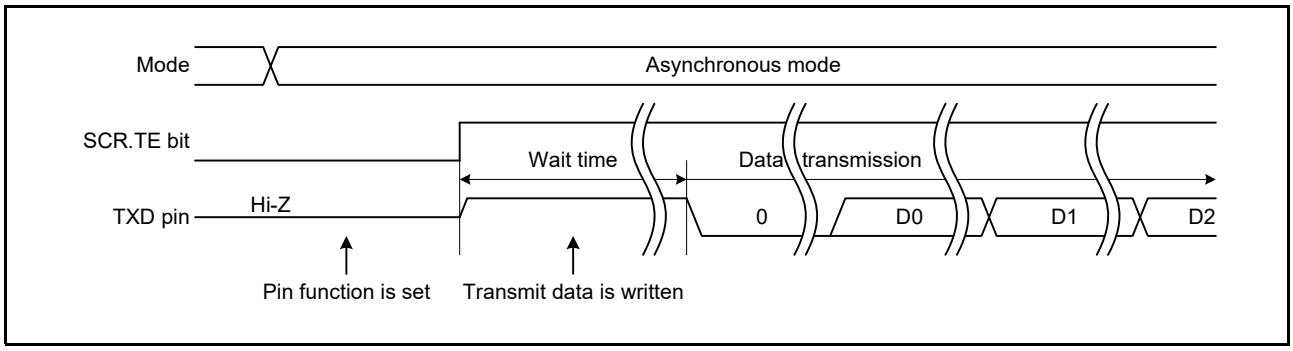


Figure 31.13 Example of Data Transmission Timing in Asynchronous Mode

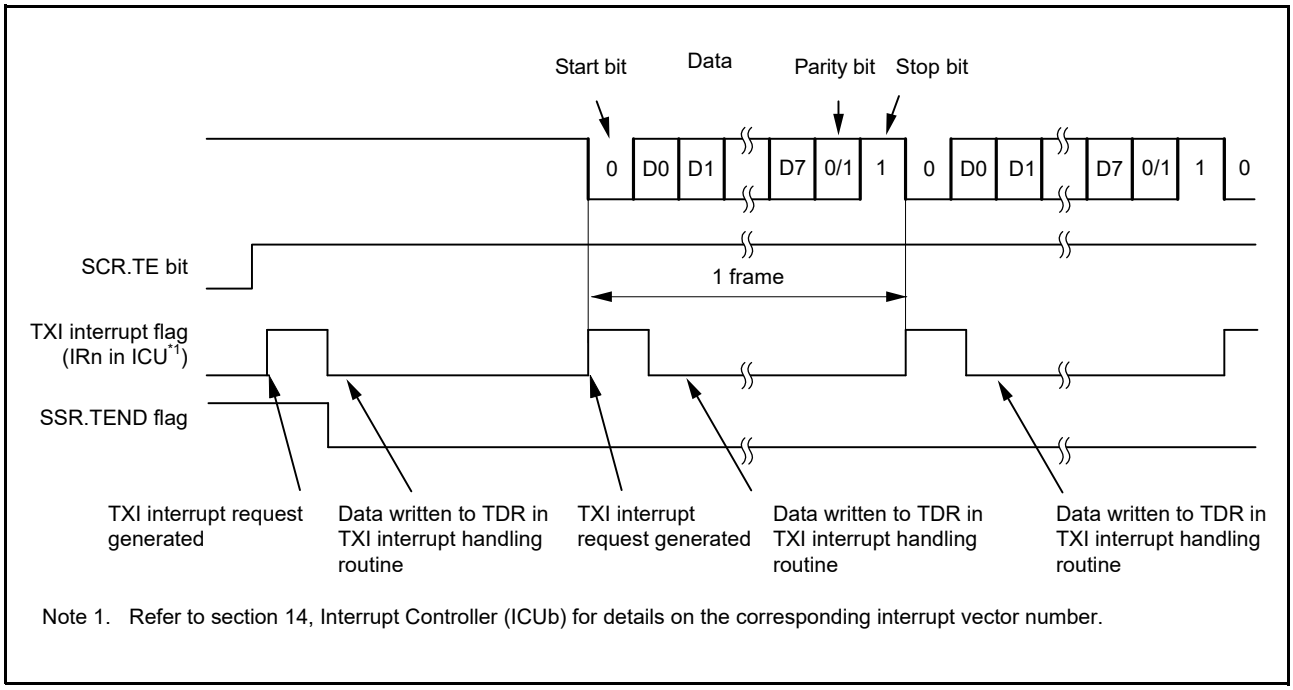
31.3.8 Serial Data Transmission (Asynchronous Mode)

Figure 31.14 to Figure 31.16 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.

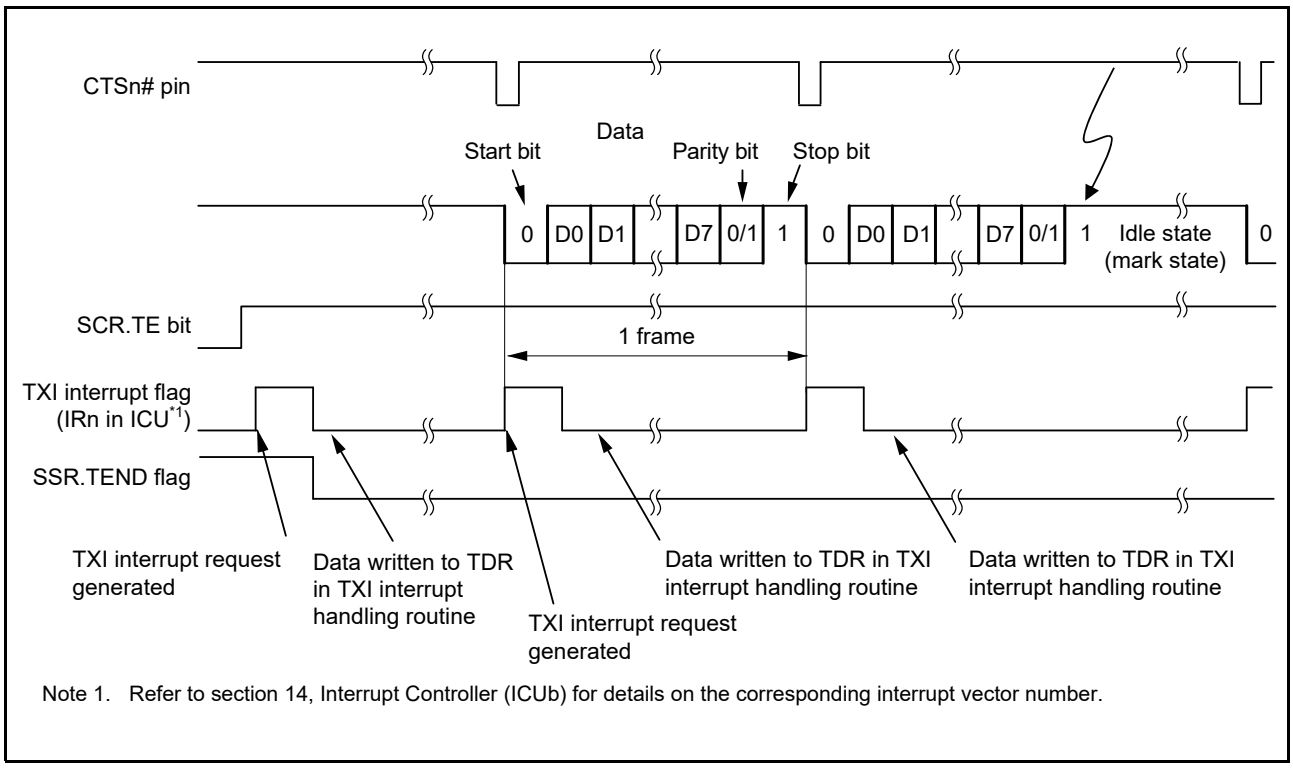
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 31.17 shows a sample flowchart for serial transmission in asynchronous mode.



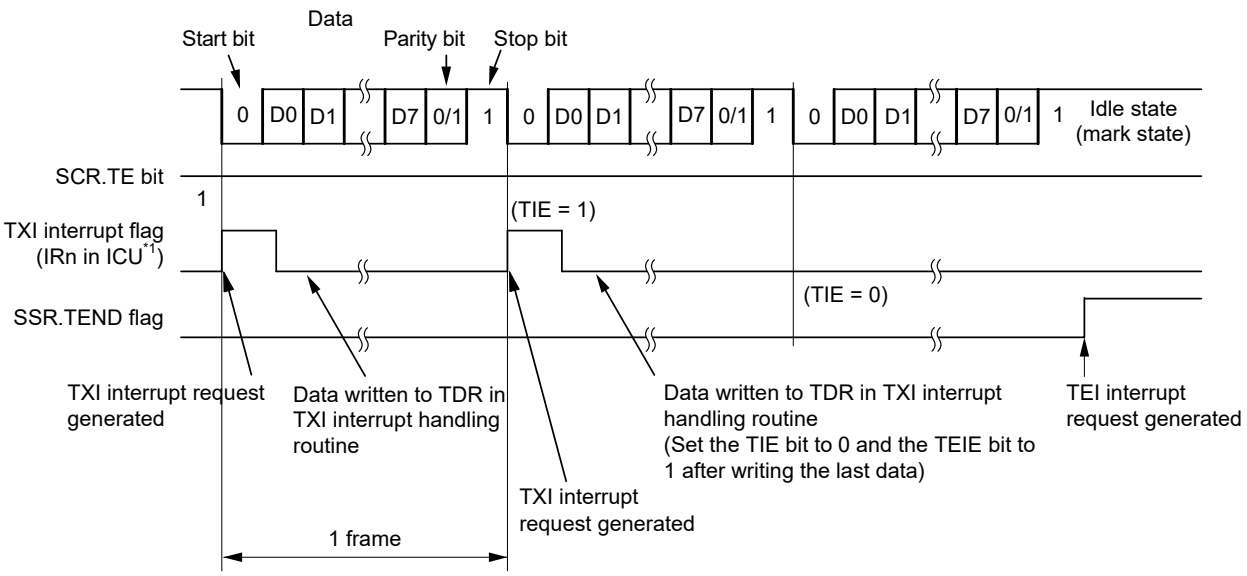
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 31.14 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)



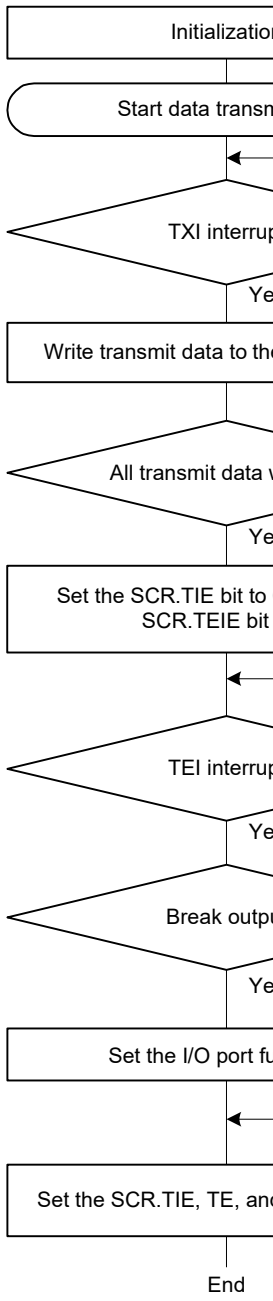
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 31.15 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 31.16 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*1, 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.

Note 1. This is when the SEMR.ITE bit is 0. When the ITE bit is 1, transmission is enabled immediately after the TE bit is set to 1.

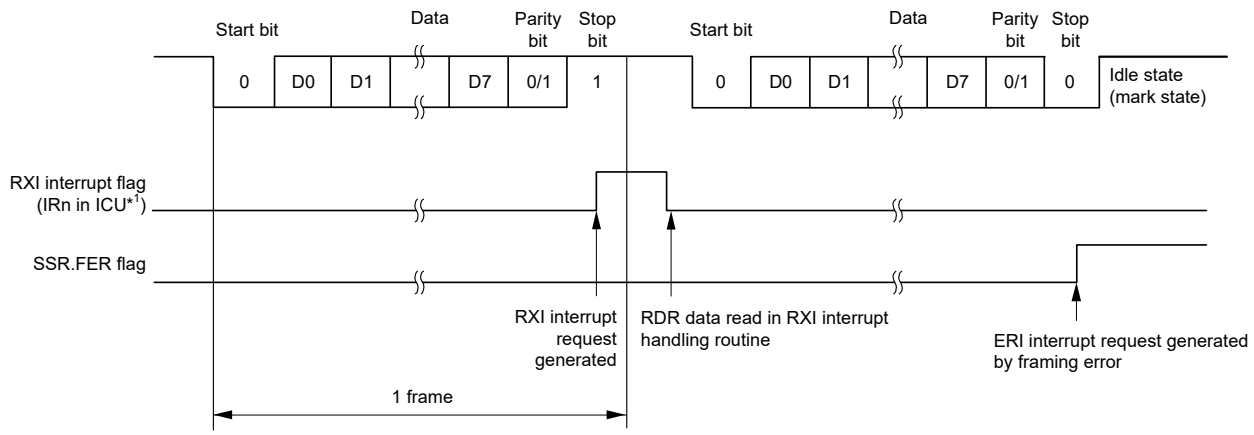
Figure 31.17 Example of Serial Transmission Flowchart in Asynchronous Mode

31.3.9 Serial Data Reception (Asynchronous Mode)

Figure 31.18 and Figure 31.19 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

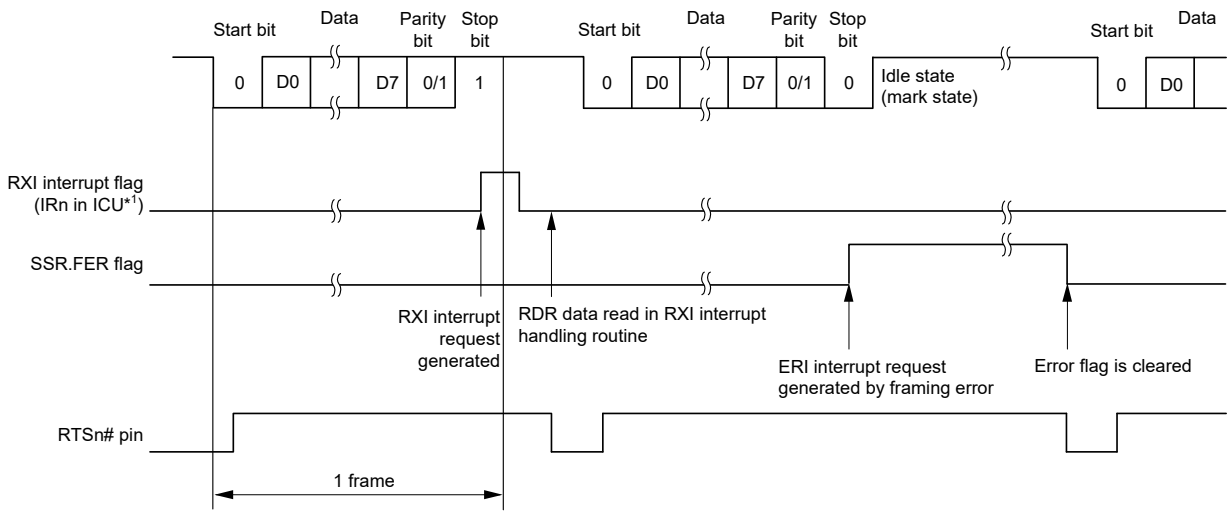
1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 31.18 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 31.19 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 31.34 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.

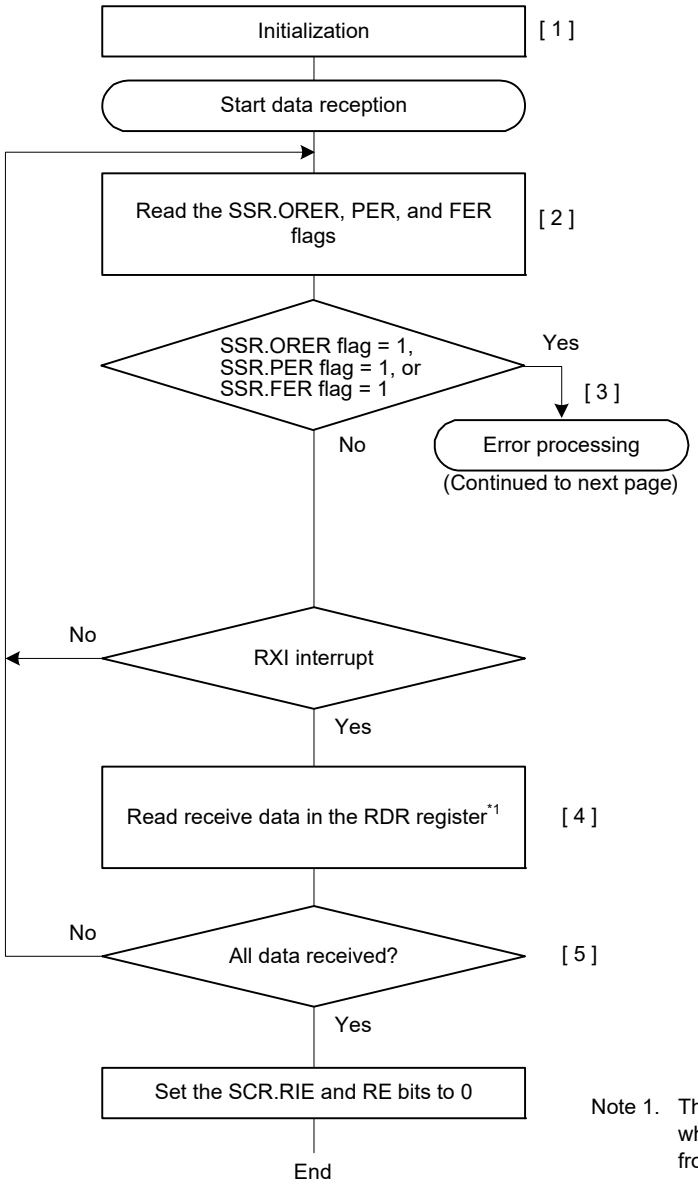
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in the RDR (or the RDRL) register.

Figure 31.20 and Figure 31.21 show samples of flowcharts for serial data reception.

Table 31.34 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 31.20 Example Flowchart of Serial Reception in Asynchronous Mode (1)

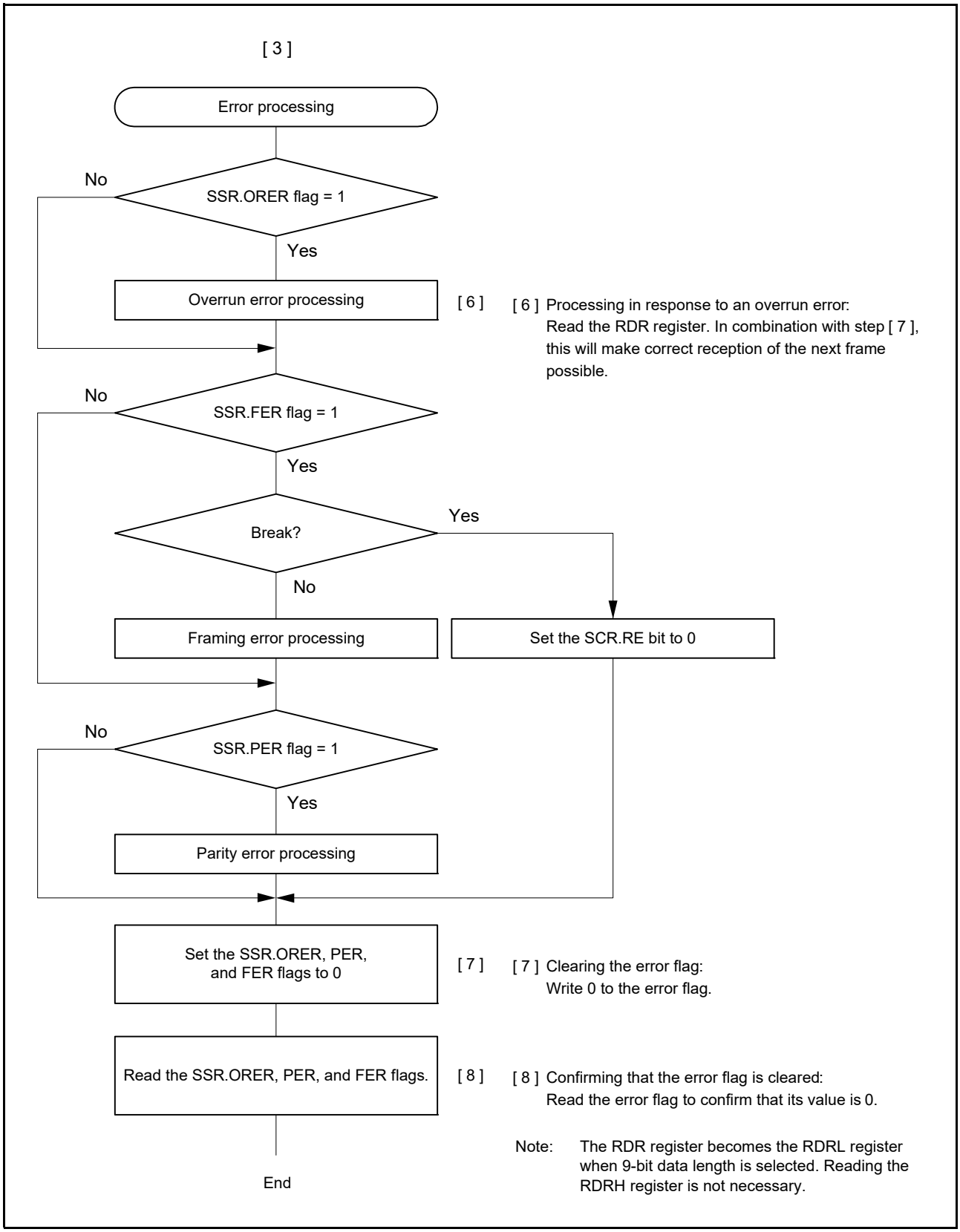


Figure 31.21 Example Flowchart of Serial Reception in Asynchronous Mode (2)

31.4 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 31.22 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags 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.

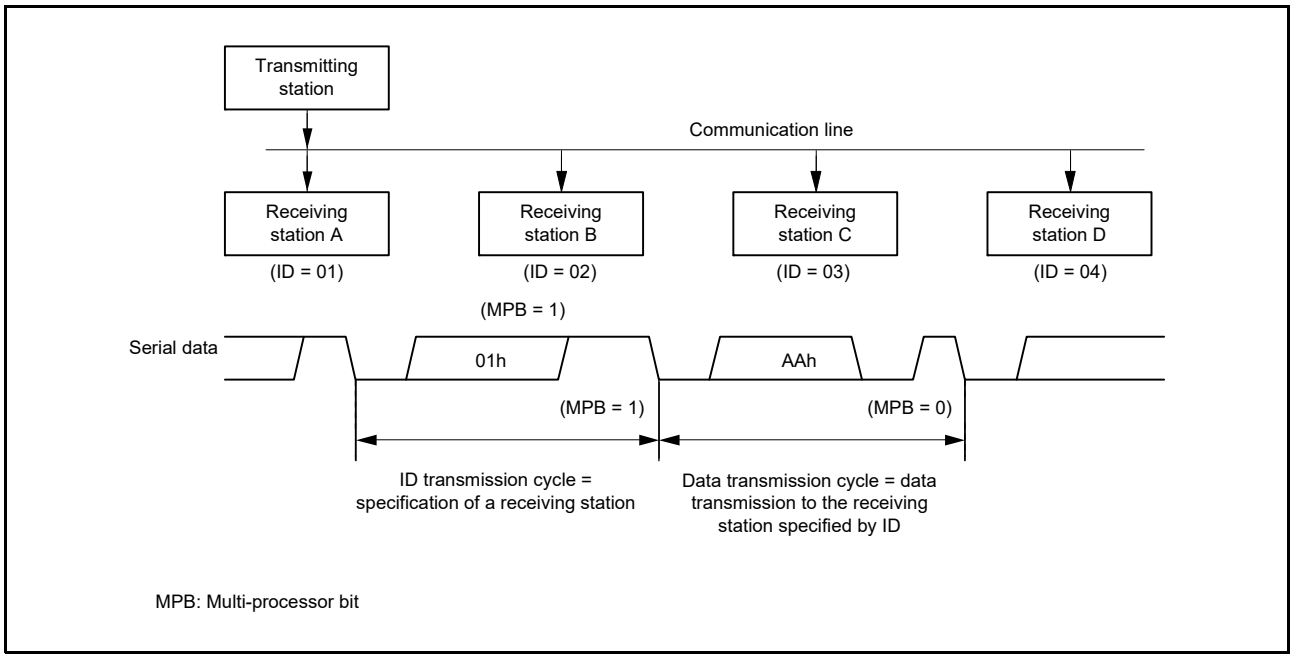


Figure 31.22 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

31.4.1 Multi-Processor Serial Data Transmission

Figure 31.23 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.

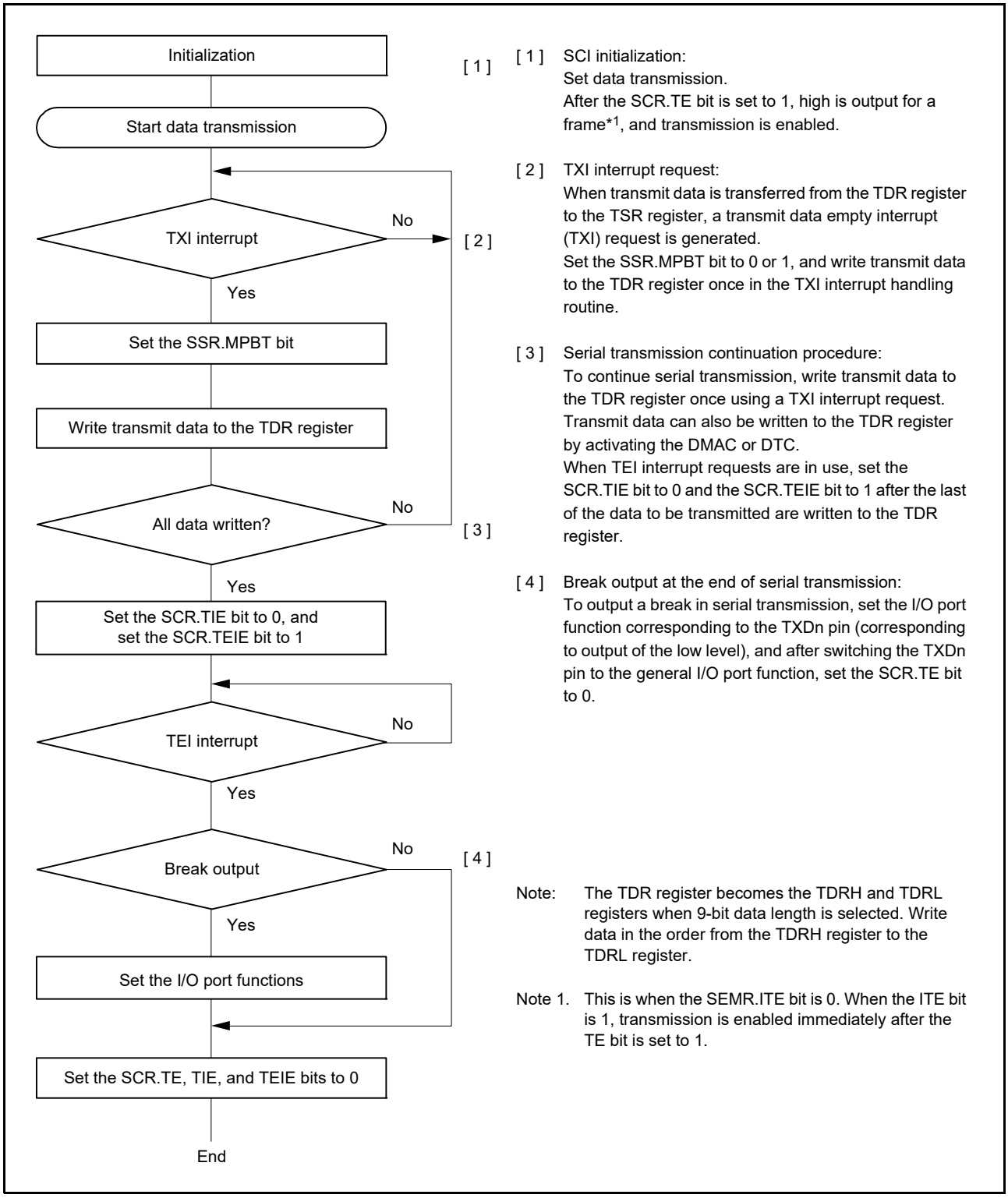
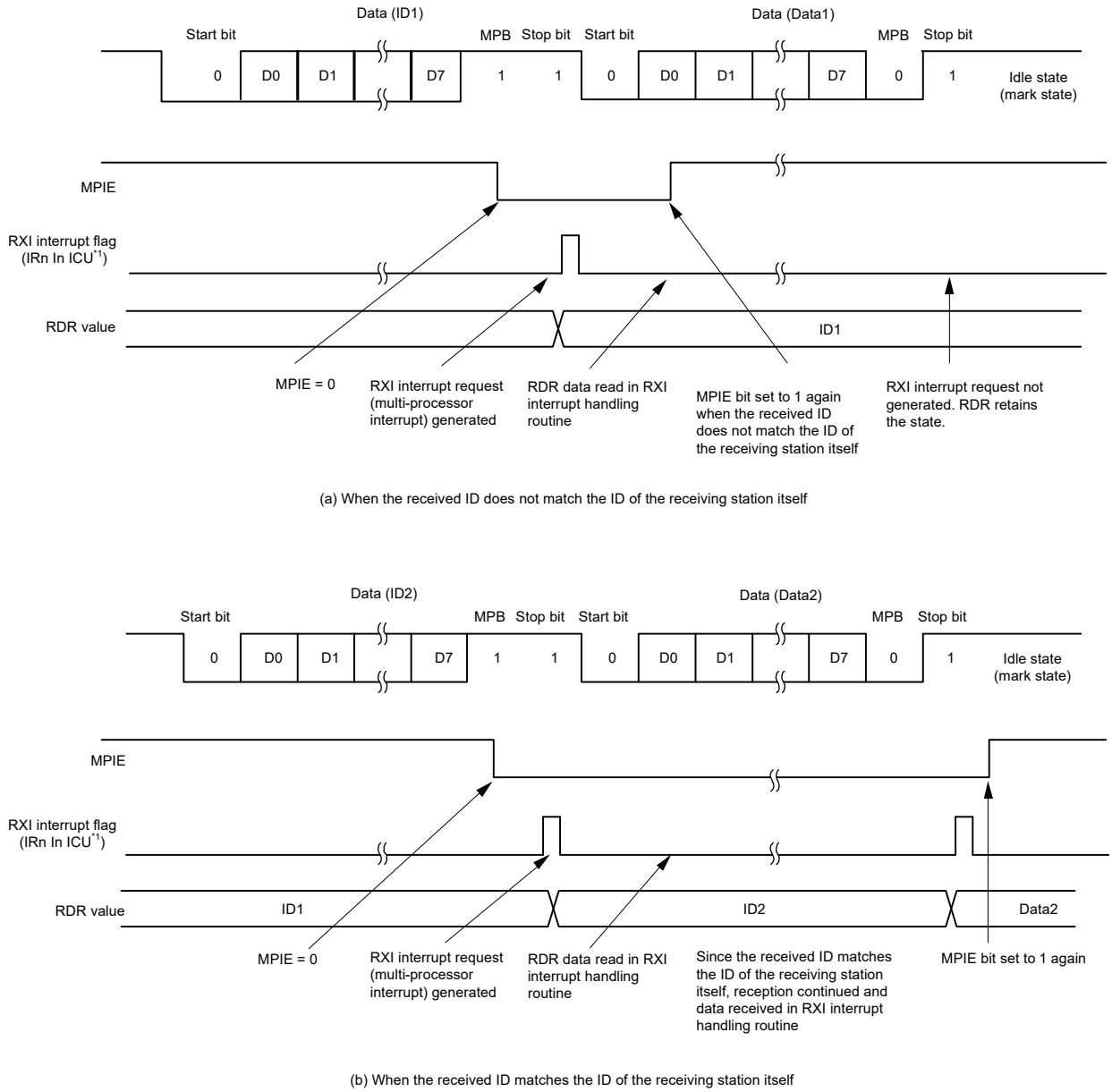


Figure 31.23 Example of Multi-Processor Serial Transmission Flowchart

31.4.2 Multi-Processor Serial Data Reception

Figure 31.25 and Figure 31.26 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 31.24 is the example of operation for reception.



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 31.24 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

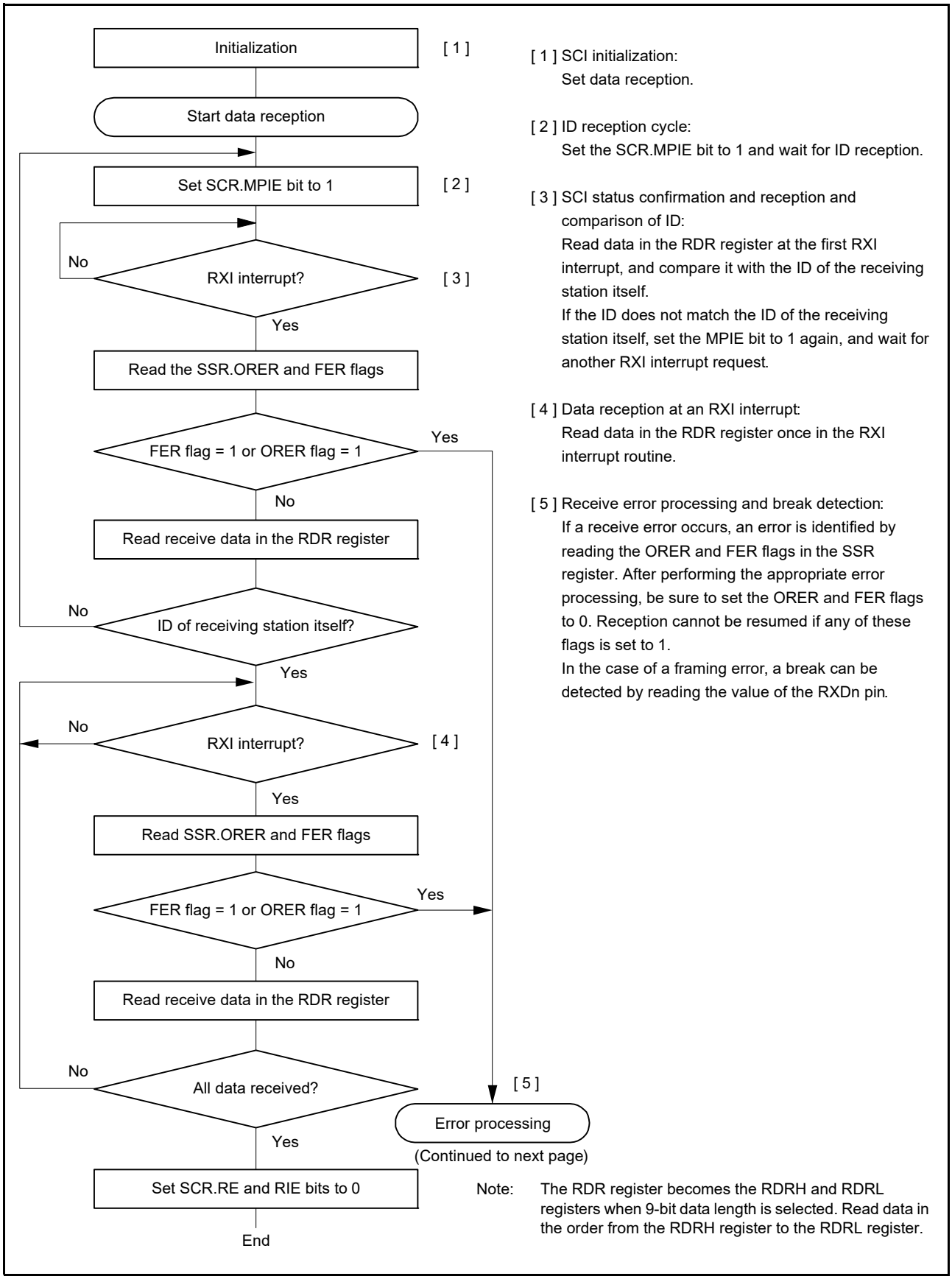


Figure 31.25 Example of Multi-Processor Serial Reception Flowchart (1)

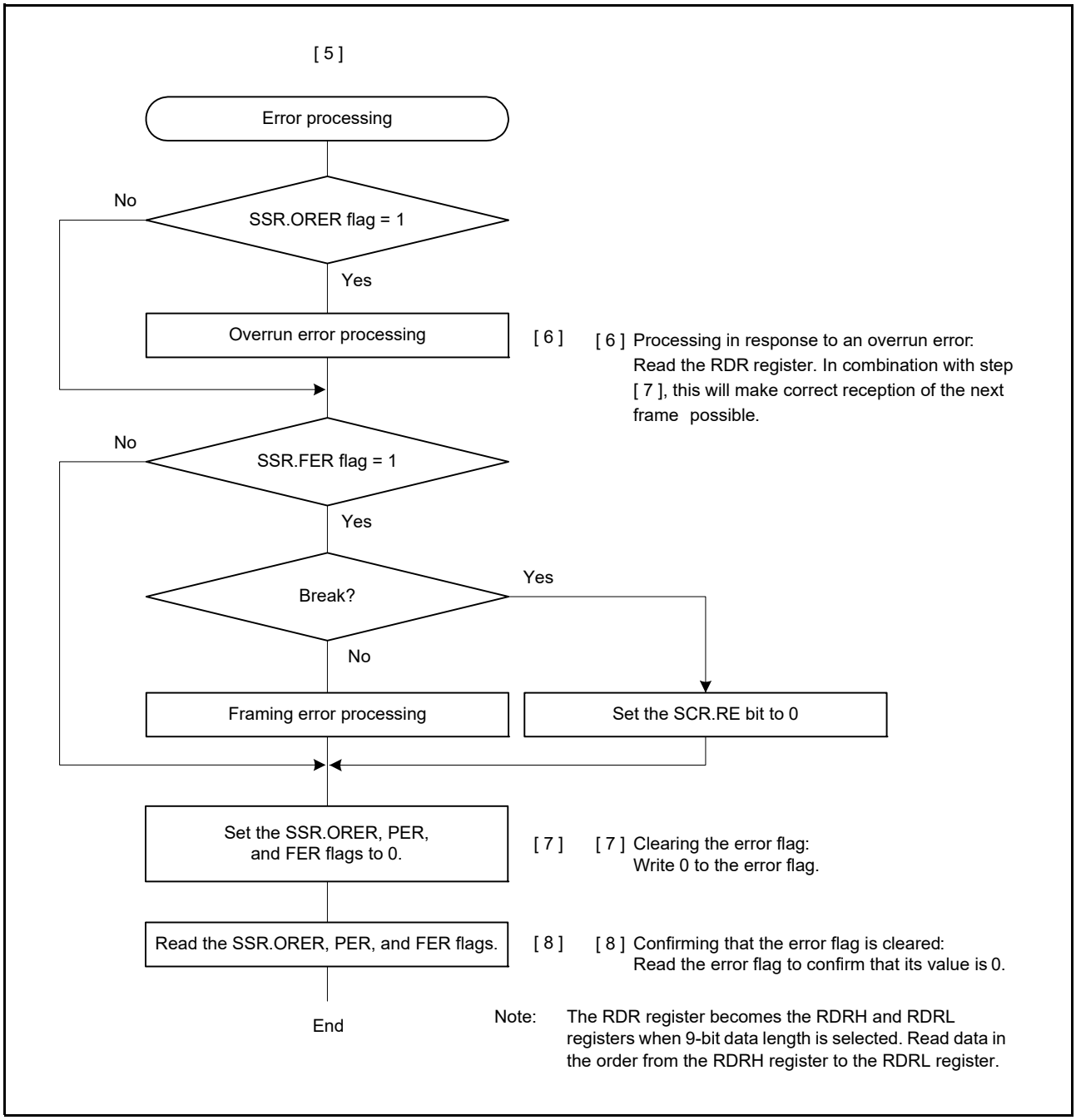


Figure 31.26 Example of Multi-Processor Serial Reception Flowchart (2)

31.5 Operation in Clock Synchronous Mode

Figure 31.27 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.

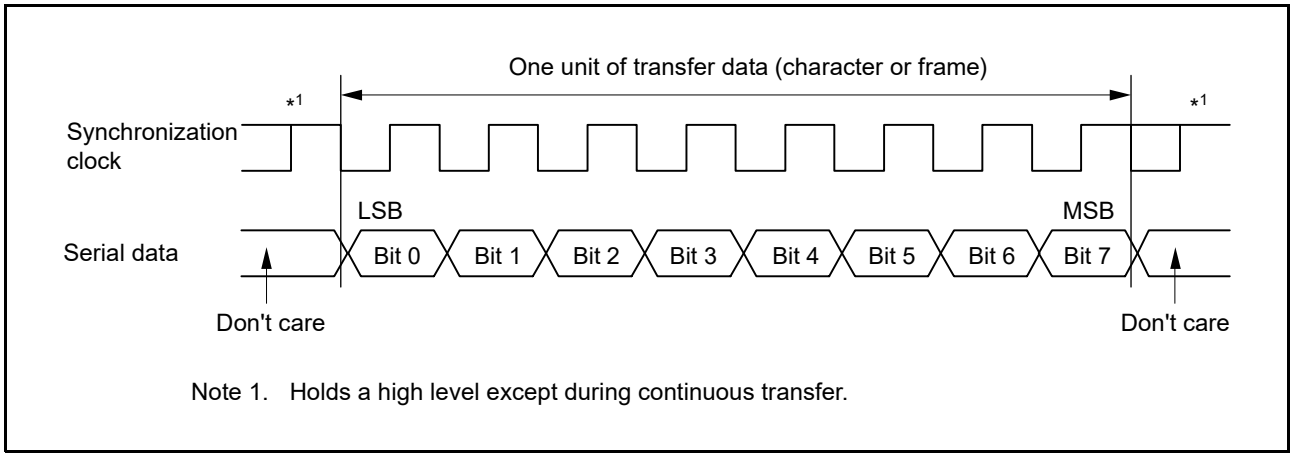


Figure 31.27 Data Format in Clock Synchronous Serial Communications (LSB First)

31.5.1 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.

31.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start. Applying the high level to the CTS# pin while reception/transmission are in progress does not affect reception/transmission of the current frame, which continues.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

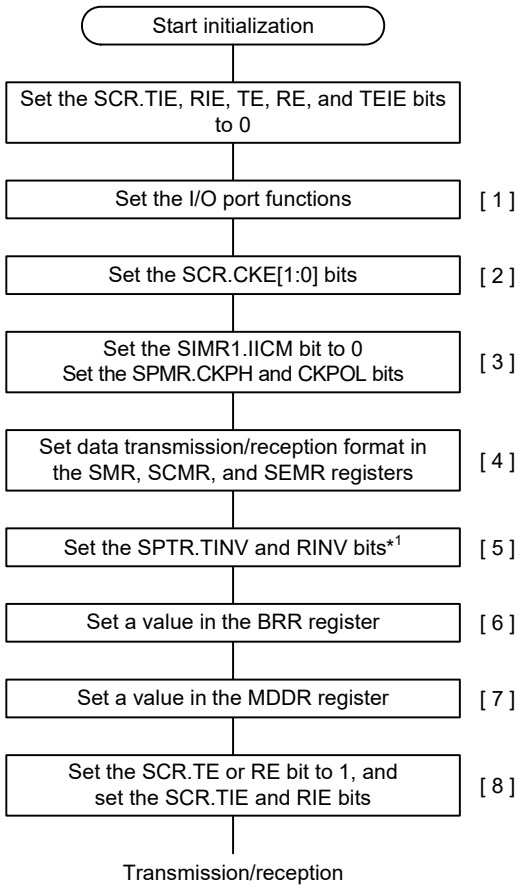
[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE or SCR.TE bit is 1.
- Transmission or reception 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.

31.5.3 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 31.28. 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 SCI1, SCI5, and SCI6.

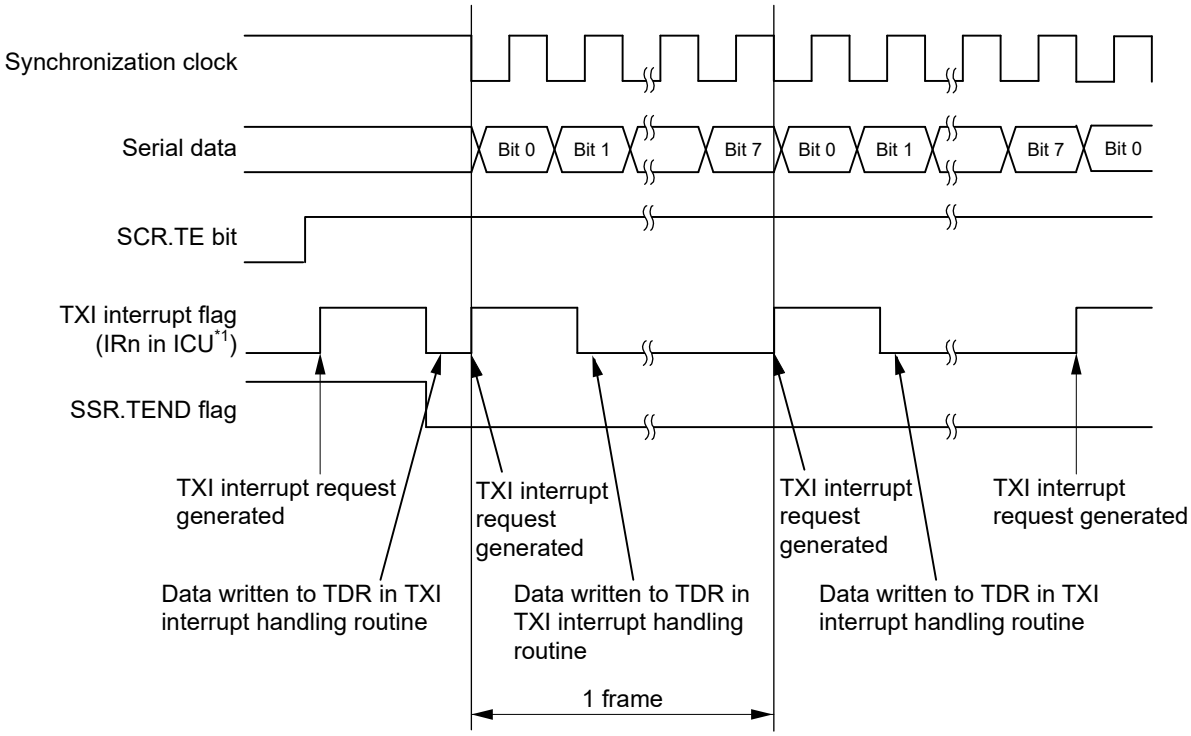
Figure 31.28 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

31.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 31.29, Figure 31.30, and Figure 31.31 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

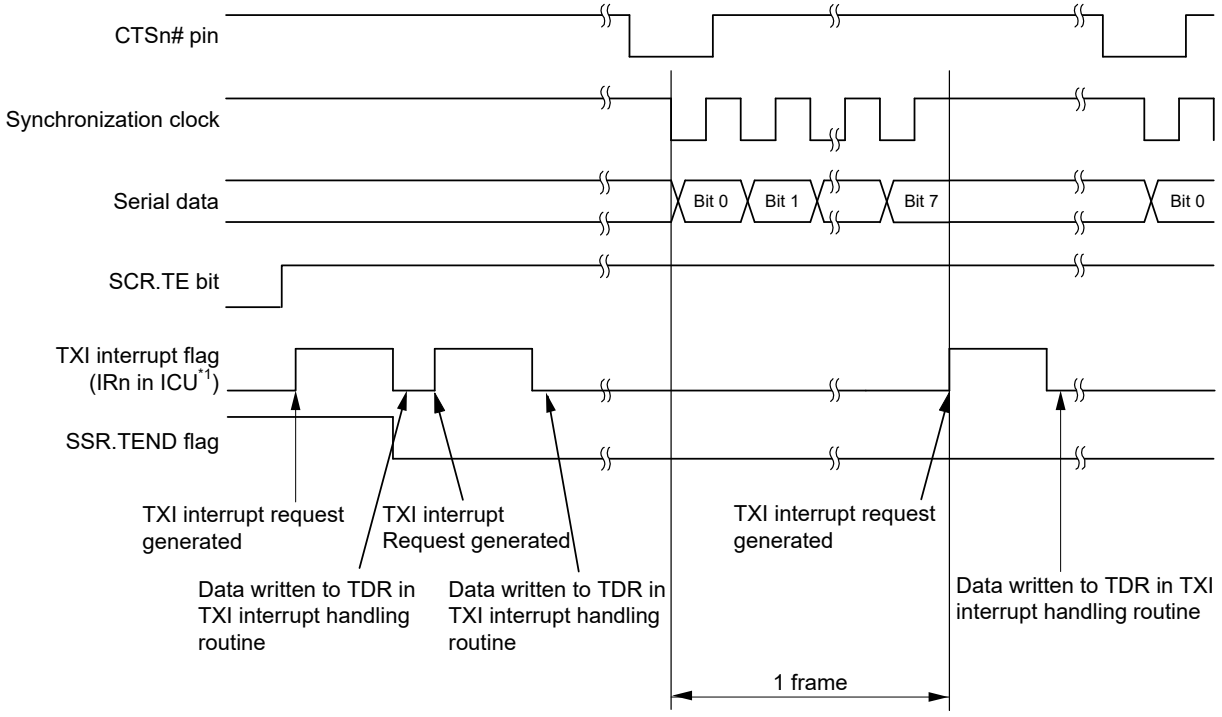
1. The SCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 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 31.32 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.



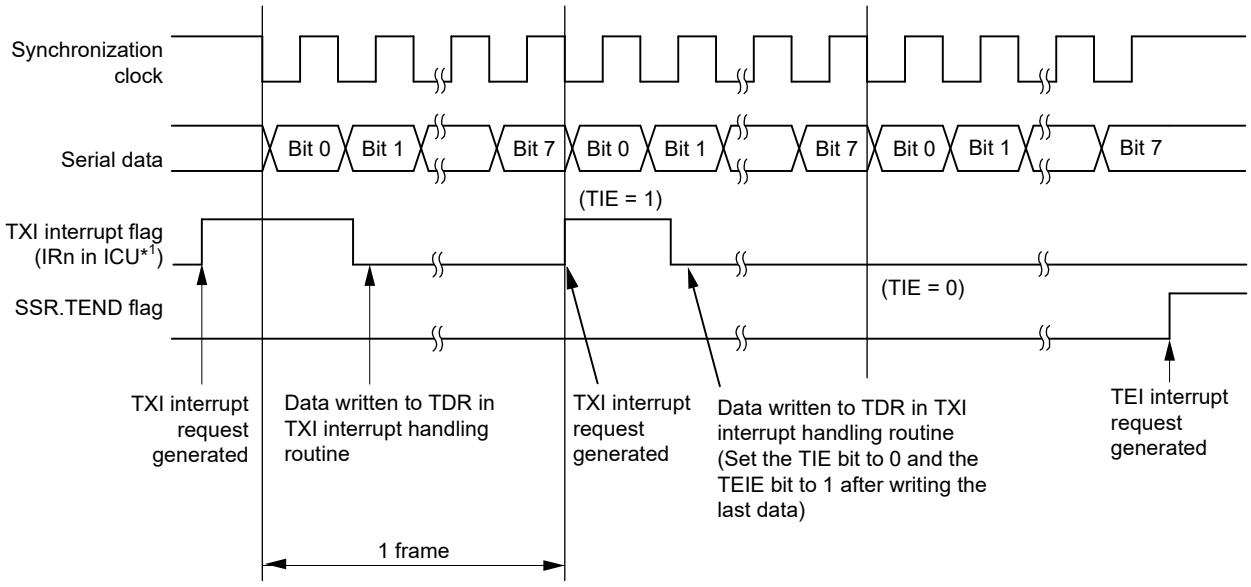
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 31.29 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



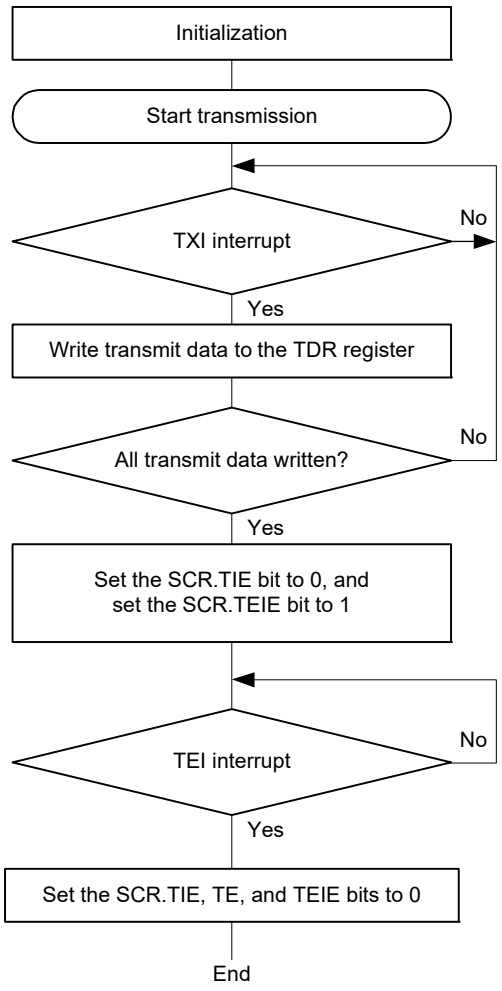
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 31.30 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 31.31 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.

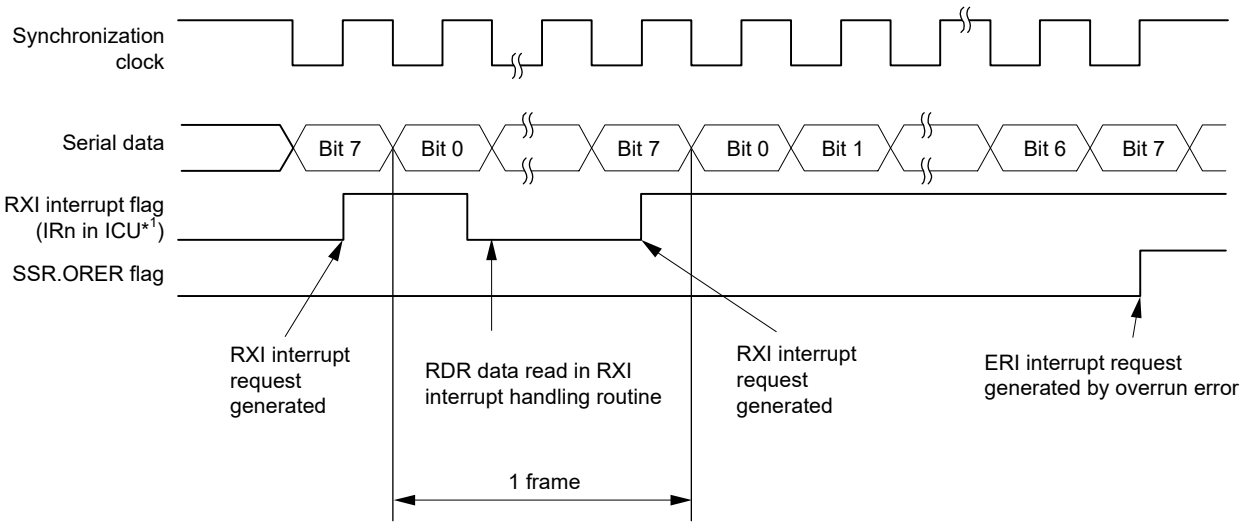
Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 31.32 Example Flowchart of Serial Transmission in Clock Synchronous Mode

31.5.5 Serial Data Reception (Clock Synchronous Mode)

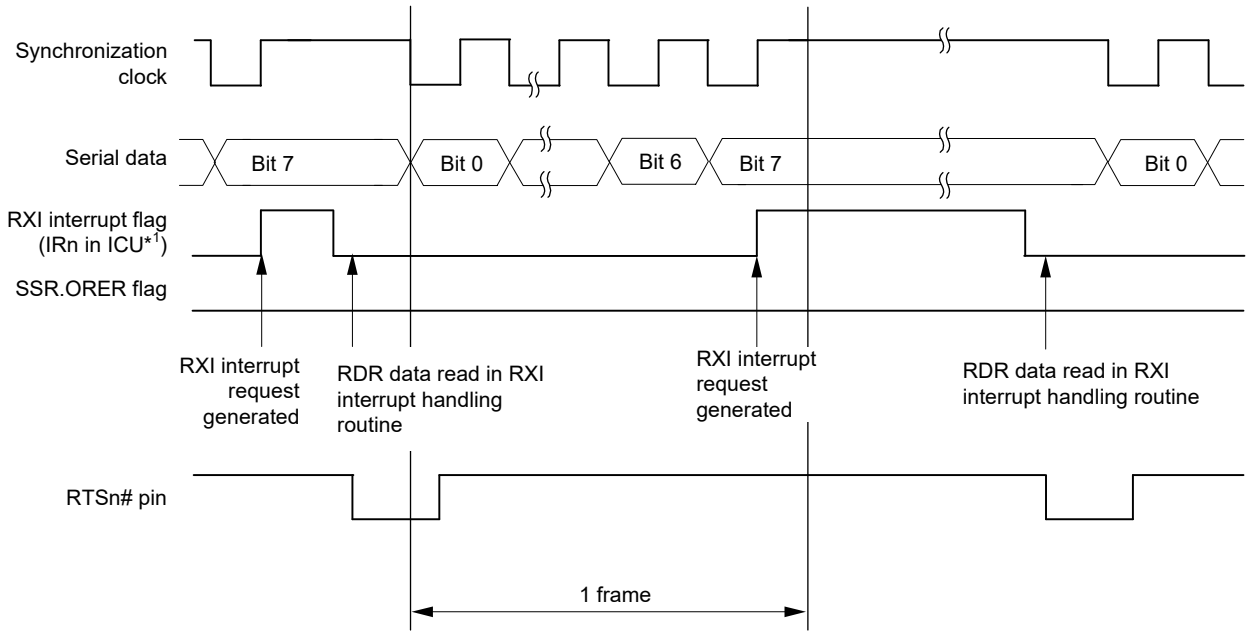
Figure 31.33 and Figure 31.34 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

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 (ICUb) for details on the corresponding interrupt vector number.

Figure 31.33 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

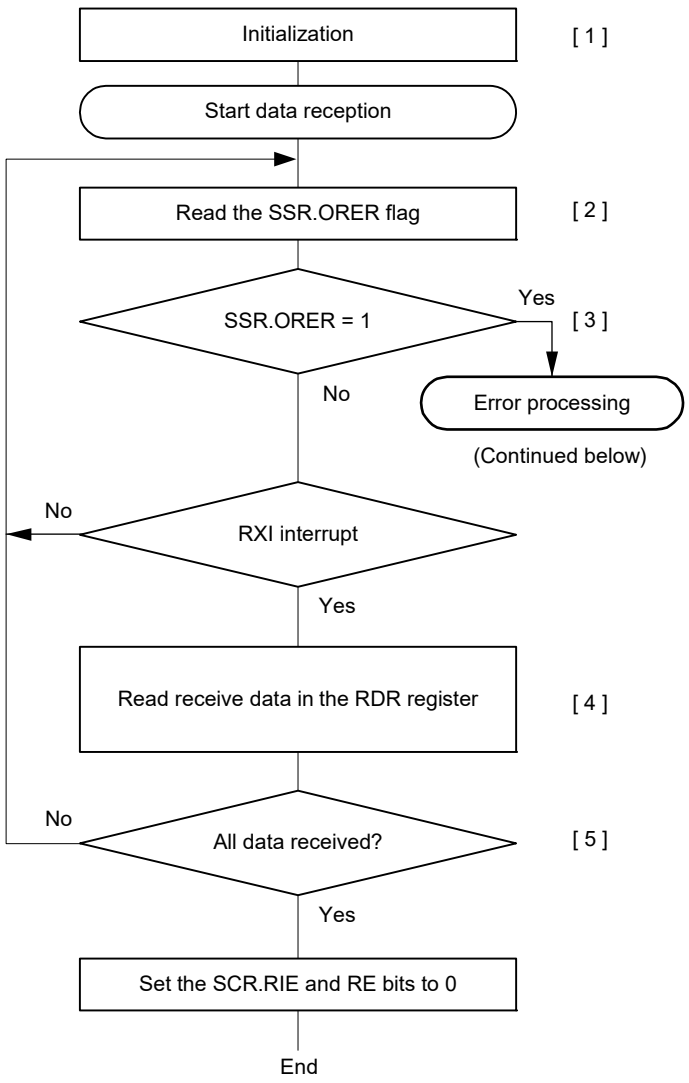


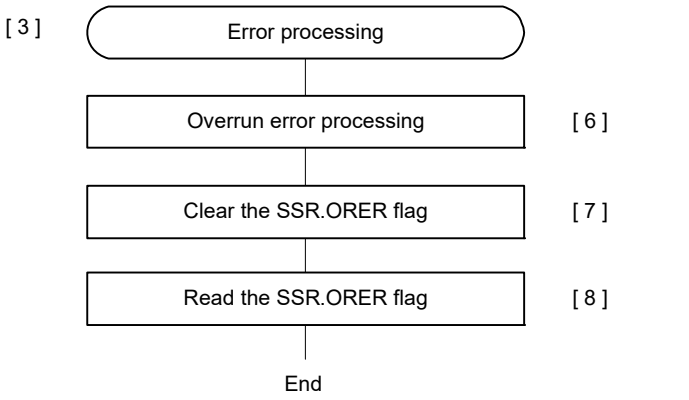
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 31.34 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 31.35 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 31.35 Example Flowchart of Serial Reception in Clock Synchronous Mode

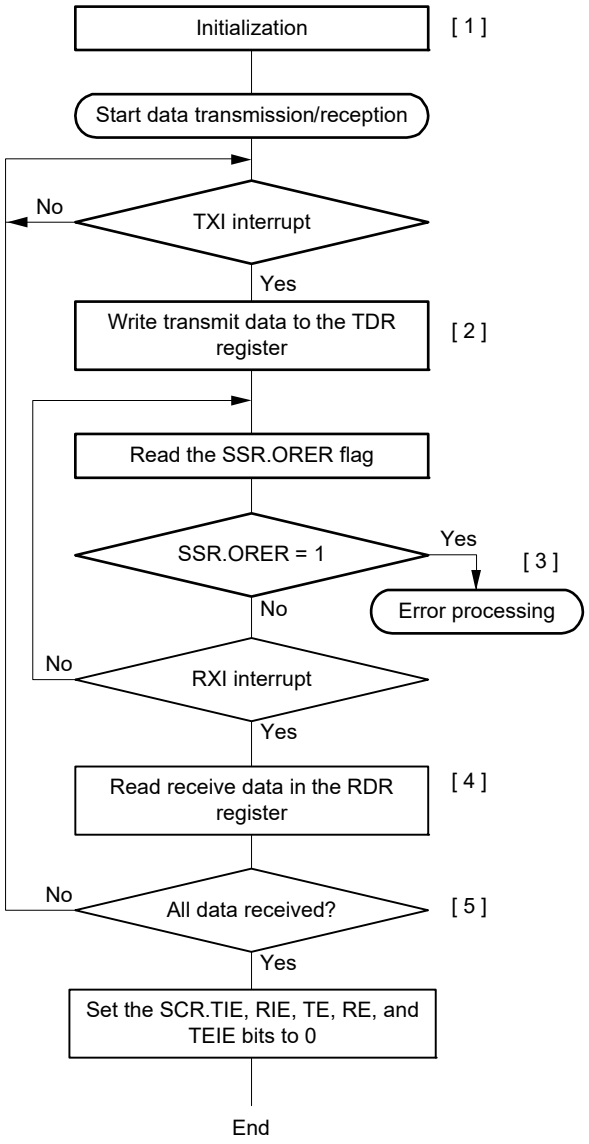
31.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 31.36 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 31.36 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

31.6 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.

31.6.1 Sample Connection

Figure 31.37 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.

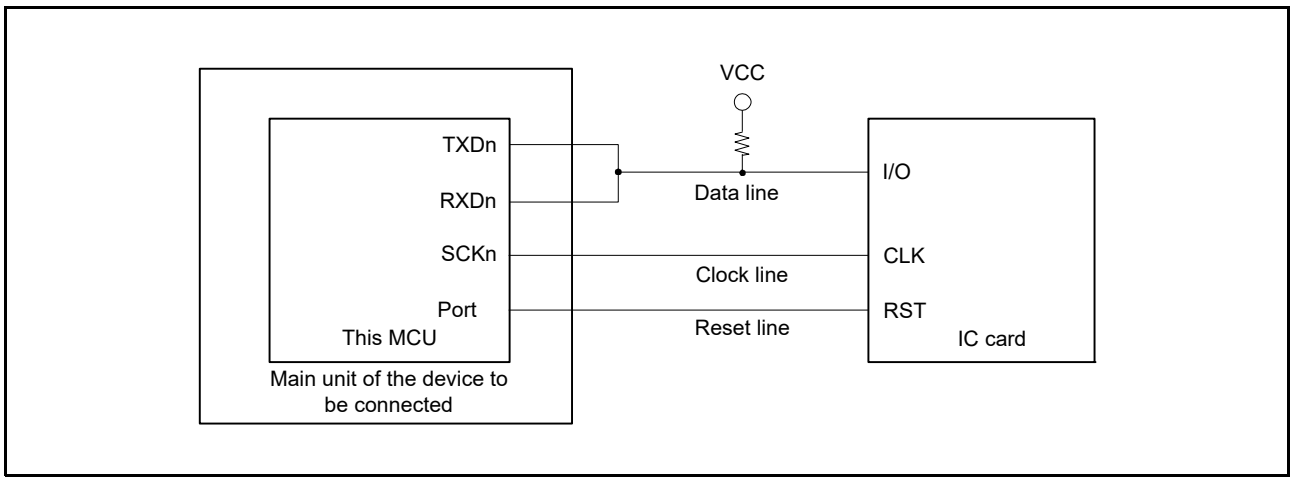


Figure 31.37 Sample Connection with a Smart Card (IC Card)

31.6.2 Data Format (Except in Block Transfer Mode)

Figure 31.38 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.

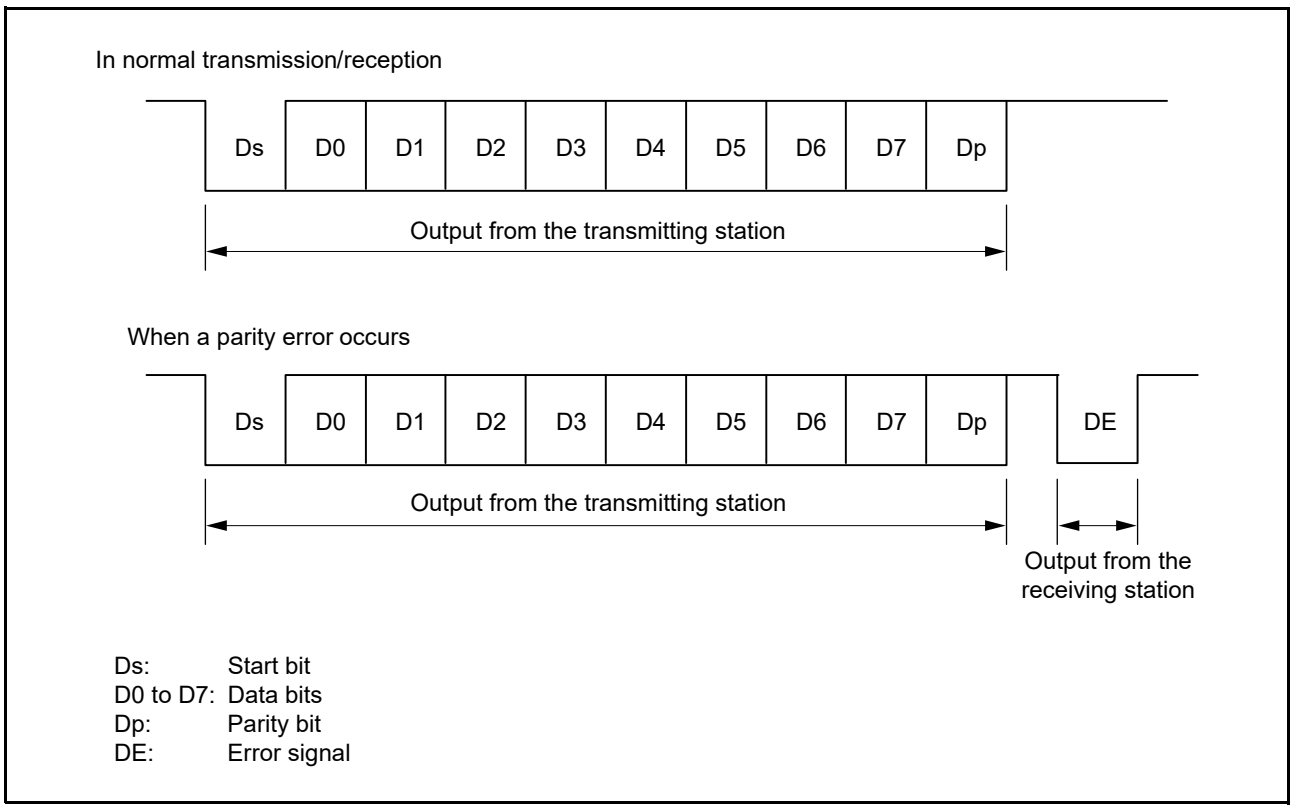


Figure 31.38 Data Formats in Smart Card Interface Mode

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 31.39. 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.

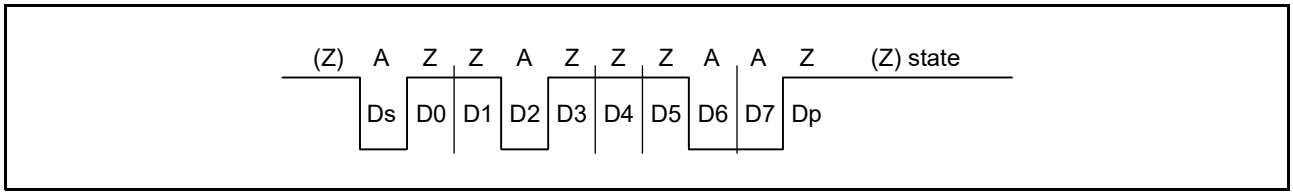


Figure 31.39 Direct Convention (SCMR.SDIR bit = 0, SCMR.SINV bit = 0, SMR.PM bit = 0)

(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 31.40. 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.

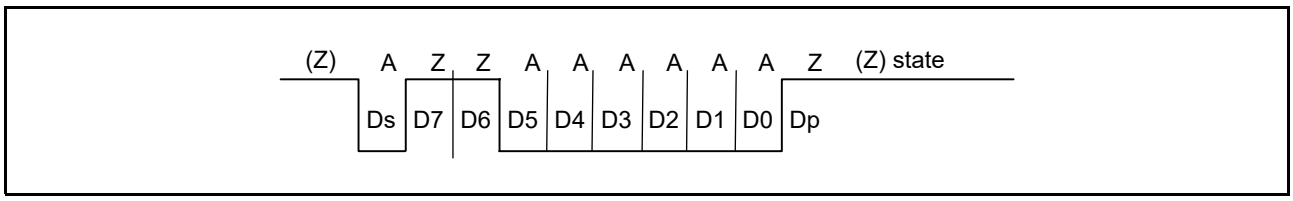


Figure 31.40 Inverse Convention (SCMR.SDIR bit = 1, SCMR.SINV bit = 1, SMR.PM bit = 1)

31.6.3 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.

31.6.4 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 31.41. 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 (%)

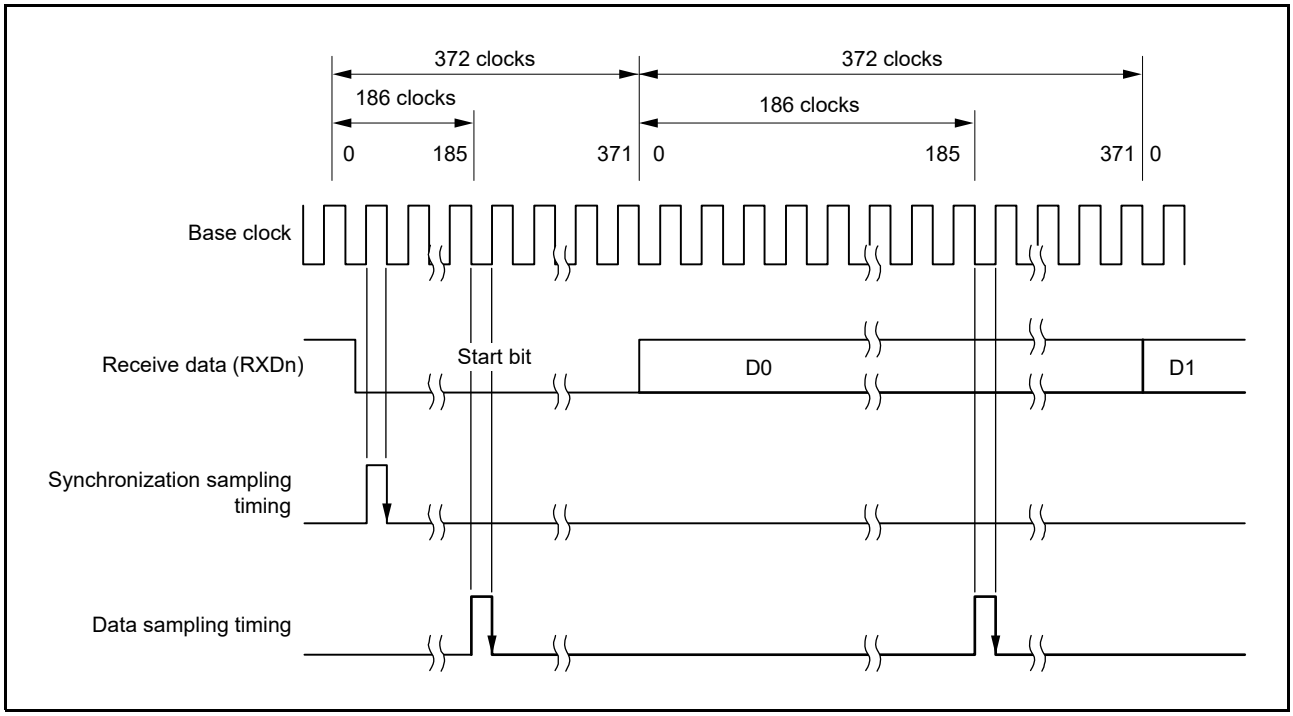
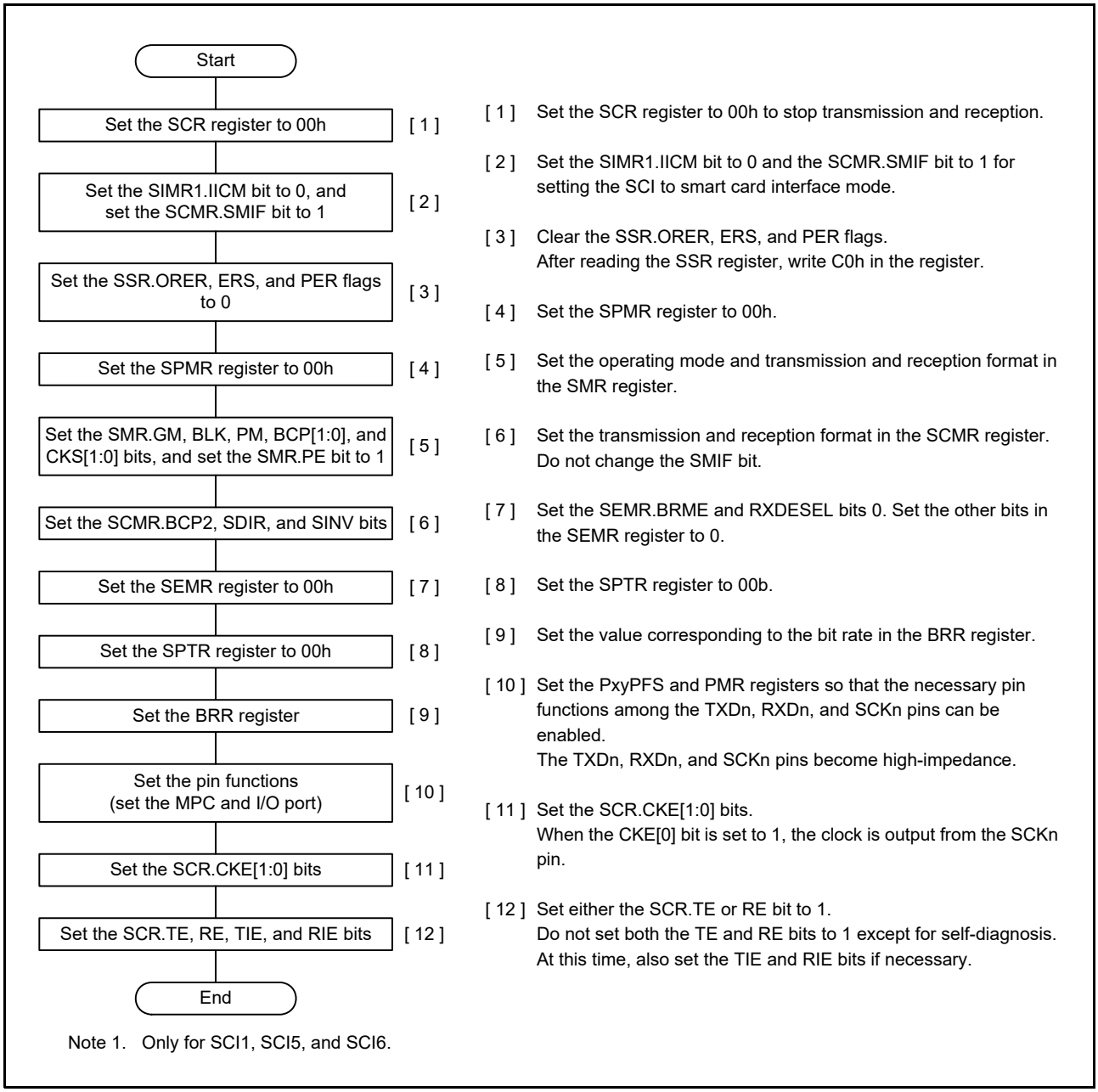


Figure 31.41 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

31.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 31.42.
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 31.42. 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 31.42. Set TE = 0 and RE = 1 at step [11]. Transmission completion can be verified by reading the SSR.TEND flag.



Note 1. Only for SCI1, SCI5, and SCI6.

Figure 31.42 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

Figure 31.43 shows an example of data transmission when the SCI is set to smart card interface mode according to the flow described in Figure 31.42 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.

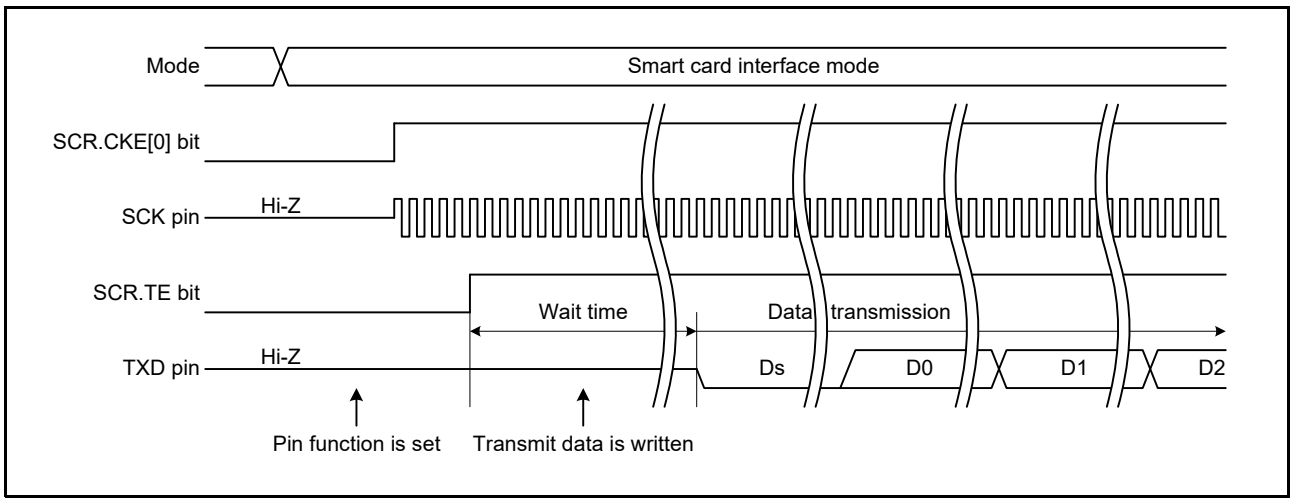


Figure 31.43 Example of Data Transmission Timing in Smart Card Interface Mode

31.6.6 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 31.44 shows the data retransmit operation during transmission.

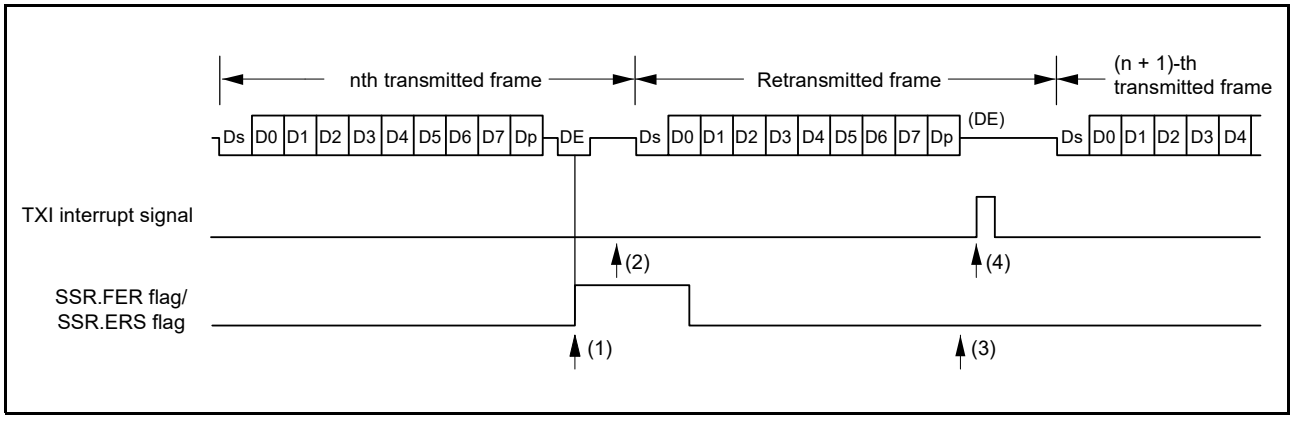


Figure 31.44 Data Retransmit Operation in SCI Transmit Mode

(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 31.45 shows a sample flowchart of serial transmission.

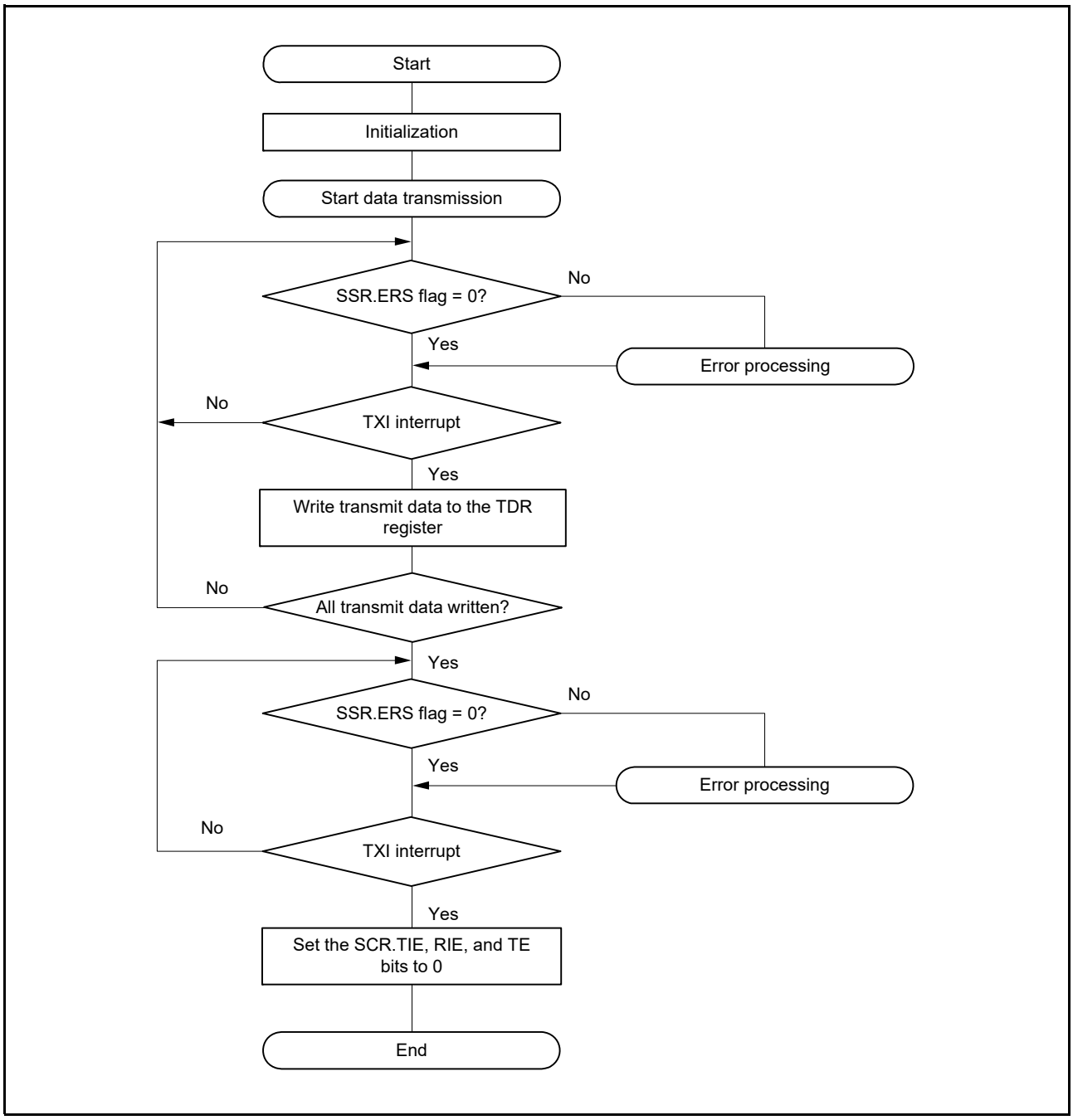


Figure 31.45 Sample Smart Card Interface Transmission Flowchart

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 (DMACA).

Note that the SSR.TEND flag is set in different timings depending on the SMR.GM bit setting. Figure 31.46 shows the TEND flag generation timing.

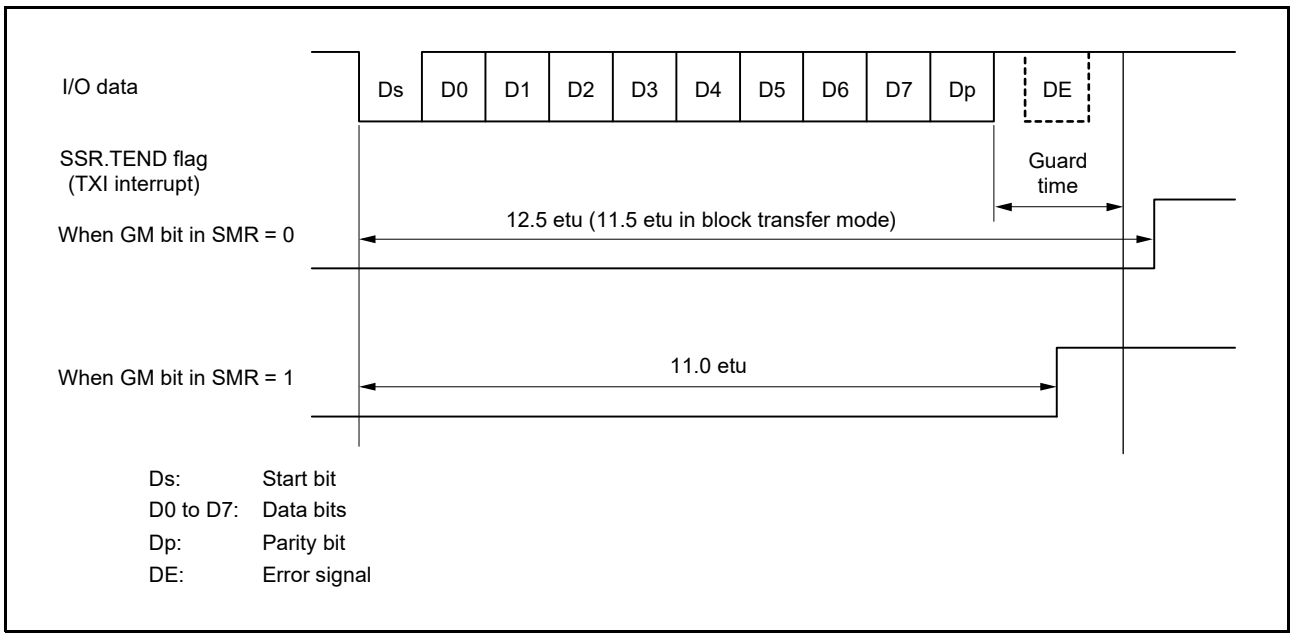


Figure 31.46 SSR.TEND Flag Generation Timing during Transmission

31.6.7 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 31.47 shows the data retransmit operation in receive mode.

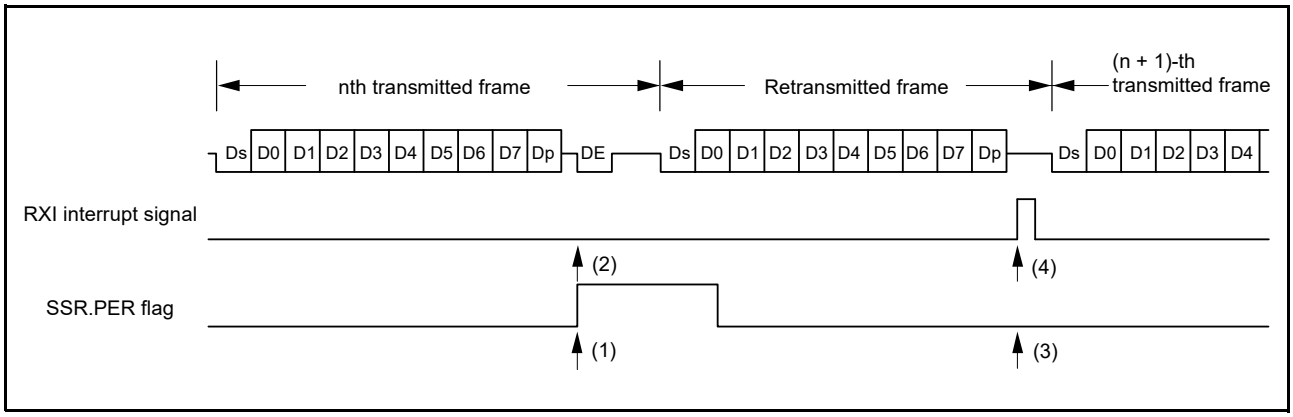


Figure 31.47 Data Retransmit Operation in SCI Receive Mode (Data Retransmit Operation during Reception)

(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 31.48 shows a sample flowchart for serial data reception.

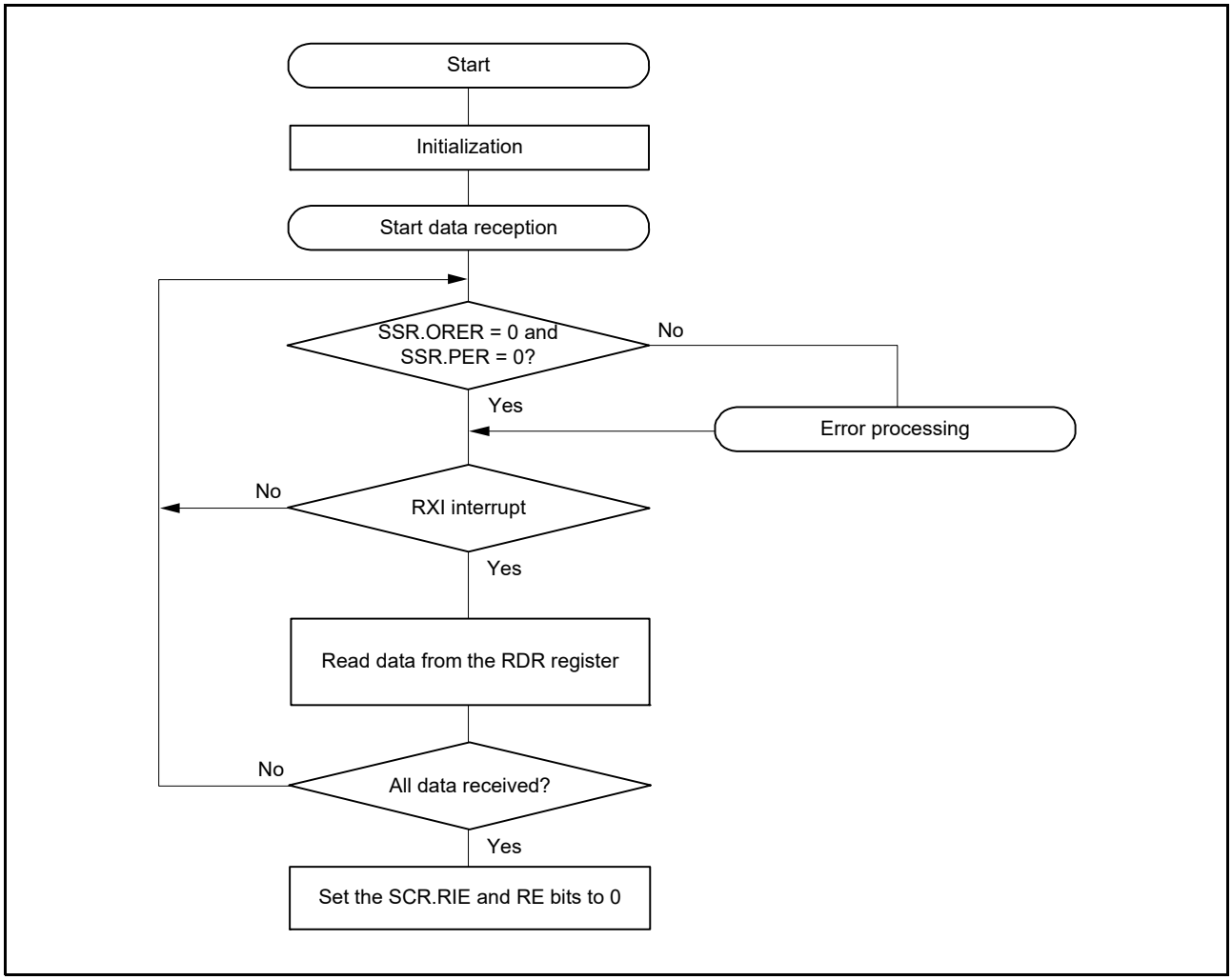


Figure 31.48 Sample Smart Card Interface Reception Flowchart

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 31.3, Operation in Asynchronous Mode.

31.6.8 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 31.2.11, 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 31.49 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.

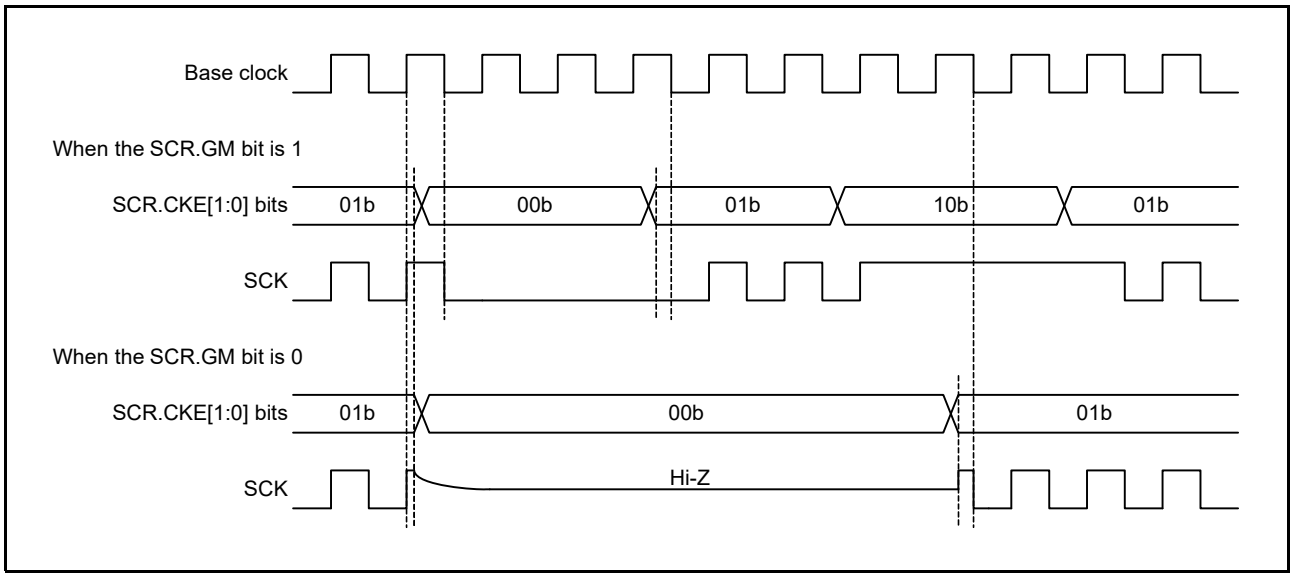


Figure 31.49 Clock Output Control

31.7 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 31.50 and Figure 31.51.

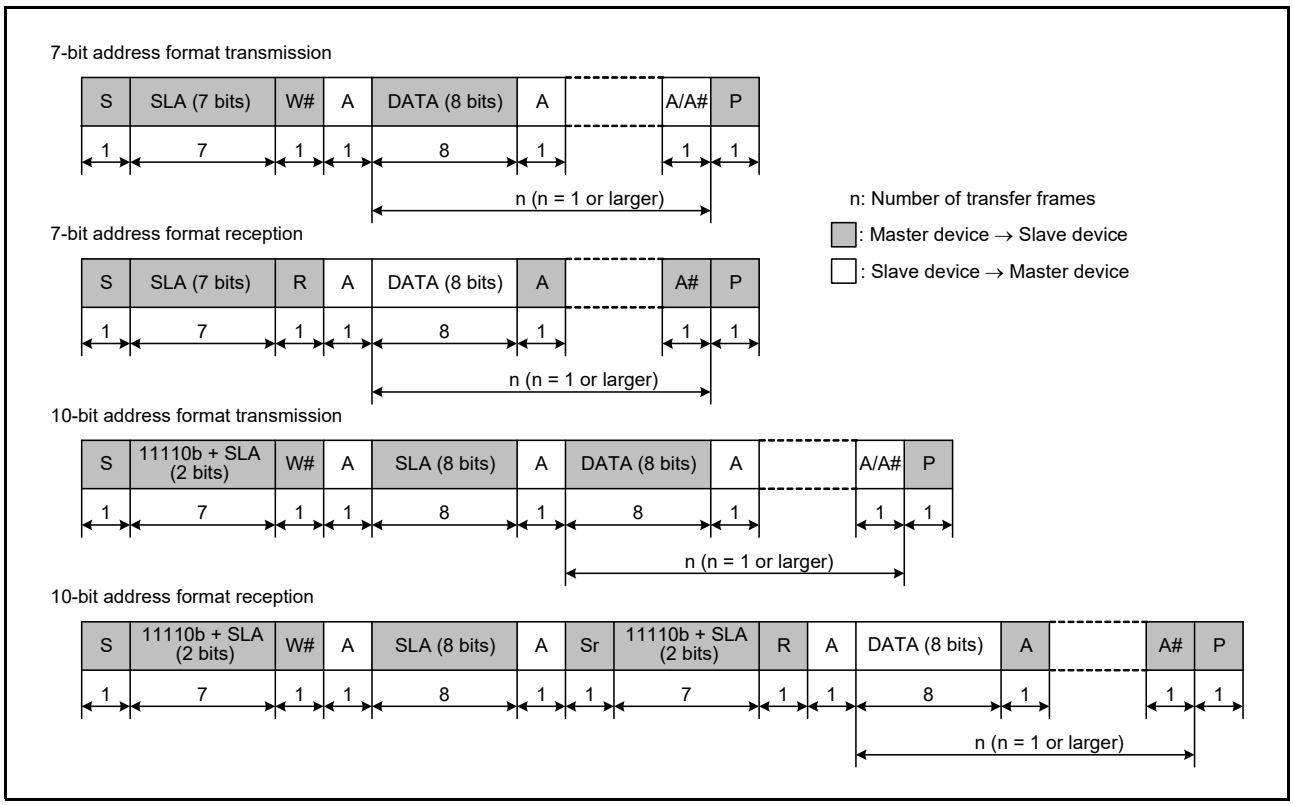


Figure 31.50 I2C-bus Format

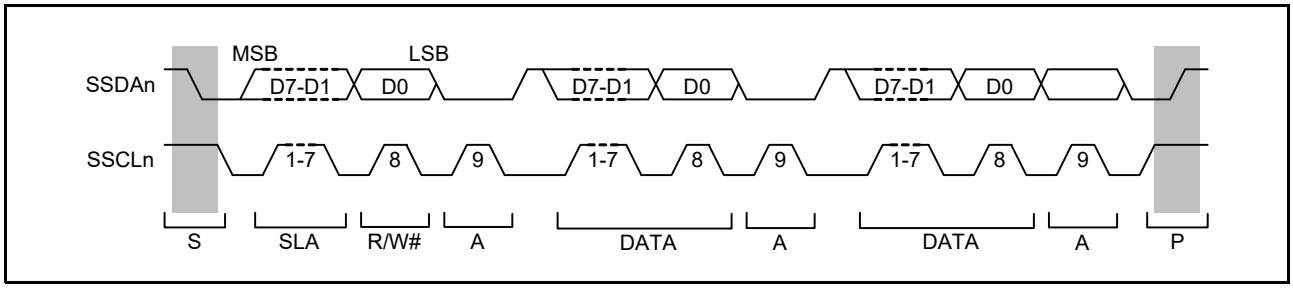


Figure 31.51 I2C-bus Timing (When SLA is 7 Bits)

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.

31.7.1 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 31.52 shows the timing of operations in the generation of start, restart, and stop conditions.

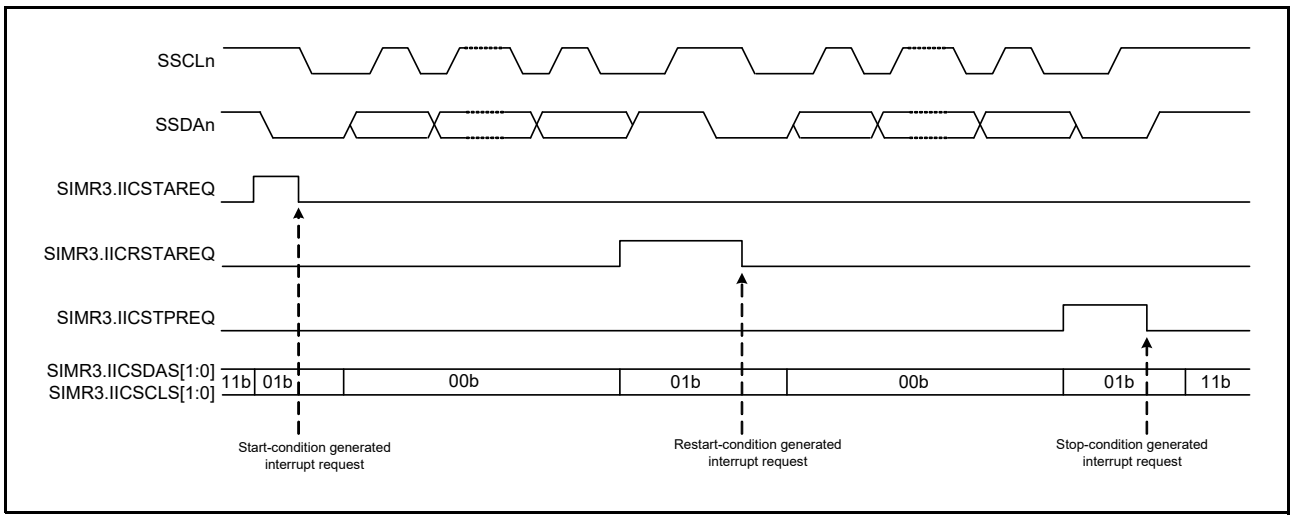


Figure 31.52 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

31.7.2 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 31.53 shows an example of operations to synchronize the clocks.

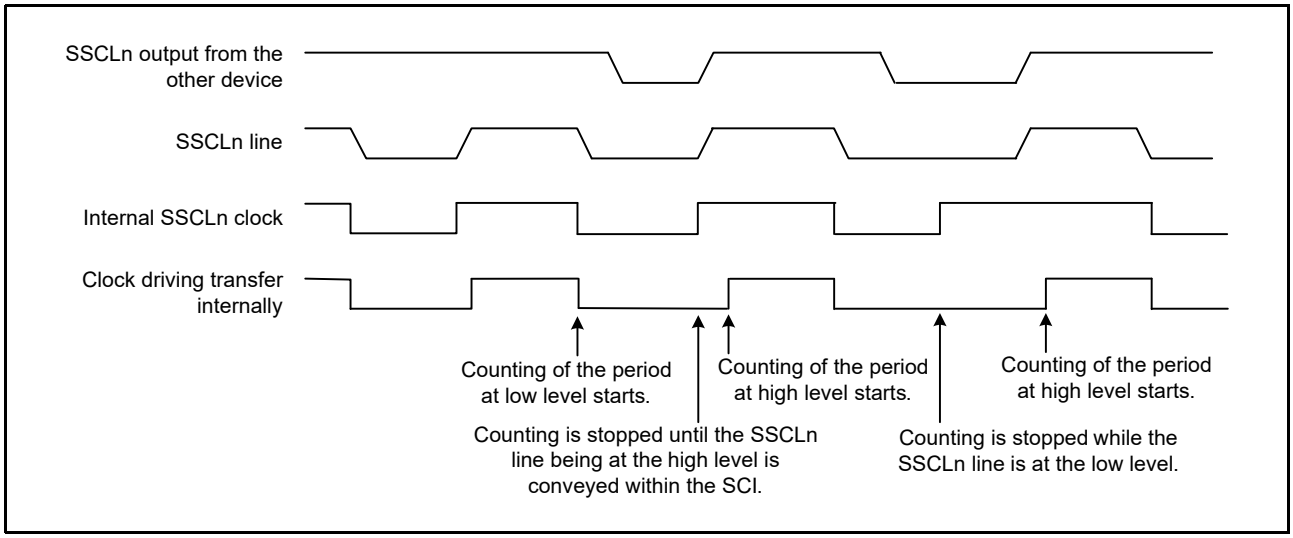


Figure 31.53 Example of Operations for Clock Synchronization

31.7.3 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 31.54 shows the timing of delays in SSDA output.

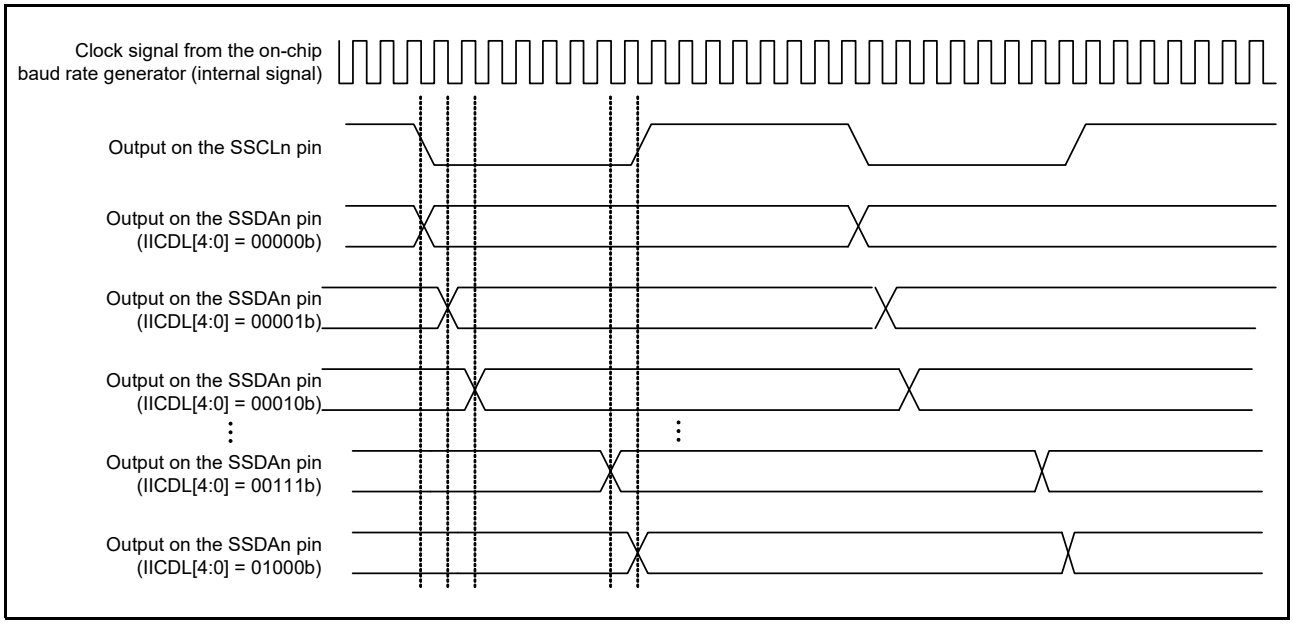


Figure 31.54 Timing of Delays in SSDA Output

31.7.4 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 31.55.
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.

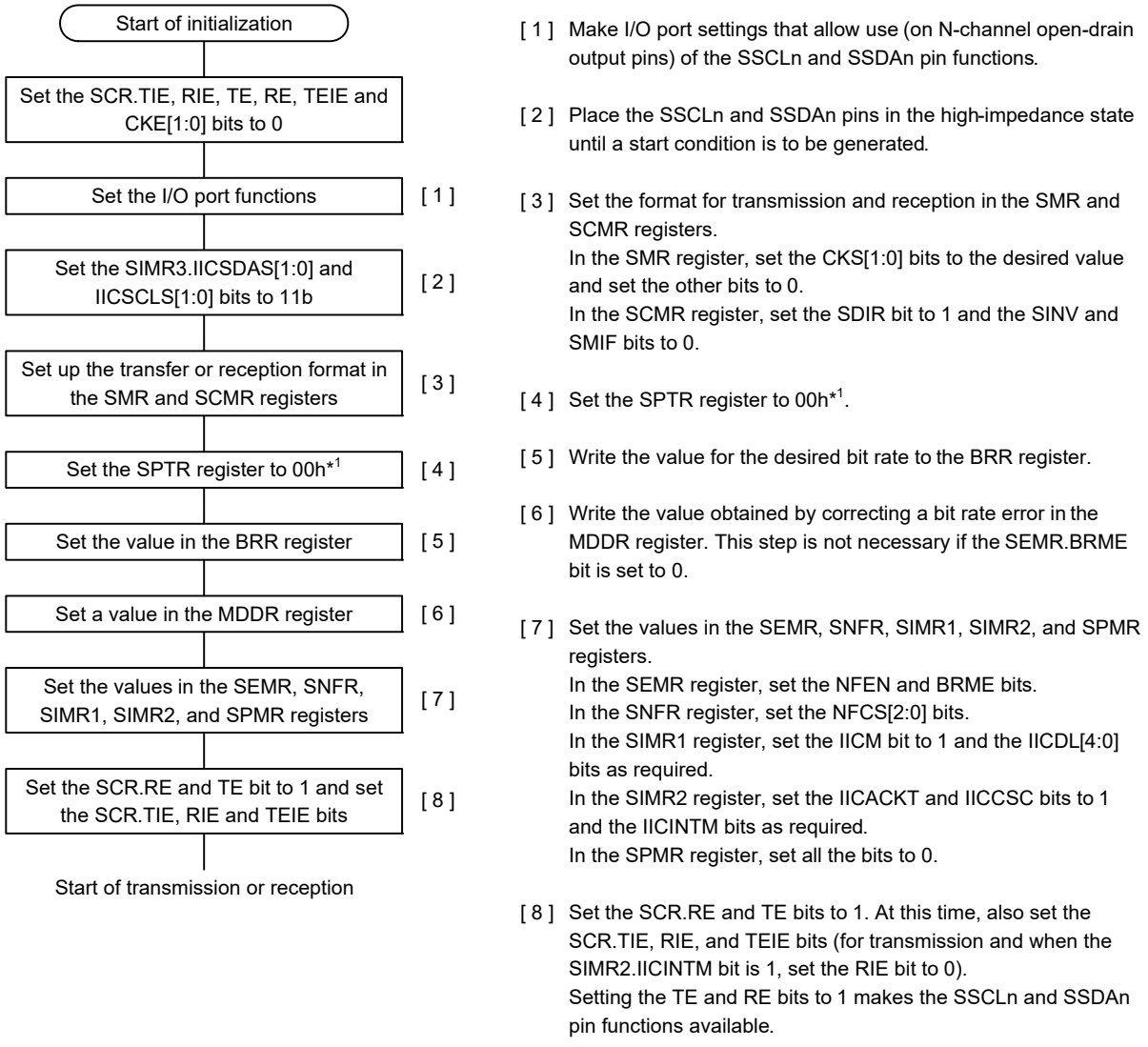
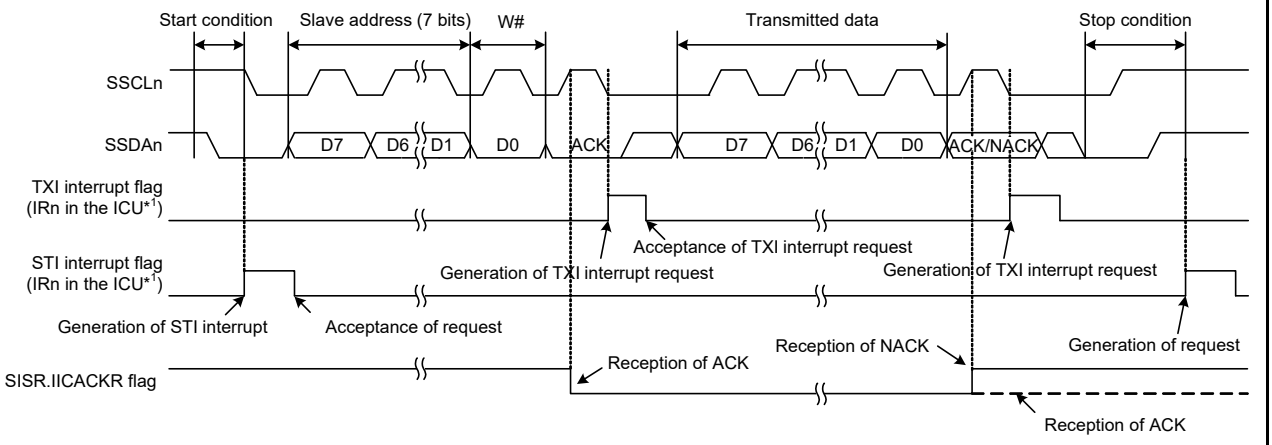


Figure 31.55 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

31.7.5 Operation in Master Transmission (Simple I2C Mode)

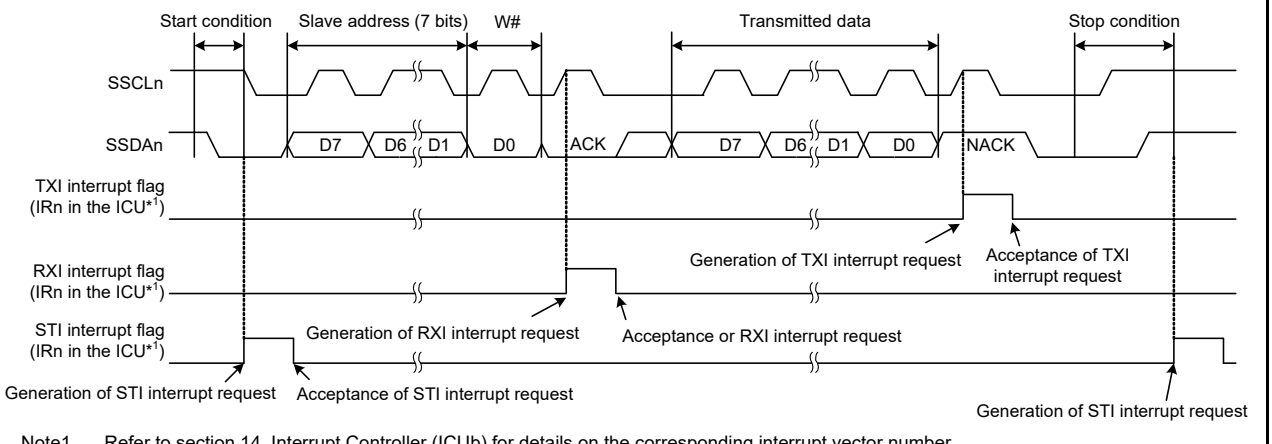
Figure 31.56 and Figure 31.57 show examples of operations in master transmission and Figure 31.58 is a flowchart showing the procedure for data transmission. Refer to Table 31.39 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 31.58 are repeated 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 (ICUb) for details on the corresponding interrupt vector number.

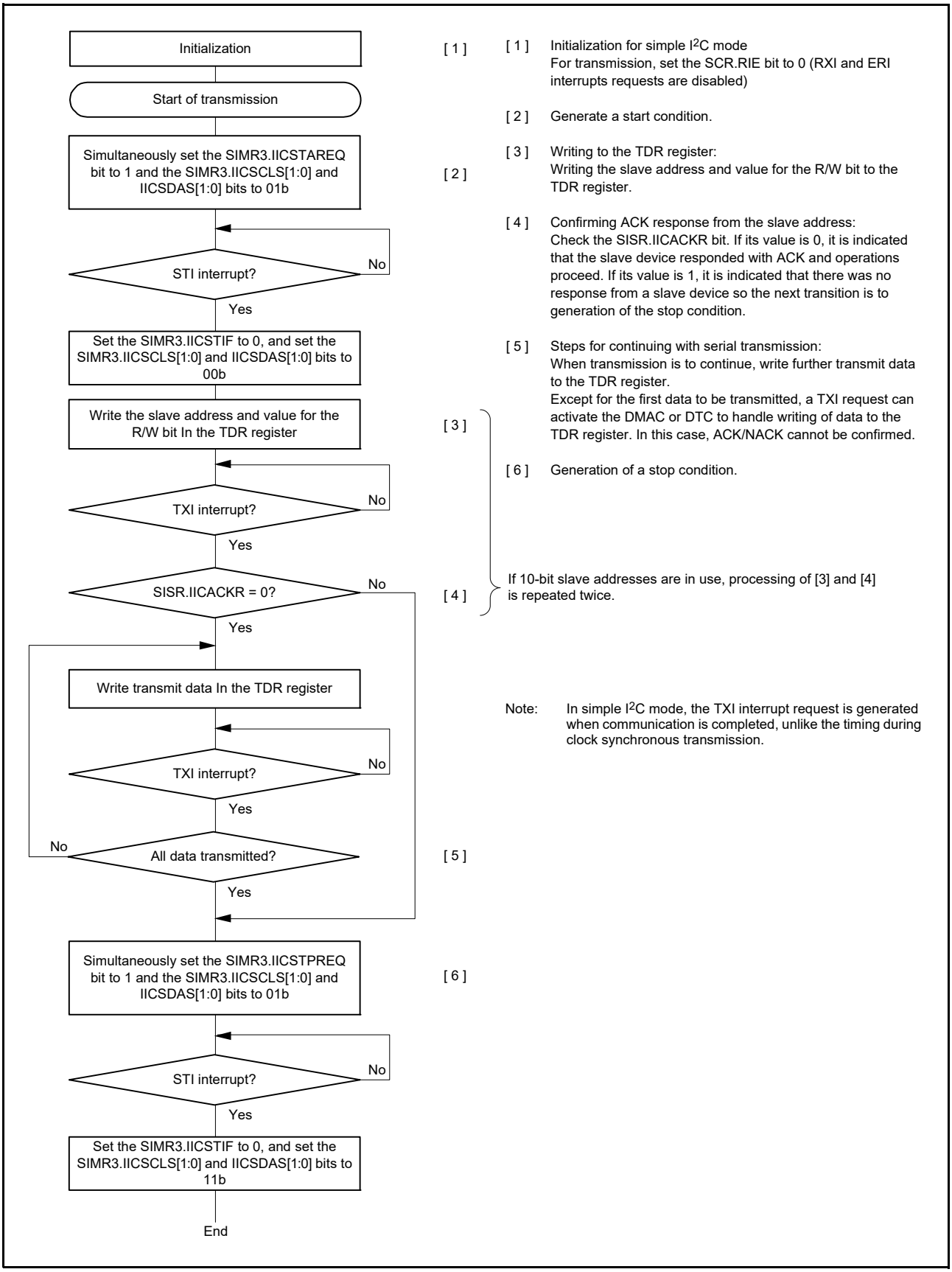
Figure 31.56 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

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 (ICUb) for details on the corresponding interrupt vector number.

Figure 31.57 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] 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:
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.

[6] Generation of a stop condition.

[3] and [4]: If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 31.58 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

31.7.6 Master Reception (Simple I2C Mode)

Figure 31.59 shows an example of operations in simple I2C mode master reception and Figure 31.60 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.

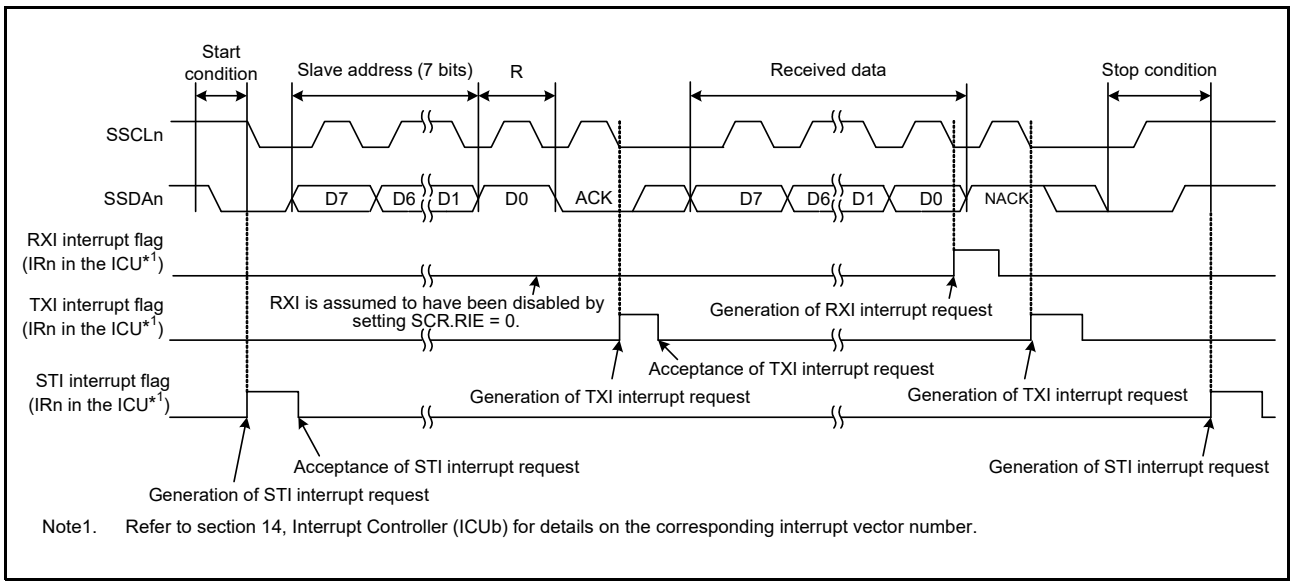
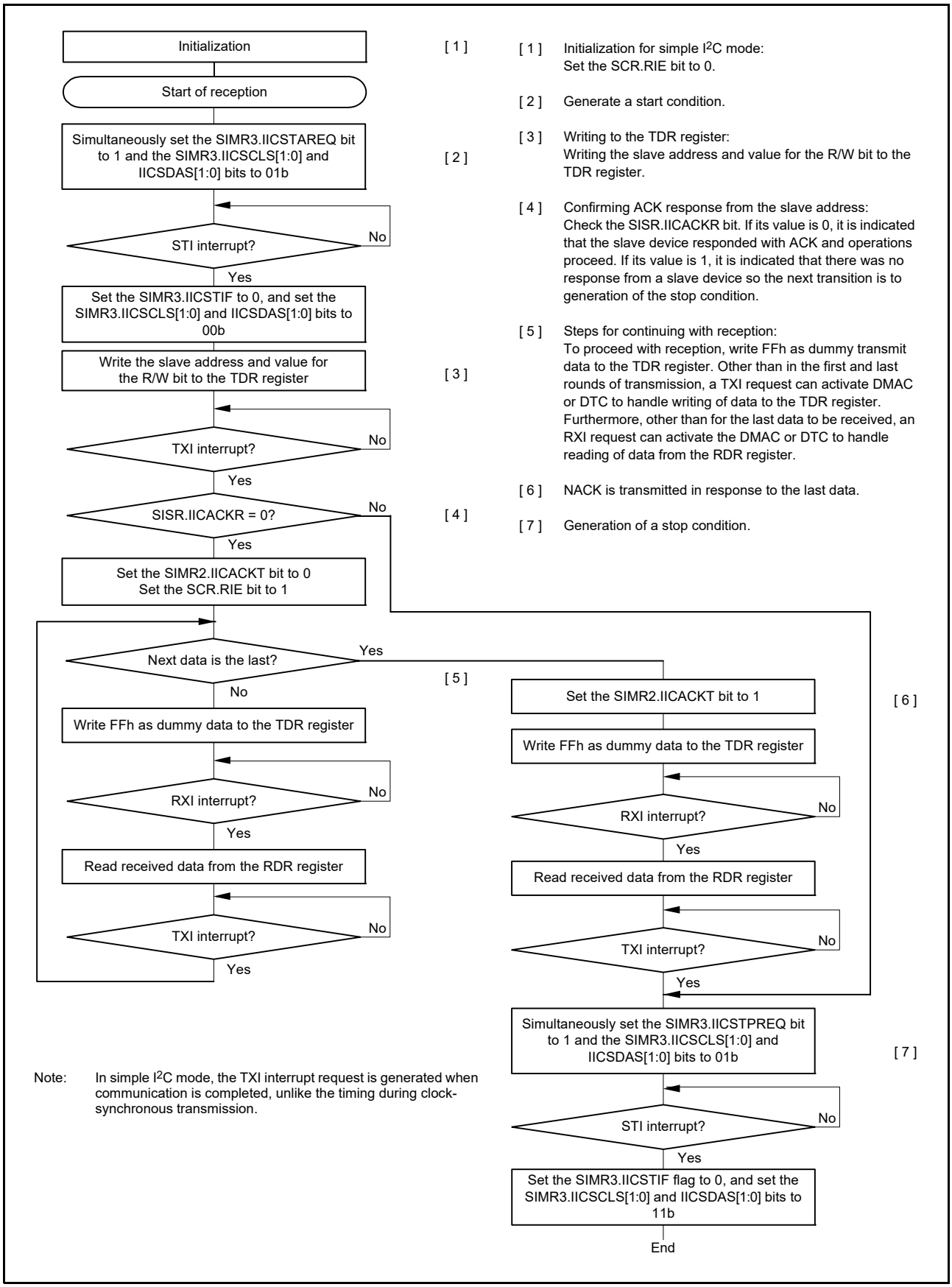


Figure 31.59 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[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 31.60 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

31.7.7 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.

31.8 Operation in Simple SPI Mode

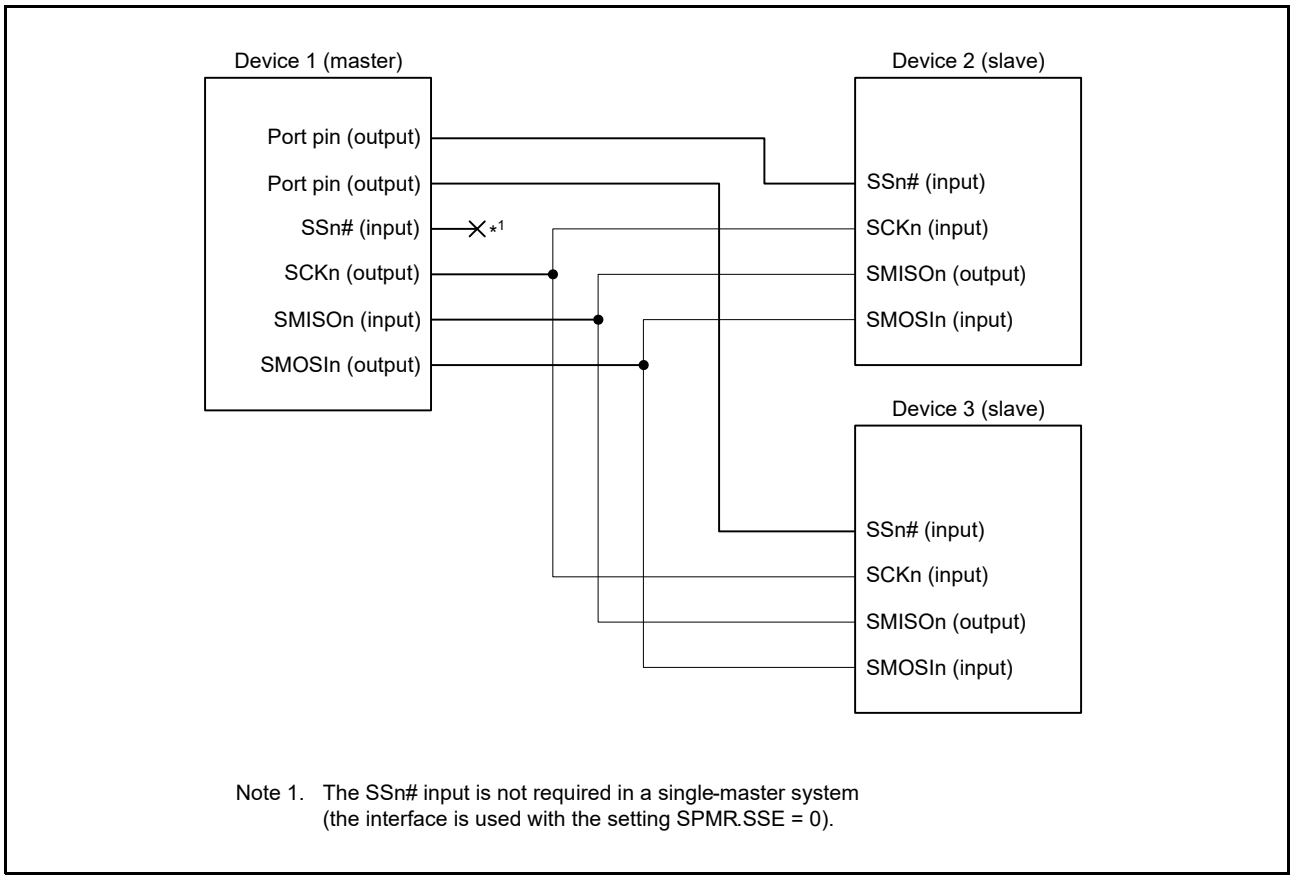
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

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 31.61 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 31.61 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

31.8.1 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 31.35 lists the states of pins according to the mode and the level on the SSn# pin.

Table 31.35 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).

31.8.2 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.

31.8.3 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.

31.8.4 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 31.62. The relation is the same for both master and slave operation.

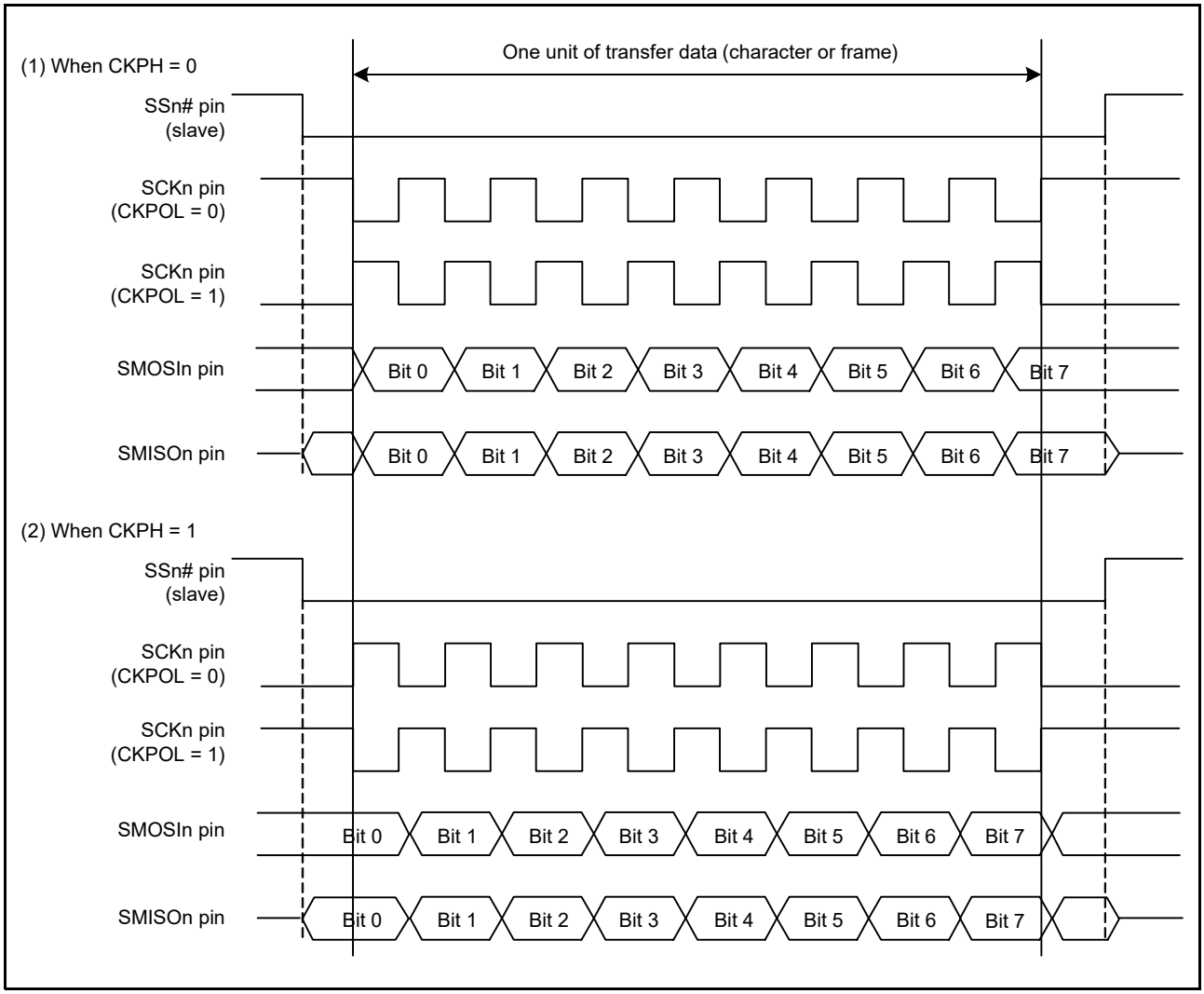


Figure 31.62 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

31.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 31.28, 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.

31.8.6 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.

31.9 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 31.63 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.

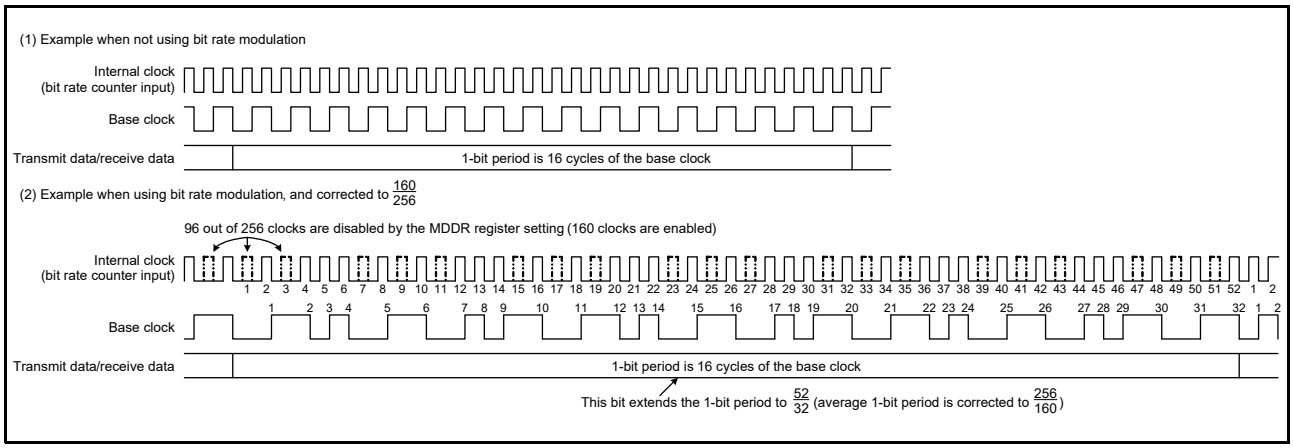


Figure 31.63 Example of the Base Clock When the Bit Rate Modulation Function is Used

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.

31.10 Extended Serial Mode Control Section: Description of Operation

31.10.1 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 31.64.
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.

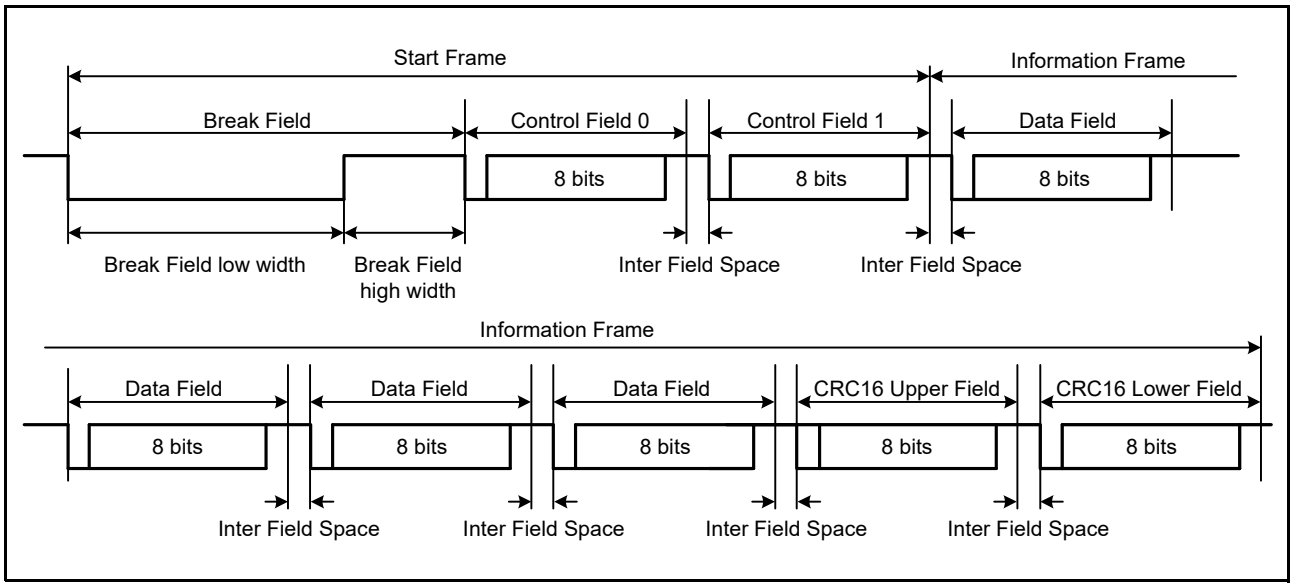


Figure 31.64 Protocol for Serial Transfer by the Extended Serial Mode Control Section

31.10.2 Transmitting a Start Frame

Figure 31.65 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 31.66 and Figure 31.67 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.

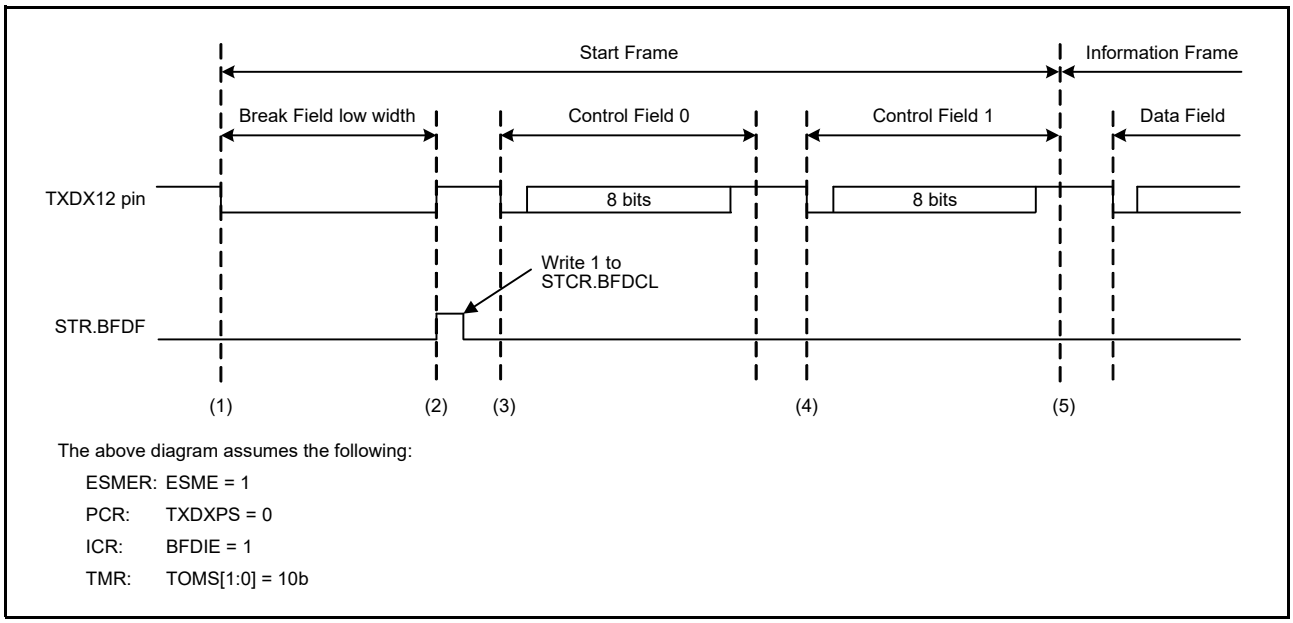


Figure 31.65 Example of Operations When Transmitting a Start Frame

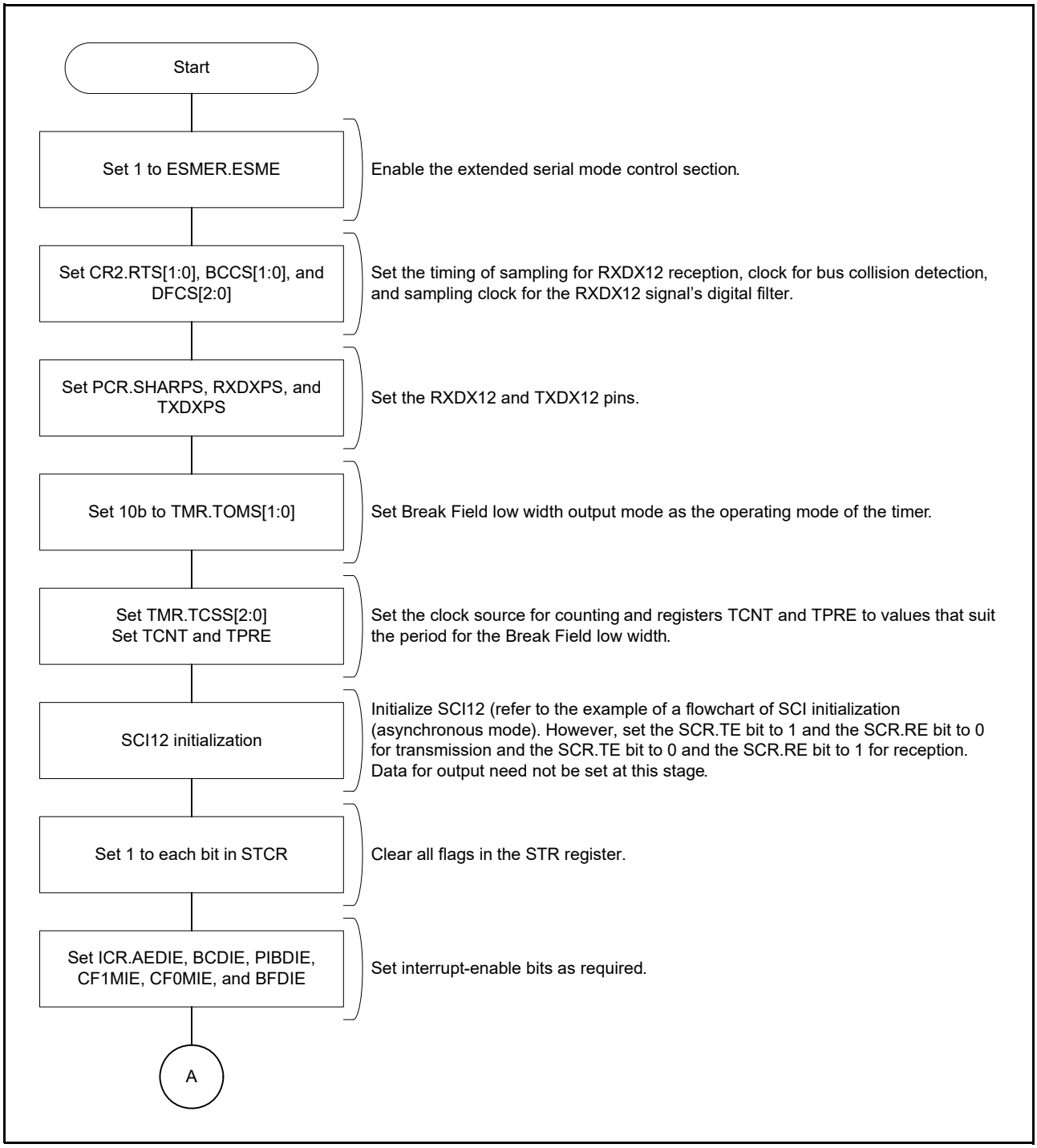


Figure 31.66 Example of Start Frame Transmission (1/2)

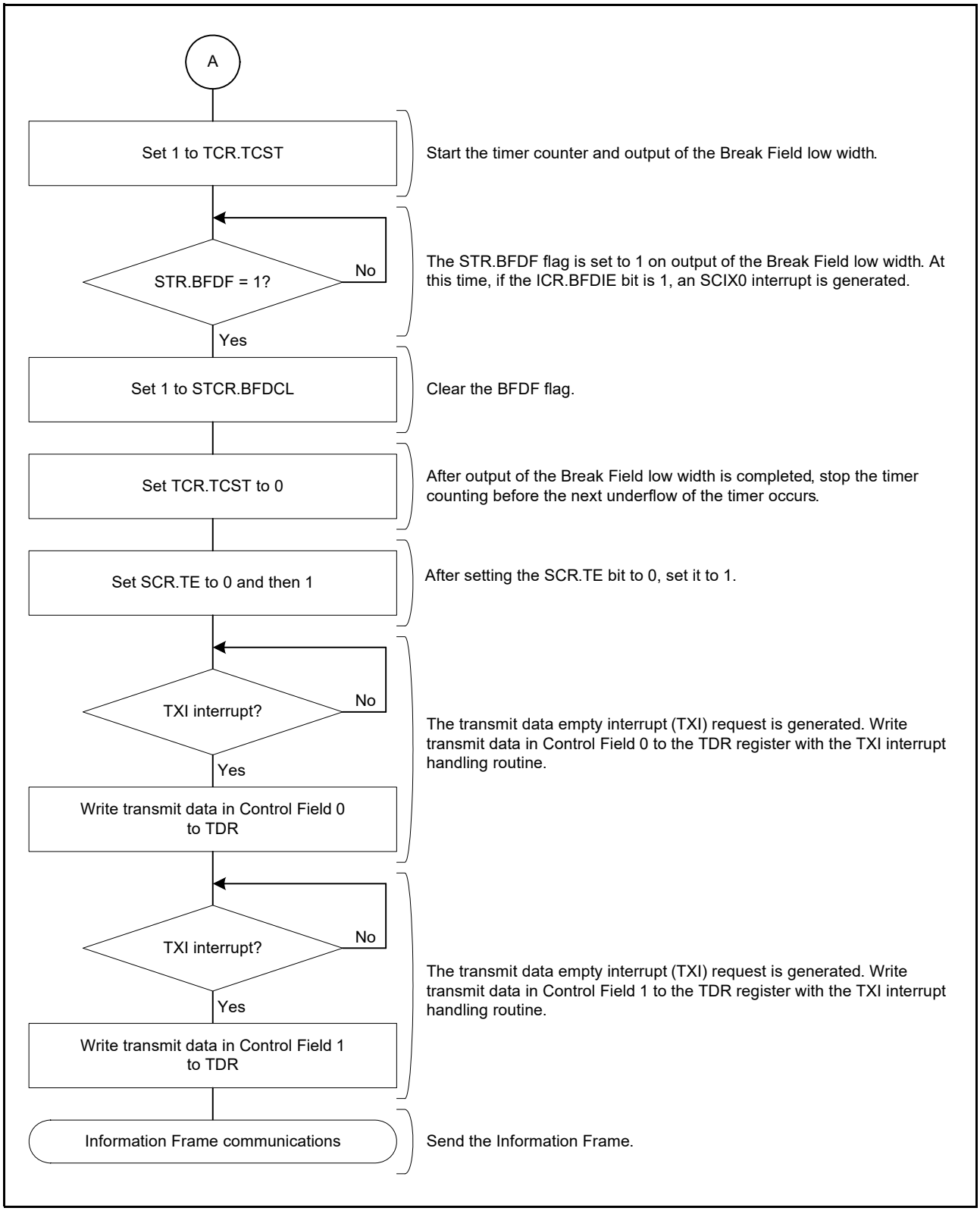


Figure 31.67 Example of Start Frame Transmission (2/2)

31.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 31.36.

Table 31.36 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 31.68 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 31.69 and Figure 31.70 are flowcharts for the reception of a Start Frame, and Figure 31.71 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.

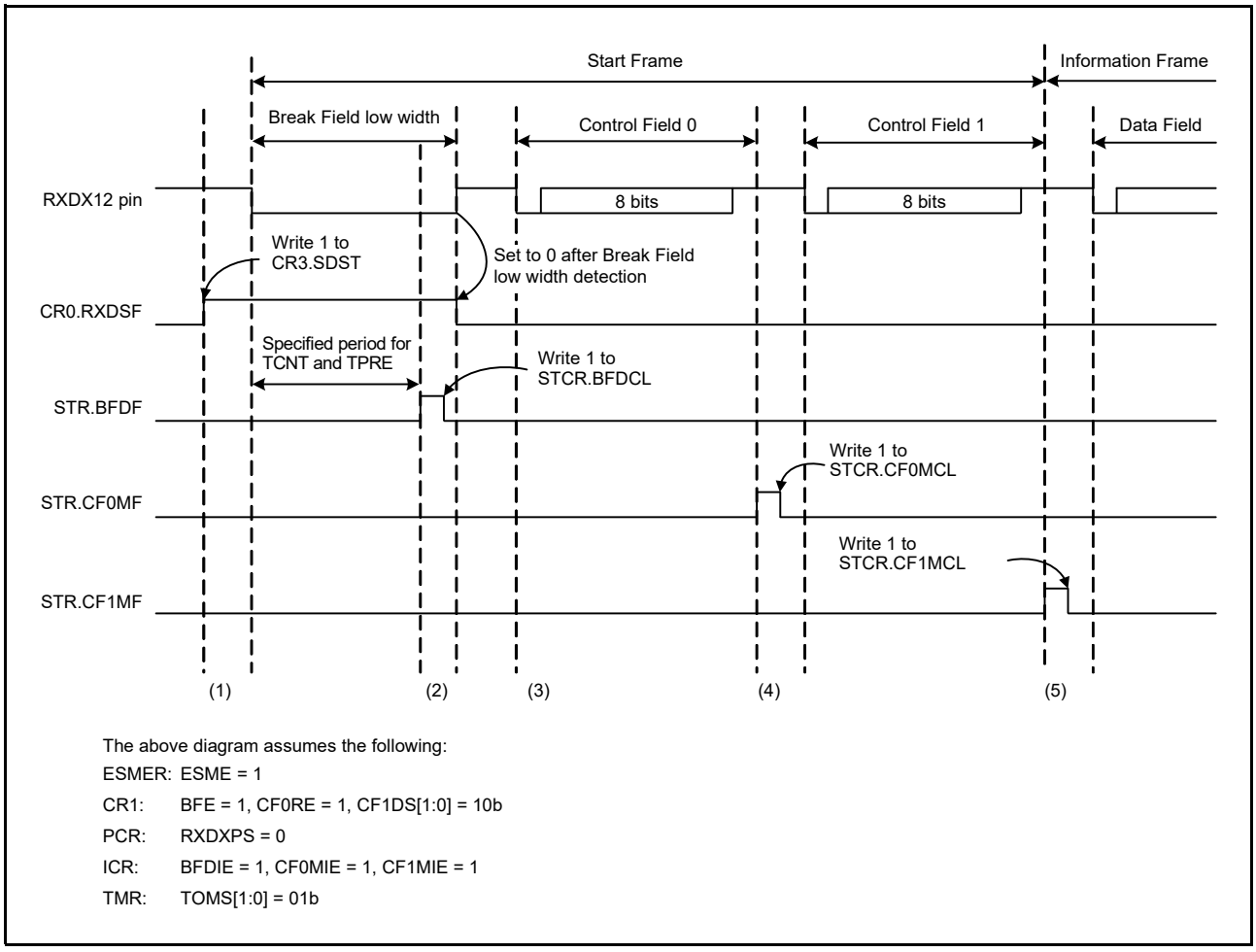


Figure 31.68 Example of Operations at the Time of Start Frame Reception

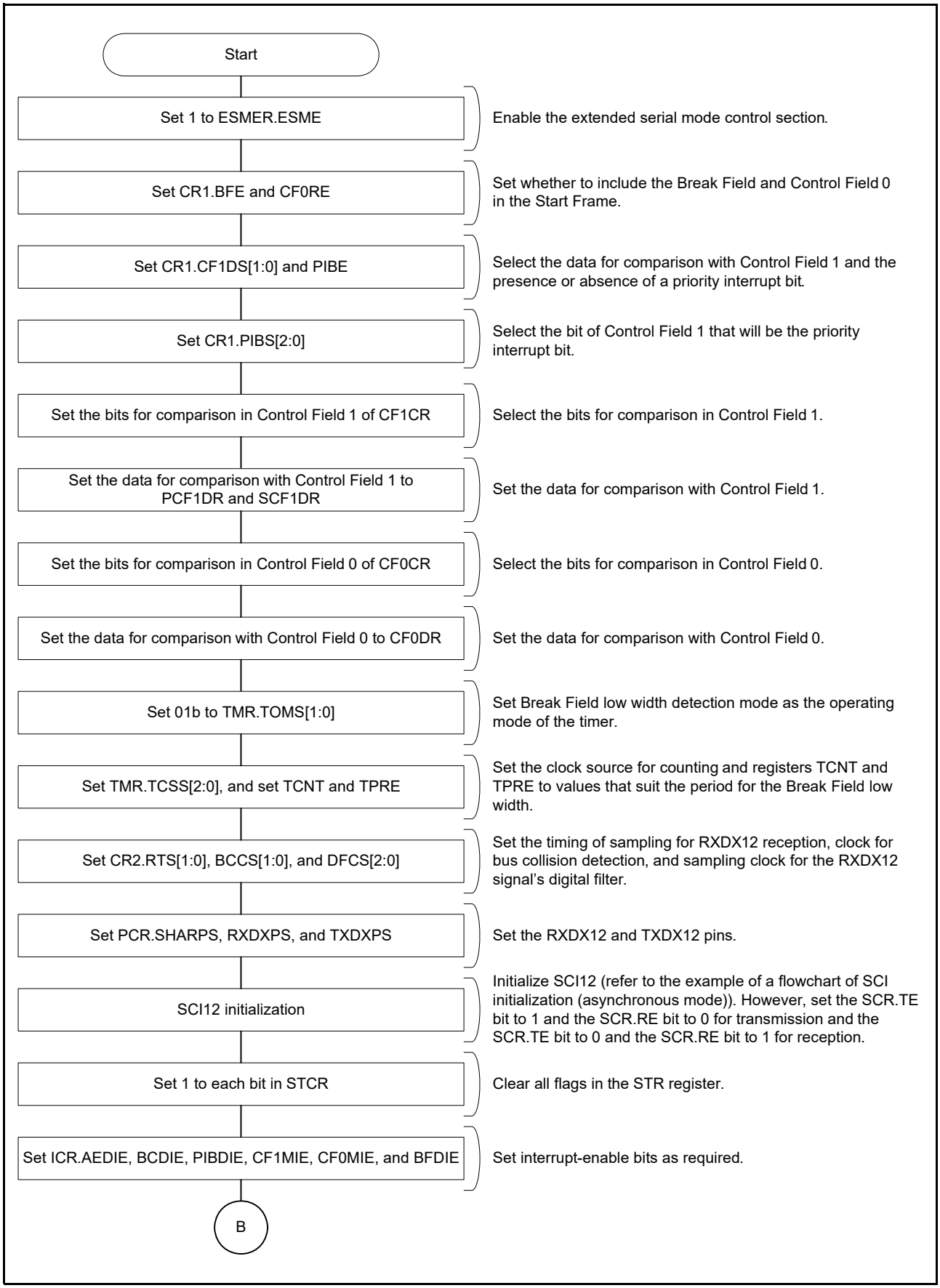


Figure 31.69 Sample Flowchart for Reception of a Start Frame (1)

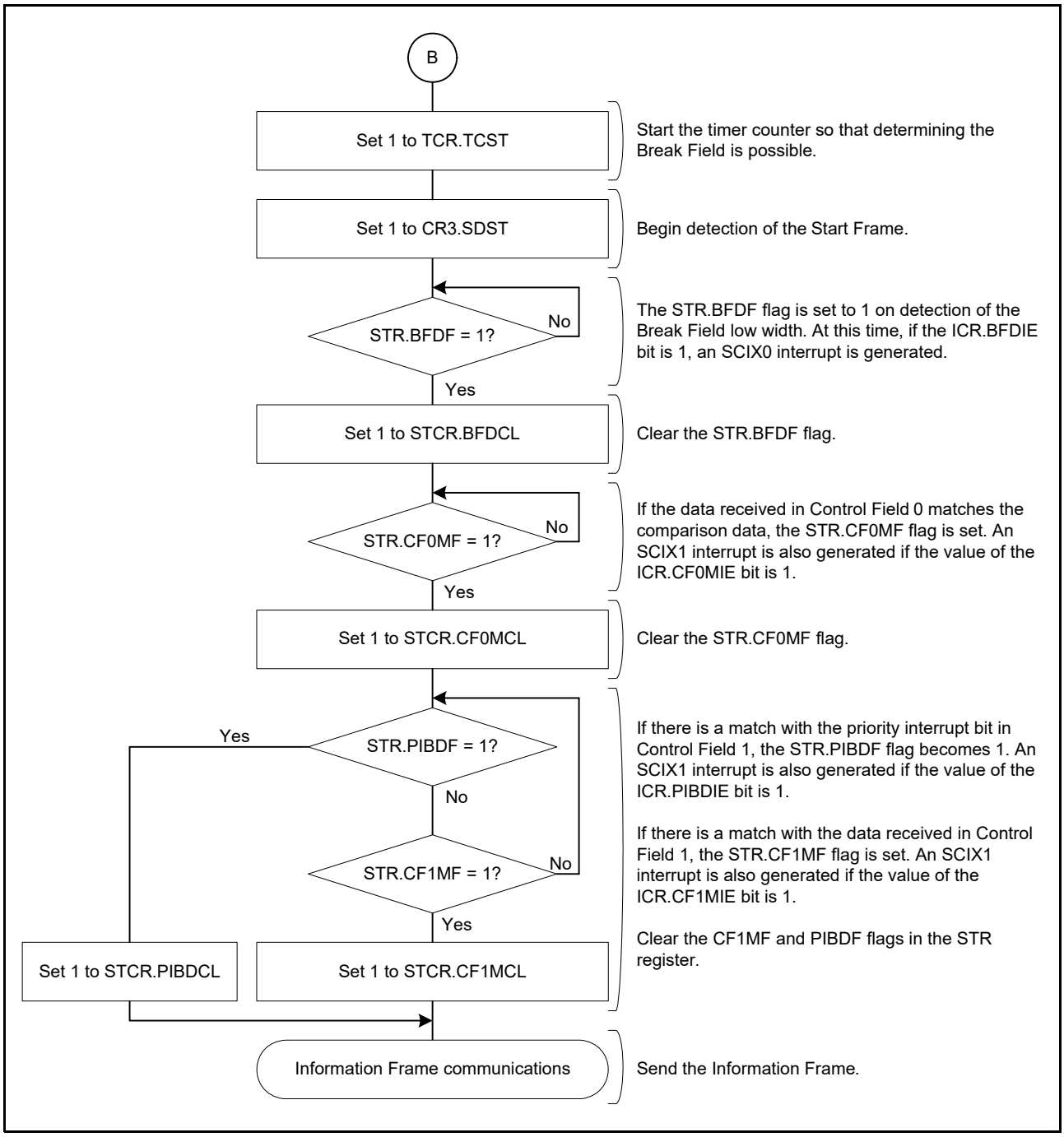


Figure 31.70 Sample Flowchart for Reception of a Start Frame (2)

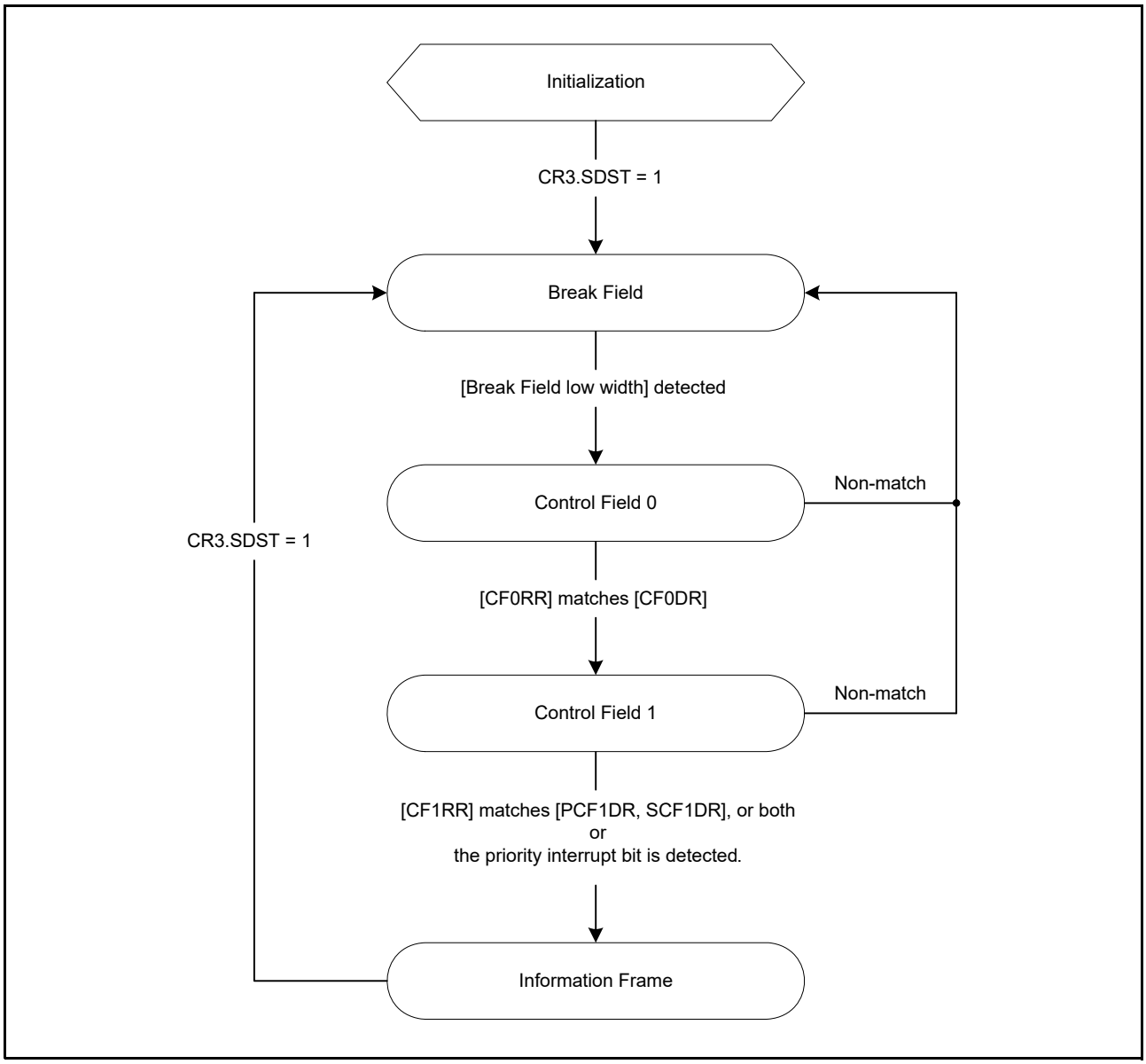


Figure 31.71 State Transitions When Receiving a Start Frame

31.10.3.1 Priority Interrupt Bit

Figure 31.72 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 31.68, 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.

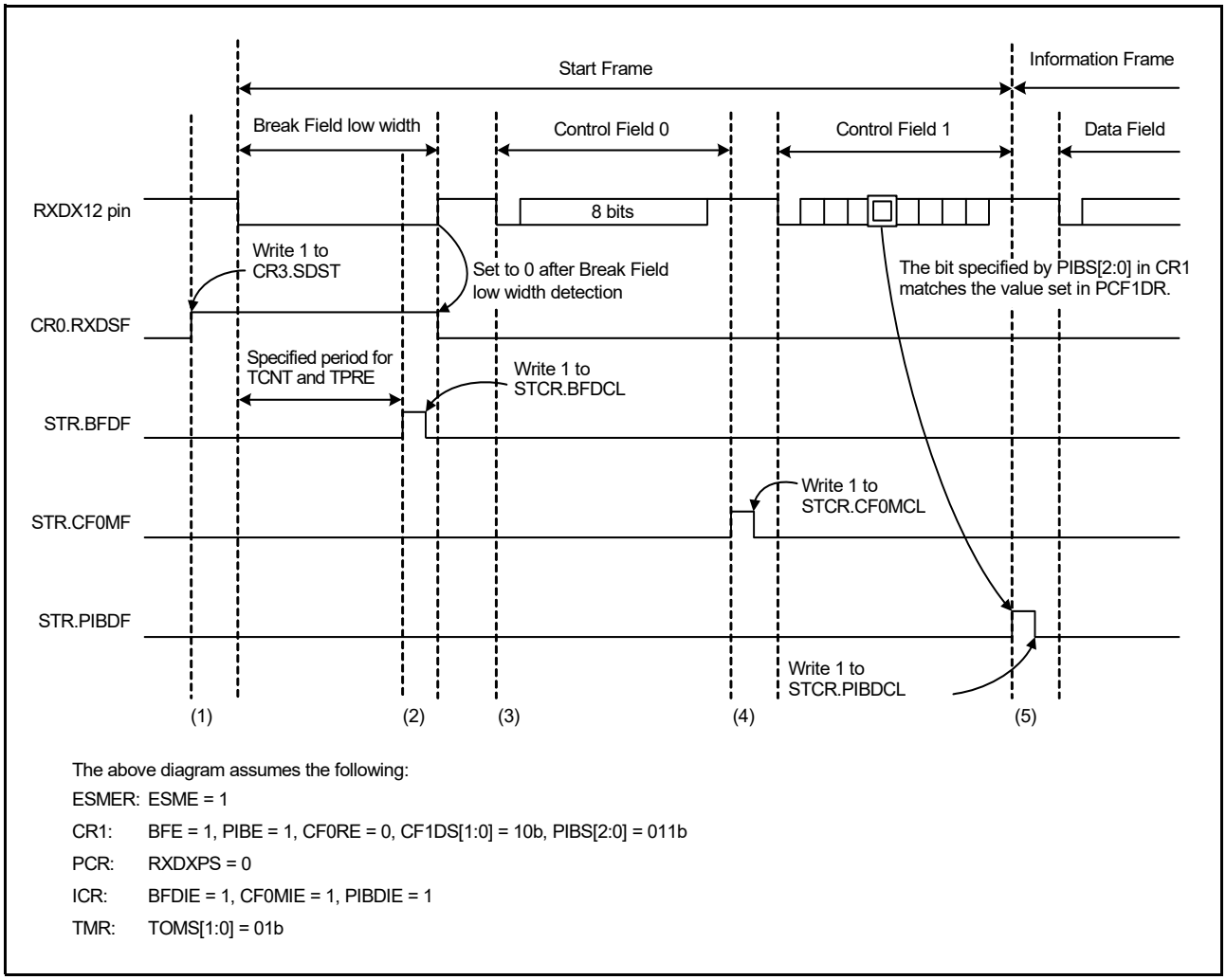


Figure 31.72 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

31.10.4 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 SCR.TE bit are set to 1.

Figure 31.73 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.

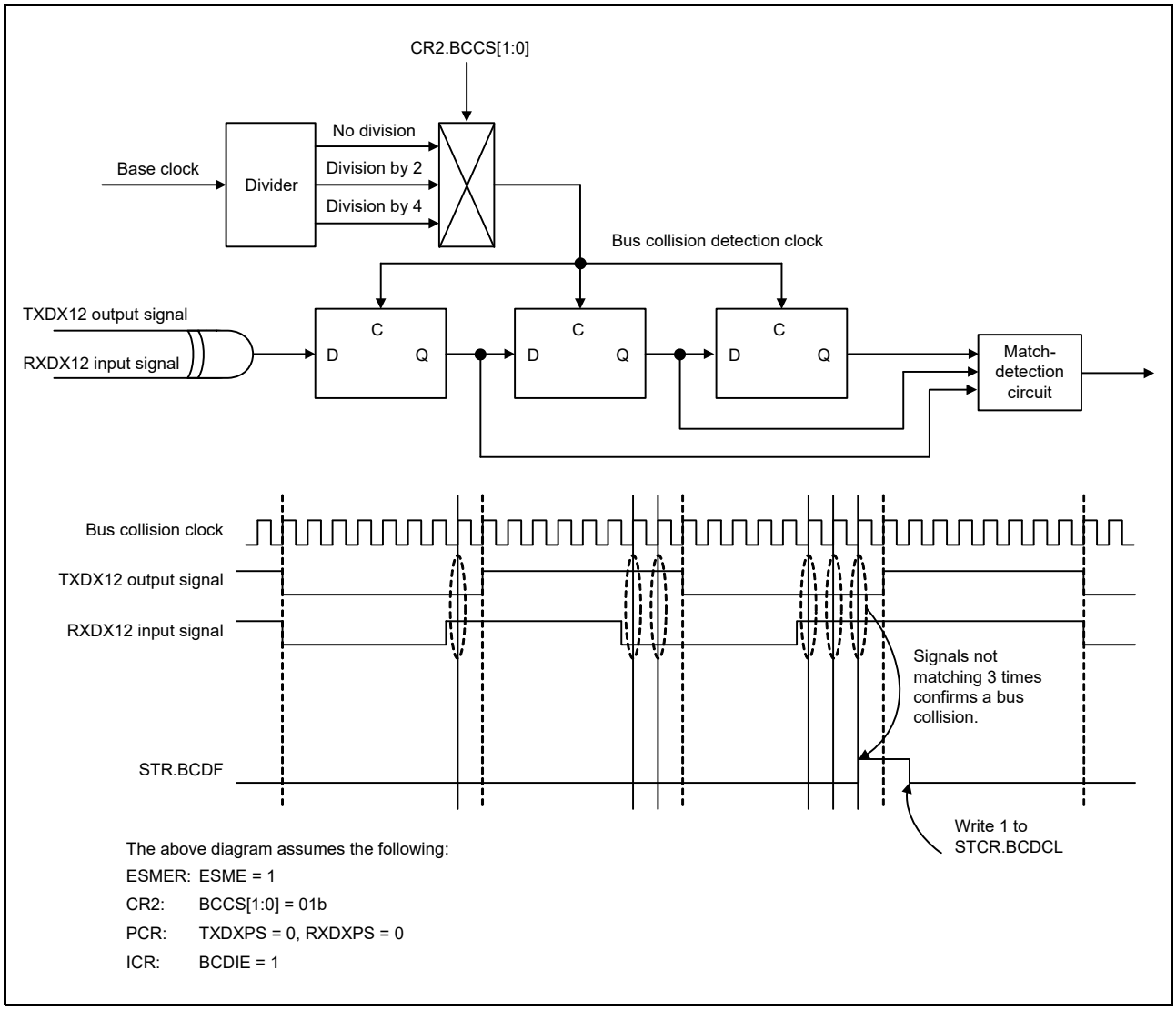


Figure 31.73 Example of Operations with Bus Collision Detection

31.10.5 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 31.74 shows an example of operations with the digital filter.

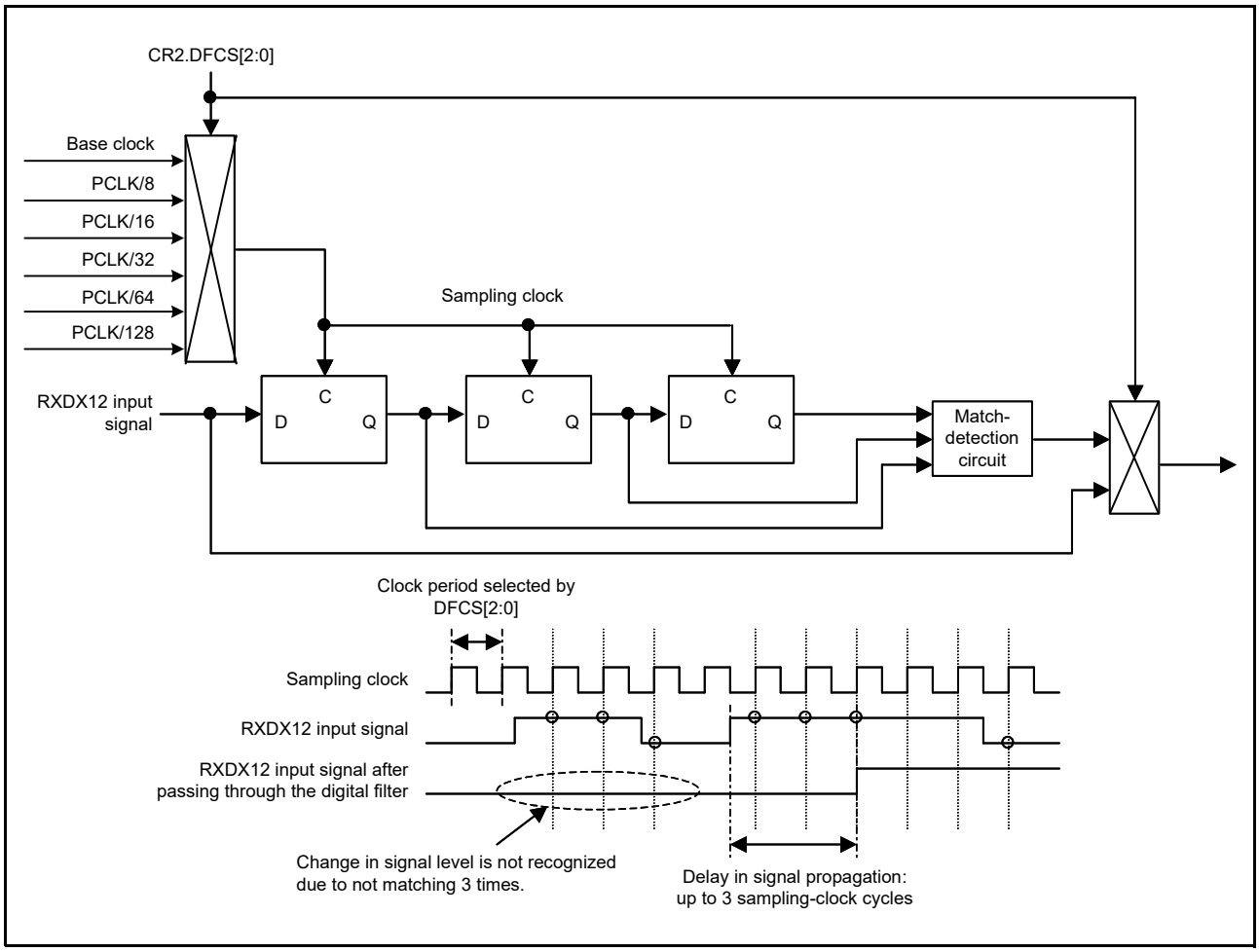
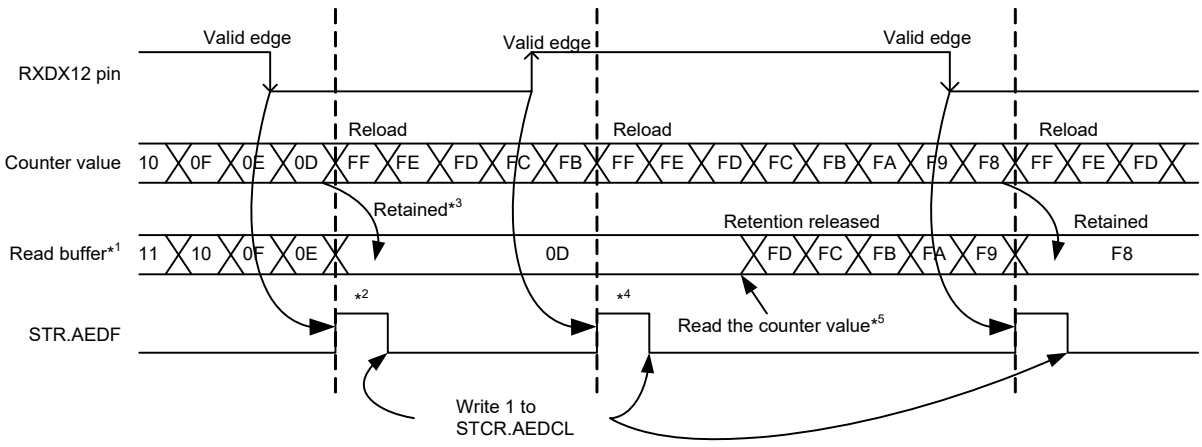


Figure 31.74 Example of Operations with the Digital Filter

31.10.6 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 31.75 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 31.75 Example of Operations for Bit Rate Measurement

31.10.7 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 31.76 shows timing for the sampling of data received through RXDX12.

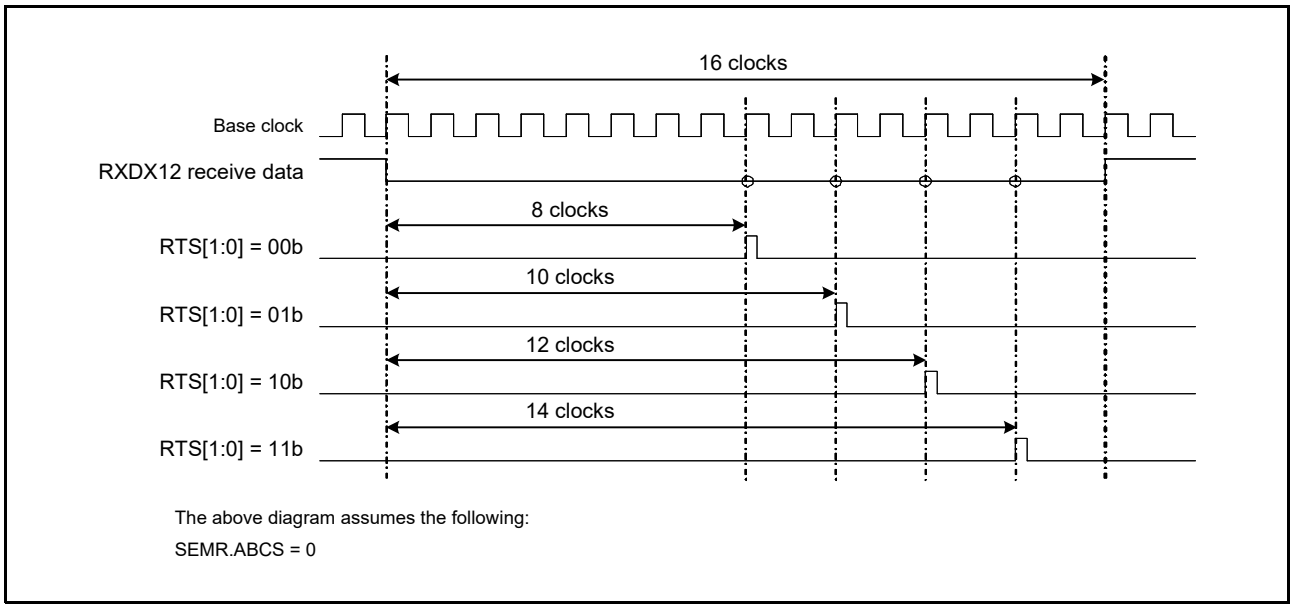


Figure 31.76 Timing for Sampling of Data Received through RXDX12

31.10.8 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 31.77 shows an example of operations in Break Field low width output mode.

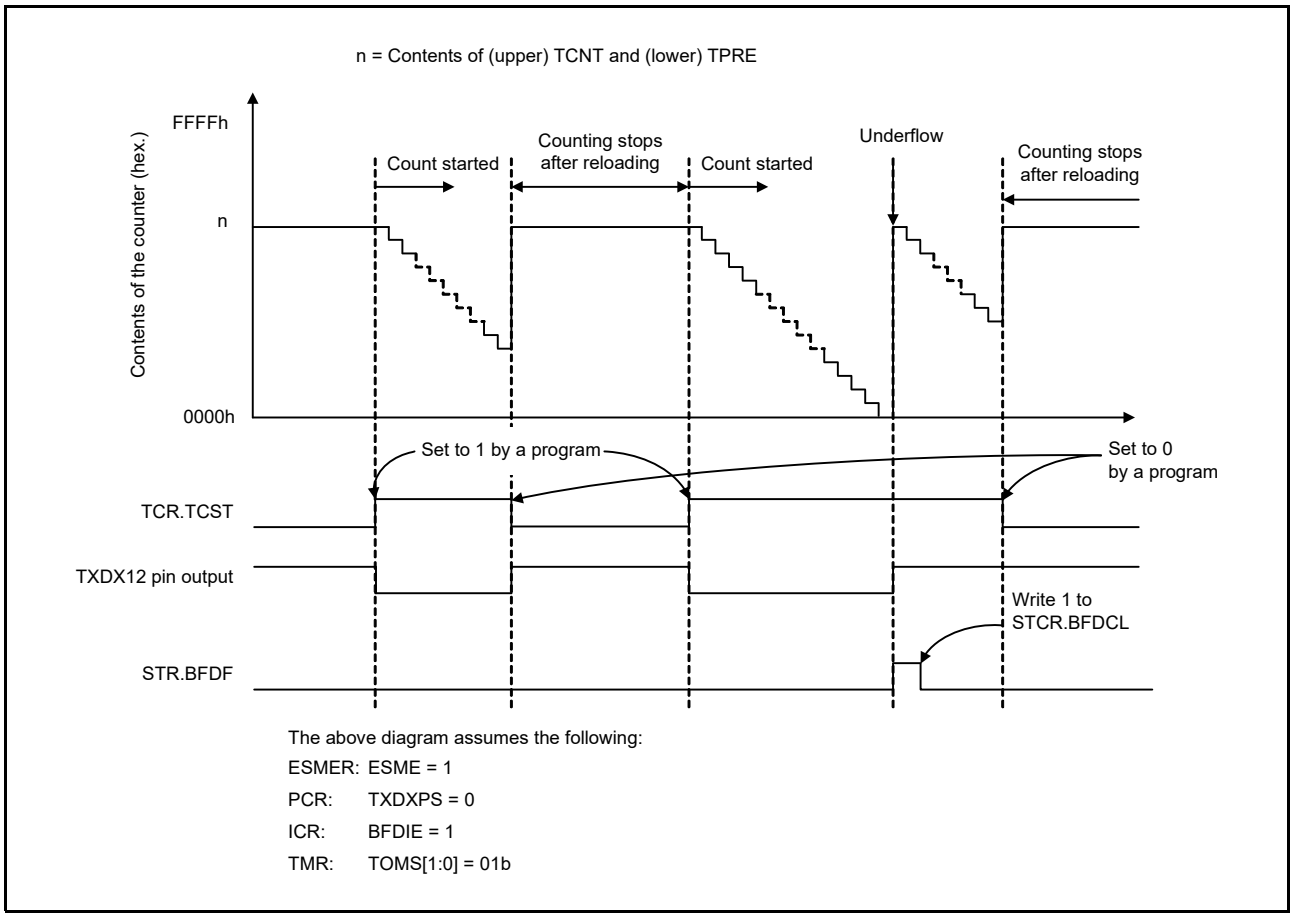


Figure 31.77 Example of Operations in Break Field Low Width Output Mode

(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 31.78 shows an example of operations in Break Field low width output mode.

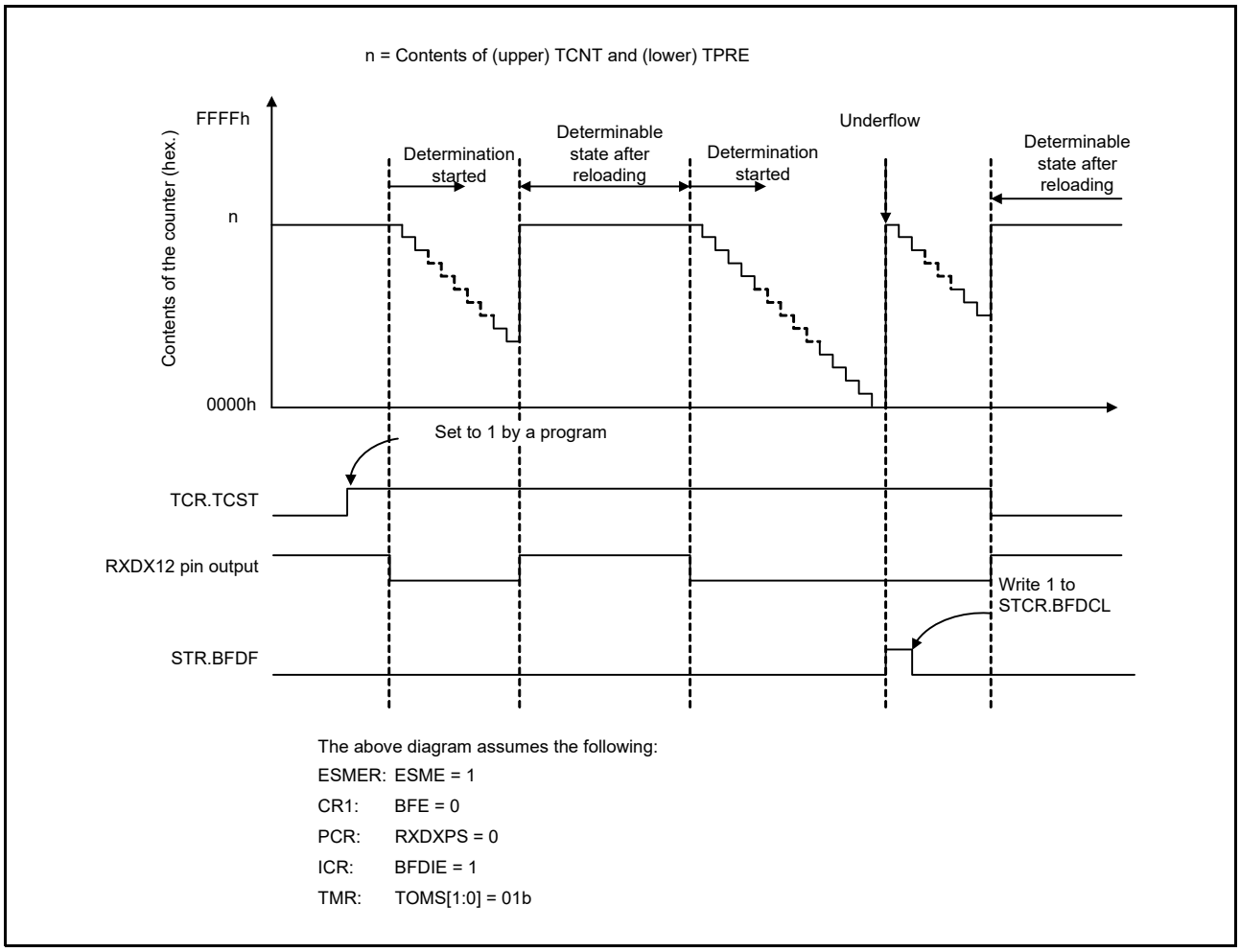


Figure 31.78 Example of Operations in Break Field Low Width Determination Mode

(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.

31.11 Noise Cancellation Function

Figure 31.79 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.

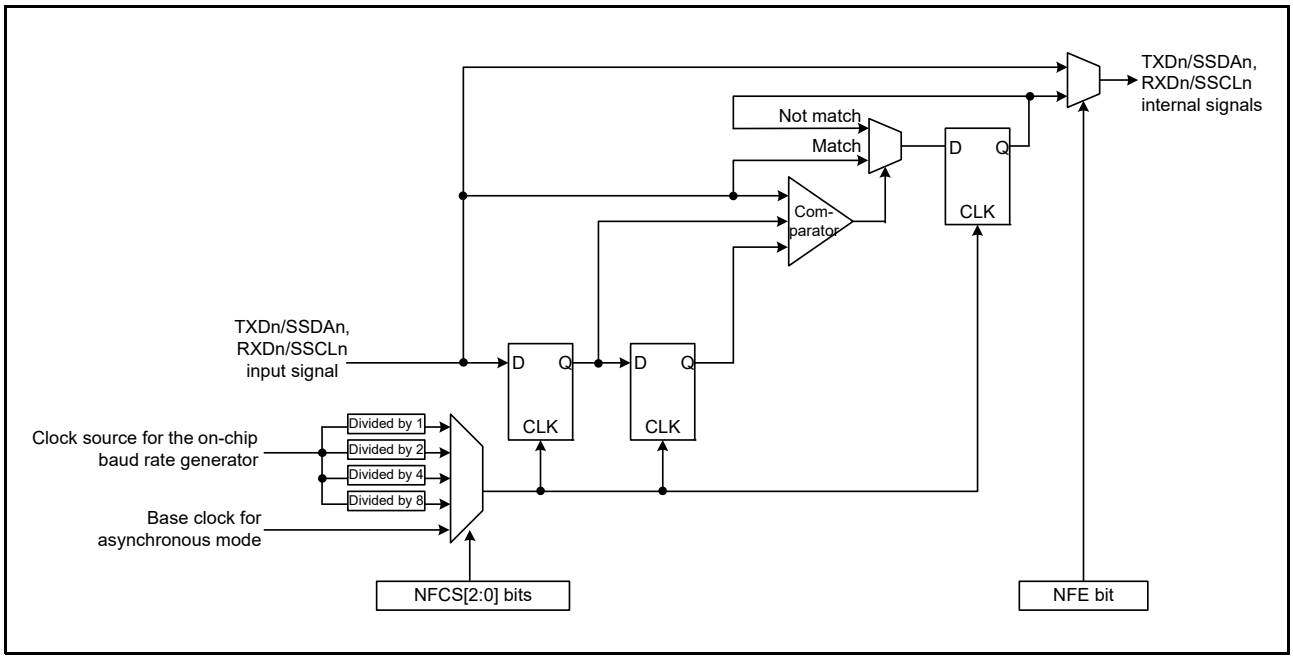


Figure 31.79 Block Diagram of Digital Noise Filter

31.12 Interrupt Sources

31.12.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR register) can also be used to discard an internally retained interrupt request.

31.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 31.37 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 31.37 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 SCI1, SCI5, and SCI6 only.

31.12.3 Interrupts in Smart Card Interface Mode

Table 31.38 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 31.38 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 (DMACA).

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.

31.12.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 31.39. 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 31.39 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).

31.12.5 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 31.40.

Table 31.40 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

31.13 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.

31.14 Usage Notes

31.14.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

31.14.2 Break Detection and Processing

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 SCI1, SCI5, and SCI6). 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.

31.14.3 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.

31.14.4 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. 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).

31.14.5 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.

31.14.6 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 31.80).

(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 31.80).

(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 31.80).

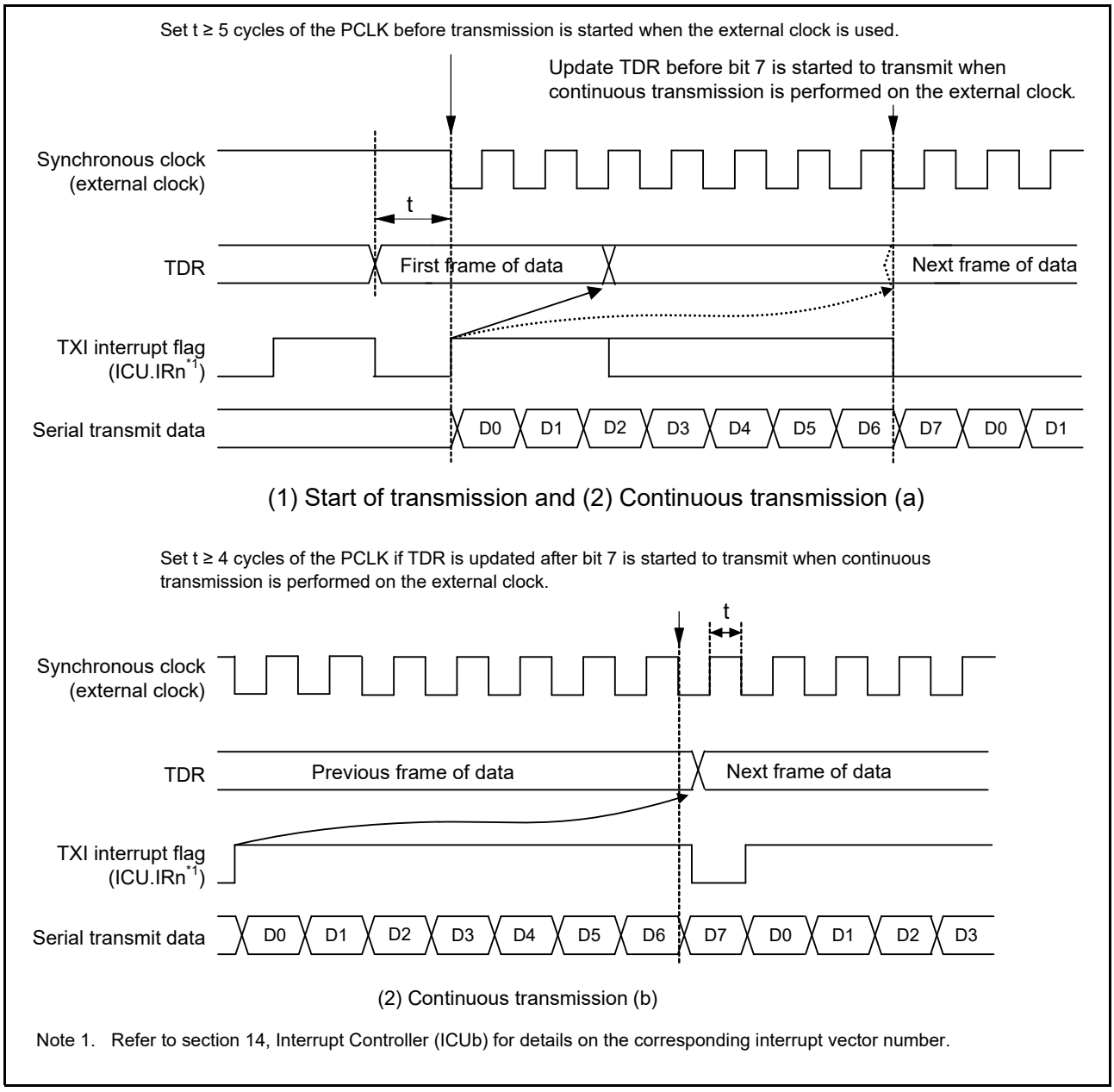


Figure 31.80 Restrictions on Use of External Clock in Clock Synchronous Transmission

31.14.7 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.

31.14.8 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 (ICUb).

- 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.

31.14.9 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 SCI1, SCI5, and SCI6). Setting the TE bit to 0 resets the TSR register and the SSR.TEND flag. Depending on the port settings or SPTR register setting (only for SCI1, SCI5, and SCI6), 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 31.81 shows a sample flowchart for transition to software standby mode during transmission. Figure 31.82 and Figure 31.83 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

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 31.84 shows a sample flowchart for transition to software standby mode during reception.

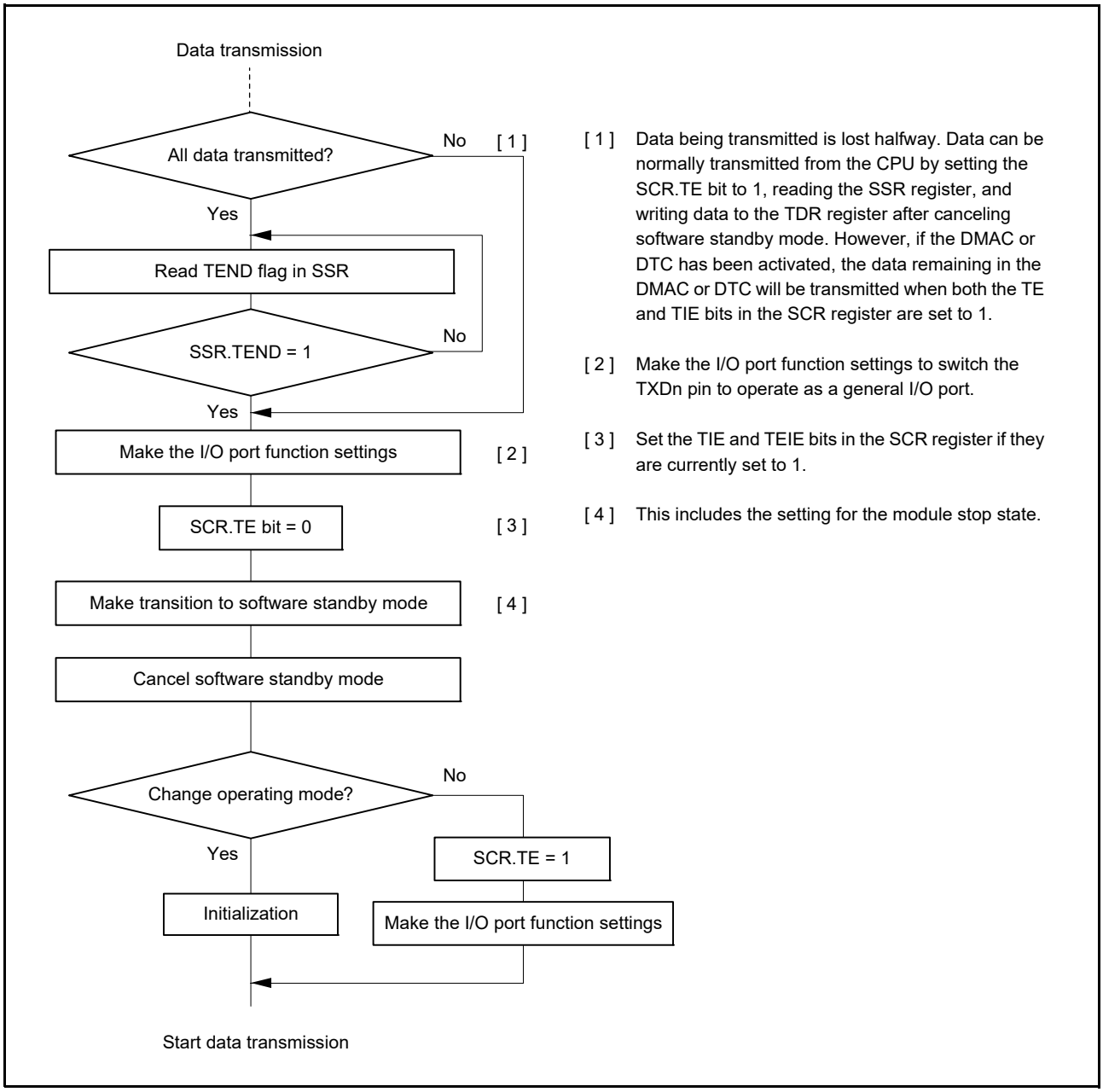


Figure 31.81 Example of Flowchart for Transition to Software Standby Mode during Transmission

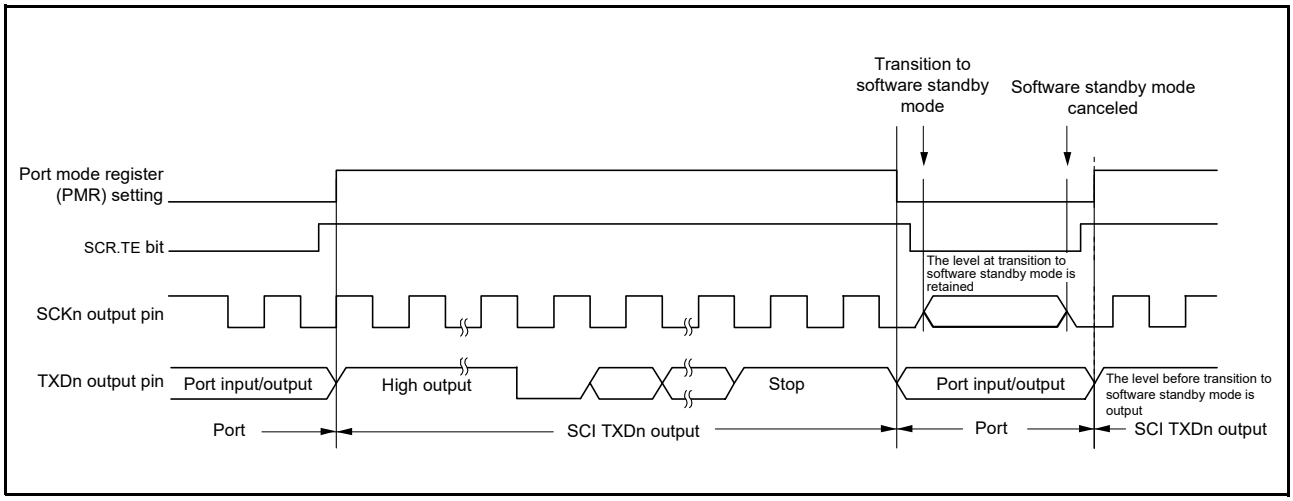


Figure 31.82 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

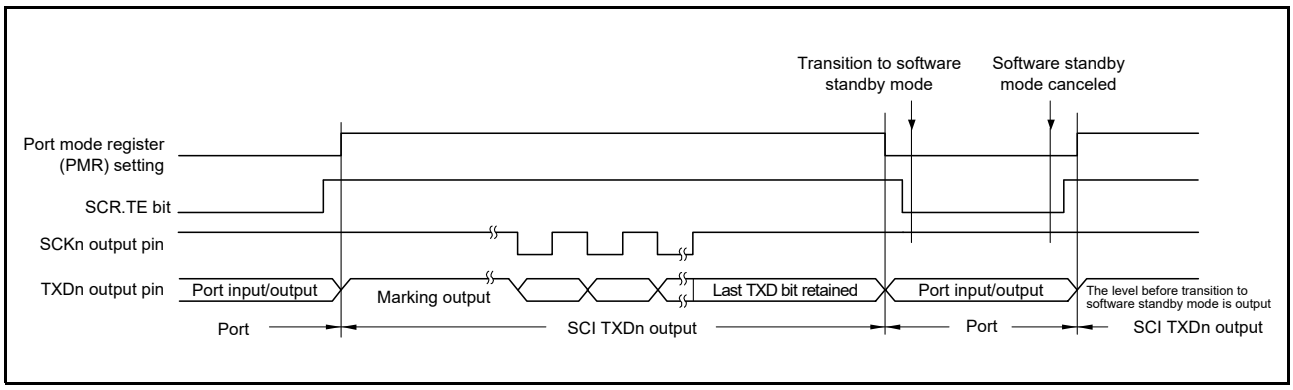


Figure 31.83 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

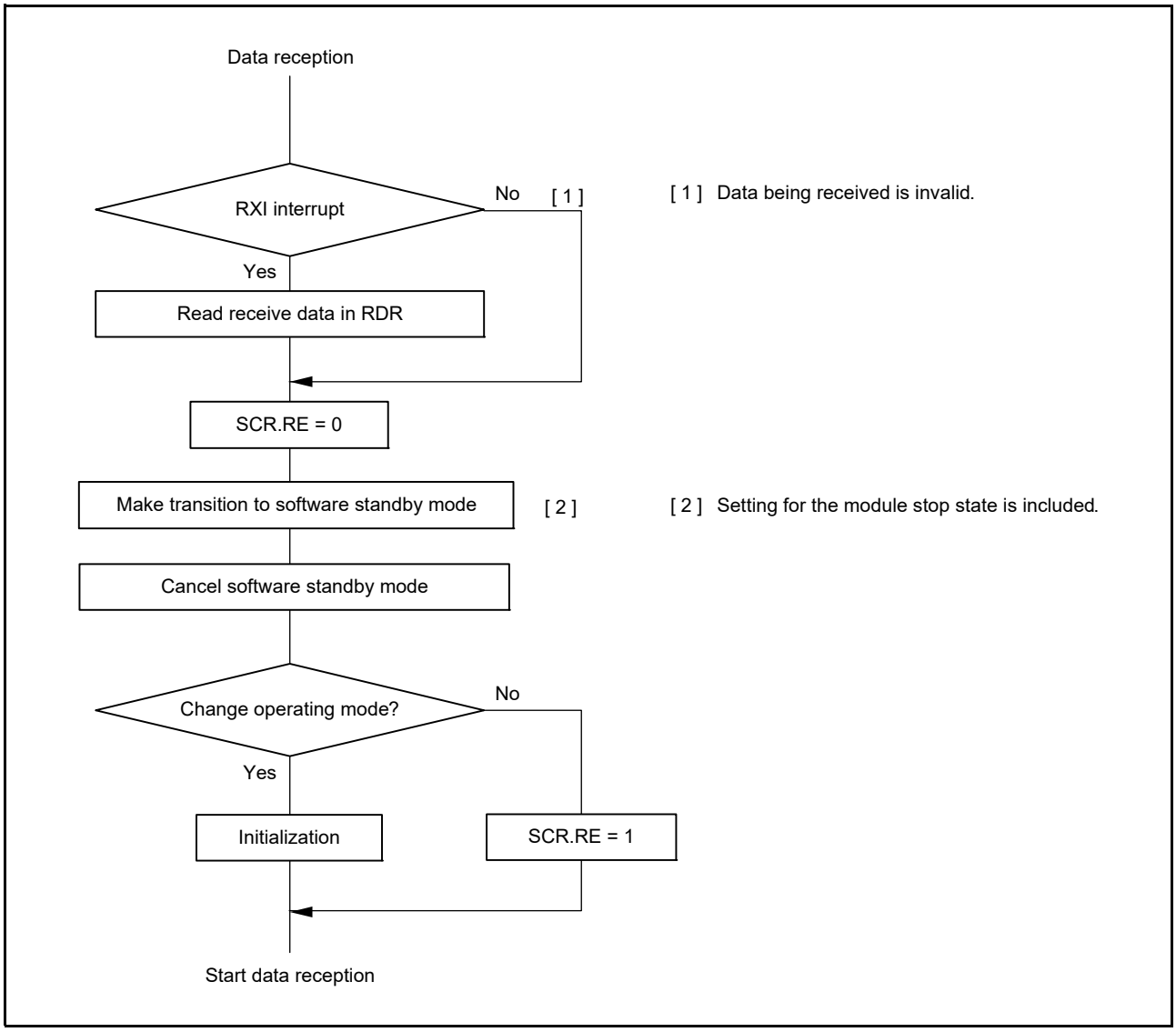


Figure 31.84 Example of Flowchart for Transition to Software Standby Mode during Reception

31.14.10 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

31.14.11 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 31.85. 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.

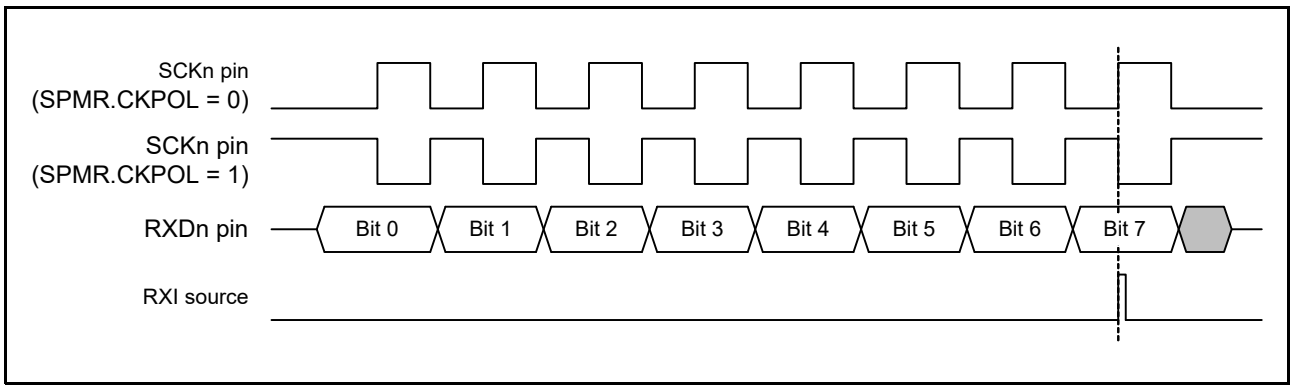


Figure 31.85 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(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.

31.14.12 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.

31.14.13 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.

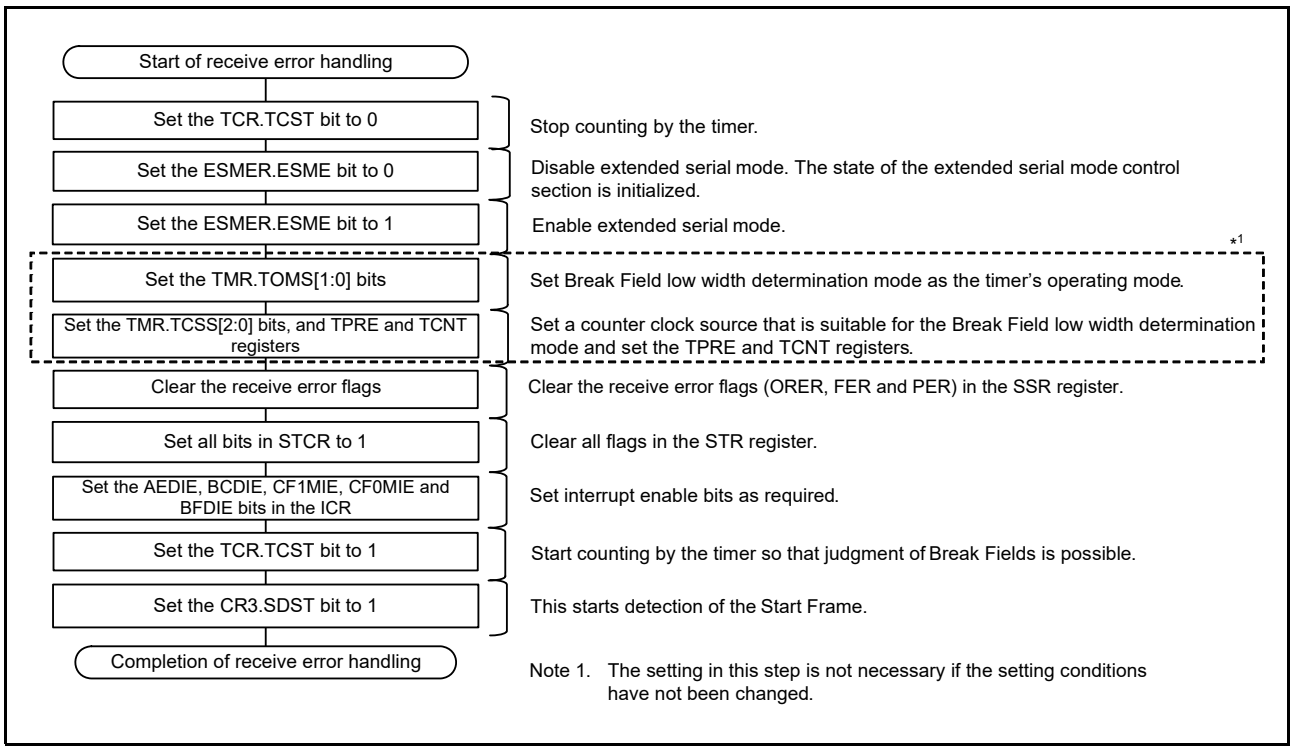


Figure 31.86 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

31.14.14 Note on Transmit Enable Bit (TE Bit)

When setting the pin function to “TXDn” while the SCR.TE bit is 0 (serial transmission is disabled) or setting the TE bit to 0 while the pin function is “TXDn”, output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(1) Connect a pull-up or pull-down resistor to the TXDn line.
(2) Set the TE bit to 1*1 before changing the pin function to “TXDn”. Before setting the TE bit to 0, change the pin function to “general-purpose I/O port” and drive the pin high or low.
(3) Set the SPTR.SPB2IO bit to 1 first, and change the pin function to “TXDn”. Leave the value of the SPB2IO bit as 1 after that (for SCI1, SCI5, and SCI6).

Note 1. An interrupt is generated when the TE bit is set to 1 while the TXI interrupt is enabled. If this creates a problem, change the pin function to “TXDn” first, and then set the corresponding ICU.IERm.IENj bit to 1.

31.14.15 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.

32. Serial Communications Interface (RSCI)

In this section, “PCLK” is used to refer to PCLKB.

32.1 Overview

RSCI 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 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 32.1 lists the RSCI specifications and Table 32.2 lists the functions of each channel.

Table 32.1 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 32.1 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	Double-buffer structure
	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 32.1 RSCI Specifications (3/3)

Item		Description
Extended serial mode	Start Frame Transmission	Break Field transmission possible, Break Field transmission complete interrupt output possible Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	Break Field detectable, Break Field detected interrupt output possible Control Field 0/1 data comparison function Control Field 1 can set two types of comparison data of primary and secondary Priority interrupt bit can be set in Control Field 1 Bit rate measurement function
Simple I²C mode	Transfer format	I²C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 32.2.7, Control Register 2 (SCR2) o 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.
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	Double-buffer structure
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	Double-buffer structure
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Table 32.2 Functions of Each Channel

Item	RSCI0	RSCI8	RSCI9
Asynchronous mode	Available	Available	Available
Manchester mode	Not available	Not available	Available
Smart card interface mode	Available	Available	Available
Extended serial mode	Not available	Not available	Available
Simple I²C mode	Available	Available	Available
Clock synchronous mode	Available	Available	Available
Simple SPI mode	Available	Available	Available
Peripheral module clock	PCLKB	PCLKB	PCLKB

Figure 32.1 and Figure 32.2 show the RSCI Block Diagrams.

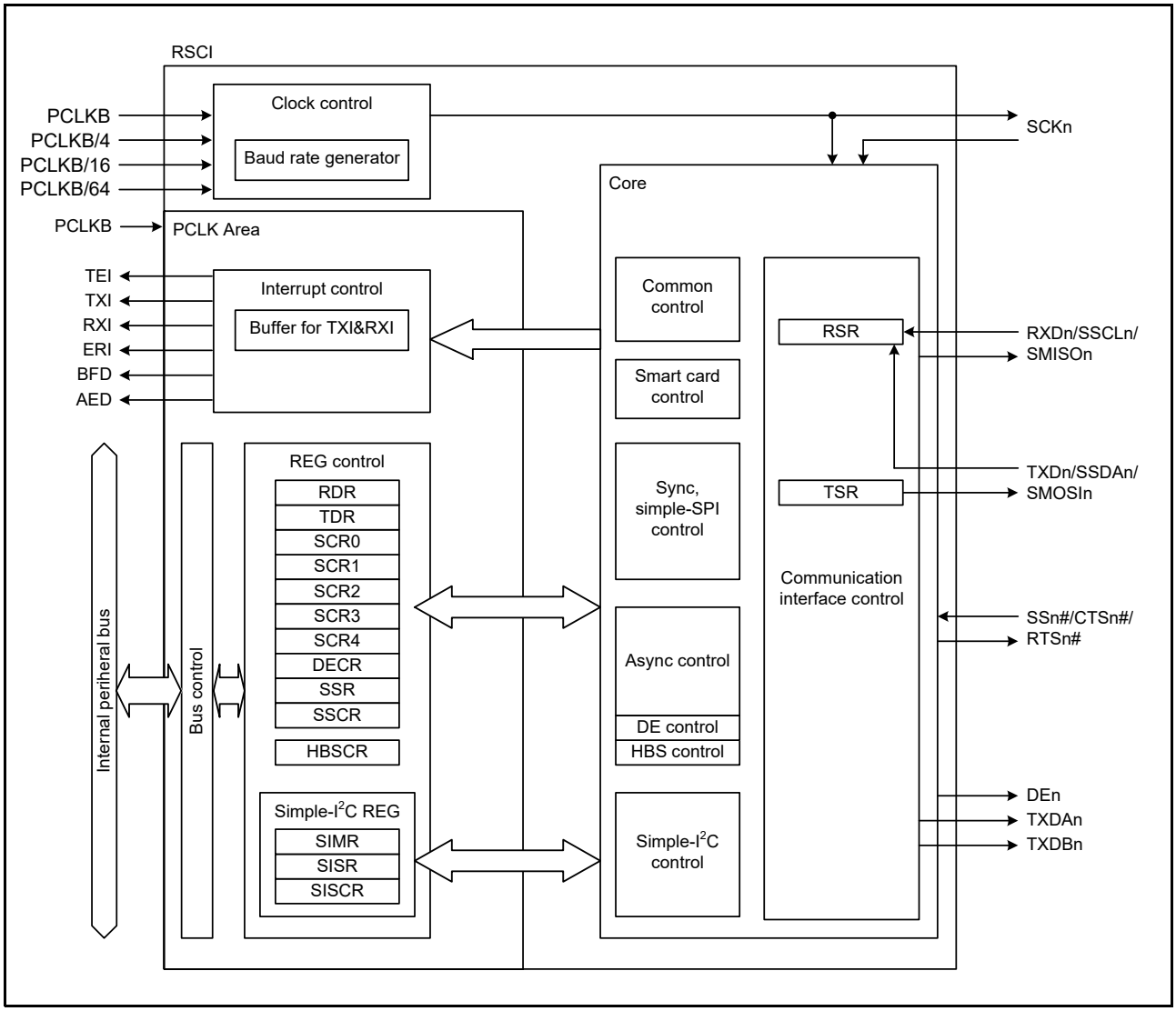


Figure 32.1 RSCI Block Diagram (n = 000, 008)

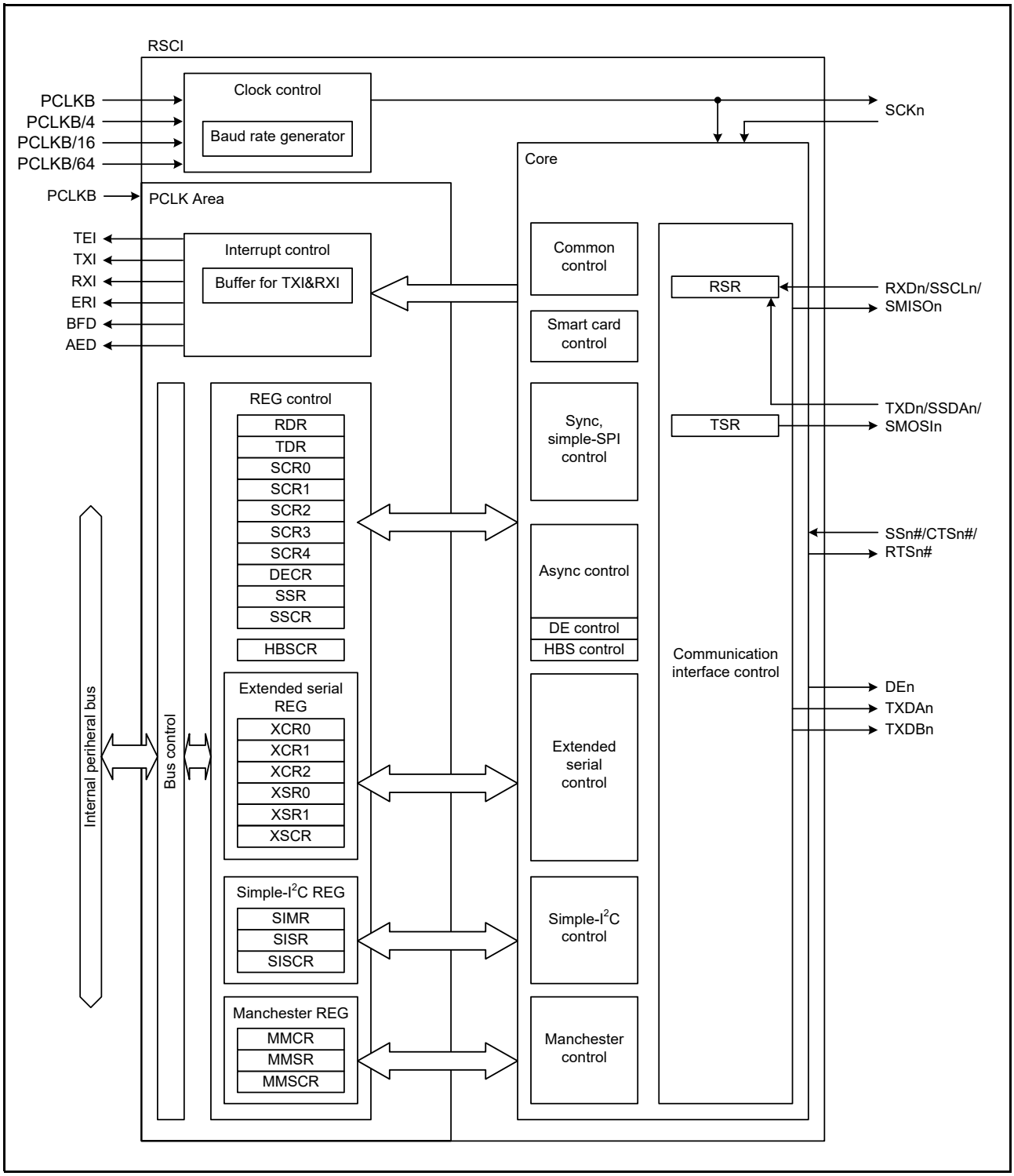


Figure 32.2 RSCI Block Diagram (n = 009)

Table 32.3 to Table 32.6 list RSCI's input/output pins.

Table 32.3 RSCI Input/Output Pin (Asynchronous Mode/Clock Synchronous Mode/Manchester Mode/ Extended Serial Mode)

Channel	Pin Name	I/O	Function
RSCI0	SCK000	I/O	RSCI0 clock input/output
	RXD000	Input	RSCI0 receive data input
	TXD000	Output	RSCI0 transmit data output
	RTS000#	Output	RSCI0 request-to-send signal output
	CTS000#	Input	RSCI0 transmission start control input
	DE000	Output	RSCI0 RS-485 driver control output
RSCI8	SCK008	I/O	RSCI8 clock input/output
	RXD008	Input	RSCI8 receive data input
	TXD008	Output	RSCI8 transmit data output
	RTS008#	Output	RSCI8 request-to-send signal output
	CTS008#	Input	RSCI8 transmission start control input
	DE008	Output	RSCI8 RS-485 driver control output
RSCI9	SCK009	I/O	RSCI9 clock input/output
	RXD009	Input	RSCI9 receive data input
	TXD009	Output	RSCI9 transmit data output
	RTS009#	Output	RSCI9 request-to-send signal output
	CTS009#	Input	RSCI9 transmission start control input
	DE009	Output	RSCI9 RS-485 driver control output

Table 32.4 RSCI Input/Output Pin (Simple I^2C Mode)

Channel	Pin Name	I/O	Function
RSCI0	SSCL000	I/O	RSCI0 I^2C clock input/output
	SSDA000	I/O	RSCI0 I^2C data input/output
RSCI8	SSCL008	I/O	RSCI8 I^2C clock input/output
	SSDA008	I/O	RSCI8 I^2C data input/output
RSCI9	SSCL009	I/O	RSCI9 I^2C clock input/output
	SSDA009	I/O	RSCI9 I^2C data input/output

Table 32.5 RSCI Input/Output Pin (Simple SPI Mode)

Channel	Pin Name	I/O	Function
RSCI0	SCK000	I/O	RSCI0 clock input/output
	SMISO000	I/O	RSCI0 slave transmit data input/output
	SMOSI000	I/O	RSCI0 master transmit data input/output
	SS000#	Input	RSCI0 slave select input
RSCI8	SCK008	I/O	RSCI8 clock input/output
	SMISO008	I/O	RSCI8 slave transmit data input/output
	SMOSI008	I/O	RSCI8 master transmit data input/output
	SS008#	Input	RSCI8 slave select input
RSCI9	SCK009	I/O	RSCI9 clock input/output
	SMISO009	I/O	RSCI9 slave transmit data input/output
	SMOSI009	I/O	RSCI9 master transmit data input/output
	SS009#	Input	RSCI9 slave select input

Table 32.6 RSCI Input/Output Pin (HBS Support Mode)

Channel	Pin Name	I/O	Function
RSCI0	RXD000	Input	RSCI0 receive data input
	TXD000	Output	RSCI0 transmit data output
	TXDA000/TXDB000	Output	RSCI0 transmit data output (in alternate output)
RSCI8	RXD008	Input	RSCI8 receive data input
	TXD008	Output	RSCI8 transmit data output
	TXDA008/TXDB008	Output	RSCI8 transmit data output (in alternate output)
RSCI9	RXD009	Input	RSCI9 receive data input
	TXD009	Output	RSCI9 transmit data output
	TXDA009/TXDB009	Output	RSCI9 transmit data output (in alternate output)

32.2 Register Descriptions

This chapter describes the RSCI registers, their functional specifications, and their operating specifications.

32.2.1 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.

32.2.2 Receive Data Register (RDR)

Address(es): RSCI0.RDR 000A 1000h, RSCI8.RDR 000A 1400h, RSCI9.RDR 000A 1480h

	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
	—	—	—	—	—	—	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
b23 to b10	—	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

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.

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.

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.

32.2.3 Transmit Data Register (TDR)

Address(es): RSCI0.TDR 000A 1004h, RSCI8.TDR 000A 1404h, RSCI9.TDR 000A 1484h

	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

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.

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.

32.2.4 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.

32.2.5 Control Register 0 (SCR0)

Address(es): RSCI0.SCR0 000A 1008h, RSCI8.SCR0 000A 1408h, RSCI9.SCR0 000A 1488h

	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 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, 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 32.4, 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 32.3.6, 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 32.2.11, Manchester Mode Control Register (MMCR) and section 32.2.18, 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.

32.2.6 Control Register 1 (SCR1)

Address(es): RSCI0.SCR1 000A 100Ch, RSCI8.SCR1 000A 140Ch, RSCI9.SCR1 000A 148Ch

	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 32.7 for the relationship between the CRSEP and CTSE bit settings and the pin functions.

Table 32.7 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 32.8.

Table 32.8 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 32.7.2, Data Format (Except in Block Transfer Mode).

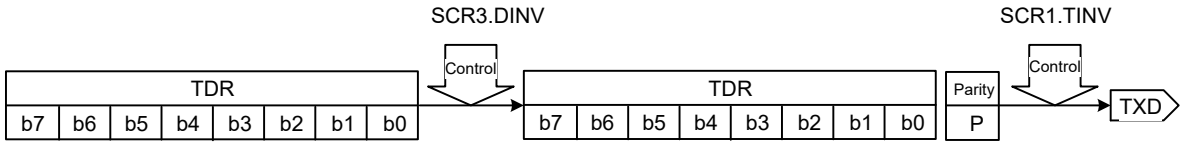
TINV Bit (Transmitter Output Invert), RINV Bit (Receiver Input Invert)

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 32.3 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.

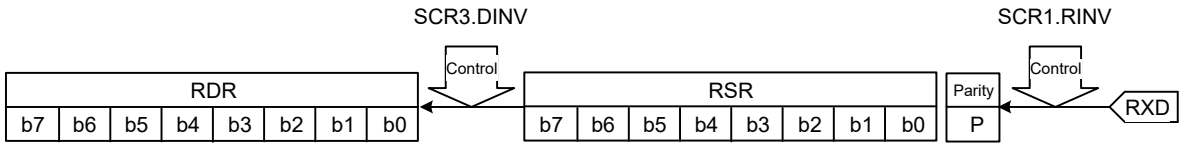
The receive/transmit data control (Data size = 8bits, Even parity, MSB first)

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 32.3 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 32.16, 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.

32.2.7 Control Register 2 (SCR2)

Address(es): RSCI0.SCR2 000A 1010h, RSCI8.SCR2 000A 1410h, RSCI9.SCR2 000A 1490h

	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: 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: PCLK (n = 0)*3 0 1: PCLK/4 (n = 1)*3 1 0: PCLK/16 (n = 2)*3 1 1: PCLK/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 32.7.4, 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 32.9 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 32.10 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 32.10 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	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	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*2	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI	0	0 (Initial value)	0 (Initial value)	$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
	1	0 (Initial value)	0 (Initial value)	$N = \frac{PCLK \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	
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[7:0] bits setting ($0 \le N \le 255$)
PCLK: 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 32.11 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 32.12 Clock Source Settings

SCR2 Setting		
CKS[1:0] Bits	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 32.13 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 32.14 and Table 32.15 list examples of N settings in BRR[7:0] in asynchronous mode and Manchester mode. Table 32.16 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 32.18. Examples of BRR[7:0] bits (N) settings in smart card interface mode are listed in Table 32.20. Examples of BRR[7:0] bits (N) settings in simple I^2C mode are listed in Table 32.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 32.7.4, Receive Data Sampling Timing and Reception Margin. Table 32.17 and Table 32.19 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode and Manchester mode, the bit rate becomes twice that listed in Table 32.14 and Table 32.15. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 32.14 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (1)

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

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 32.15 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2)

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	20			25			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	–0.35
300	2	129	0.16	2	162	–0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35
1200	1	129	0.16	1	162	–0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35
4800	0	129	0.16	0	162	–0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35
31250	0	19	0.00	0	24	0.00	0	29	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73

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 32.16 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester Mode) (1)

PCLK (MHz)	SCR2 Settings					Maximum Bit Rate (bps)	PCLK (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	9.8304	0	0	0	0	0	307200
		1	0	0	0	500000			1	0	0	0	614400
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	1228800
	0 or 1	0 or 1	1	0	0	1333333		0 or 1	0 or 1	1	0	0	1638400
10	0	0	0	0	0	312500	12	0	0	0	0	0	375000
		1	0	0	0	625000			1	0	0	0	750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	1500000
	0 or 1	0 or 1	1	0	0	1666667		0 or 1	0 or 1	1	0	0	2000000
12.288	0	0	0	0	0	384000	14	0	0	0	0	0	437500
		1	0	0	0	768000			1	0	0	0	875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	1750000
	0 or 1	0 or 1	1	0	0	2048000		0 or 1	0 or 1	1	0	0	2333333
16	0	0	0	0	0	500000	17.2032	0	0	0	0	0	537600
		1	0	0	0	1000000			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	2150400
	0 or 1	0 or 1	1	0	0	2666667		0 or 1	0 or 1	1	0	0	2867200
18	0	0	0	0	0	562500	19.6608	0	0	0	0	0	614400
		1	0	0	0	1125000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2250000			1	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	3000000		0 or 1	0 or 1	1	0	0	3276800
20	0	0	0	0	0	625000	25	0	0	0	0	0	781250
		1	0	0	0	1250000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2500000			1	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	3333333		0 or 1	0 or 1	1	0	0	4166667
30	0	0	0	0	0	937500							
		1	0	0	0	1875000							
	1	0	0	0	0								
		1	0	0	0	3750000							
	0 or 1	0 or 1	1	0	0	5000000							

Table 32.17 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (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

Table 32.18 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	8			10			30		
	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
2.5 k	0	2	49	0	1	249	0	3	46
5 k	0	2	24	0	1	124	0	2	93
10 k	0	1	49	0	0	249	0	2	46
25 k	0	2	4	0	1	24	0	1	74
50 k	0	1	9	0	0	49	0	0	149
100 k	0	1	4	0	0	24	0	0	74
250 k	0	1	1	0	0	9	0	0	29
500 k	0	1	0	0	0	4	0	0	14
1 M	0	0	1	1	0	4	1	0	14
2.5 M	—	—	—	0	0	0	0	0	2
5 M	—	—	—	1	0	0	1	0	2
7.5 M	—	—	—	—	—	—	0	0	0

—: Can be set, but an error over 10% will occur.

Table 32.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	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

Table 32.20 BRR[7:0] Bits 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

Table 32.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0

Table 32.22 BRR[7:0] Bits 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	30	0.8	0	49	0.0	0	62	–0.8	0	77	0.2
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	0	24	0.0	0	30	0.8
50 k	0	4	0.0	0	5	4.2	0	9	0.0	0	12	–3.8	2	0	–2.3
100 k	0	2	–16.7	1	0	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	0	25.0	0	1	0.0	0	2	–16.7	0	2	4.2
350 k										0	1	–10.7	0	2	–25.6
384 k										0	1	–18.6	0	2	–32.2

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	30		
	n	N	Error (%)
10 k	0	93	–0.3
25 k	0	37	–1.3
50 k	0	18	–1.3
100 k	0	9	–6.3
250 k	1	0	–6.3
350 k	0	2	–10.7
384 k	0	2	–18.6

Table 32.23 Minimum High and Low Periods of the SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)
10 k	0	24	43.75/50.00	0	30	43.40/49.60	0	49	43.75/50.00	0	62	44.10/50.40
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	0	24	17.50/20.00
50 k	0	4	8.75/10.00	0	5	8.40/9.60	0	9	8.75/10.00	0	12	9.10/10.40
100 k	0	2	5.25/6.00	1	0	5.60/6.40	0	4	4.38/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	0	1.40/1.60	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60
384 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	25			30		
	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)
10 k	0	77	43.68/49.92	0	93	43.87/50.13
25 k	0	30	17.36/19.84	0	37	17.73/20.27
50 k	2	0	8.96/10.24	0	18	8.87/10.13
100 k	1	1	4.48/5.12	0	9	4.67/5.33
250 k	0	2	1.68/1.92	1	0	1.87/2.13
350 k	0	2	1.68/1.92	0	2	1.40/1.60
384 k	0	2	1.68/1.92	0	2	1.40/1.60

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 32.24.

The initial value of MDDR[7:0] bits is FFh. Bit 7 in this register is fixed to 1.

Table 32.24 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	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	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	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
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[7:0] bits setting (128 ≤ M ≤ 255)
N: BRR[7:0] bits setting (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n: Determined by the settings of the SCR2.CKS[1:0] bits as listed in Table 32.12, 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 32.25 and Table 32.26 list examples of N settings in BRR[7:0] bits and M settings in MDDR[7:0] bits in asynchronous mode.

Table 32.25 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency PCLK (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 PCLK (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 PCLK (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 32.26 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (2)

Bit Rate (bps)	Operating Frequency PCLK (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 PCLK (MHz)				
	30				
	n	N	M	BGDM Bit	Error (%)
38400	0	36	194	1	0.01
57600	0	10	173	0	–0.01
115200	0	10	173	1	–0.01
230400	0	6	220	1	–0.09
460800	0	3	252	1	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)

32.2.8 Control Register 3 (SCR3)

Address(es): RSCI0.SCR3 000A 1014h, RSCI8.SCR3 000A 1414h, RSCI9.SCR3 000A 1494h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	BLK	GM	—	—	CKE[1:0]		—	—	DEEN	—	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]		SYNDIS	—	—	—	—	—	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
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	SYNDIS	Synchronizer Disable	Set this bit to 1.	R/W
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 32.3 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	—	Reserved	This bit is read as 0. The write value should be 0.	R
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
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 32.99 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 32.99 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 32.27 Relationship between Communication Mode Selection Bits (MOD[2:0]), Other Operation Mode Setting Bits

Communication mode	Asynchronous				Smart Card I/F	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.DEEN	0	1	0	1	—	—	—		—	—		—
SCR3.SSE	—				—	—	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.

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 32.7.8, 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 32.7.6, Serial Data Transmission (Except in Block Transfer Mode) and section 32.7.8, Clock Output Control.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 32.7.3, Block Transfer Mode.

32.2.9 Control Register 4 (SCR4)

Address(es): RSCI0.SCR4 000A 1018h, RSCI8.SCR4 000A 1418h, RSCI9.SCR4 000A 1498h

	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 PCLK delay 0 0 0 1: 2 PCLK delay 0 0 1 0: 3 PCLK delay 0 0 1 1: 4 PCLK 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 32.3.10, 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 32.10.7, 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 32.3.11, 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.

32.2.10 I²C Mode Register (SIMR)

Address(es): RSCI0.SIMR 000A 1020h, RSCI8.SIMR 000A 1420h, RSCI9.SIMR 000A 14A0h

	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 32.72 and Figure 32.73 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 PCLK 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.

32.2.11 Manchester Mode Control Register (MMCR)

Address(es): RSCI9.MMCR 000A 14ACh

	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 32.5.7, 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 32.5.6, 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 32.5.9, 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 32.30 and Figure 32.31.

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 32.29.
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 32.29.
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.

32.2.12 DE Signal Control Register (DECR)

Address(es): RSCI0.DECR 000A 1030h, RSCI8.DECR 000A 1430h, RSCI9.DECR 000A 14B0h

	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 (1/6, 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 (1/6, 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.).

32.2.13 Extended Serial Mode Control Register 0 (XCR0)

Address(es): RSCI9.XCR0 000A 14B4h

	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: PCLK 0 1: PCLK/4 1 0: PCLK/16 1 1: PCLK/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.

32.2.14 Extended Serial Mode Control Register 1 (XCR1)

Address(es): RSCI9.XCR1 000A 14B8h

	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.

32.2.15 Extended Serial Mode Control Register 2 (XCR2)

Address(es): RSCI9.XCR2 000A 14BCh

	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.

32.2.16 Status Register (SSR)

Address(es): RSCI0.SSR 000A 1048h, RSCI8.SSR 000A 1448h, RSCI9.SSR 000A 14C8h

	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 1: Mode fault	R
b27	APER	Aggregate Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R
b28	AFER	Aggregate Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R
b29	TDRE	Transmit Data Empty Flag	0: Transmit data is in TDR register 1: No transmit data is in TDR register	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	0: No received data is in RDR register 1: Received data is in RDR register	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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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]

- 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.

[Clearing condition]

- When write 1 to SSCR.ORERC bit.

Refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts to exit from the interrupt handling routine after setting the ORER flag to 0.

MFF Flag (Mode Fault Flag)

This bit indicates mode faults. In a multi-master configuration, determine the mode fault 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]

- When a parity error is detected during reception.

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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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]

- When 0 is sampled as the stop bit during reception.
- 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.

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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts to exit from the interrupt handling routine after setting the AFER flag to 0.

TDRE Flag (Transmit Data Empty Flag)

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.

TEND Flag (Transmit End Flag)

(1) 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts to exit from the interrupt handling routine after setting the TEND flag to 0.
(2) 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.

RDRF Flag (Receive Data Full Flag)

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.

Note: Do not clear the RDRF and TDRE flags by the SSCR register except when the communication is to be aborted.

32.2.17 I²C Status Register (SISR)

Address(es): RSCI0.SISR 000A 104Ch, RSCI8.SISR 000A 144Ch, RSCI9.SISR 000A 14CCh

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

32.2.18 Manchester Mode Status Register (MMSR)

Address(es): RSCI9.MMSR 000A 14D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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 32.5.11, Errors in Manchester Mode.
[Clearing condition]
Write 1 to MMSCR.MCERC bit.
Refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts 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.

32.2.19 Extended Serial Mode Status Register 0 (XSR0)

Address(es): RSCI9.XSR0 000A 14DCh

	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 PCLK 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.

32.2.20 Extended Serial Mode Status Register 1 (XSR1)

Address(es): RSCI9.XSR1 000A 14E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	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 32.67), 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.

32.2.21 Status Clear Register (SSCR)

Address(es): RSCI0.SSCR 000A 1068h, RSCI8.SSCR 000A 1468h, RSCI9.SSCR 000A 14E8h

	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

32.2.22 I²C Status Clear Register (SISCR)

Address(es): RSCI0.SISCR 000A 106Ch, RSCI8.SISCR 000A 146Ch, RSCI9.SISCR 000A 14ECh

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

32.2.23 Manchester Mode Status Clear Register (MMSCR)

Address(es): RSCI9.MMSCR 000A 14F4h

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

32.2.24 Extended Serial Mode Status Clear Register (XSCR)

Address(es): RSCI9.XSCR 000A 14F8h

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

32.2.25 HBS Support Mode Control Register (HBSCR)

Address(es): RSCI0.HBSCR 000A 101Eh, RSCI8.HBSCR 000A 141Eh, RSCI9.HBSCR 000A 149Eh

	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 32.6, HBS Support Mode.

32.3 Operation in Asynchronous Mode

Figure 32.4 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, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

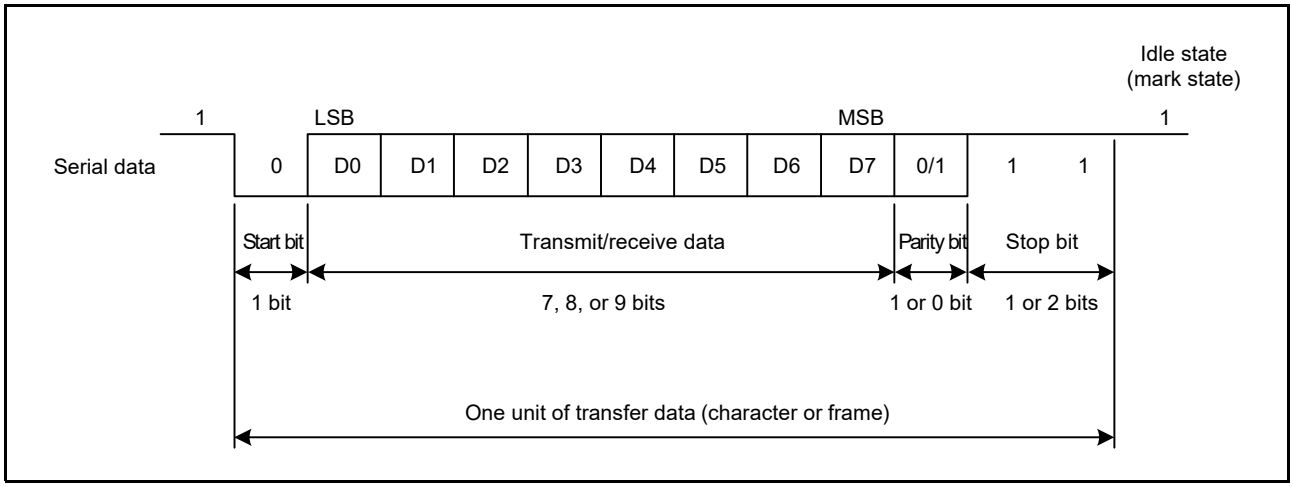


Figure 32.4 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

32.3.1 Serial Data Transfer Format

Table 32.28 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 32.4, Multi-Processor Communication Function.

Table 32.28 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

32.3.2 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 32.5. 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 32.5, 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.

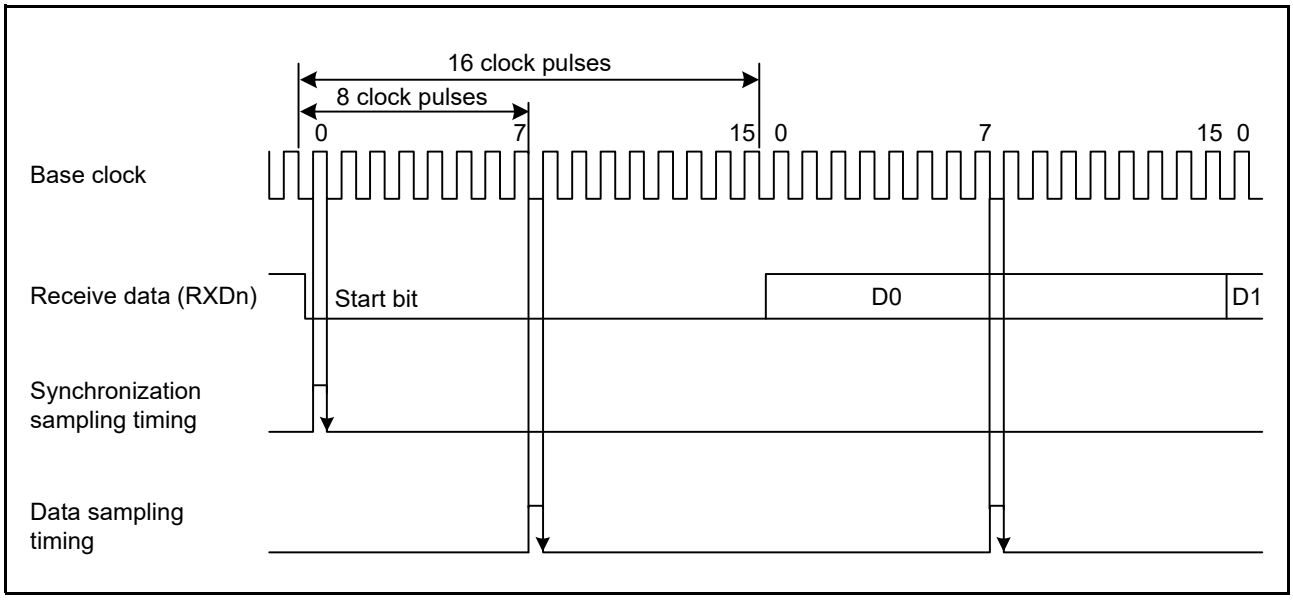


Figure 32.5 Receive Data Sampling Timing in Asynchronous Mode

32.3.3 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 32.6.

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.

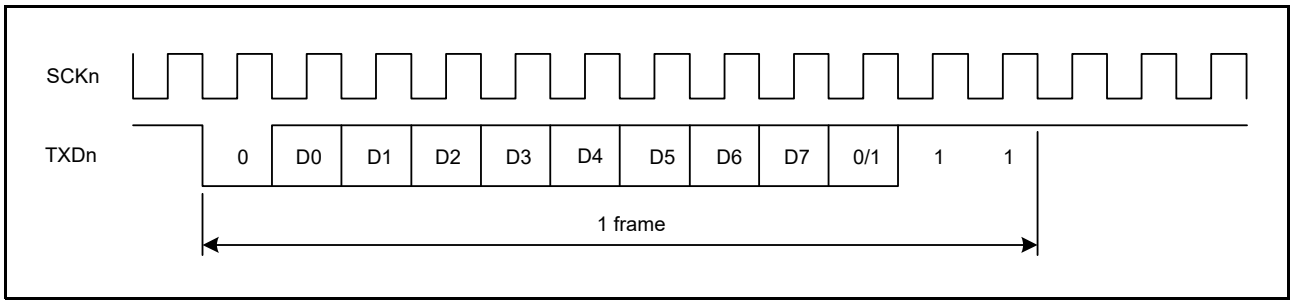


Figure 32.6 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SCR1.PE Bit = 1, SCR3.CHR[1:0] Bits = 10b, MP Bit = 0, STOP Bit = 1)

32.3.4 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 32.3.2, 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.

32.3.5 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.

[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

32.3.6 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 32.7 and Figure 32.8 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.

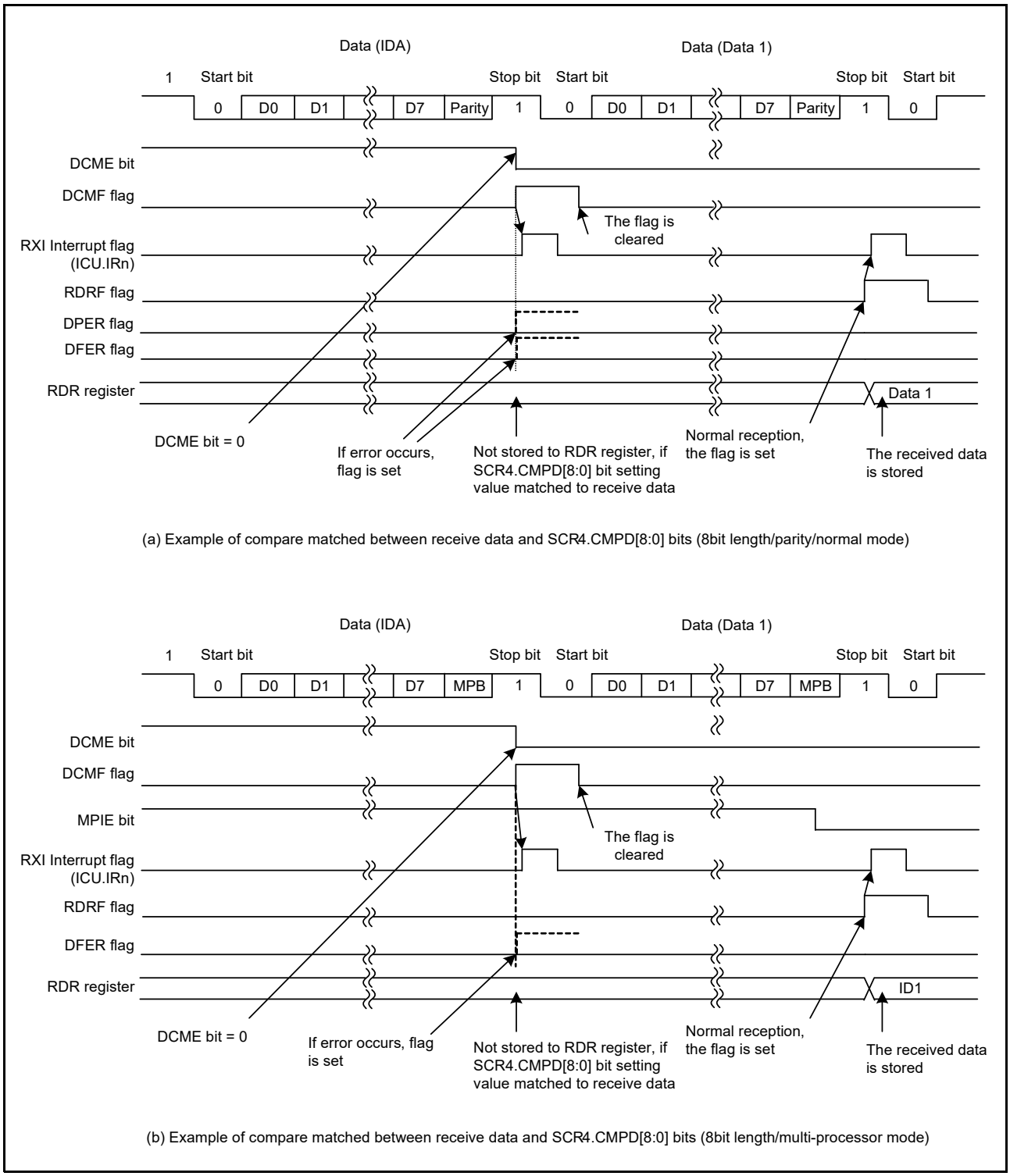
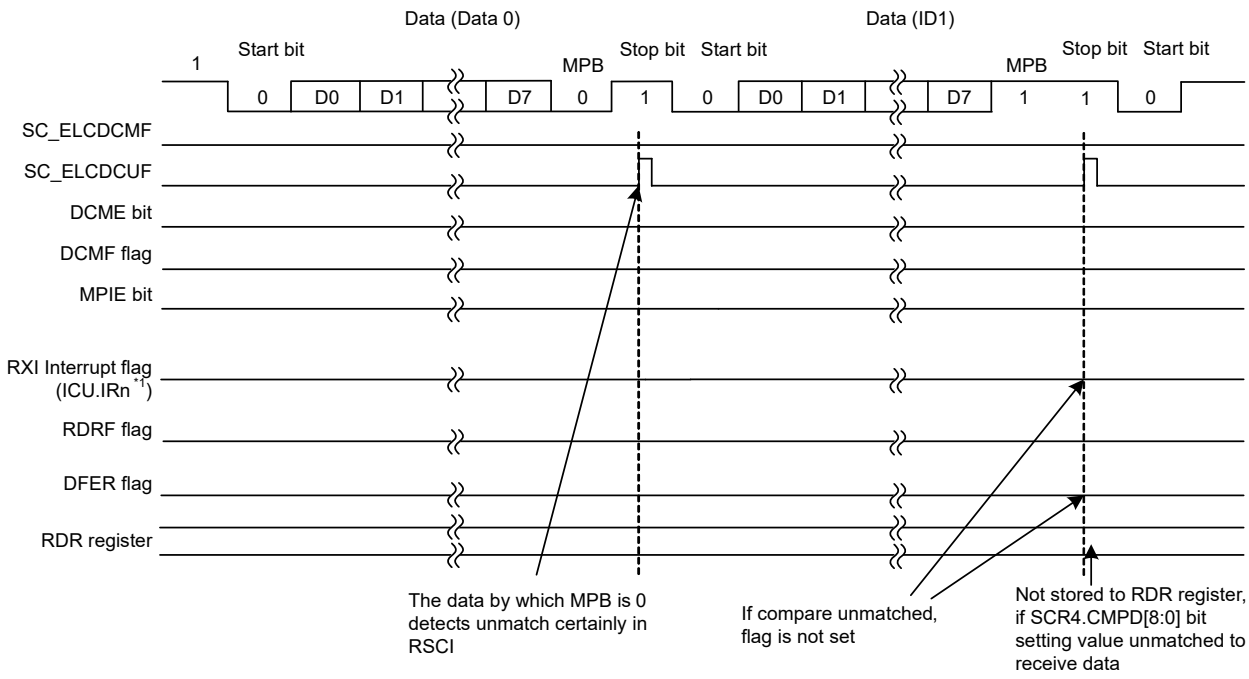
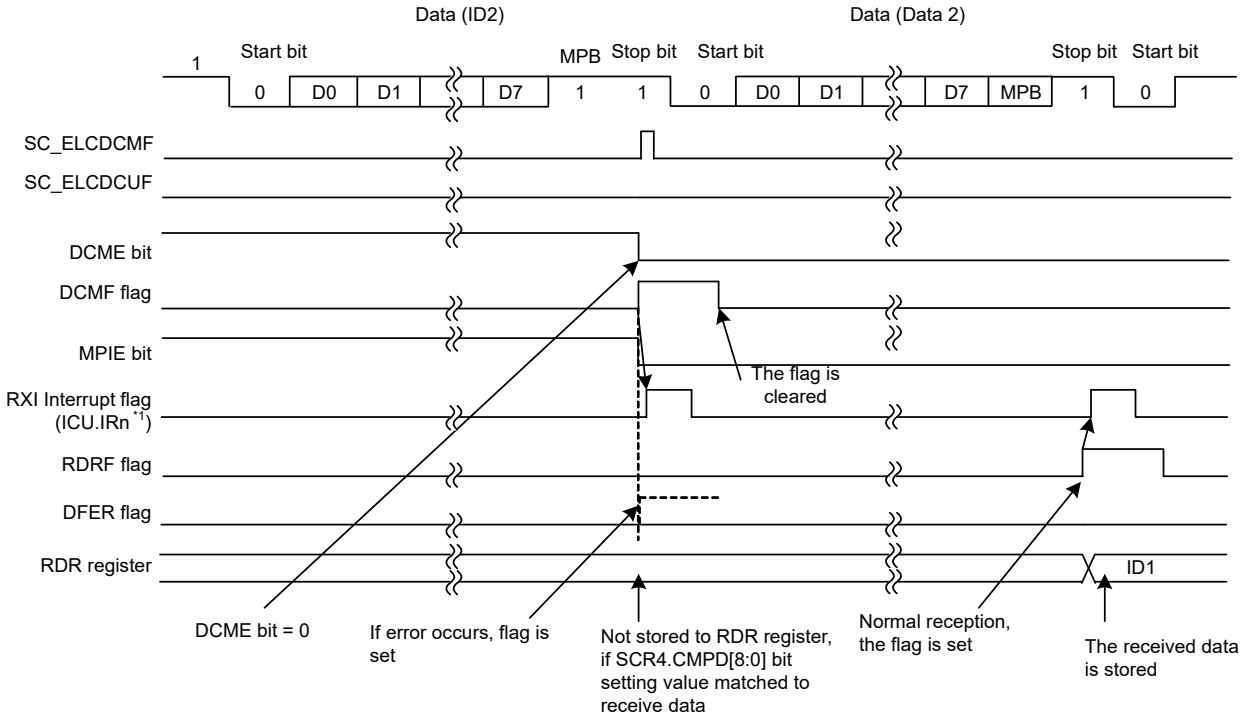


Figure 32.7 Example of Data Match Detection (1) 8bit-Data



(a) Example of compare unmatched between receive data and SCR4.CMPD[8:0] bits (8bit length/IDSEL = 1)



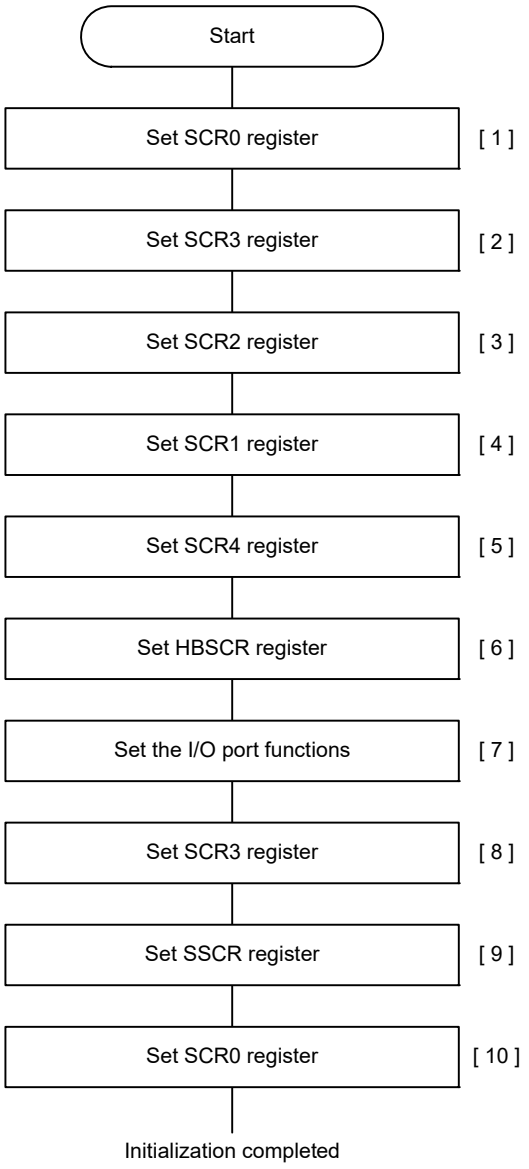
(b) Example of compare matched between receive data and SCR4.CMPD[8:0] bits (8bit length/IDSEL = 1)

Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.8 Example of Data Match Detection (2) Multi-Processor Mode/8bit-Data

32.3.7 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 for RSCI given in Figure 32.9. 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, and RDR register. 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
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 32.9 Sample RSCI Initialization Flowchart (Asynchronous Mode)

Figure 32.10 shows an example of the timing when data is transmitted after reset is released, and RSCI is set to asynchronous mode according to Figure 32.9. 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.

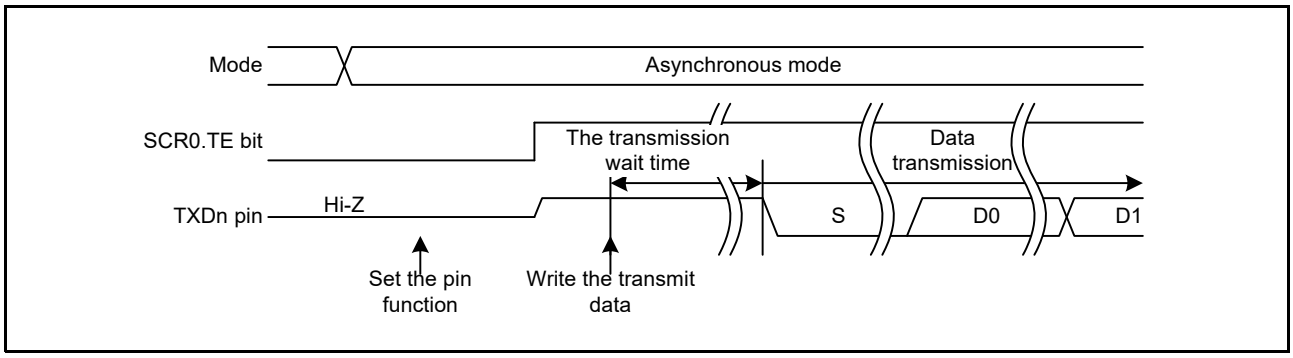


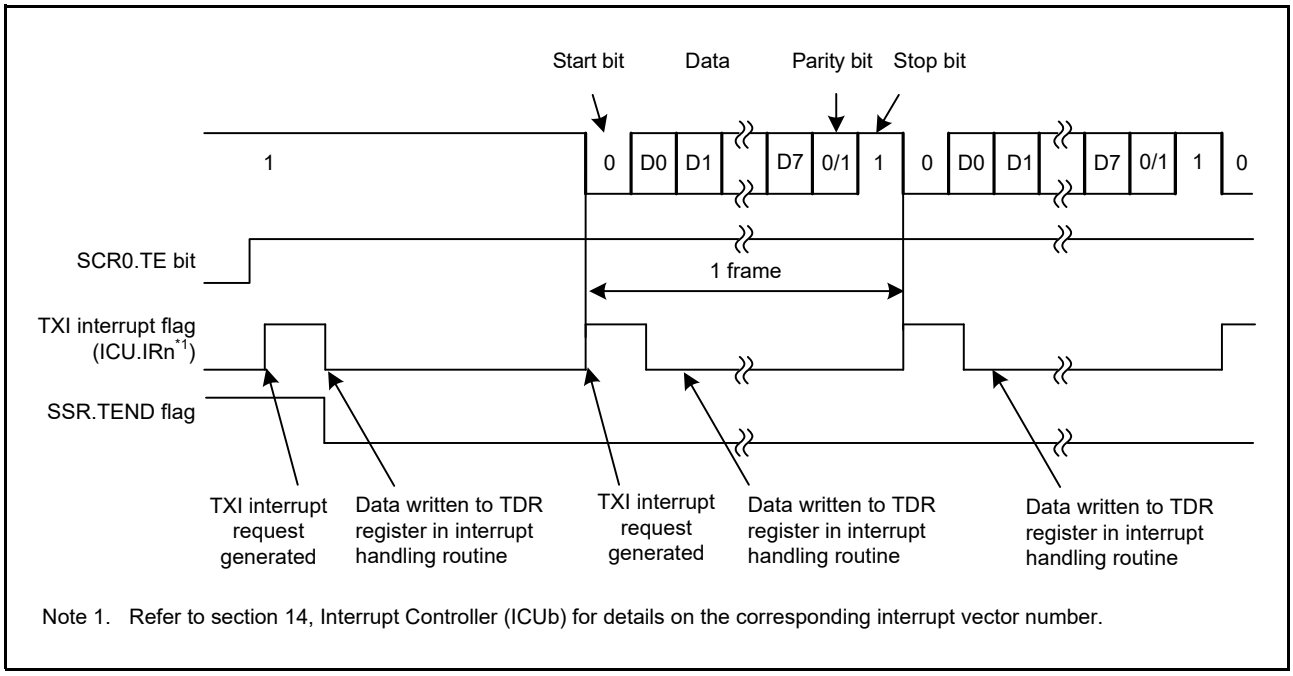
Figure 32.10 Example of Data Transmission Timing in Asynchronous Mode

32.3.8 Serial Data Transmission (Asynchronous Mode)

Figure 32.11 to Figure 32.13 show an example of the operation for serial transmission in asynchronous mode. In serial transmission, the RSCI operates as described below.

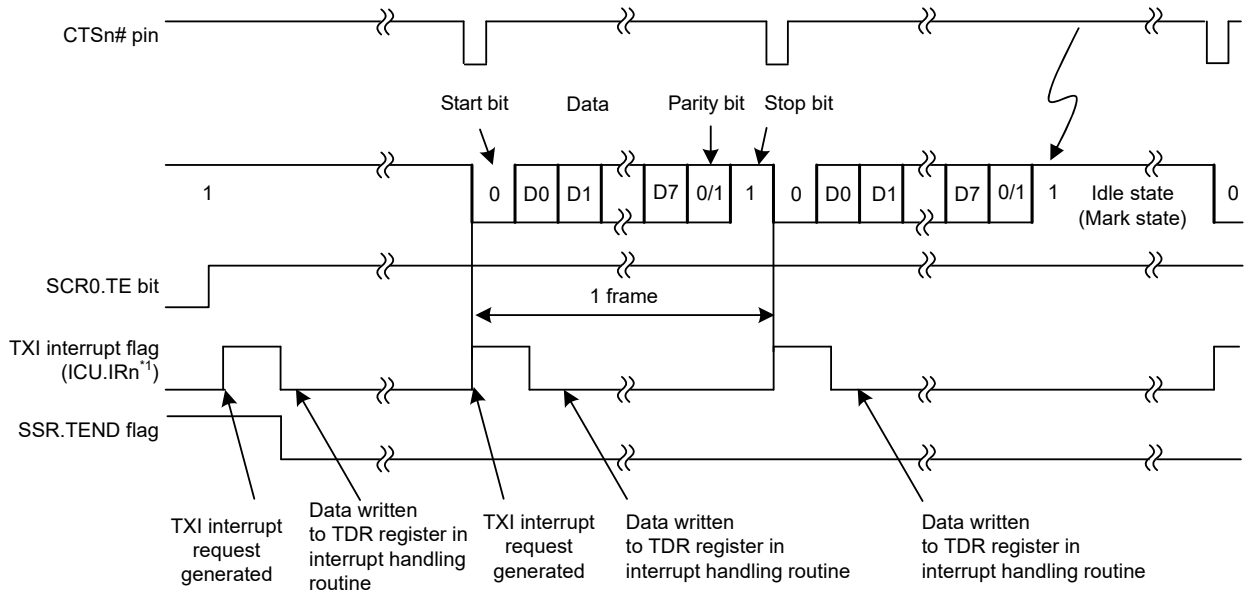
1. The RSCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. At the beginning of transmission, set 1 to the SCR0.TE bit and the SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.
2. Transmission 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 32.15 shows the example of Serial Transmission Flowchart.



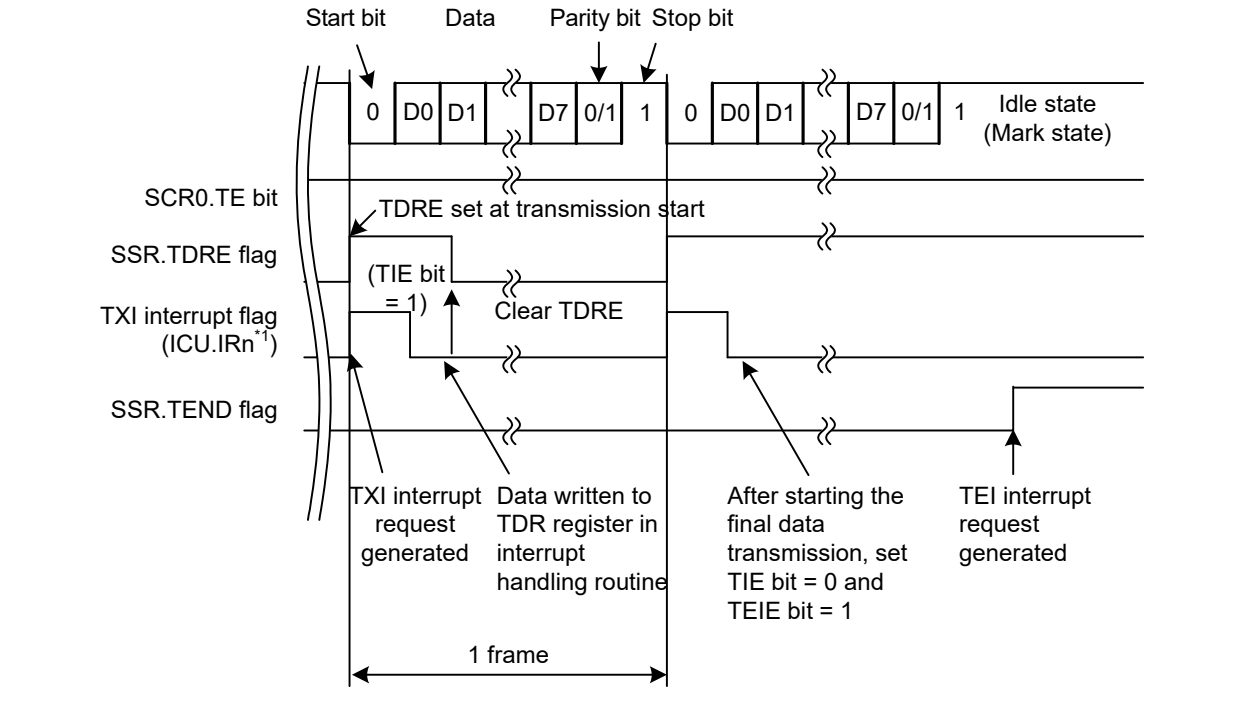
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.11 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)



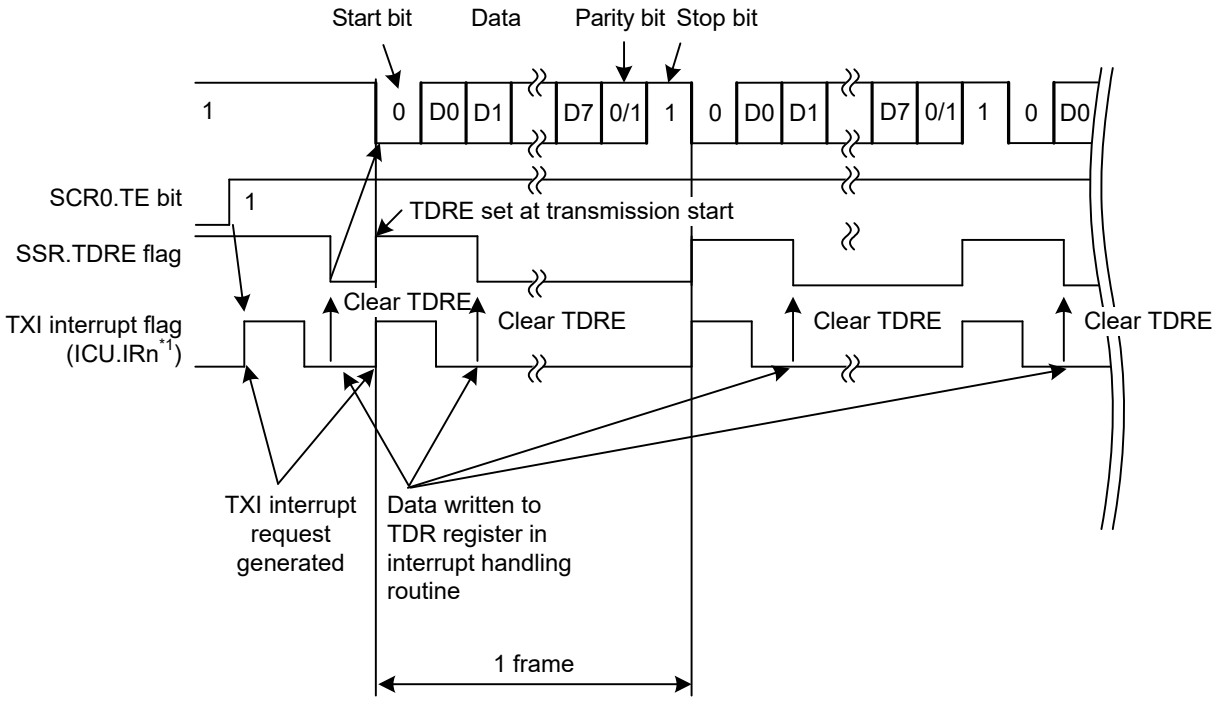
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.12 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.13 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.14 Example of Operation for Serial Transmission in Asynchronous Mode (4) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, during Transmission)

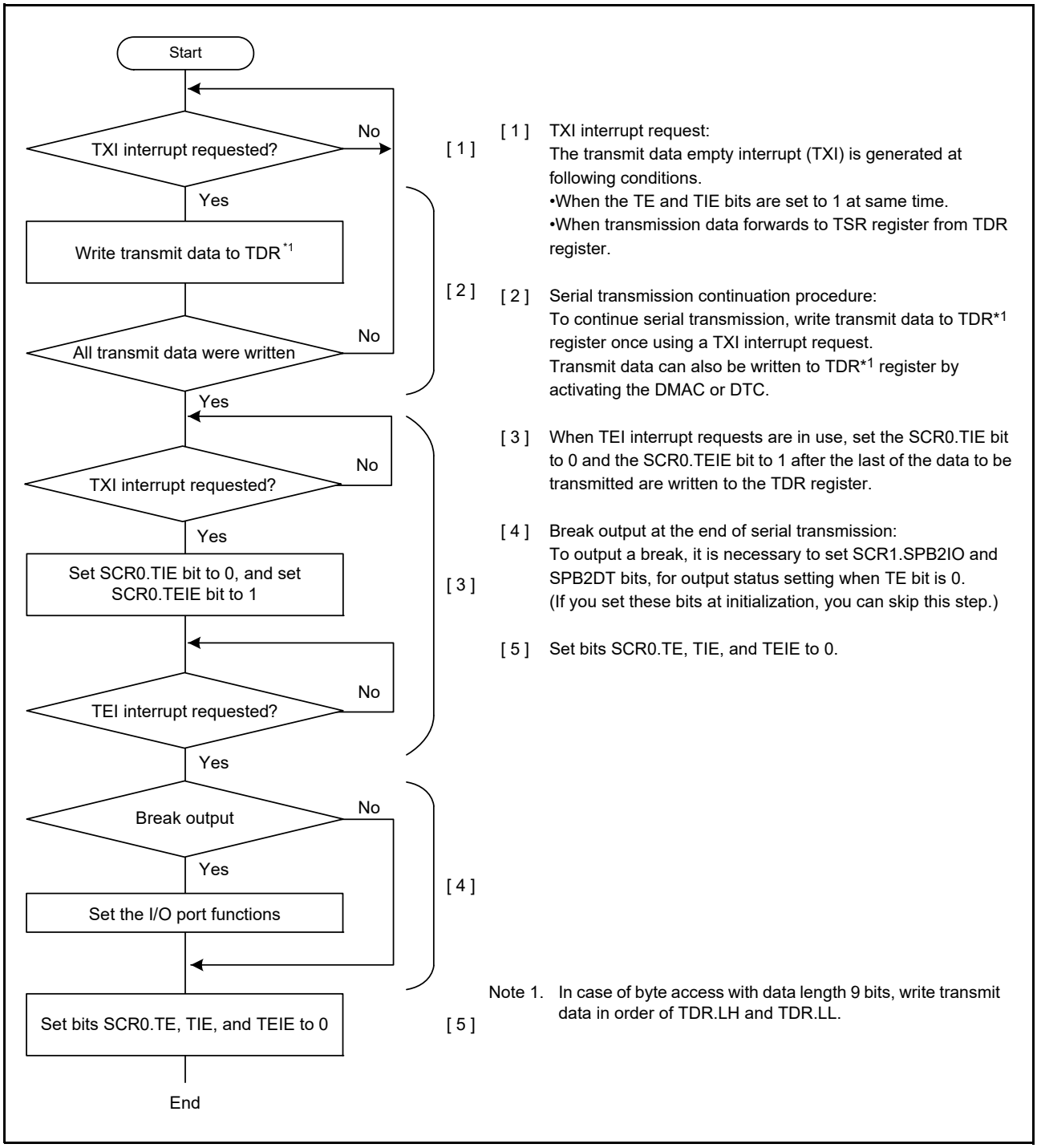
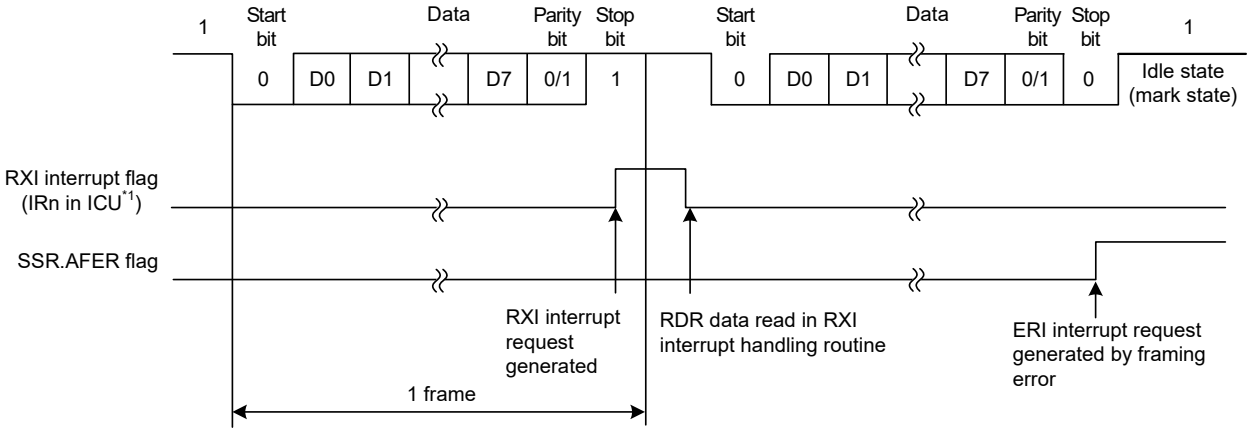


Figure 32.15 Example of Serial Transmission Flowchart in Asynchronous Mode

32.3.9 Serial Data Reception (Asynchronous Mode)

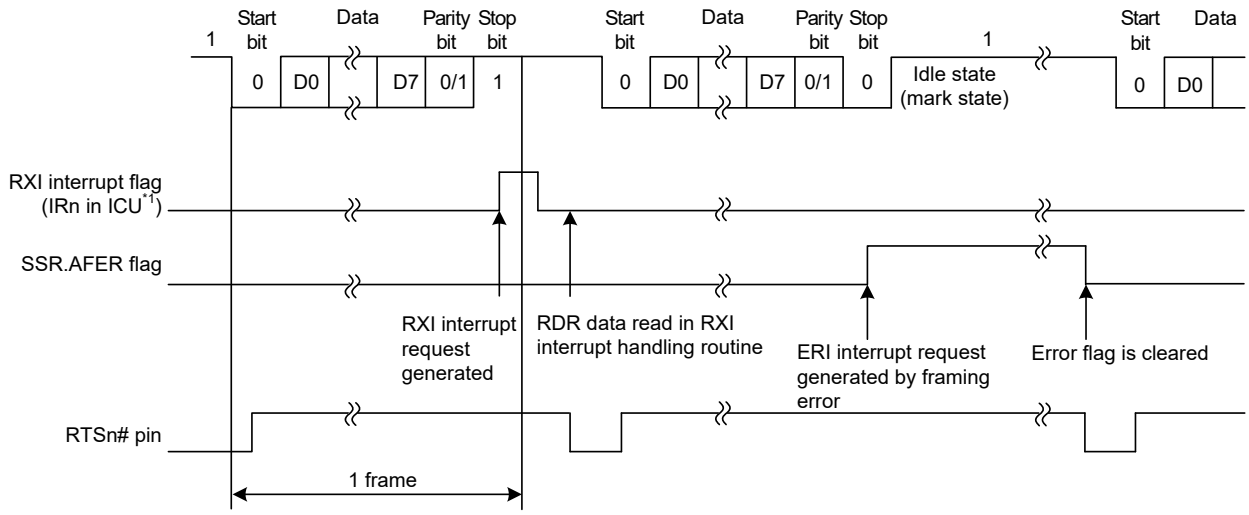
Figure 32.16 and Figure 32.17 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the RSCI operates as described below.

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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.16 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.17 Example of RSCI Operation for Serial Reception in Asynchronous Mode (2) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Used)

Table 32.29 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 32.18 and Figure 32.19 show samples of flowcharts for serial data reception.

Table 32.29 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

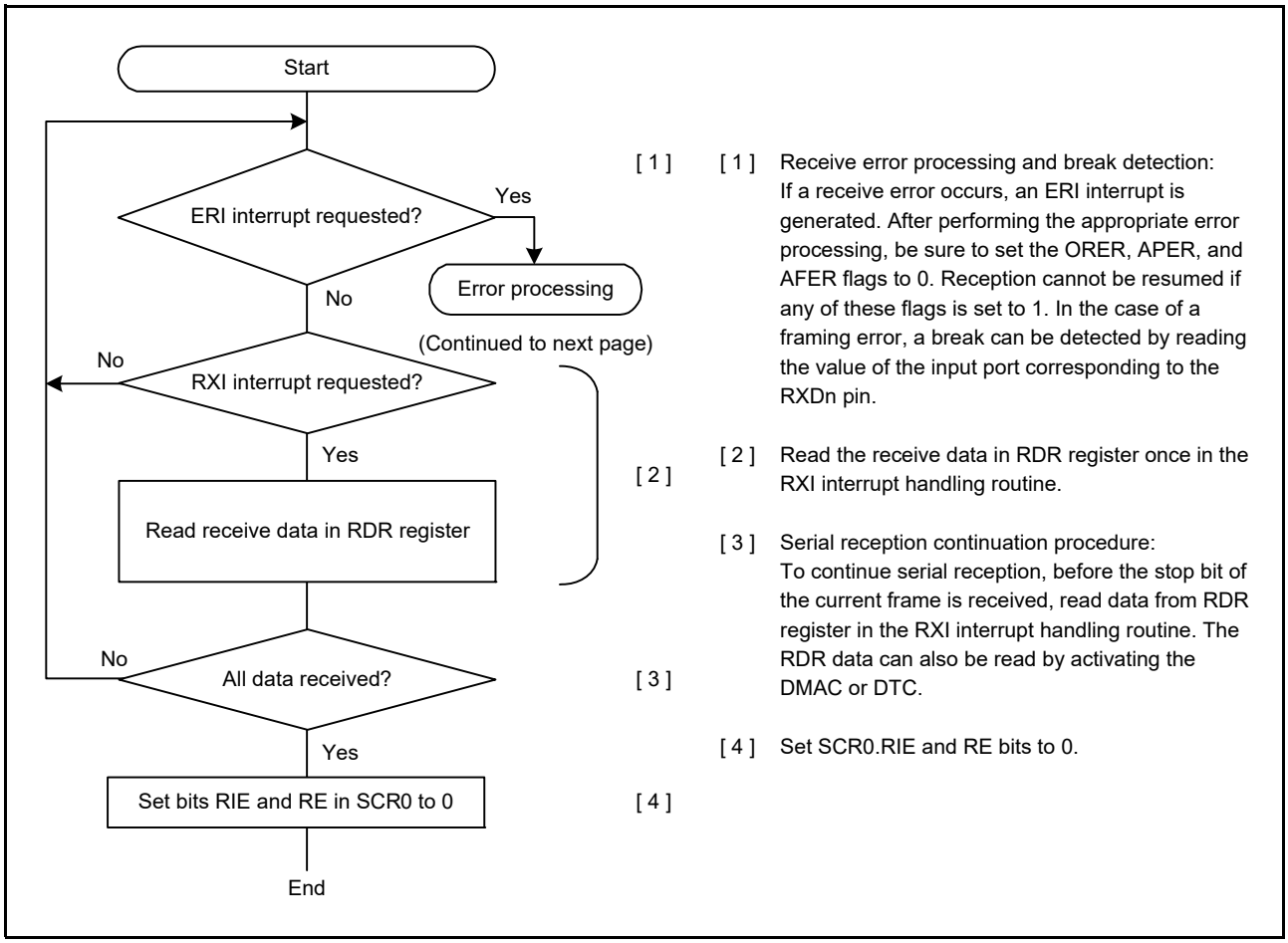


Figure 32.18 Example Flowchart of Serial Reception in Asynchronous Mode (1)

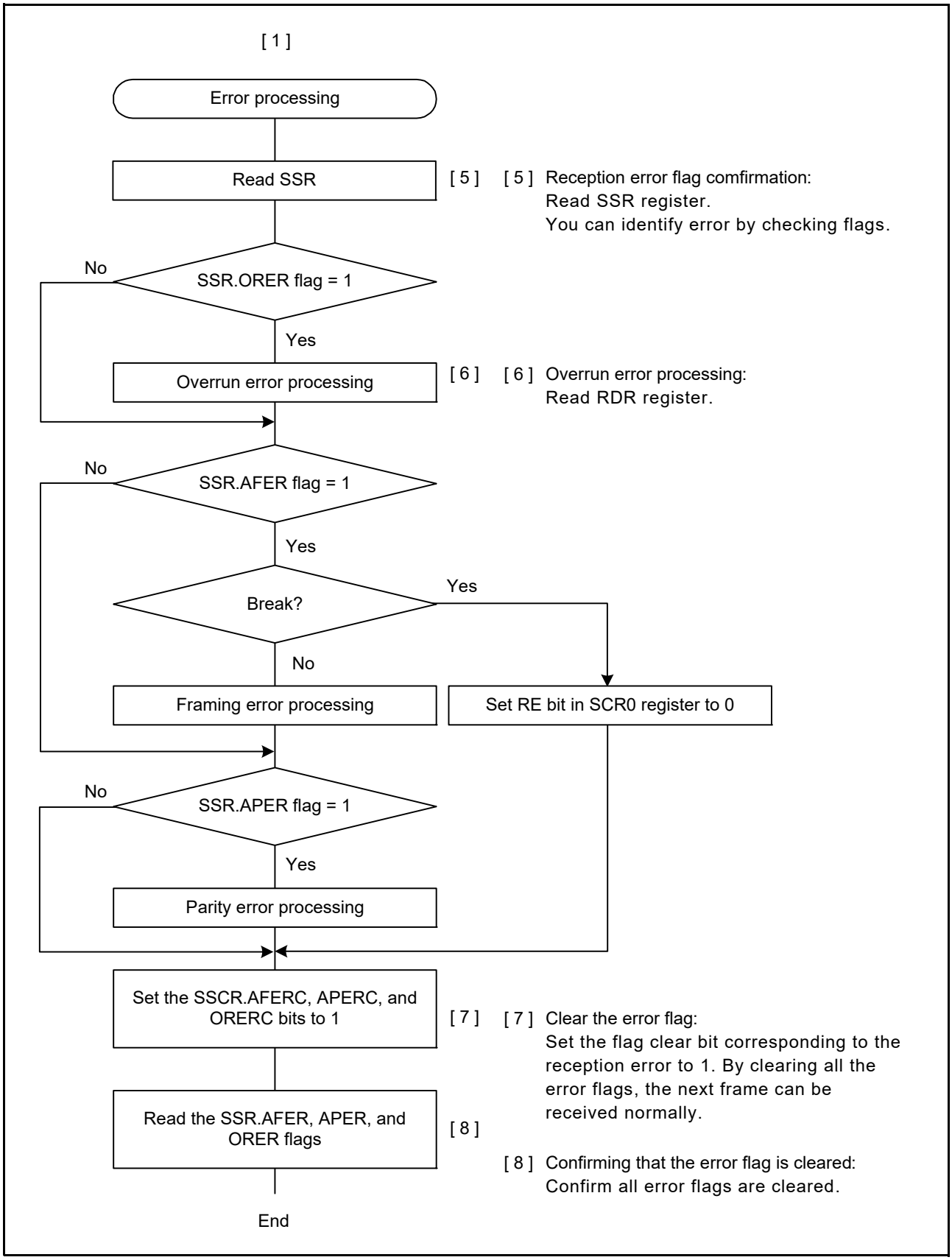


Figure 32.19 Example Flowchart of Serial Reception in Asynchronous Mode (2)

32.3.10 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 32.20 shows an example of the sampling timing adjustment.

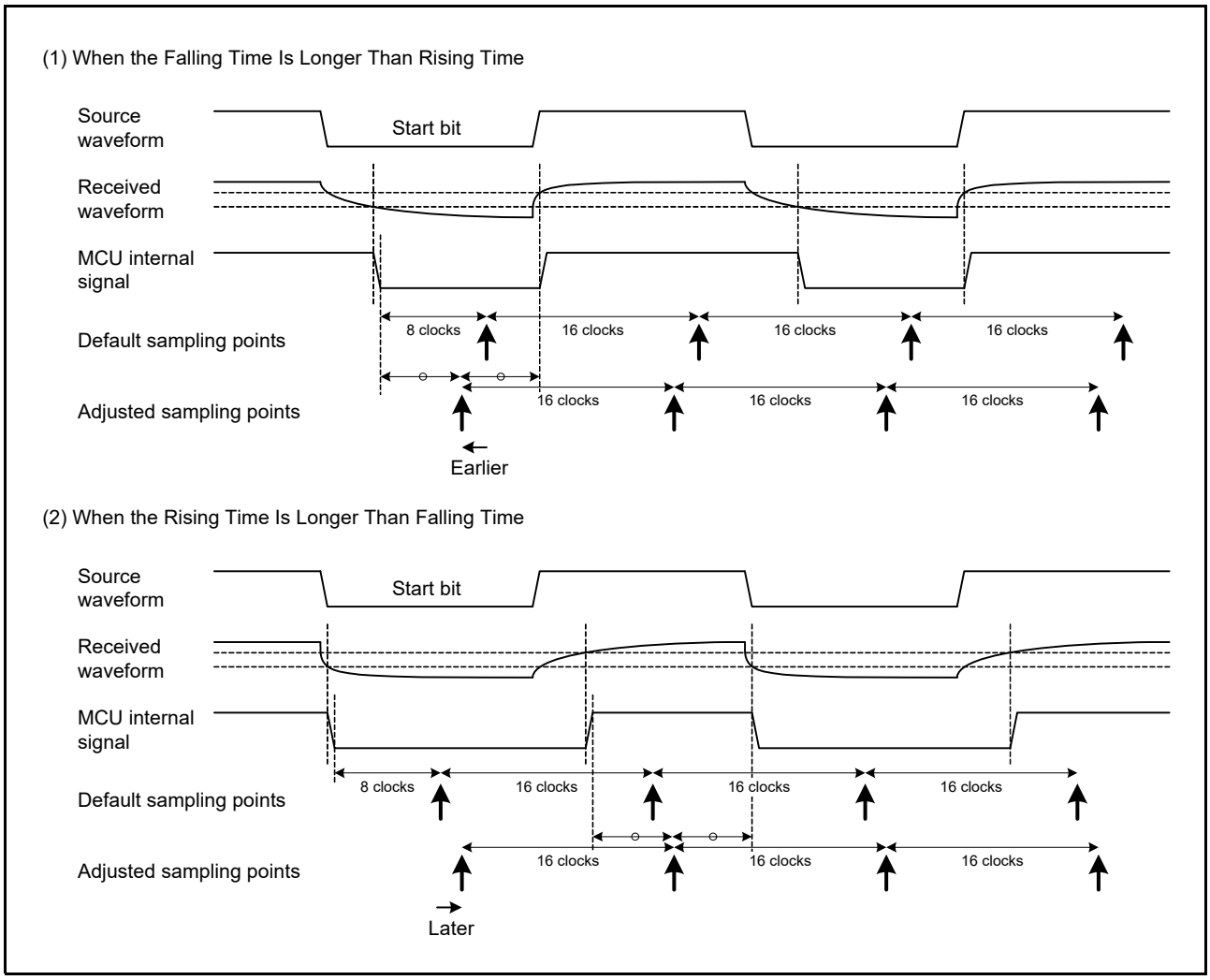


Figure 32.20 Example of Sampling Timing Adjustment (SCR2.ABCSE Bit = 0 and SCR2.ABCS Bit = 0)

32.3.11 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 32.21 shows an example of the transition timing adjustment.

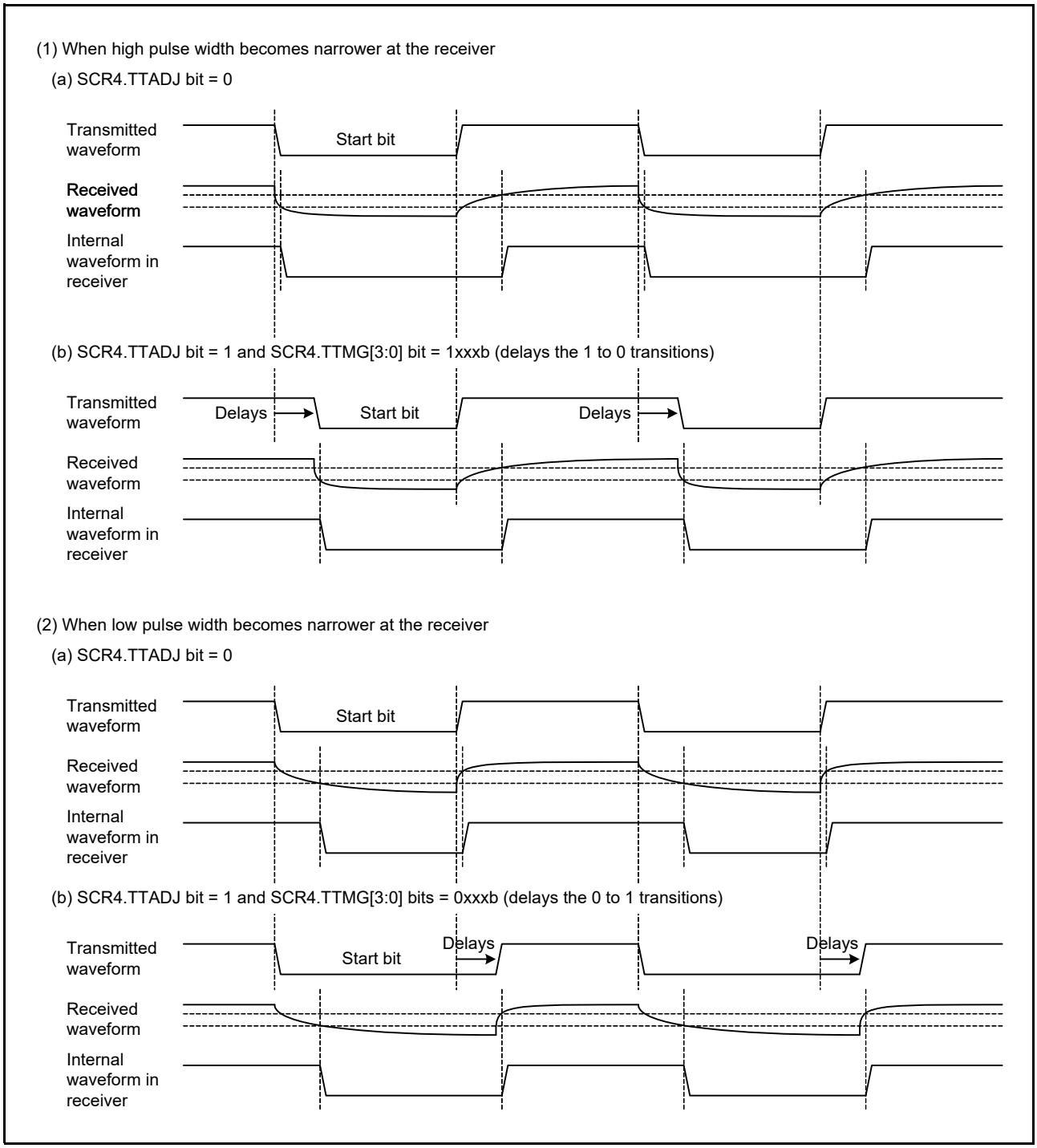


Figure 32.21 Example of Transition Timing Adjustment

32.4 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 32.22 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.

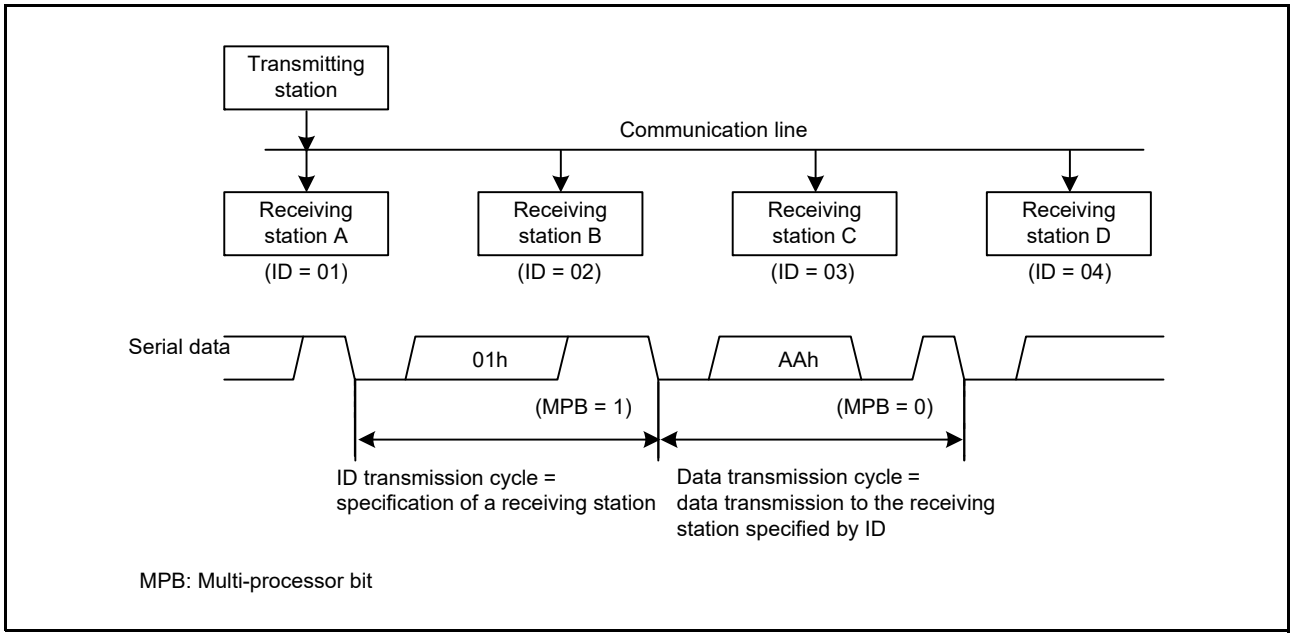


Figure 32.22 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

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.

32.4.1 Multi-Processor Serial Data Transmission

Figure 32.23 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.

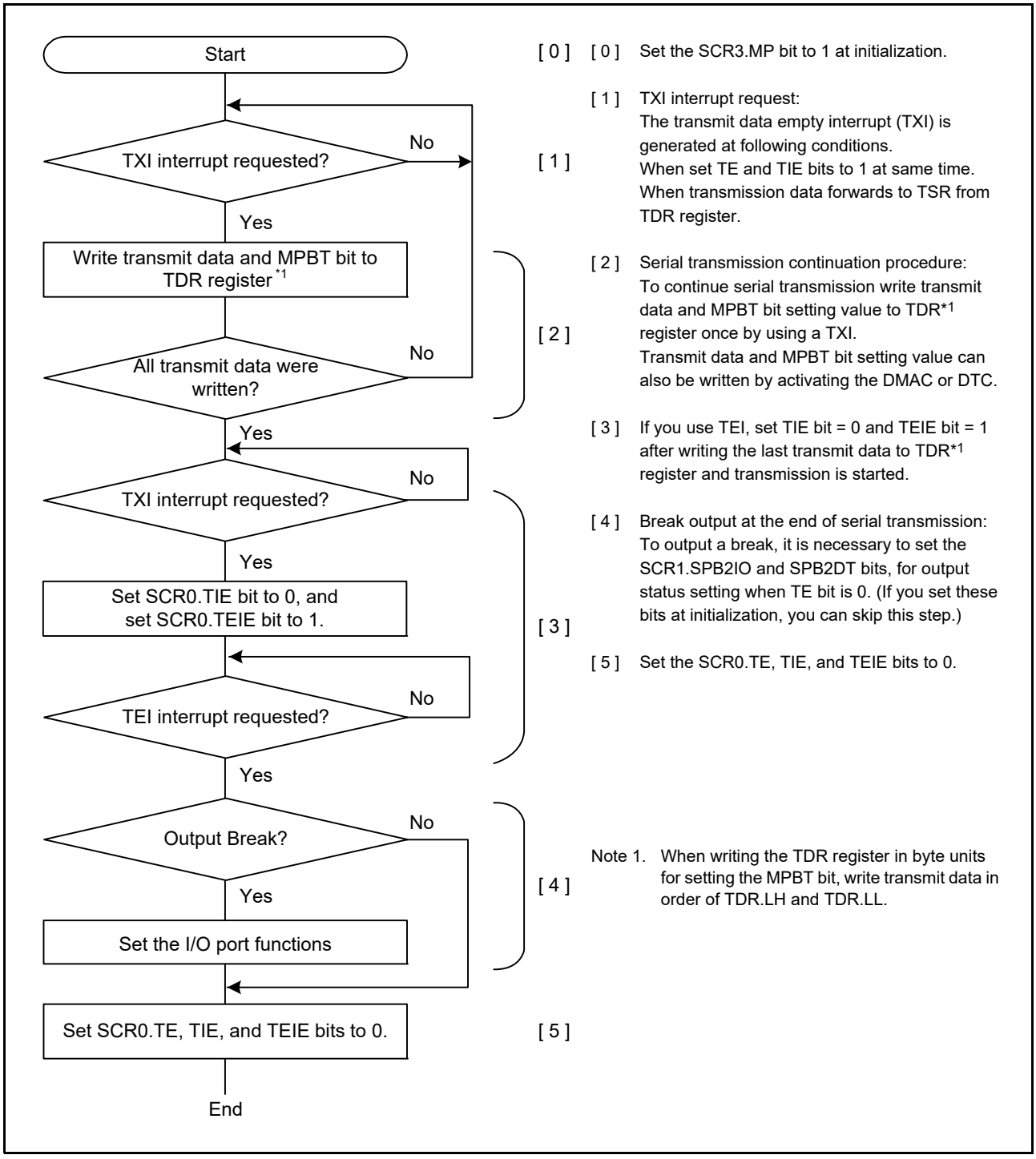
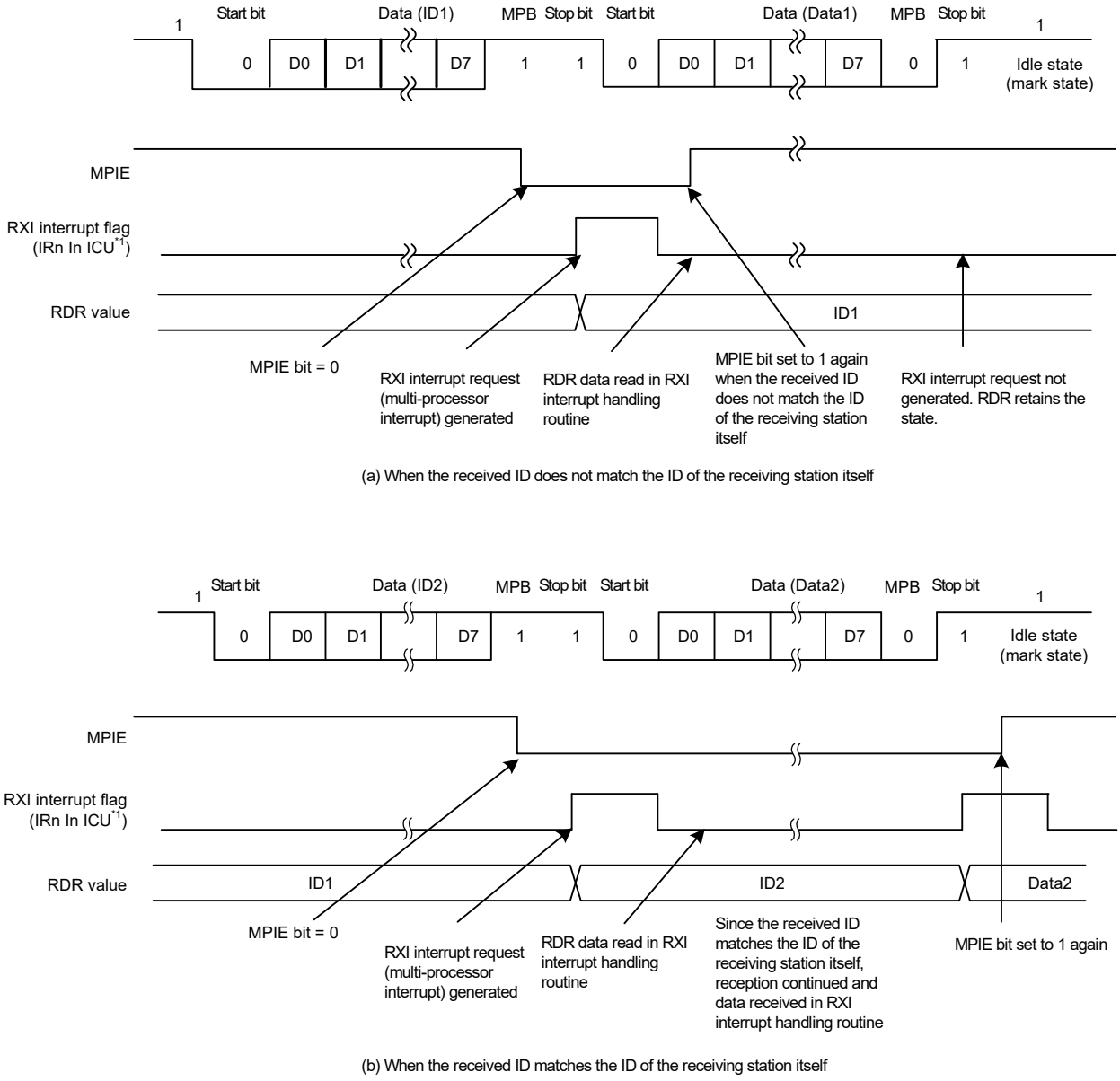


Figure 32.23 Example of Multi-Processor Serial Transmission Flowchart

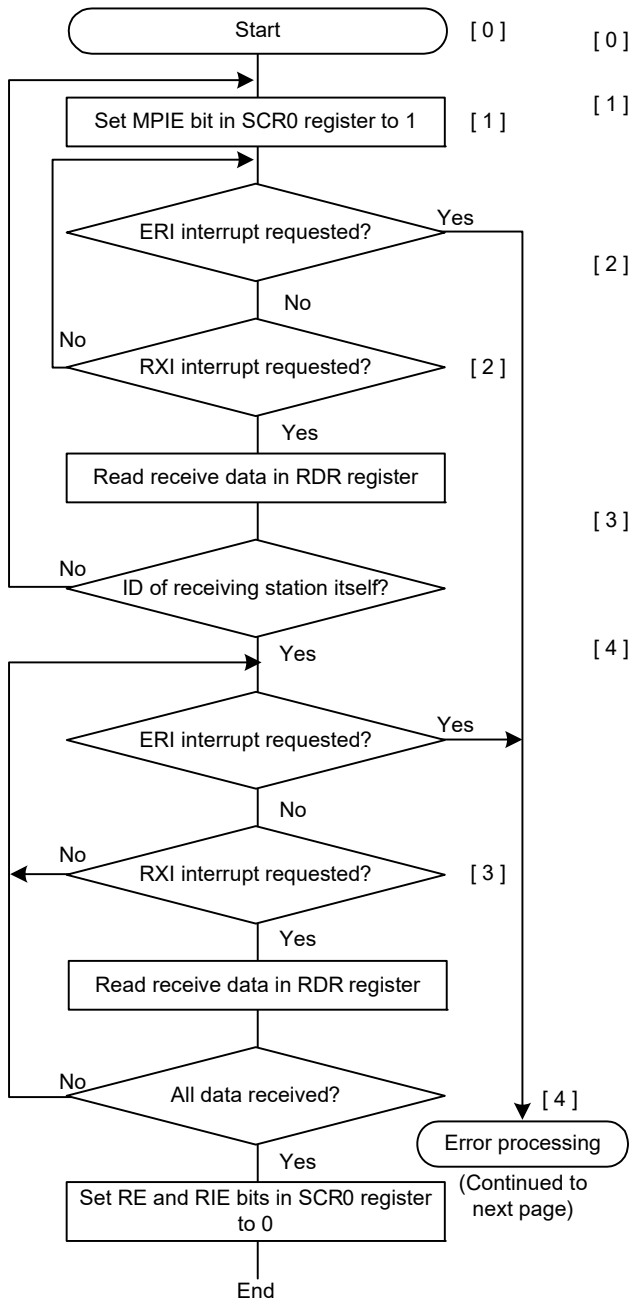
32.4.2 Multi-Processor Serial Data Reception

Figure 32.25 and Figure 32.26 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 32.24 is the example of operation for reception.



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.24 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 32.25 Example of Multi-Processor Serial Reception Flowchart (1)

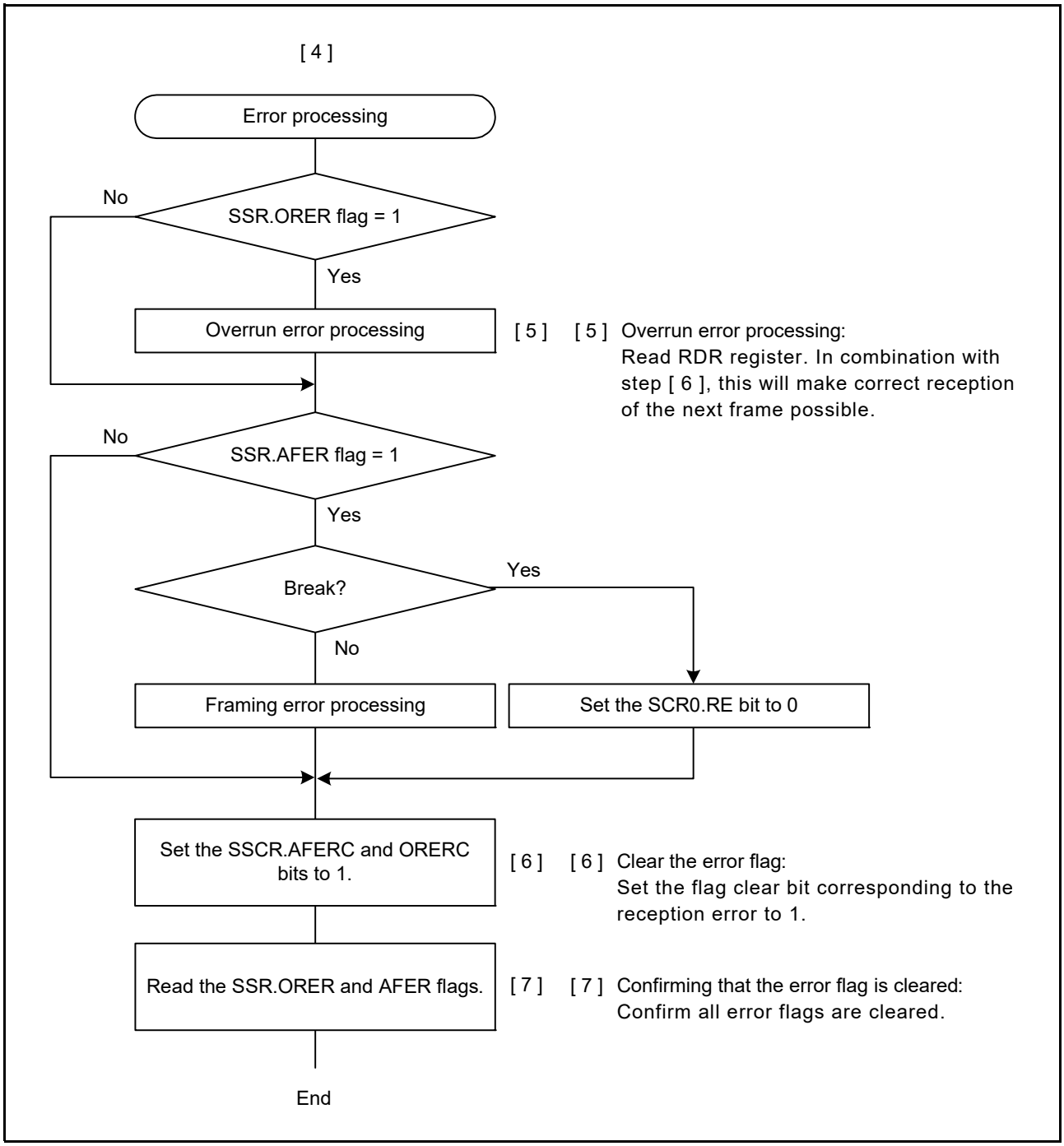


Figure 32.26 Example of Multi-Processor Serial Reception Flowchart (2)

32.5 Manchester Mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.
Figure 32.27 shows the conceptual image of Manchester encoding.

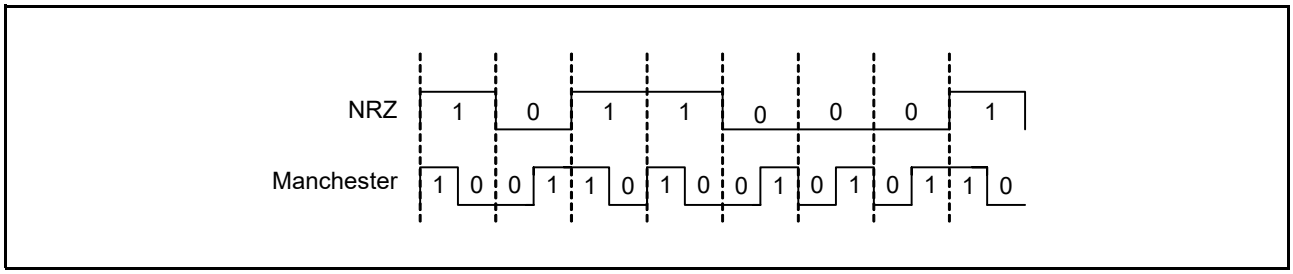


Figure 32.27 Example of Manchester Encoding

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 32.5.1, Frame Format.

32.5.1 Frame Format

Figure 32.28 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.

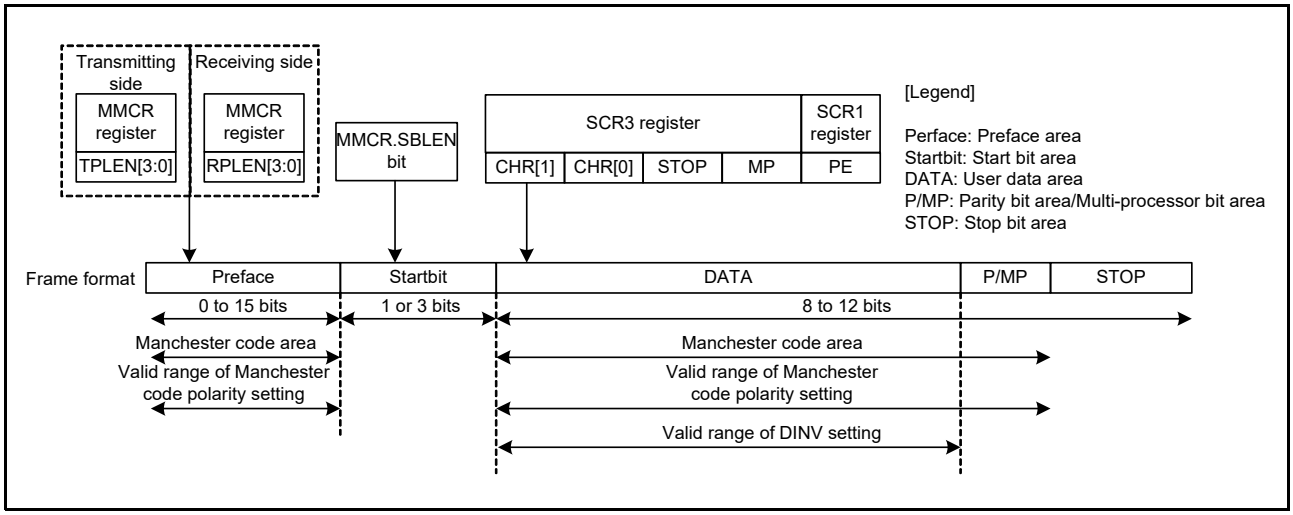


Figure 32.28 Frame Format in Manchester Mode

(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 32.29 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

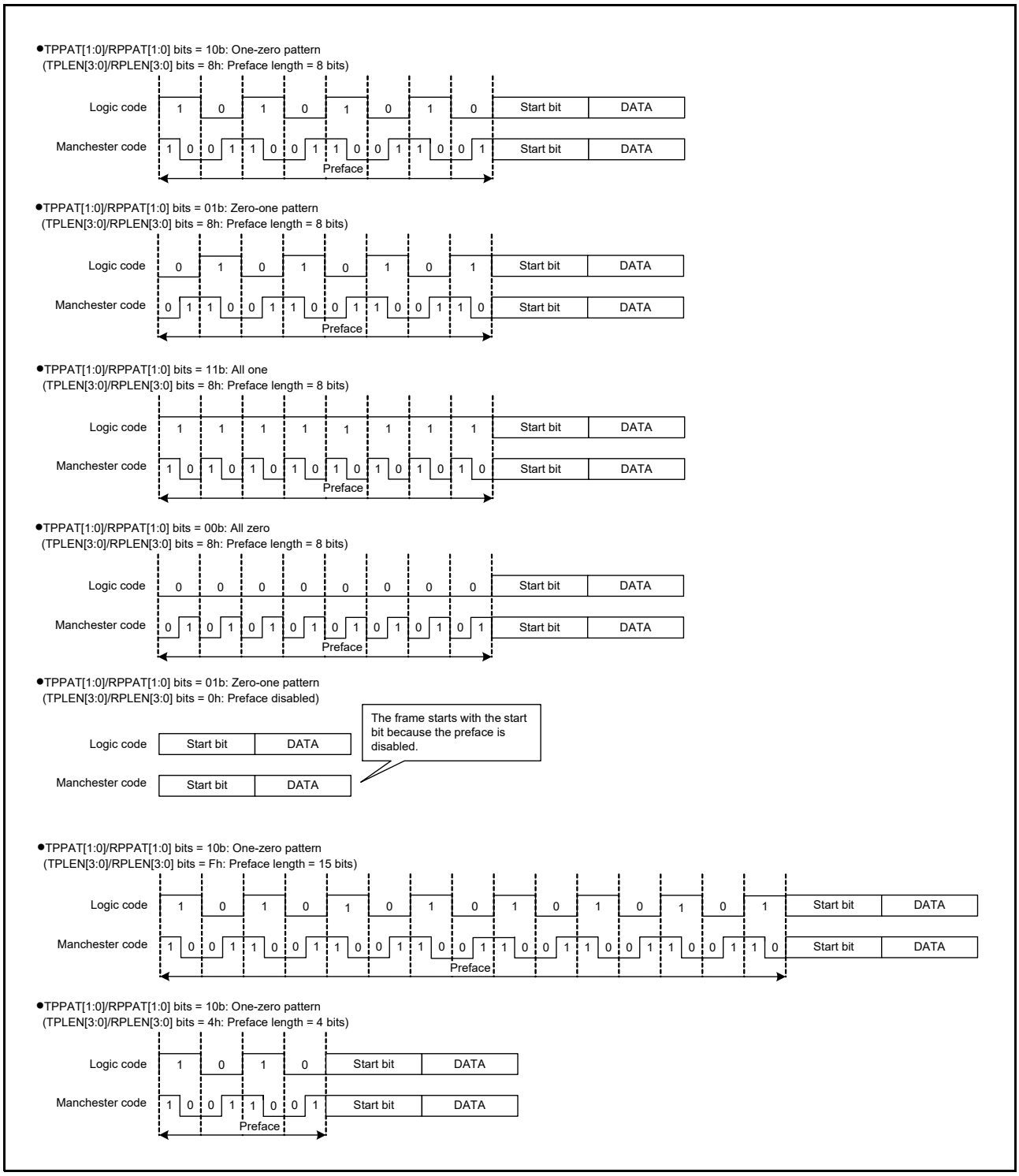
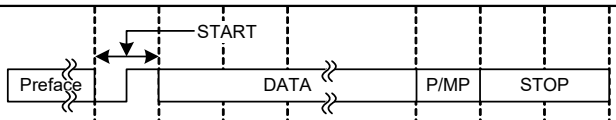
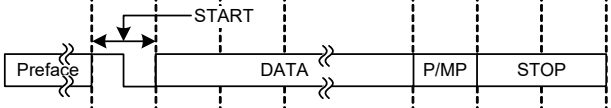
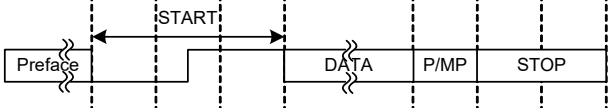
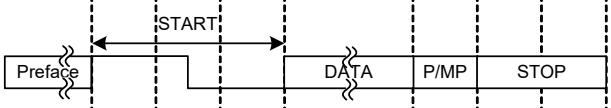
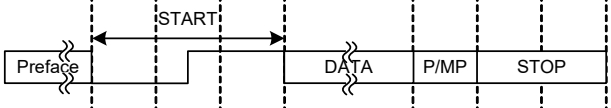
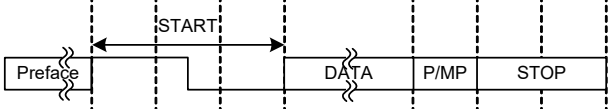


Figure 32.29 Preface Pattern Setting Example

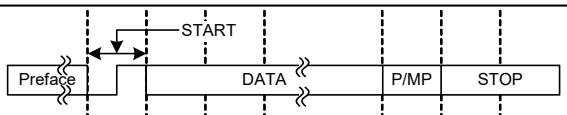
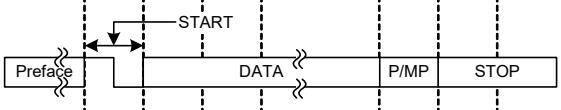
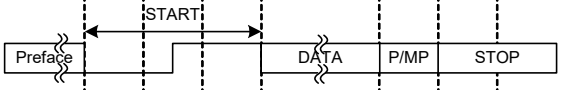
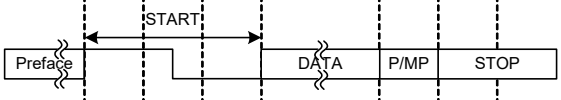
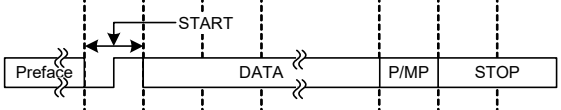
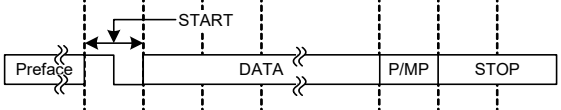
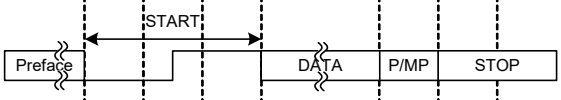
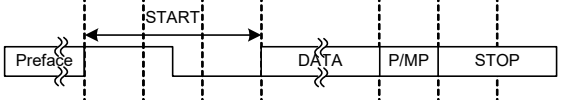
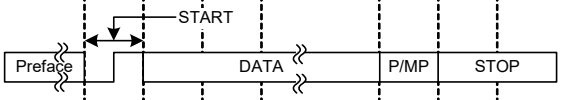
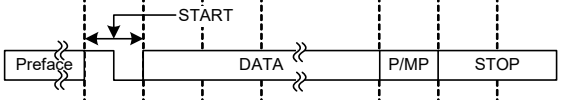
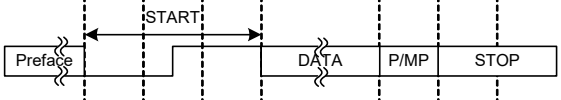
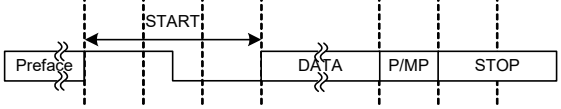
(2) Start Bit Area

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 32.30 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 32.31 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 32.30 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 32.31 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 32.3.1, Serial Data Transfer Format.
As shown in Figure 32.28, Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

32.5.2 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.

32.5.3 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 32.32.
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.

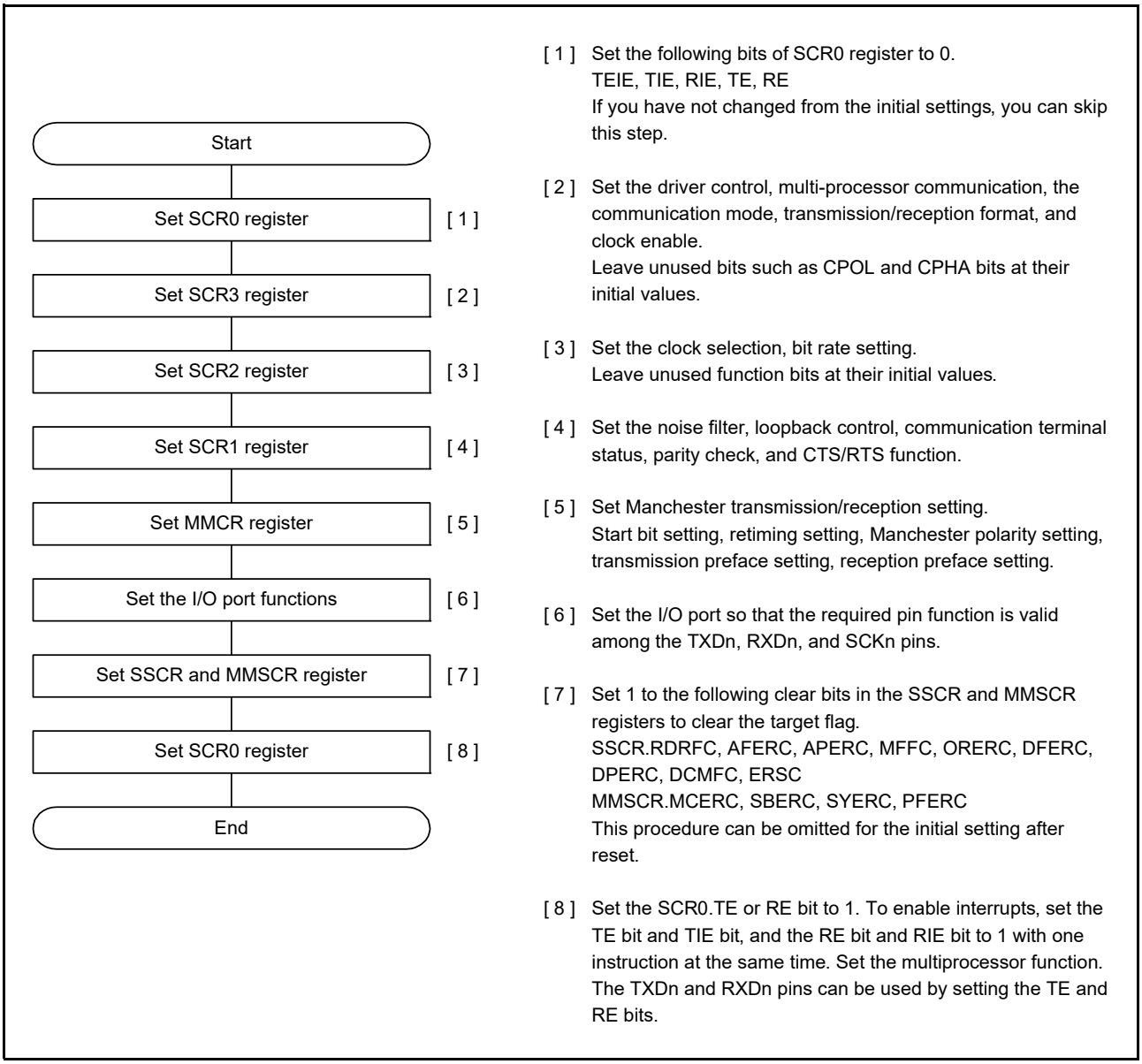


Figure 32.32 RSCI Initialization Flow in Manchester Mode

32.5.4 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.

32.5.5 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.

32.5.6 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 32.27)
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 32.5.1, Frame Format.
Figure 32.33 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 32.34 to Figure 32.36 show examples of the operation for serial transmission in Manchester mode.

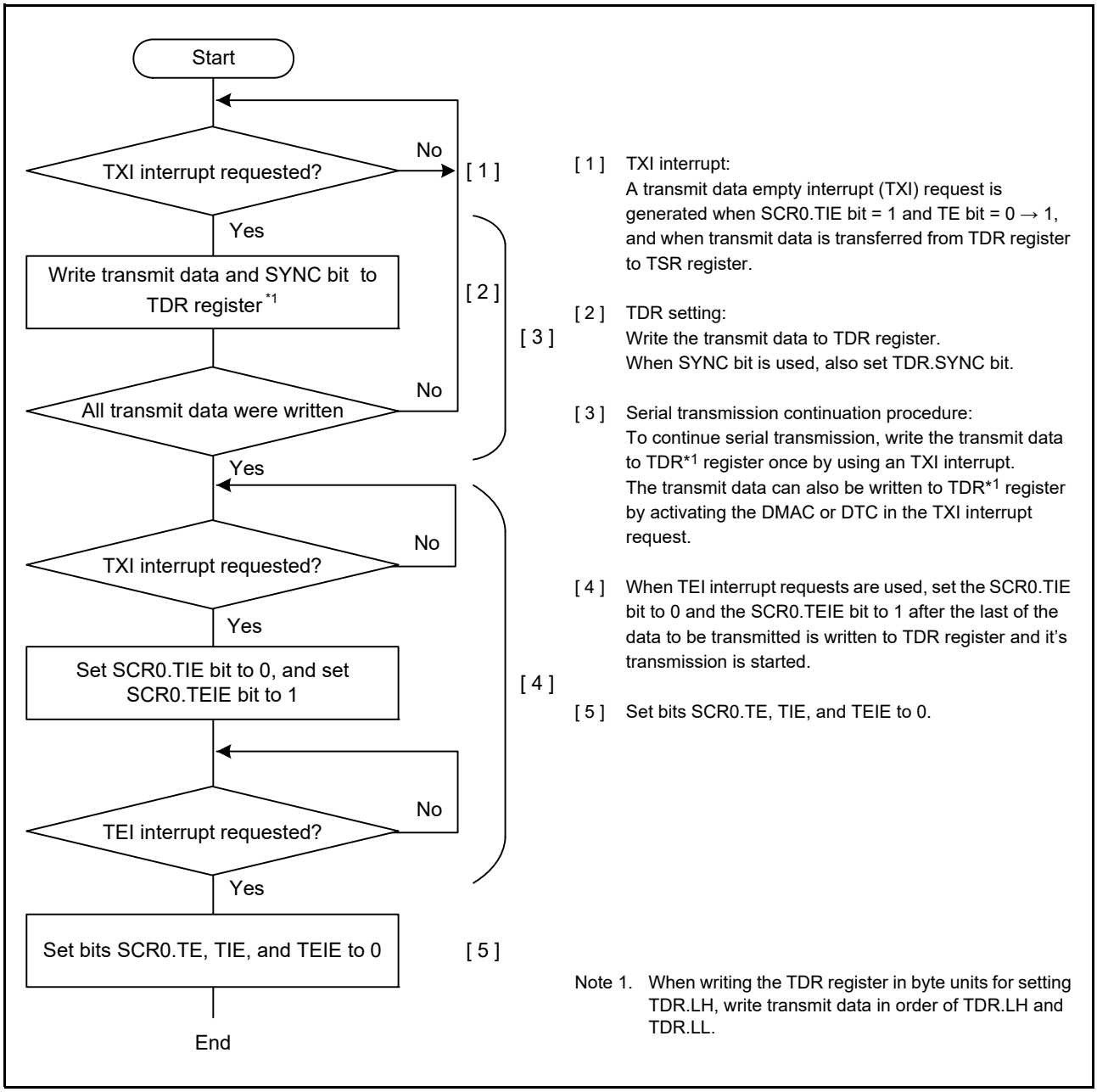
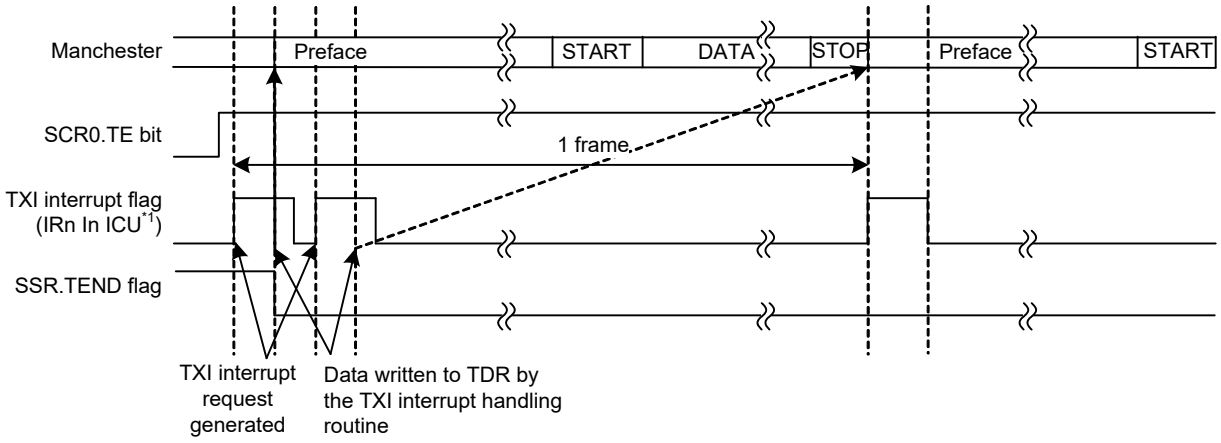
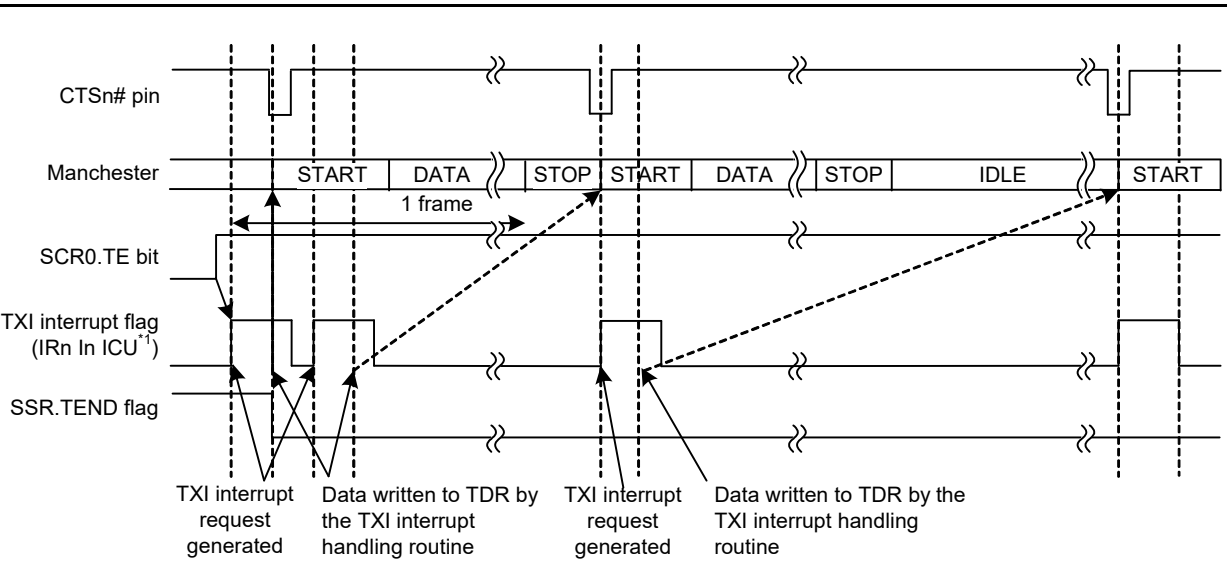


Figure 32.33 Serial Transmission Flowchart in Manchester Mode



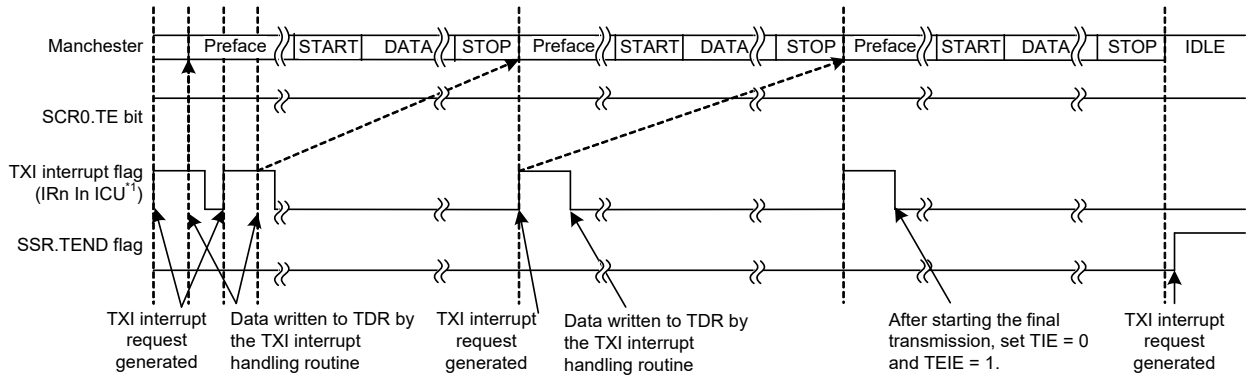
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.34 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.35 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.36 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)

32.5.7 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 32.37, 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.

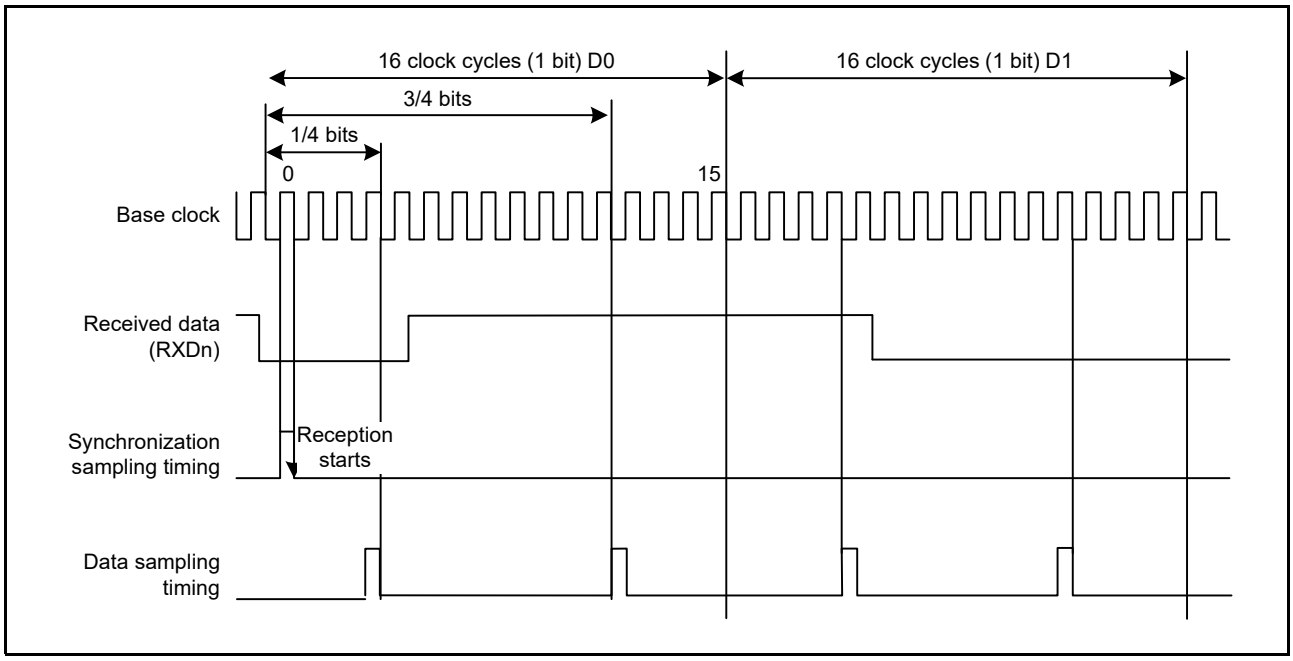


Figure 32.37 Data Reception Sampling Timing in Manchester Mode

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.

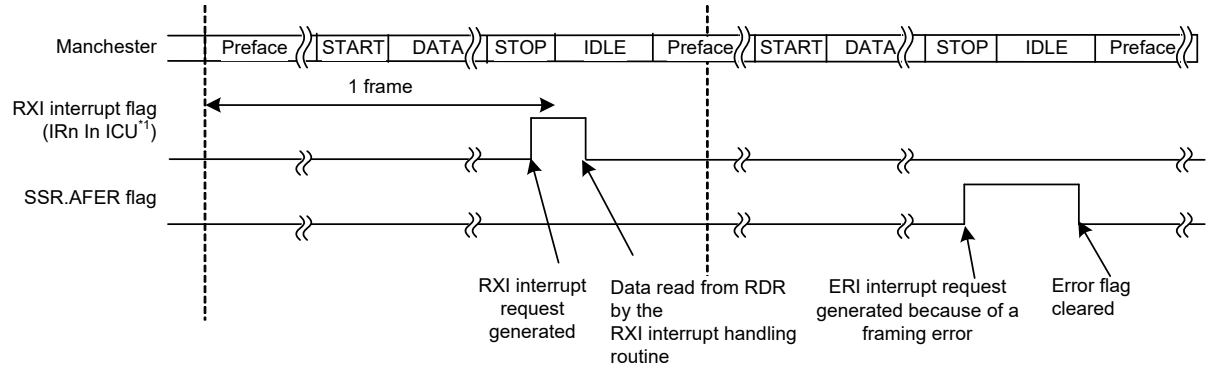
Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

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 32.5.11, 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 32.38 shows an example of the operation for serial data reception in Manchester mode.

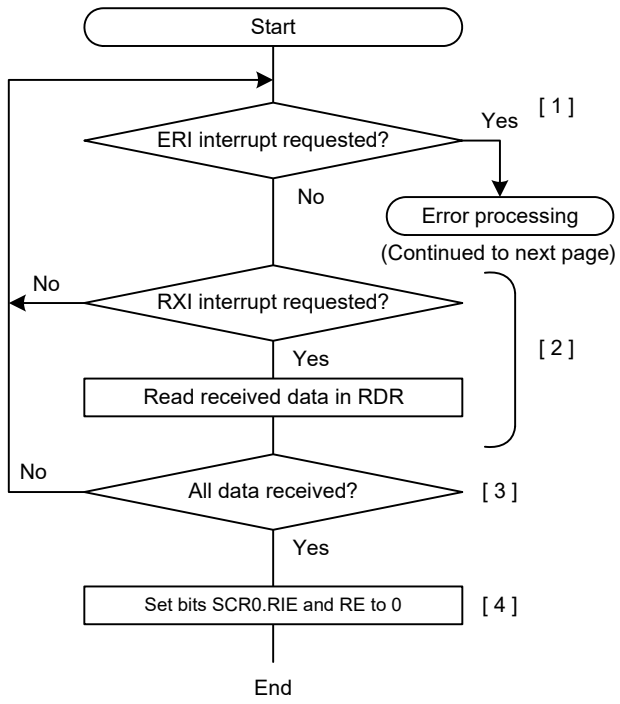


Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.38 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 32.5.11, 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 32.39 and Figure 32.40 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 32.39 Example of Serial Data Reception in Manchester Mode (Normal)

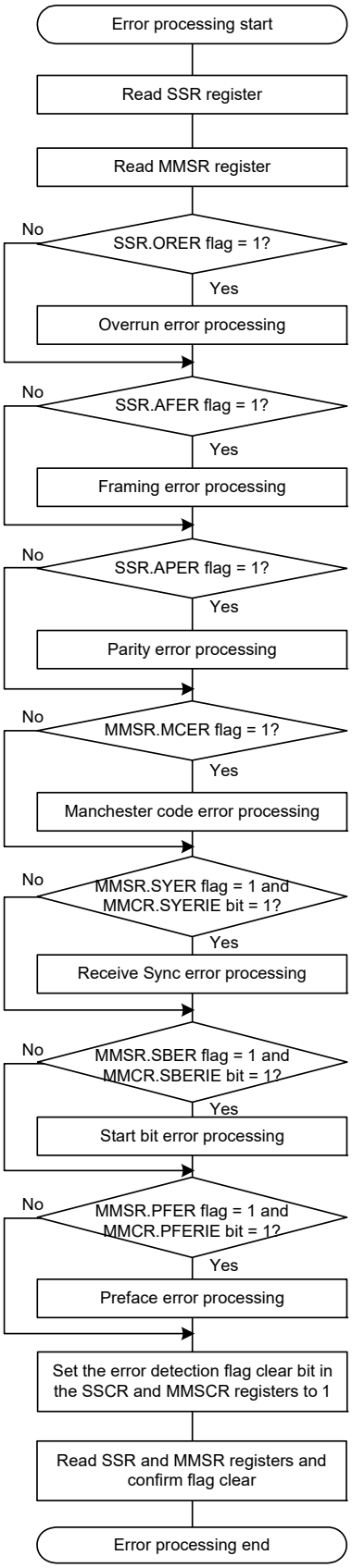


Figure 32.40 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

32.5.8 Operation When Multi-Processor Bit is Used

See section 32.4, 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 32.40 for error processing in Manchester mode for the reception flowchart (Figure 32.25). Refer to Table 32.32 for the operation status when detecting various errors.

32.5.9 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 32.41 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.

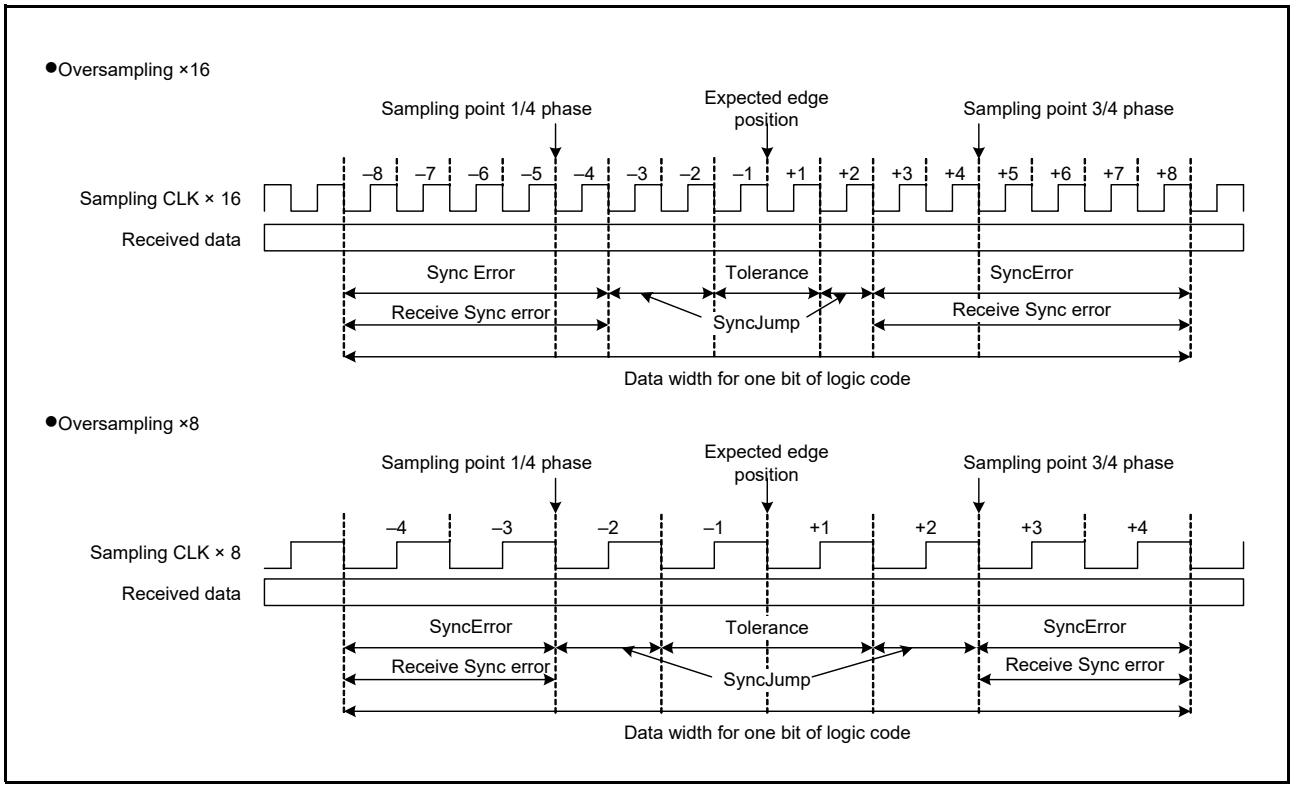


Figure 32.41 Conceptual Image of Reception Retiming Range

32.5.10 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 32.42 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 32.5.1, Frame Format (2).

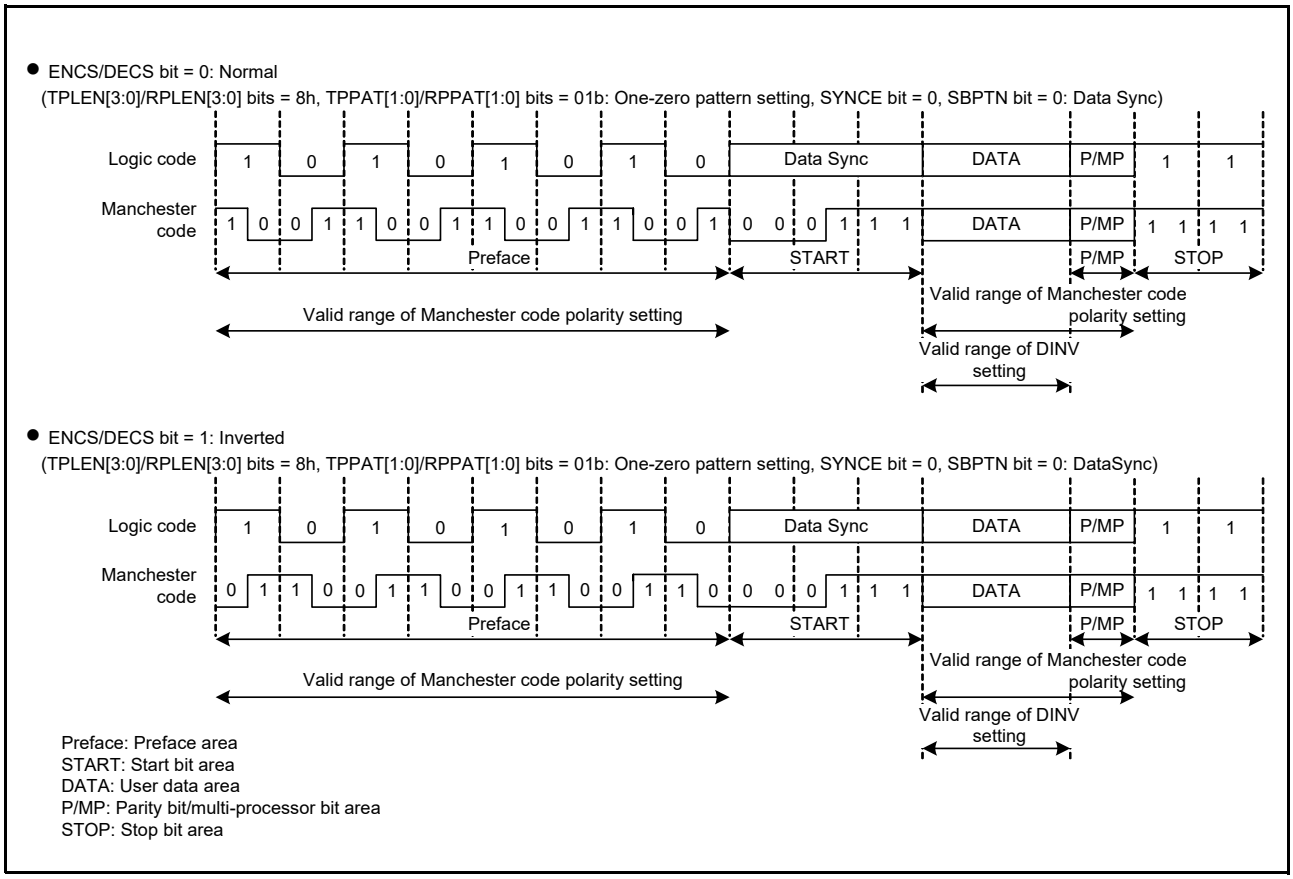


Figure 32.42 Valid Range of the Manchester Code Polarity Setting

32.5.11 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 32.3.9, 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 32.30 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR register.
Table 32.31 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 32.32 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 32.5.9, Receive Retiming is enabled, the receive retiming operation is performed.
If no edges are detected within the receive retiming range (SyncError area in Figure 32.41) 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 32.30 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 32.31 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 32.5.11, Errors in Manchester Mode (7)

Table 32.32 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 32.33 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.

32.6 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.

32.6.1 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.

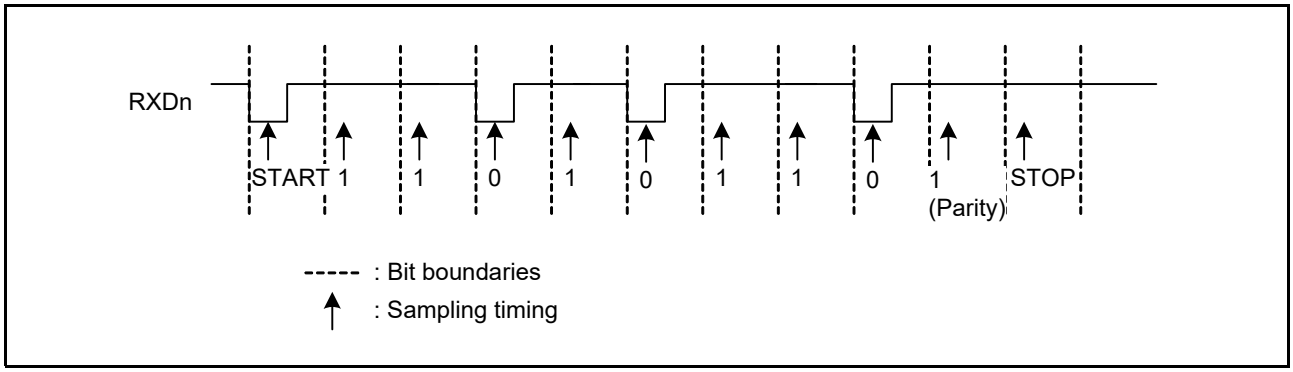


Figure 32.43 HBS Support Mode Reception Timing Chart

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 32.3.2, 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.

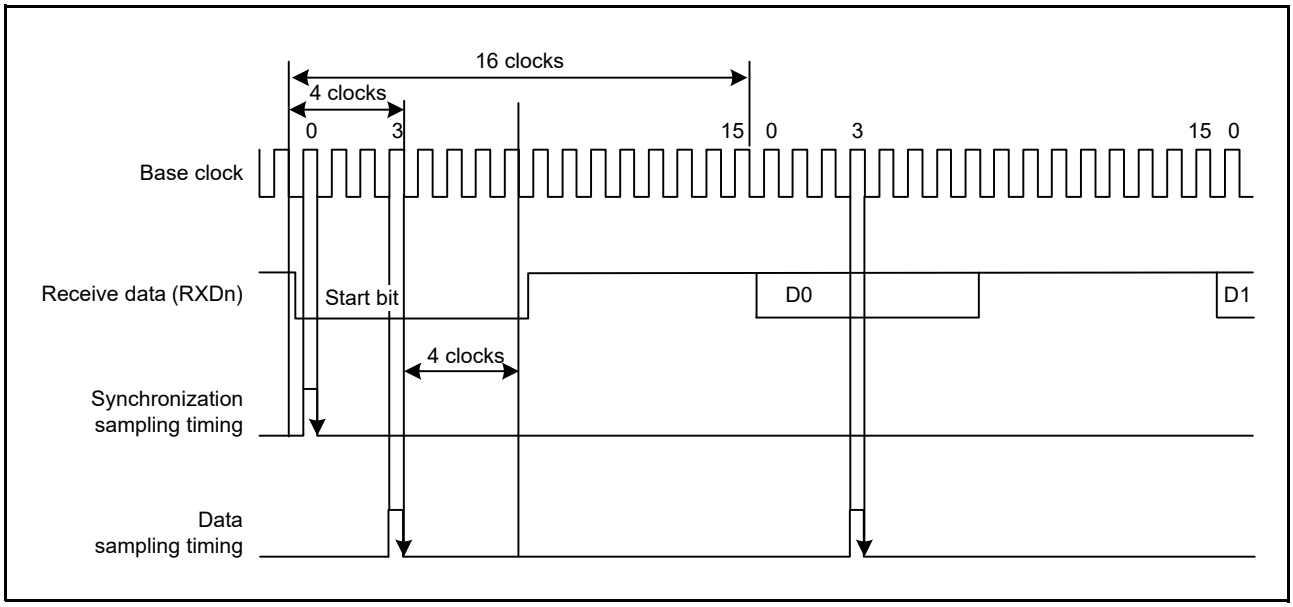


Figure 32.44 Details of Reception Sampling Timing in HBS Support Mode

32.6.2 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.

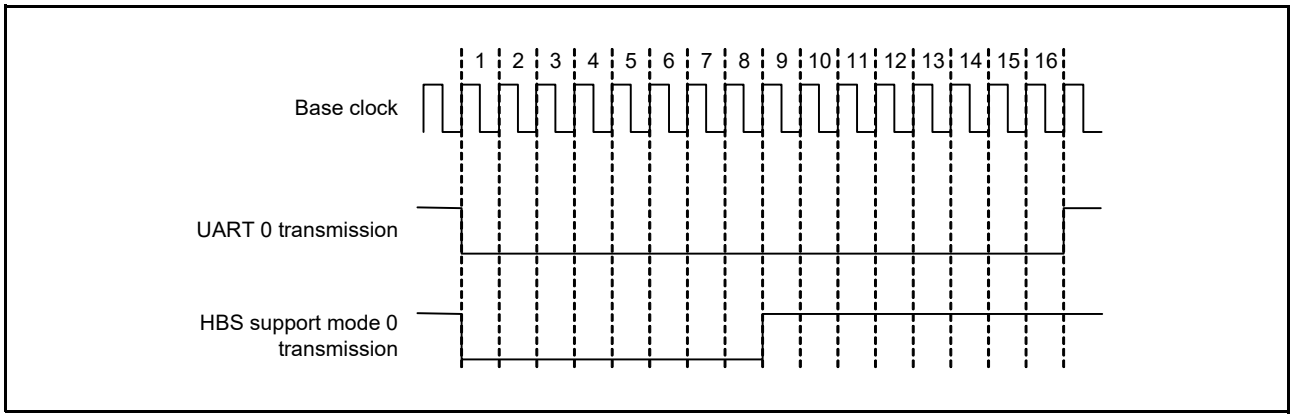


Figure 32.45 Transmission Waveform of HBS Support Mode

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.

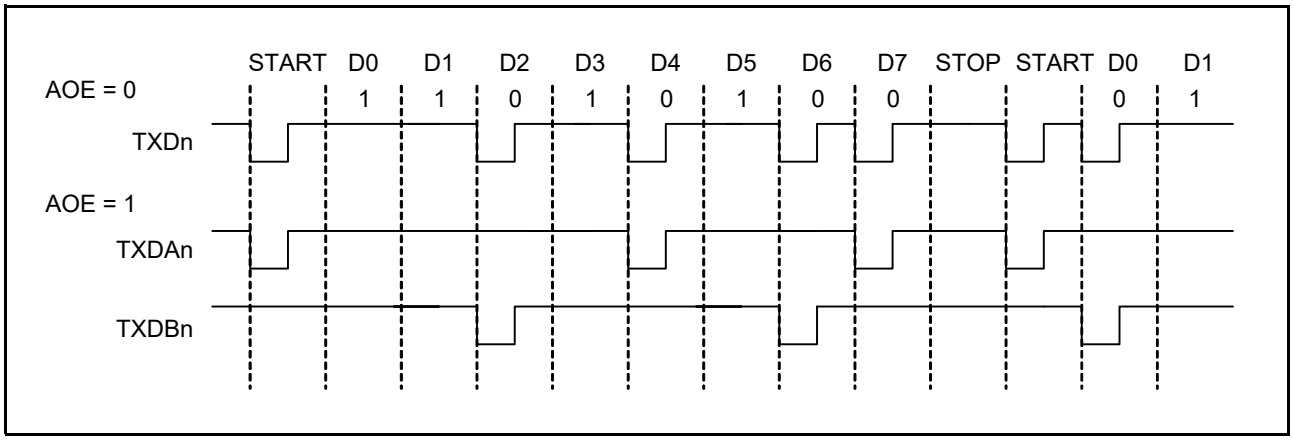


Figure 32.46 Difference in Transmission Waveform Depending on AOE Bit (When LPS Bit = 0)

Figure 32.46 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.

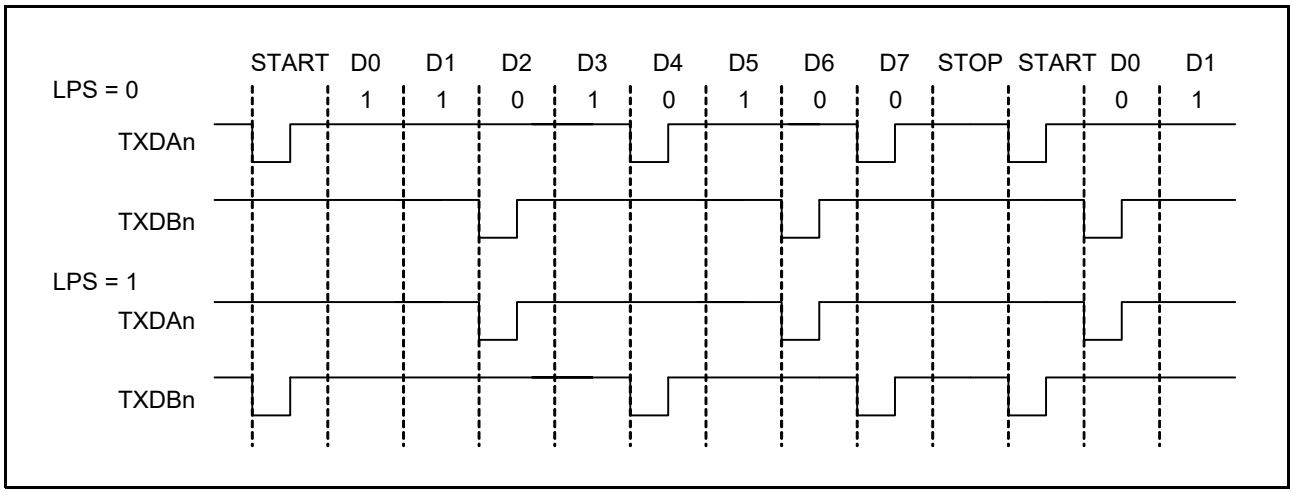


Figure 32.47 Difference in Transmission Waveform Depending on LPS Bit (When AOE Bit = 1)

Figure 32.47 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.

32.6.3 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 32.34 before use. Register bits not described can be set in the same way as the asynchronous mode.

Table 32.34 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.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.

32.7 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.

32.7.1 Sample Connection

Figure 32.48 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.

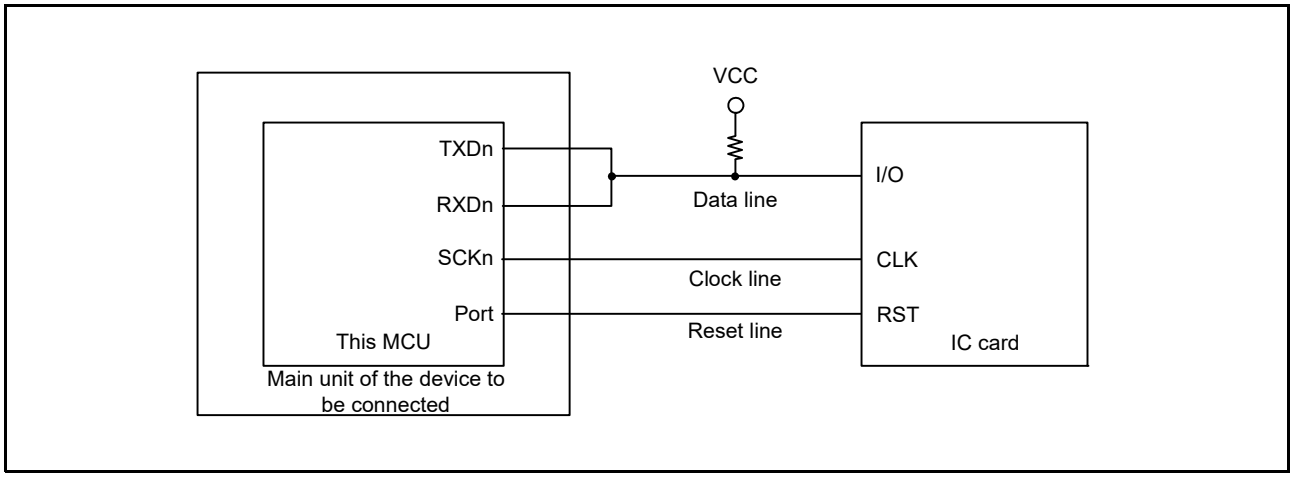


Figure 32.48 Sample Connection with a Smart Card (IC Card)

32.7.2 Data Format (Except in Block Transfer Mode)

Figure 32.49 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.

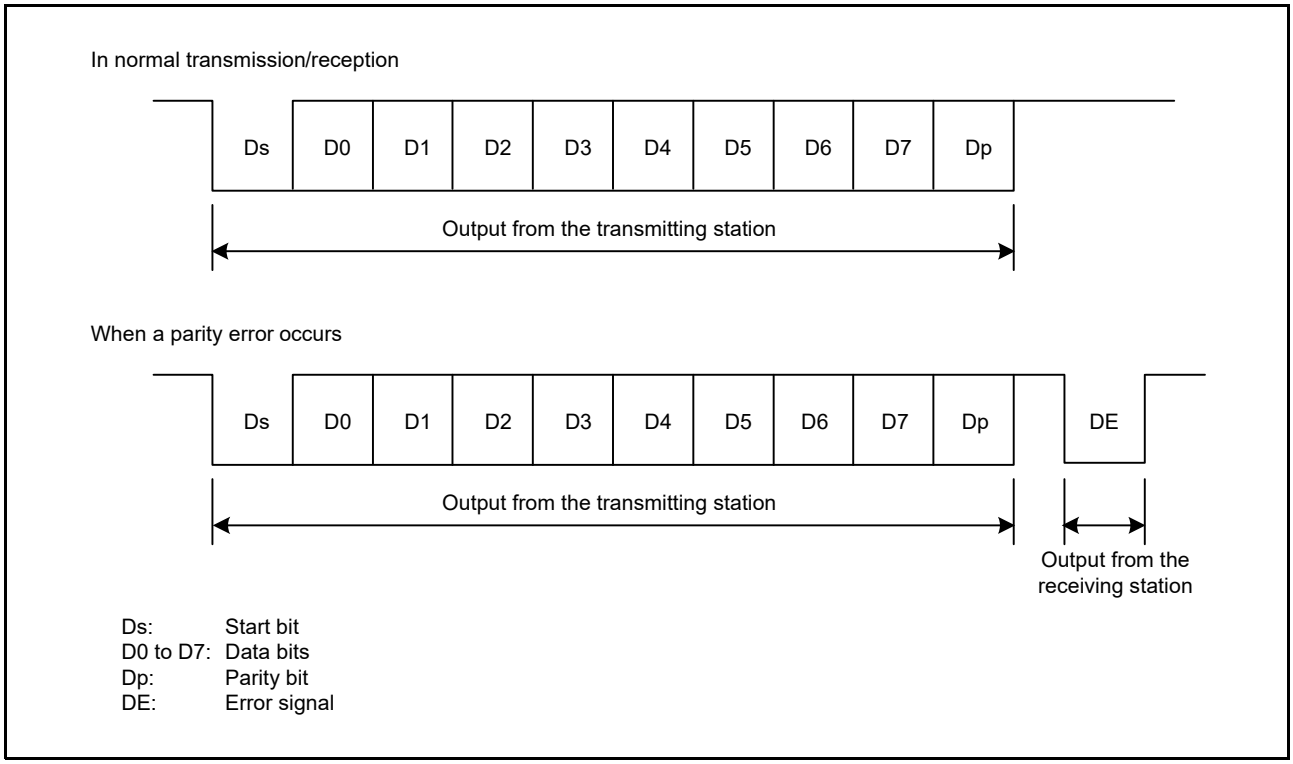


Figure 32.49 Data Formats in Smart Card Interface Mode

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 32.50. 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.

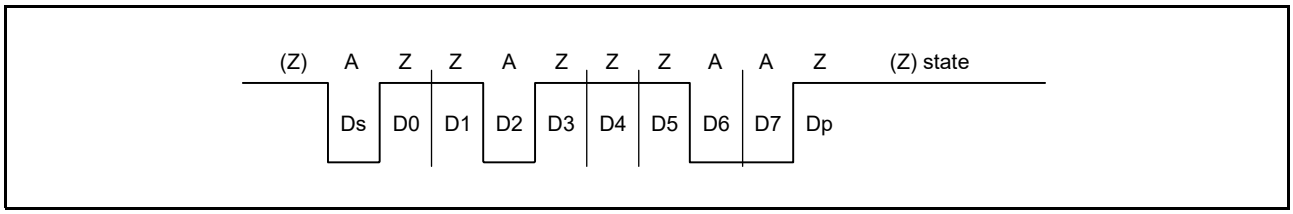


Figure 32.50 Direct Convention (SCR3.DDIR Bit = 1, SCR3.DINV Bit = 0, SCR1.PM Bit = 0)

(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 32.51. 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.

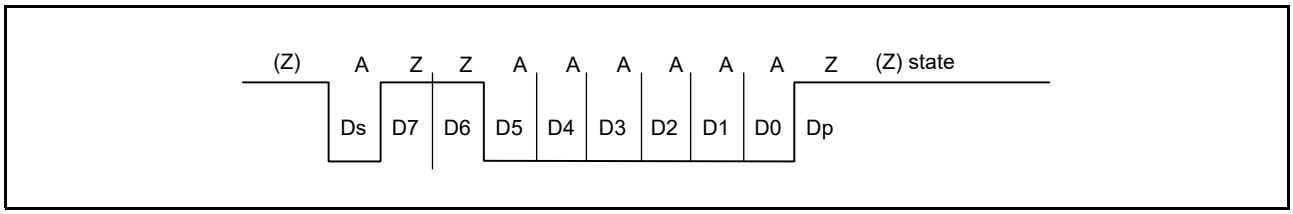


Figure 32.51 Inverse Convention (SCR3.DDIR Bit = 0, SCR3.DINV Bit = 1, SCR1.PM Bit = 1)

32.7.3 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.

32.7.4 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 32.52. 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 (%)

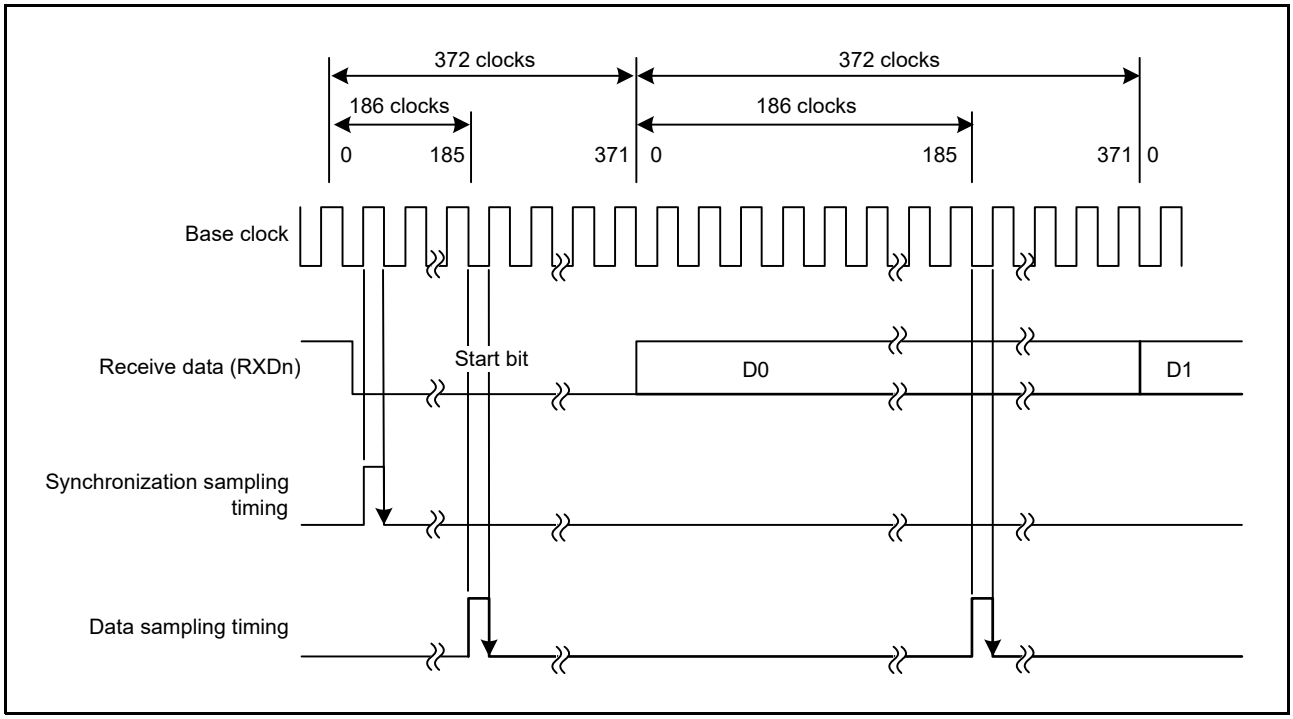


Figure 32.52 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

32.7.5 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 32.53.
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.

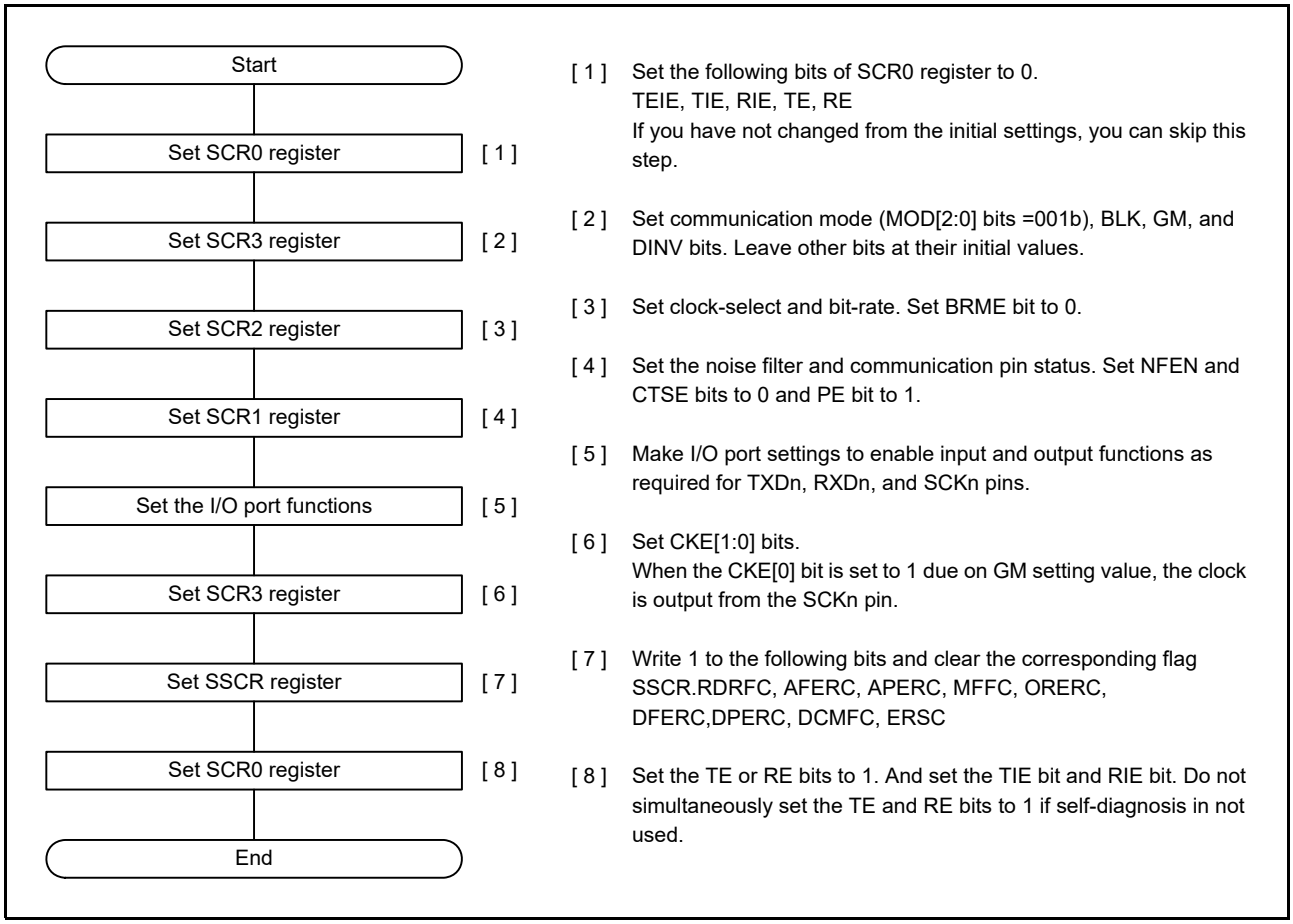


Figure 32.53 Example of RSCI Initialization Flowchart (Smart Card Interface Mode)

Figure 32.54 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.

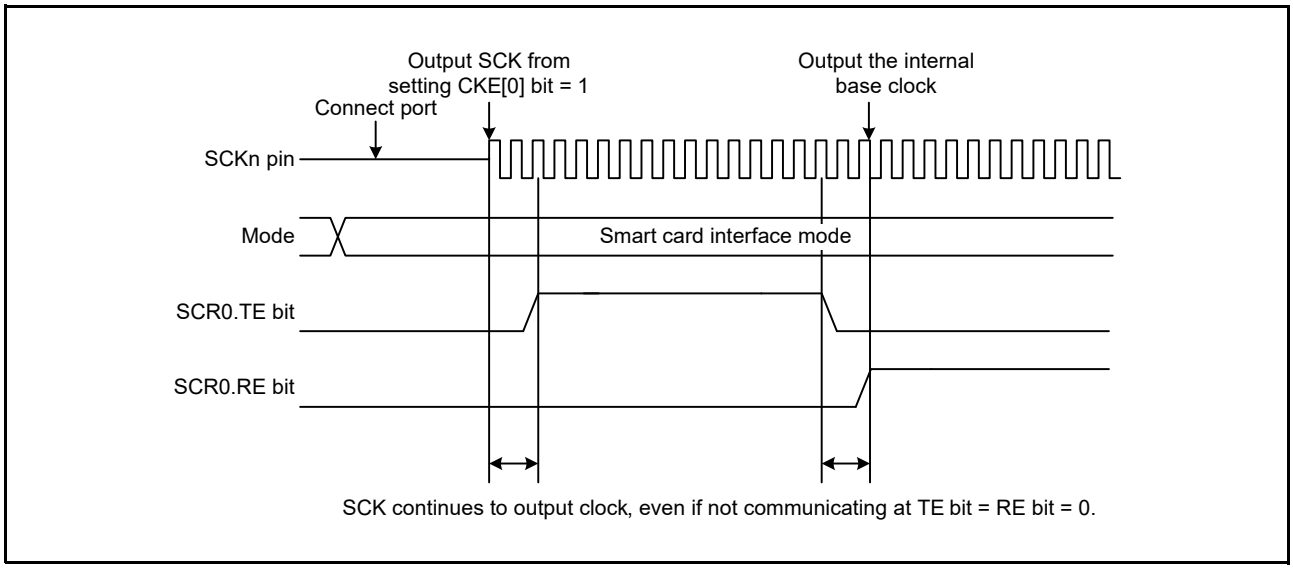
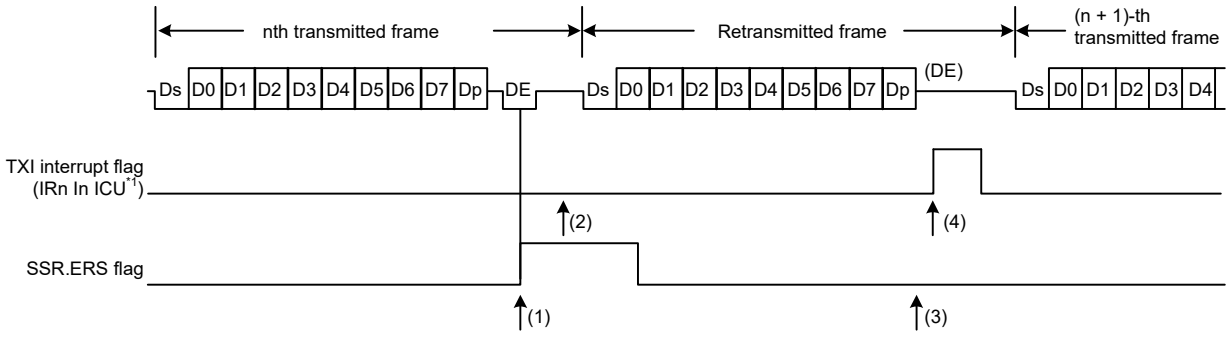


Figure 32.54 Example of Data Transmission Timing in Smart Card Interface Mode

32.7.6 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 32.55 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.55 Data Retransmit Operation in RSCI Transmit Mode

Figure 32.57 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 (DMACA) 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 32.56 shows the timing for setting the TEND flag.

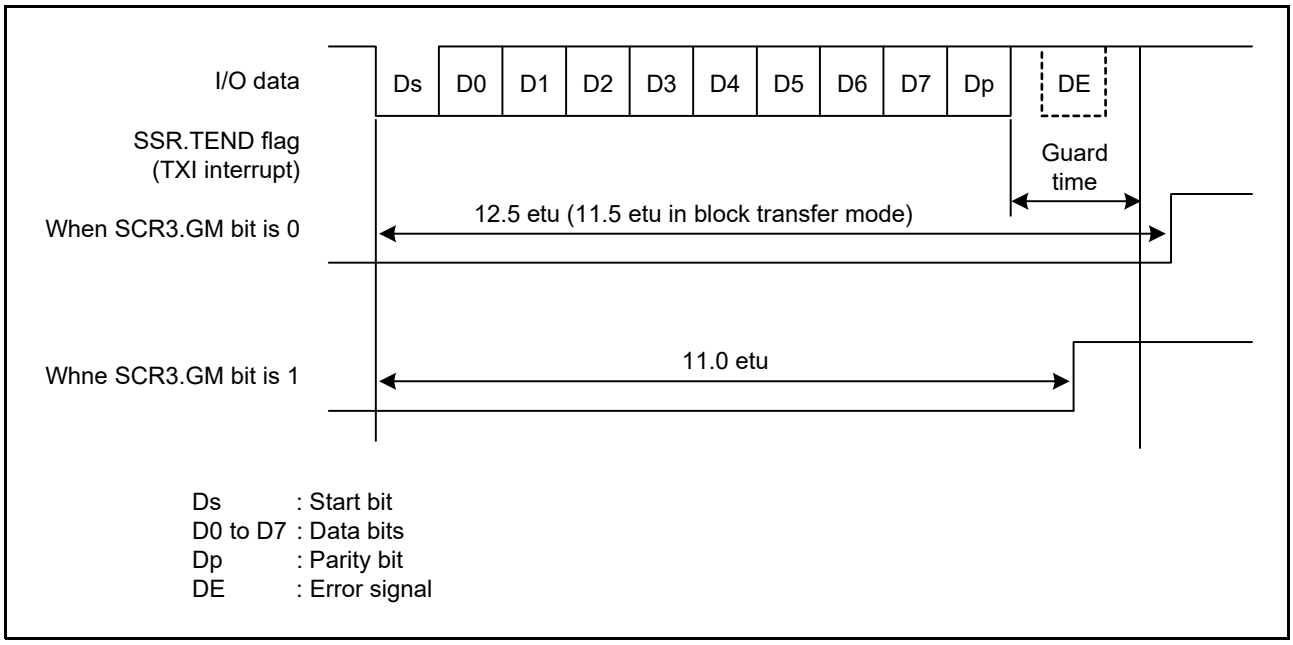


Figure 32.56 Timing for Setting the SSR.TEND Flag during Transmission

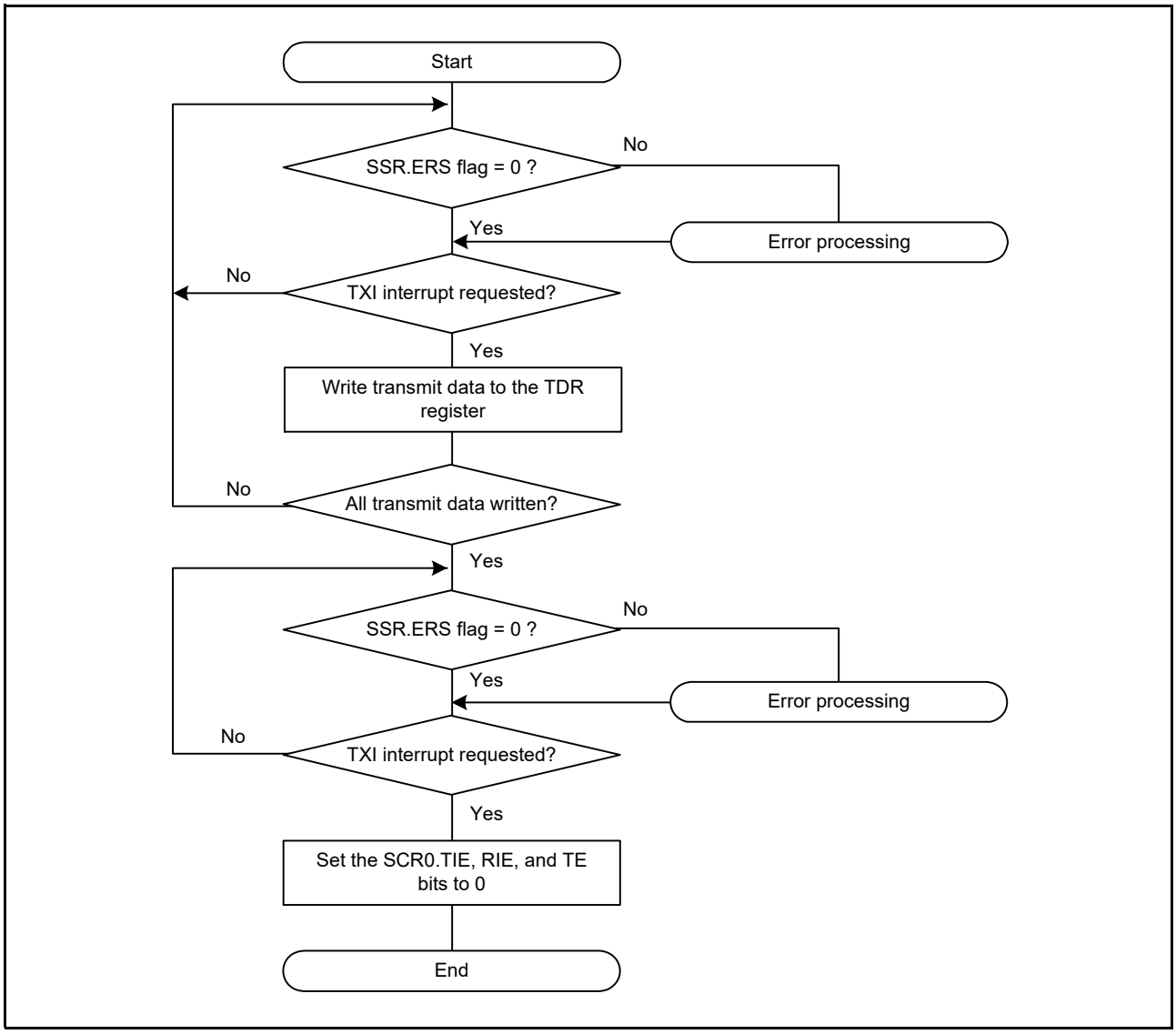


Figure 32.57 Sample Smart Card Interface Transmission Flowchart

32.7.7 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 32.58 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.

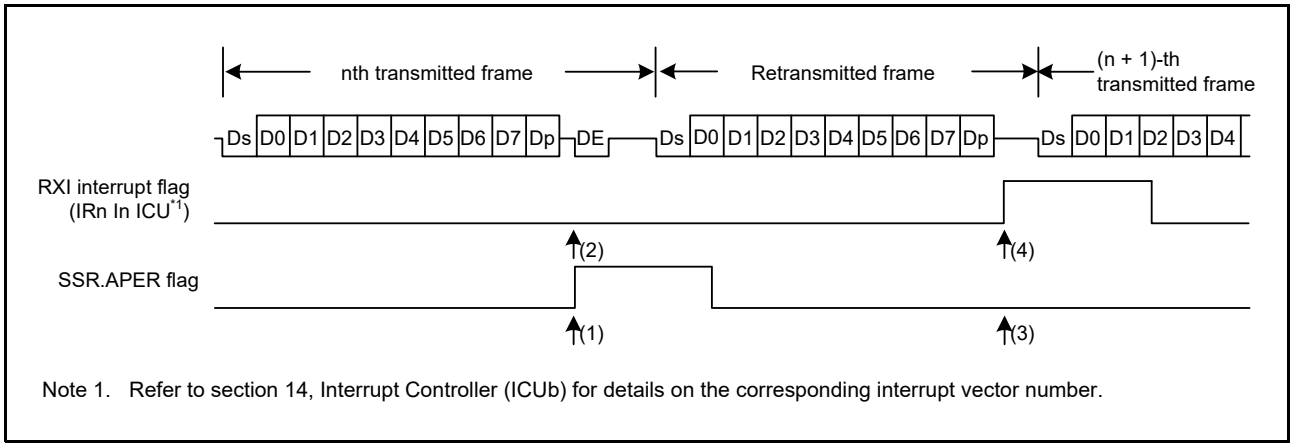


Figure 32.58 Data Retransmit Operation in RSCI Receive Mode (Data Retransmit Operation during Reception)

Figure 32.59 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 32.3, Operation in Asynchronous Mode.

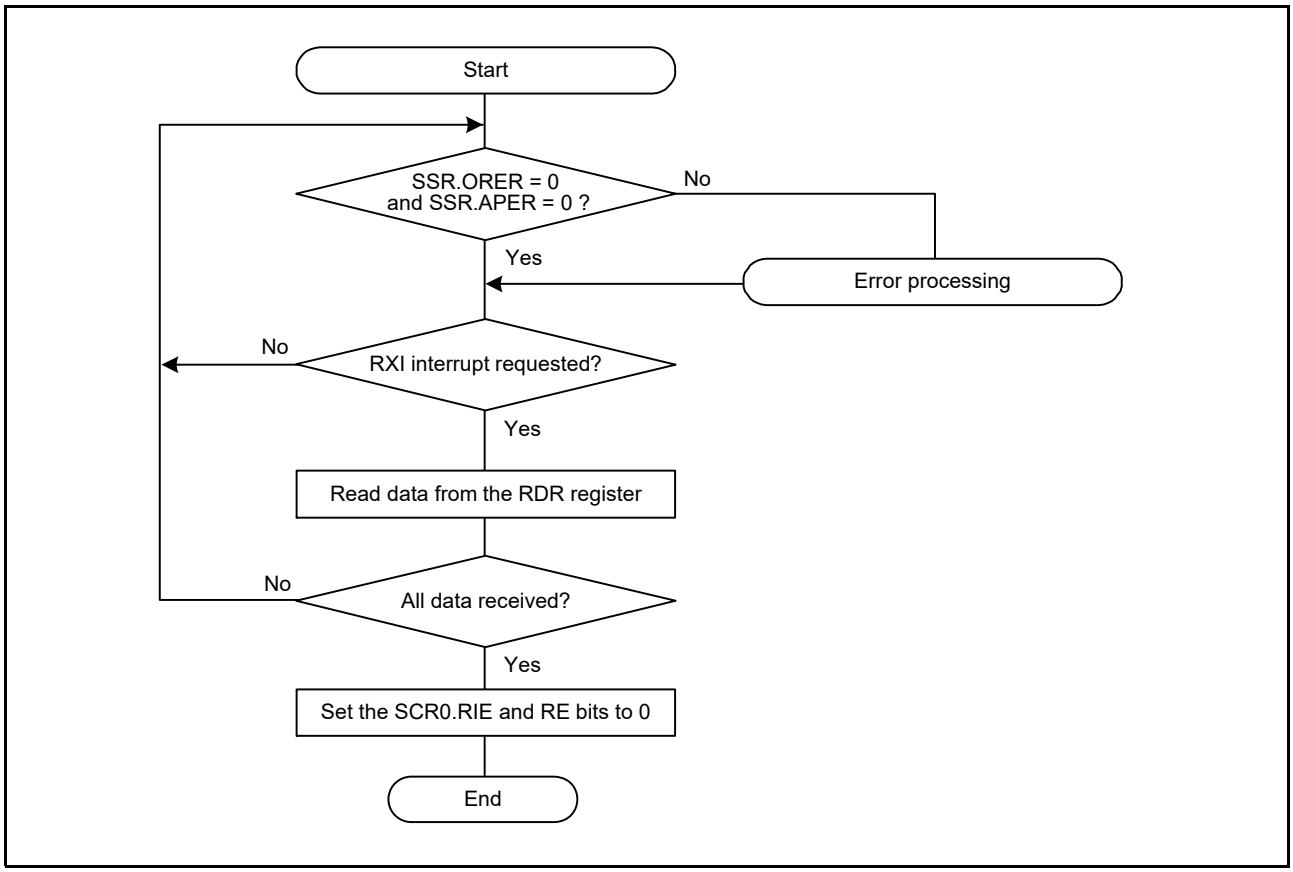


Figure 32.59 Sample Smart Card Interface Reception Flowchart

32.7.8 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 32.2.8, Control Register 3 (SCR3). When setting the clock output, the base clock described in section 32.7.4, 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 32.2.7, 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 32.60 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.

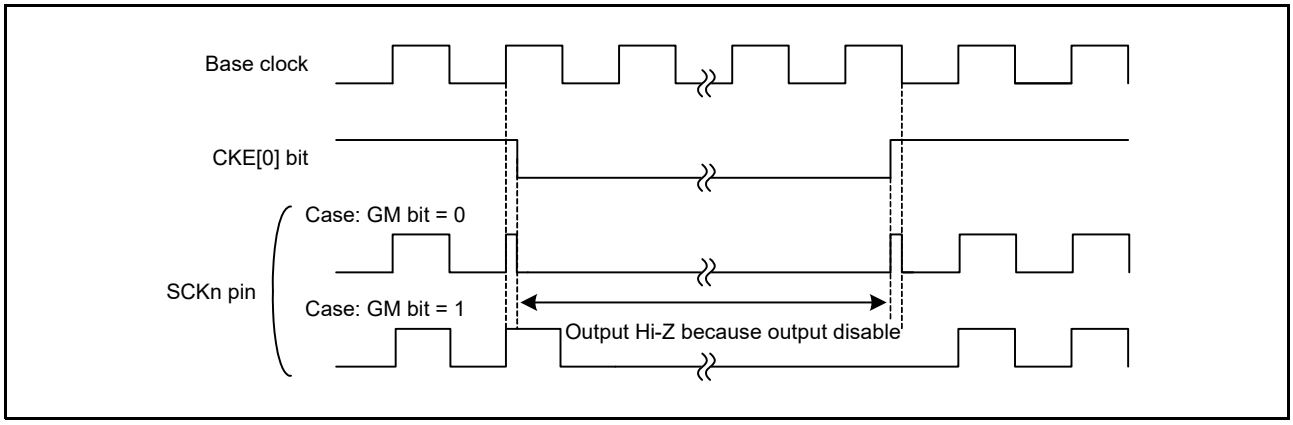


Figure 32.60 Clock Control Timing Chart by SCR3.GM Bit

32.8 Extended Serial Mode

32.8.1 Serial Transfer Protocol

As an extended function of the RSCI, the RSCI supports the serial communication protocol (Figure 32.61) 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.

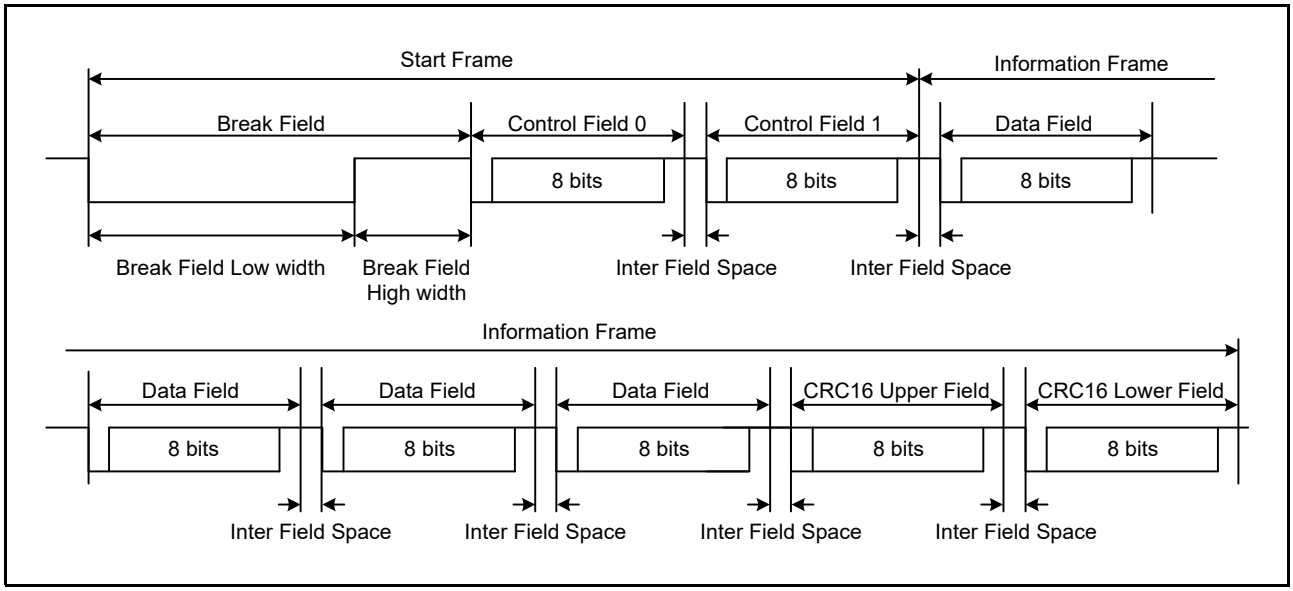


Figure 32.61 Protocol for Serial Transfer by the Extended Serial Mode

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.

32.8.2 Transmitting a Start Frame

Figure 32.62 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 32.63 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 32.9) 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)

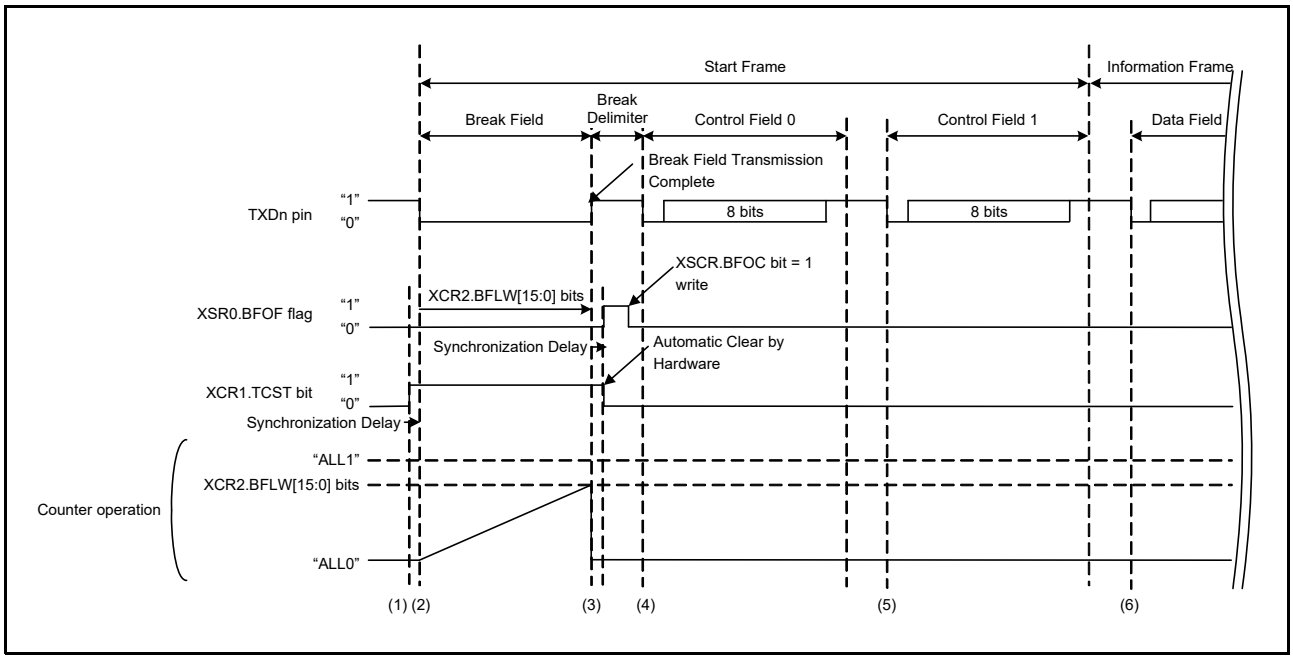


Figure 32.62 Example of Operations When Transmitting a Start Frame

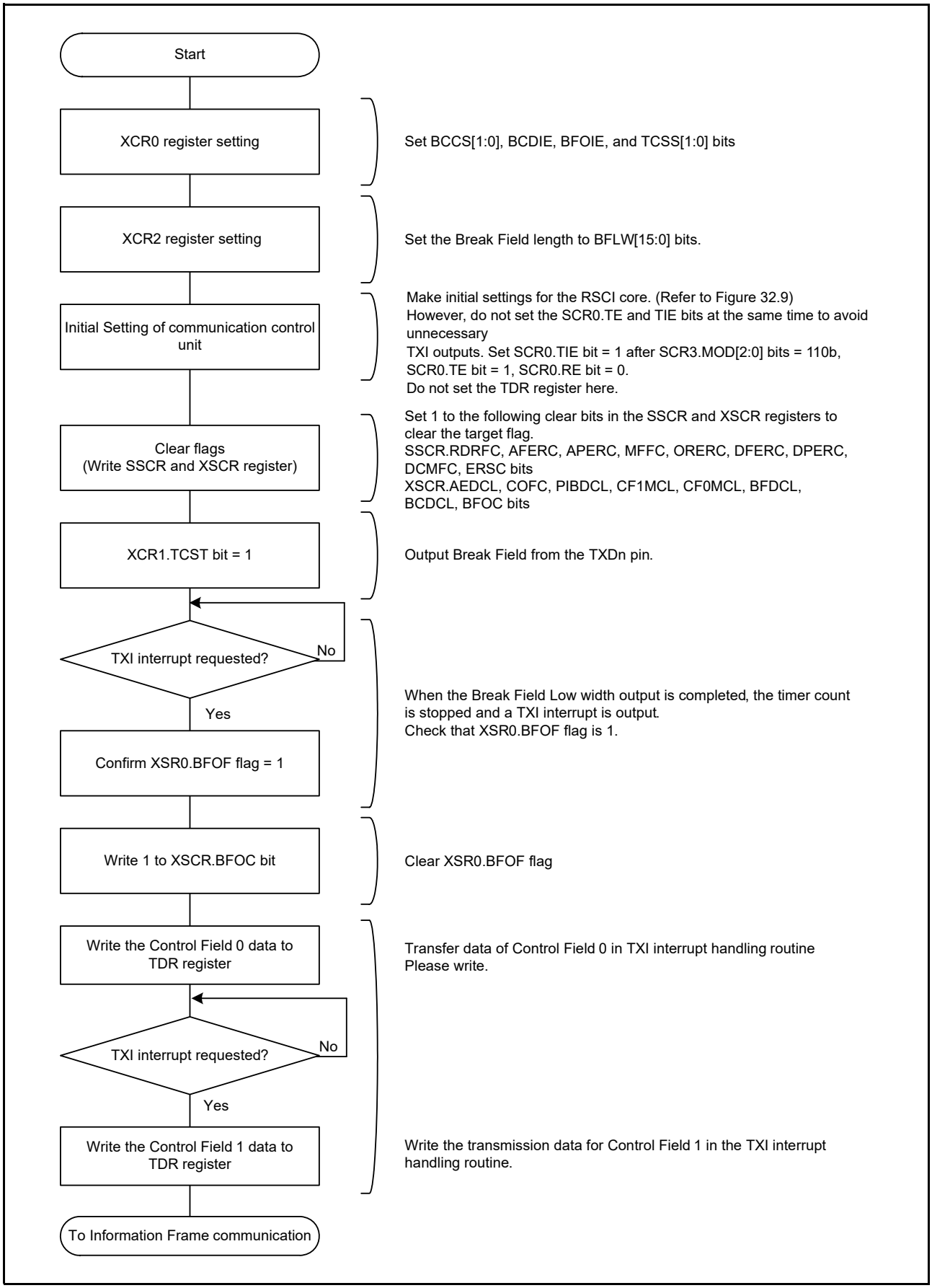


Figure 32.63 Example of Start Frame Transmission

32.8.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 32.35.

Table 32.35 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

32.8.3.1 Extended Serial Normal Reception of Start Frame (PIB not Used)

Figure 32.64 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 32.65 shows an example of reception to detect the Break Field during Control Field 1. Figure 32.66 shows a flowchart to receive the Start Frame, and Figure 32.67 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 32.65).
(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.

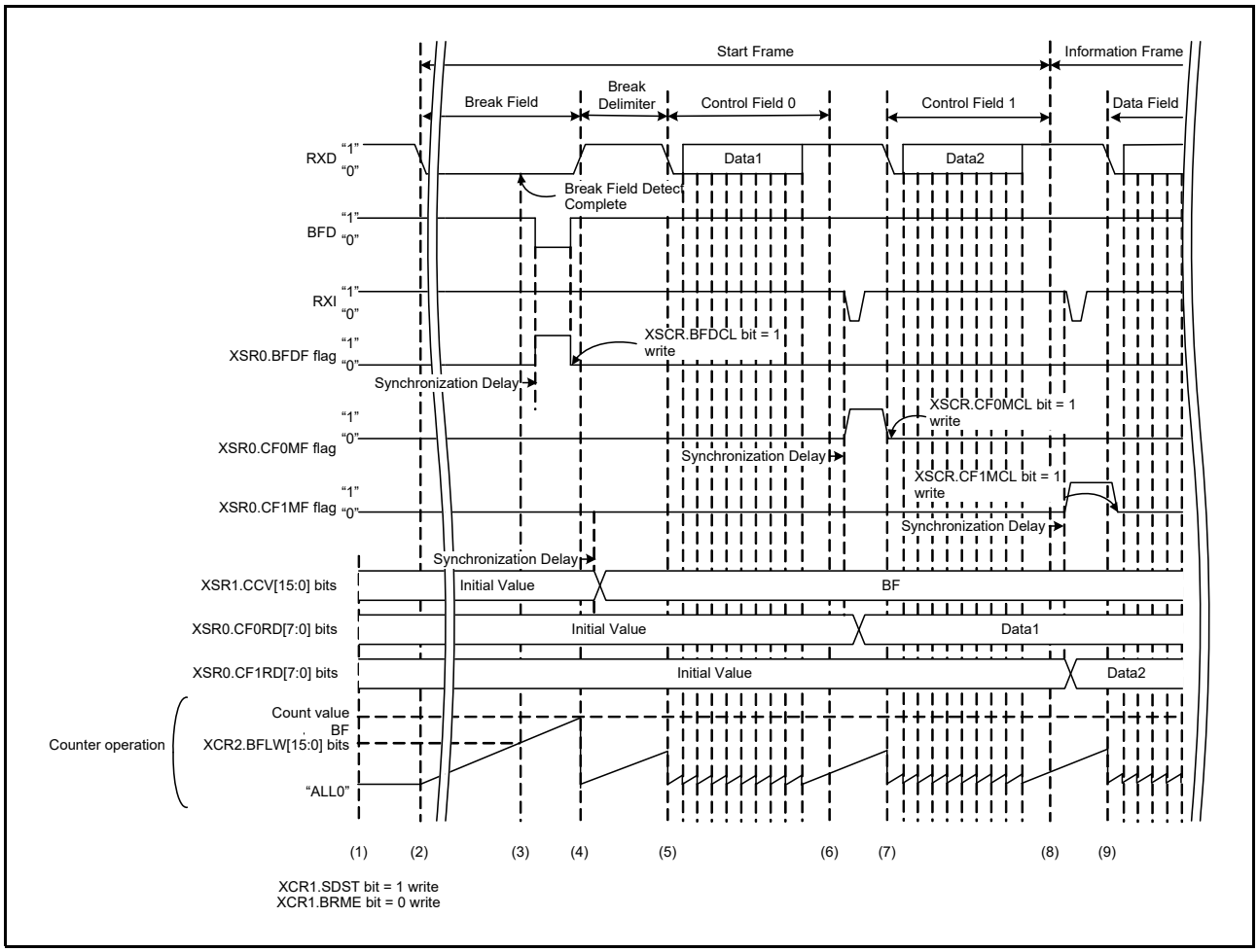


Figure 32.64 Normal Reception Example of Start Frame (PIB Not Used)

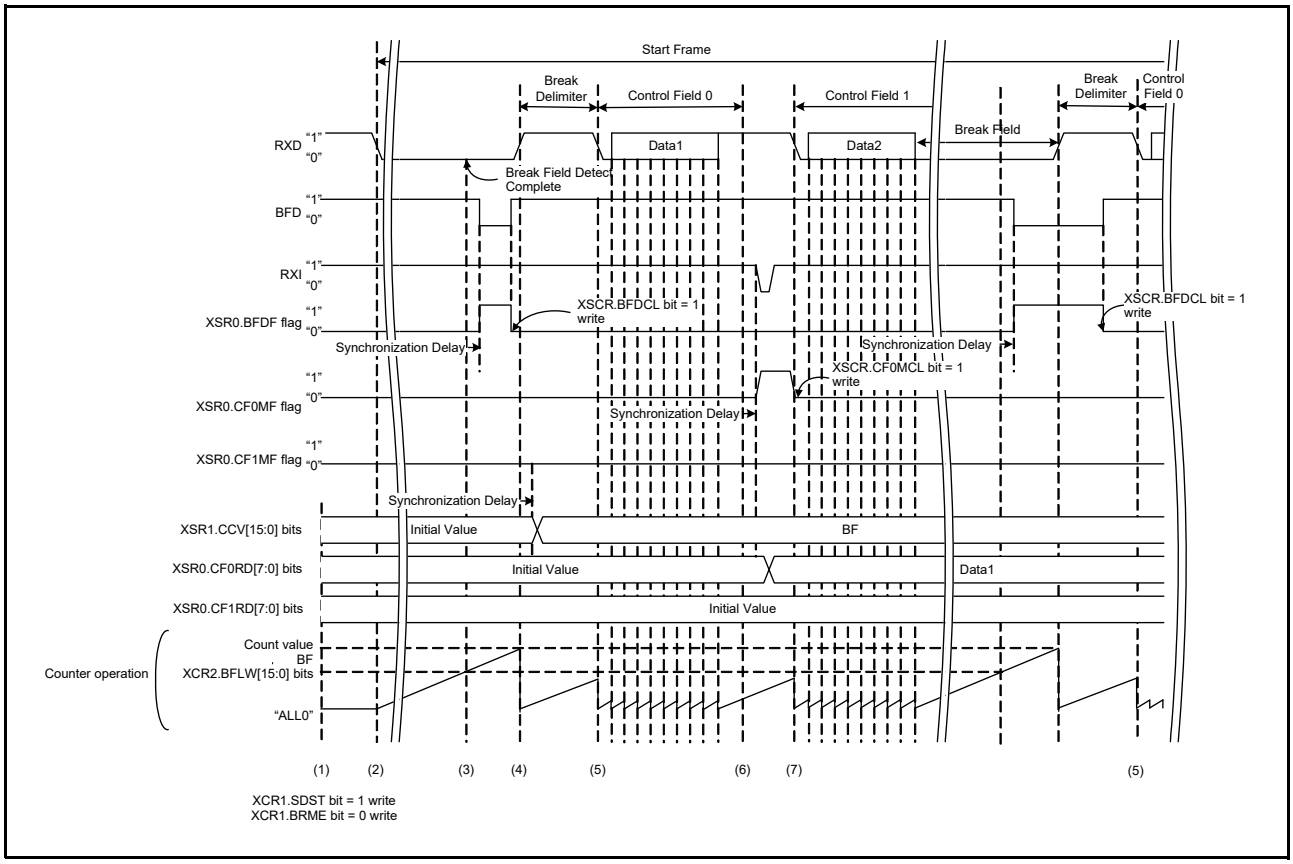


Figure 32.65 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

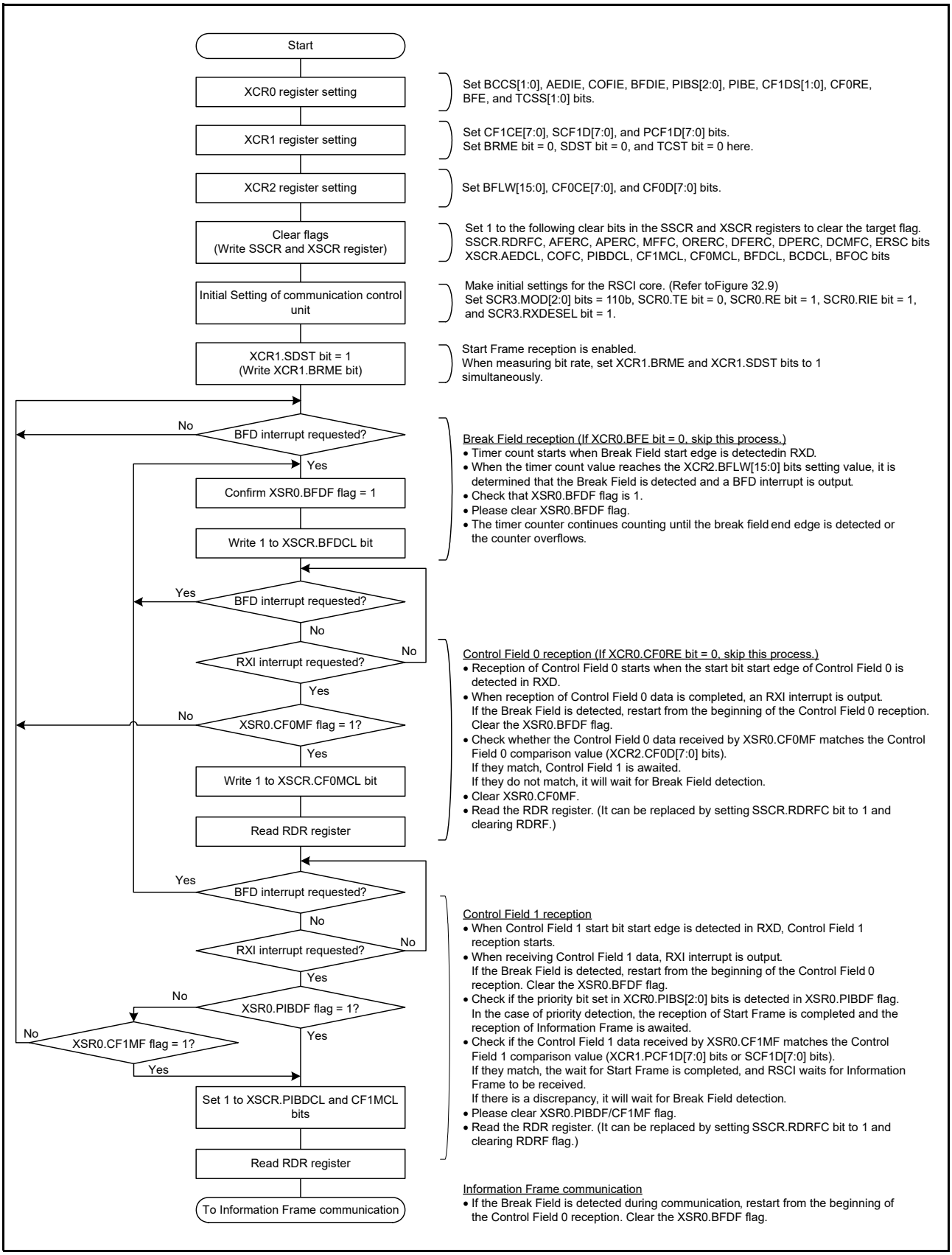


Figure 32.66 Sample Flowchart for Reception of a Start Frame

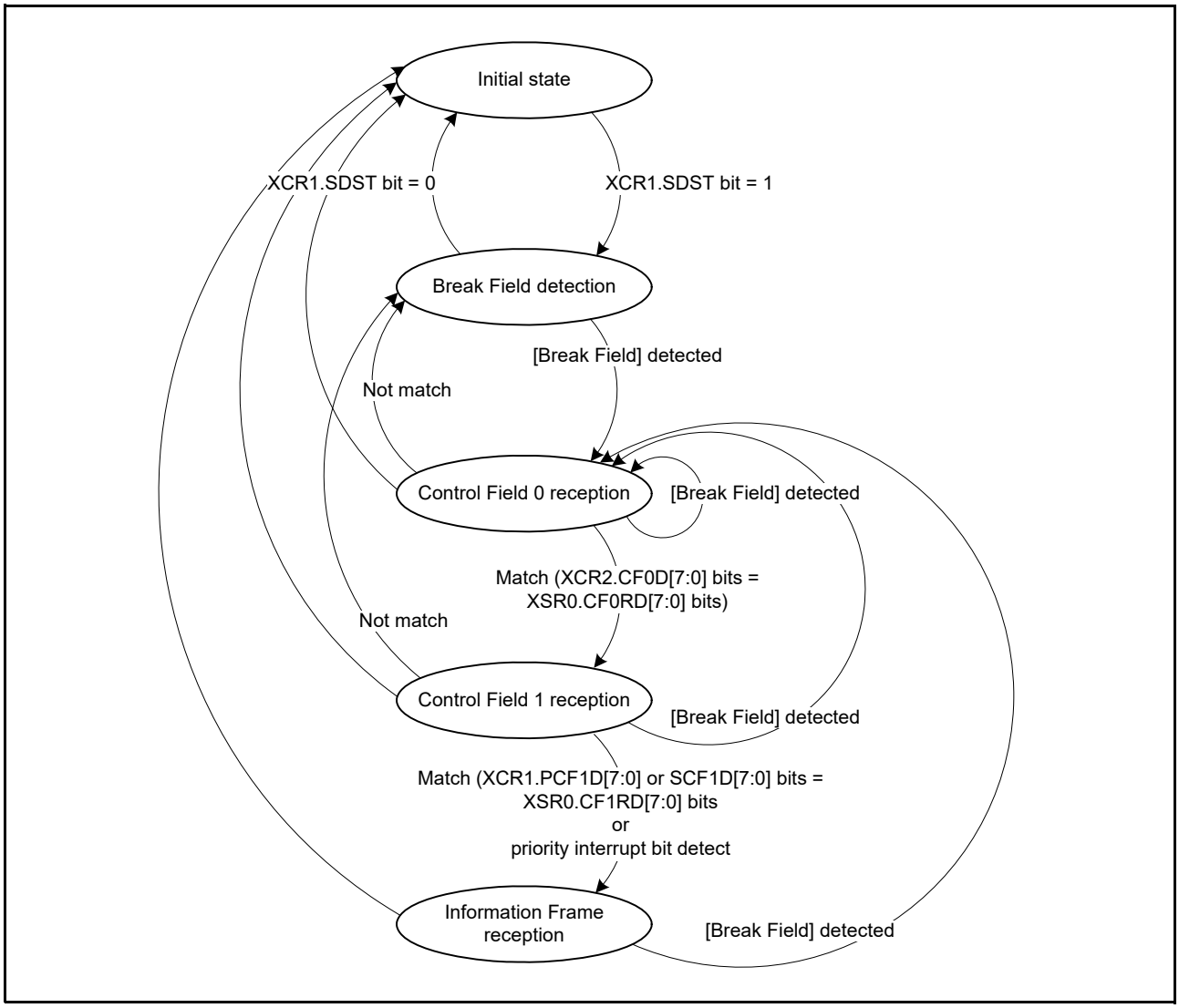


Figure 32.67 State Transition Diagram of Start Frame Reception

32.8.3.2 Priority Interrupt Bit

Figure 32.68 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 32.64, 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.

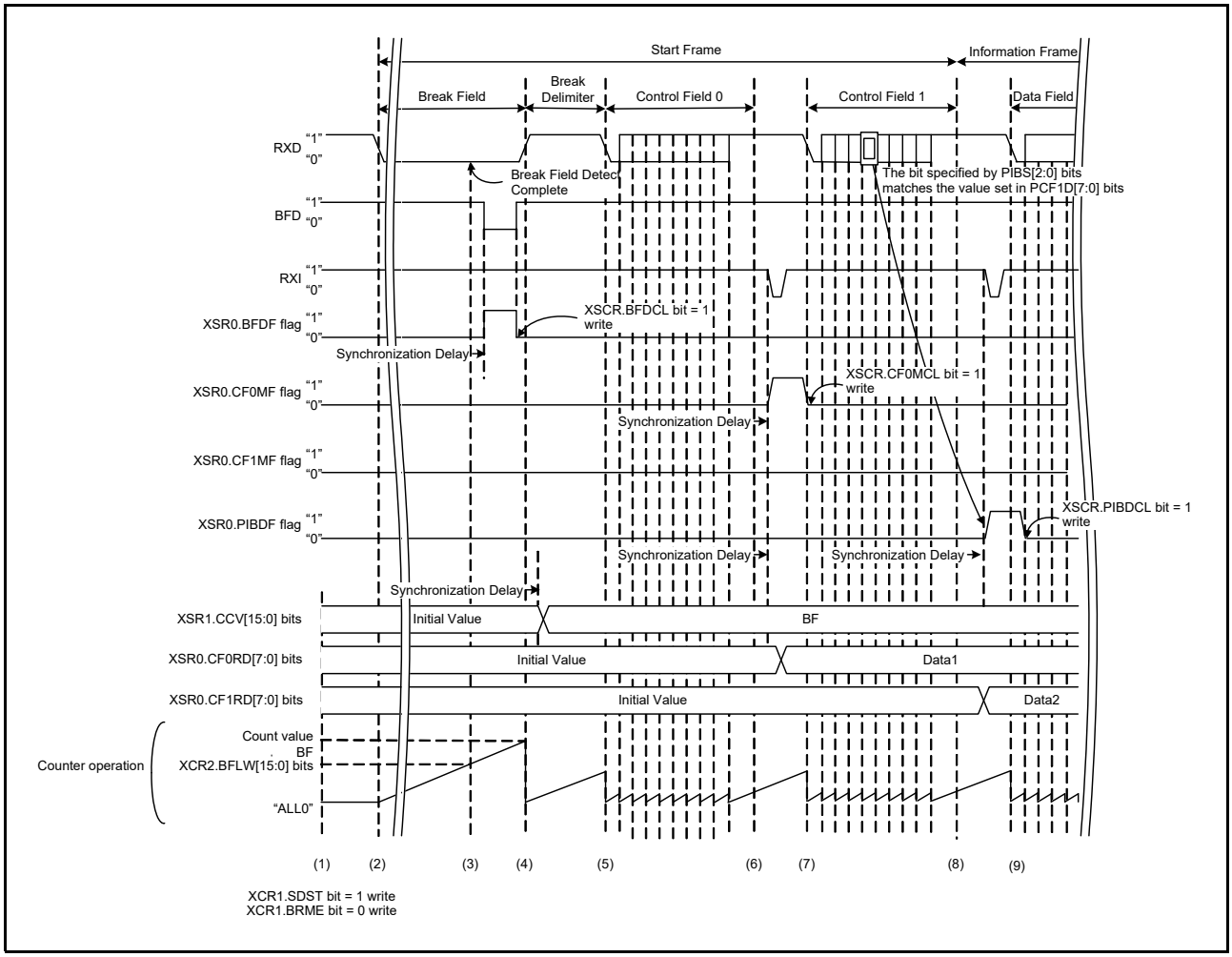


Figure 32.68 Start Frame Reception Example (Priority Interrupt Bit Used)

32.8.4 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 32.69 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 32.70. Check the bus state to decide whether to resume transmission.

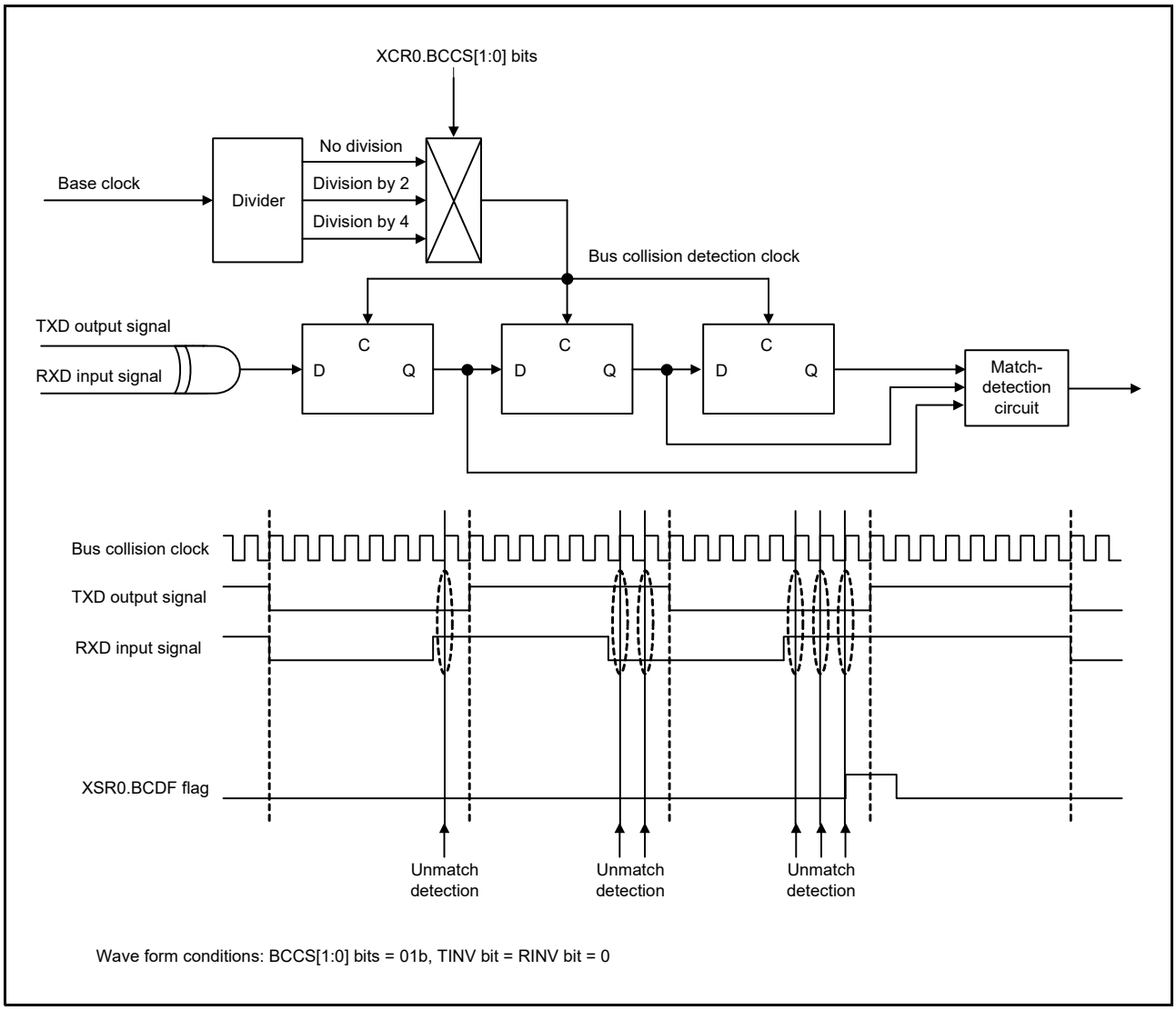


Figure 32.69 Example of Operations with Bus Collision Detection

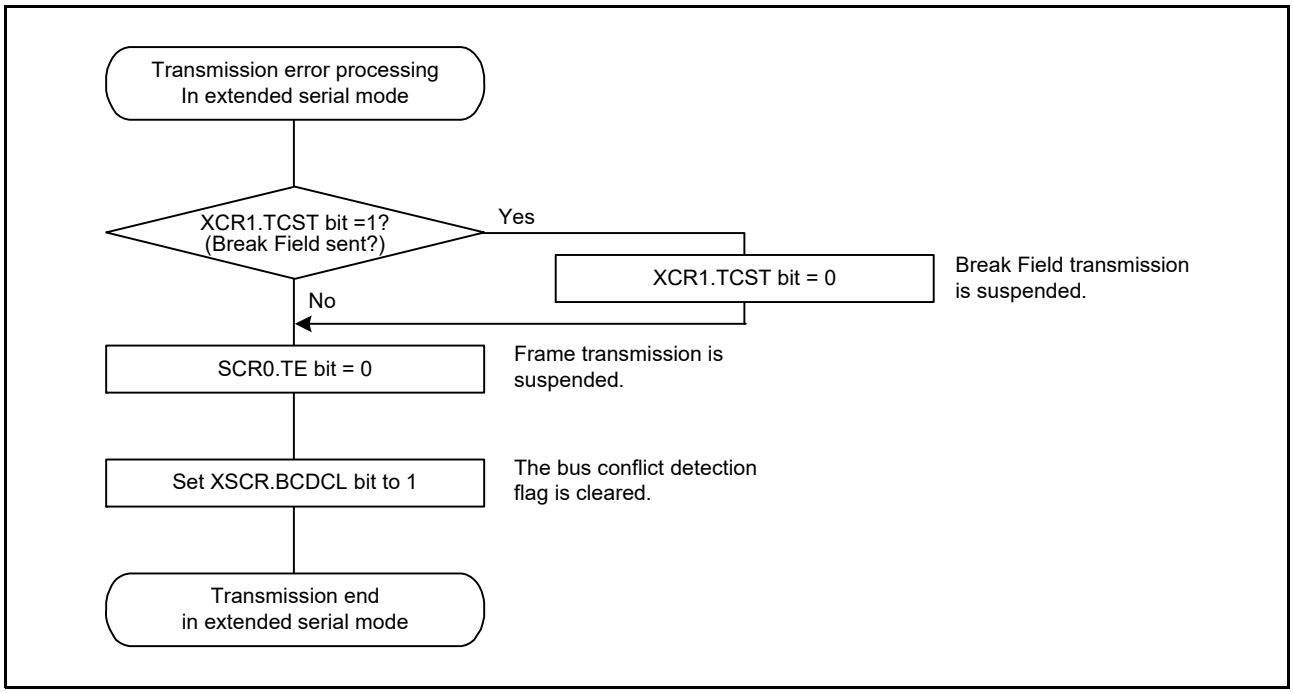


Figure 32.70 ERI Interrupt Handling Flow at Transmission in Extended Serial Mode

32.8.5 Bit Rate Measurement

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 32.71 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 XSR1.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.

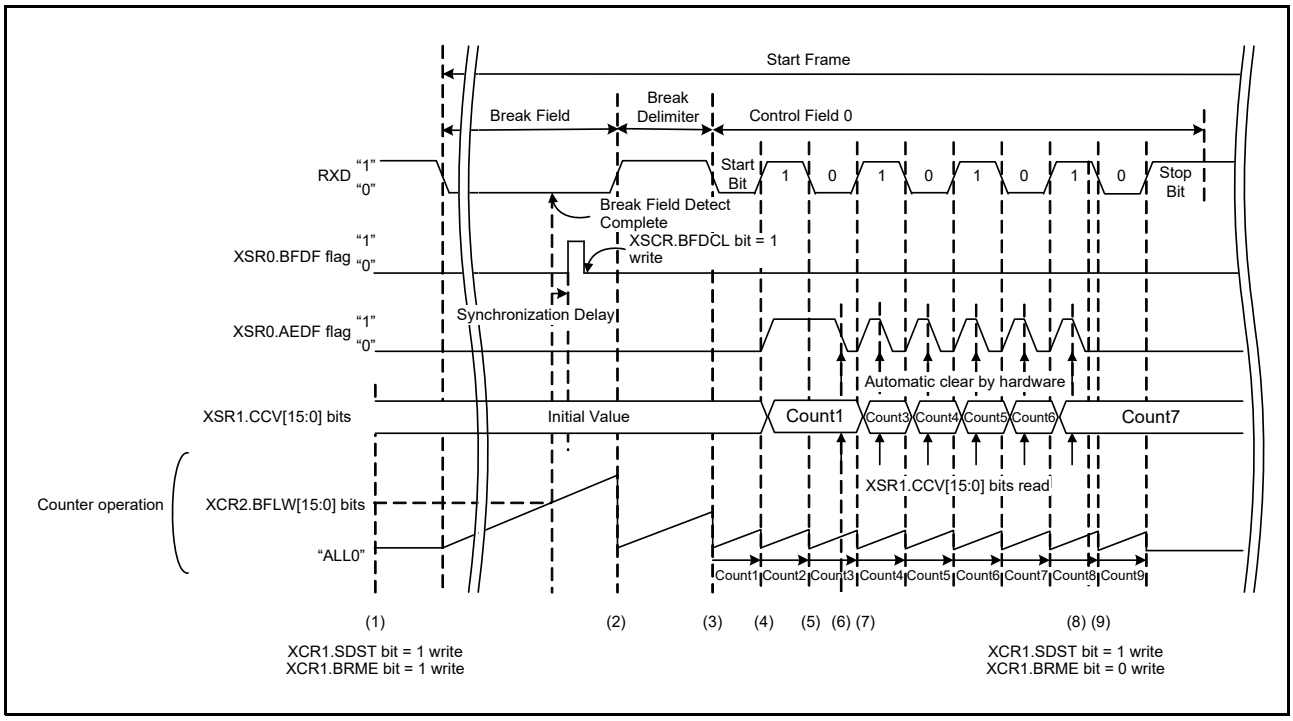


Figure 32.71 Operation Example of the Bit Rate Measurement Function

32.9 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 32.72 shows I2C bus format, and Figure 32.73 shows the timing of I2C.

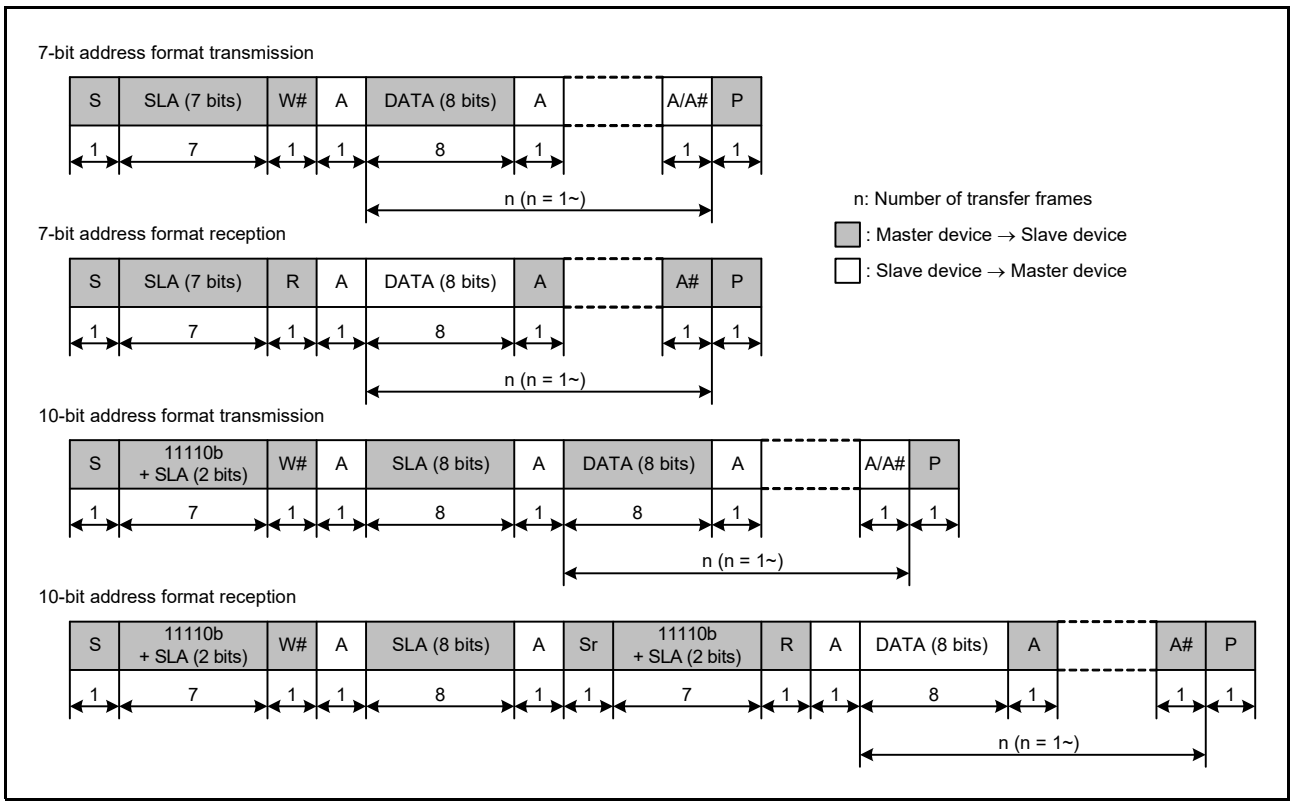


Figure 32.72 I2C-Bus Format

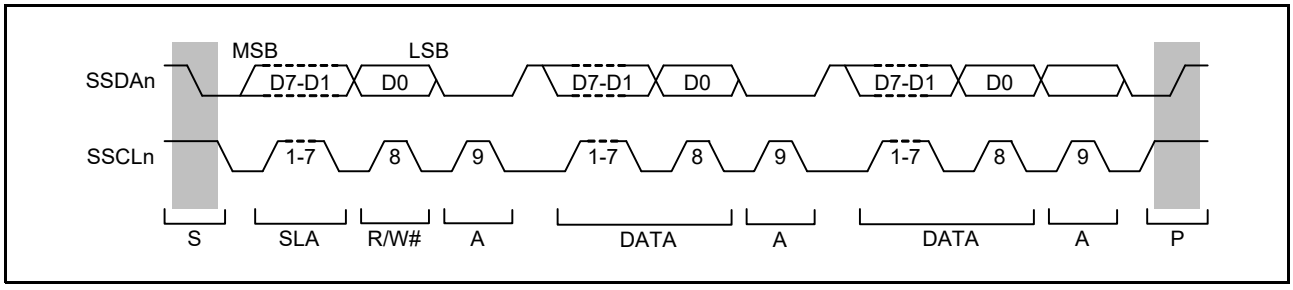


Figure 32.73 I2C-Bus Timing (When SLA is 7 Bits)

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.

32.9.1 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 32.74 shows the timing of operations in the generation of start, restart and stop conditions.

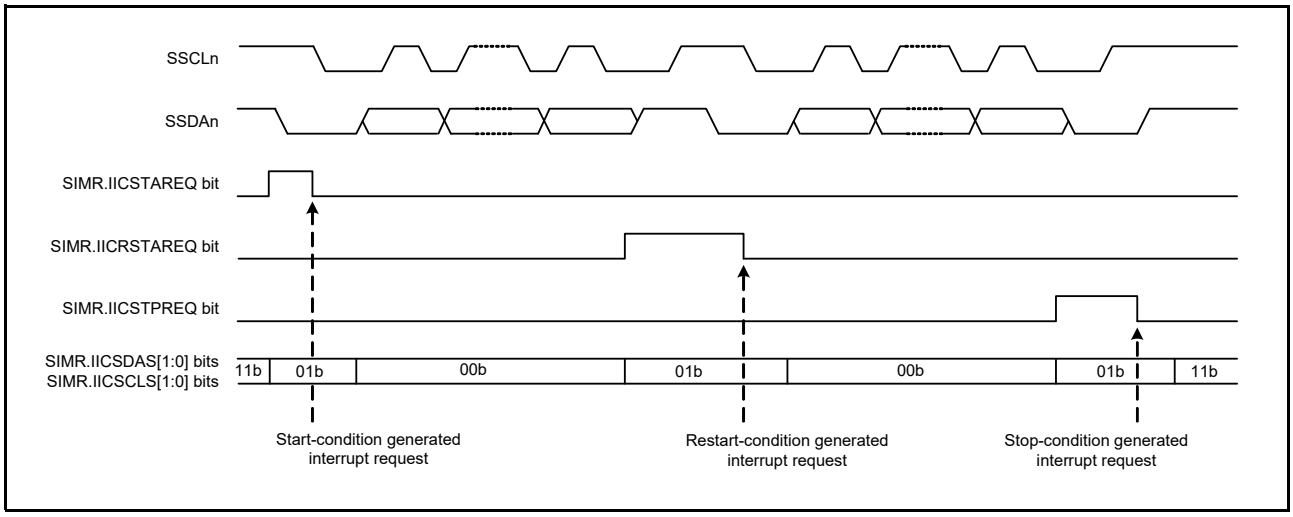


Figure 32.74 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

32.9.2 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 PCLK). 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 32.75 shows an example of operations to synchronize the clocks.

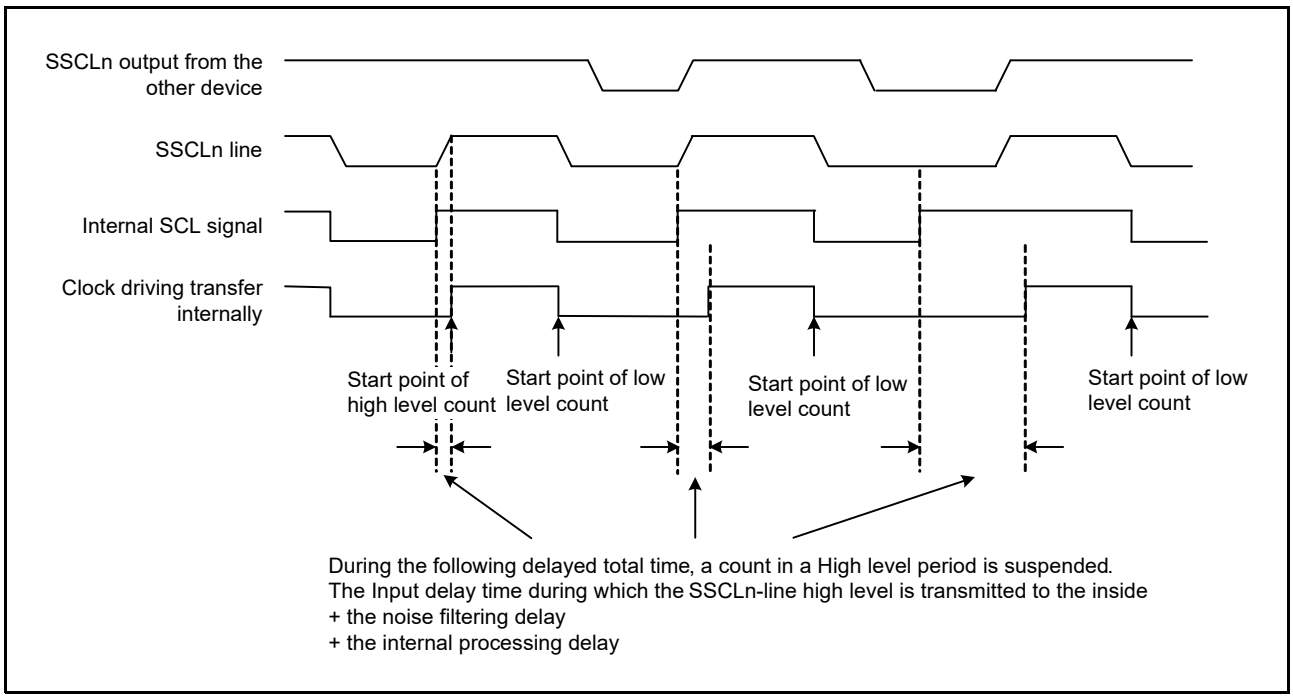


Figure 32.75 Example of Operations for Clock Synchronization

32.9.3 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 PCLK 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 32.76 shows the timing of delays in SDA output.

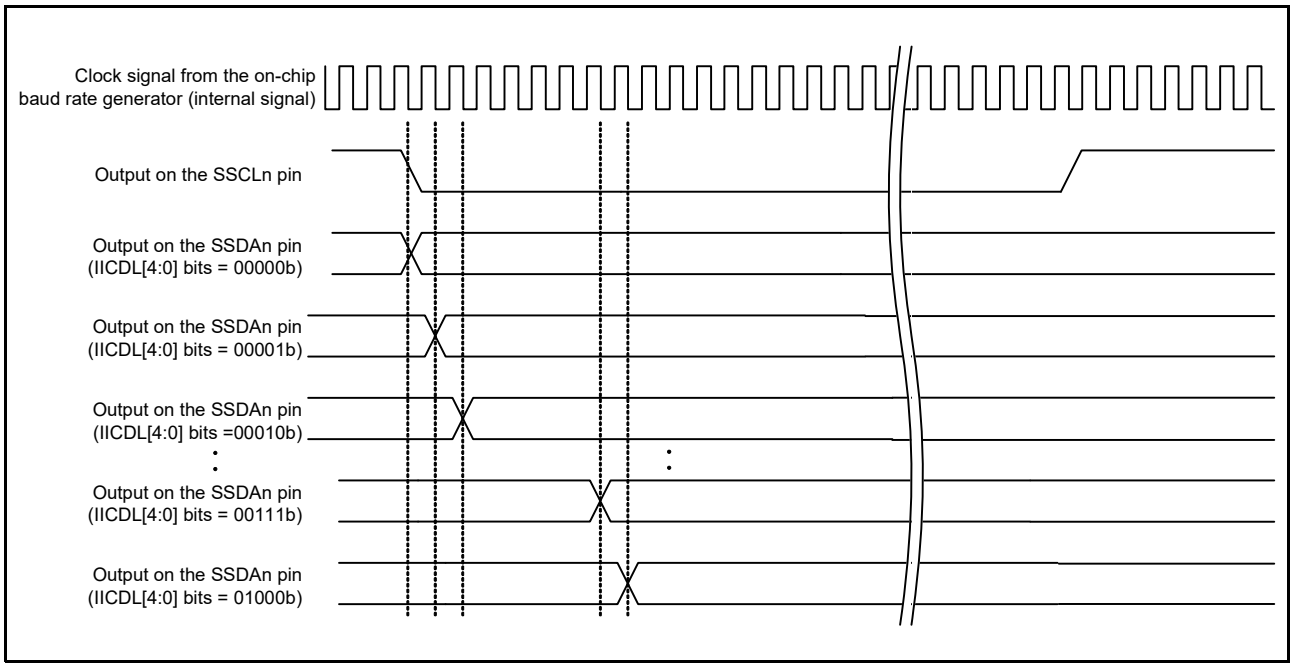
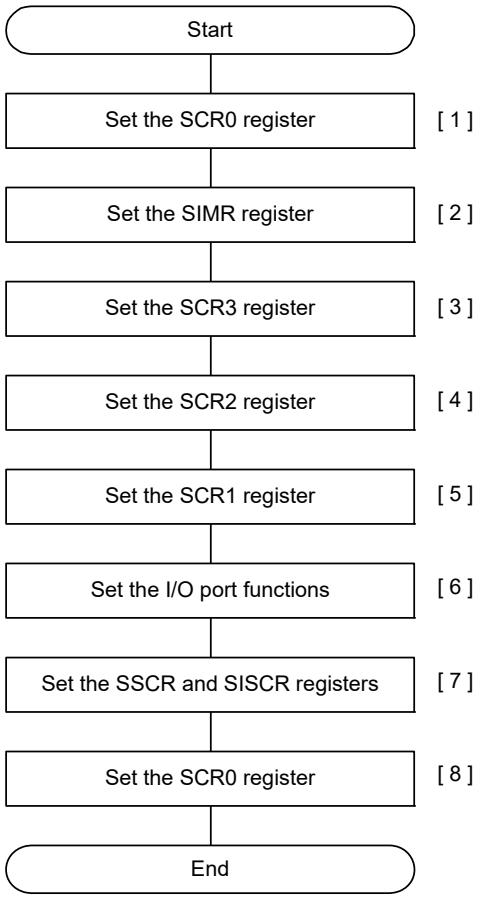


Figure 32.76 Timing of Delays in SDA Output

32.9.4 RSCI Initialization (Simple I2C Mode)

Write initial value (0000_0000h) to SCR0, then initialize RSCI according to Figure 32.77.
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.

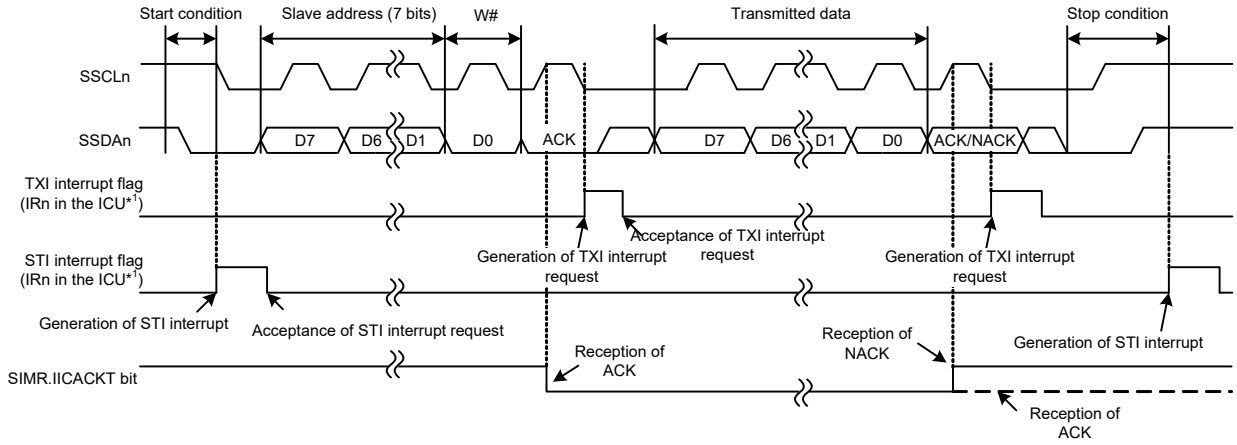
Figure 32.77 Example of the Flowchart of RSCI Initialization (for Simple I2C Mode)

32.9.5 Operation in Master Transmission (Simple I2C Mode)

Figure 32.78 and Figure 32.79 show examples of operations in master transmission, Figure 32.80 to Figure 32.82 show the example flowcharts. See Table 32.41 about the STI interrupt.

Figure 32.78 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 32.80 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 32.81 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 (ICUb) for details on the corresponding interrupt vector number.

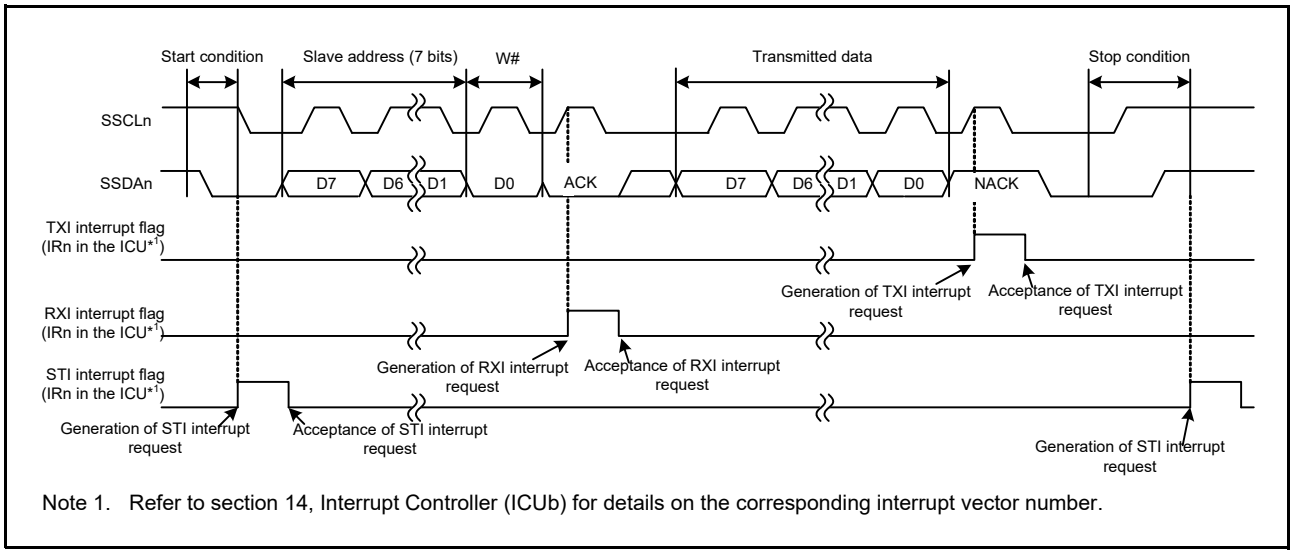
Figure 32.78 Example 1 of Operations for Master Transmission in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))

Figure 32.79 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 32.82 shows a flow chart of SIMR.IICINTM bit is 0

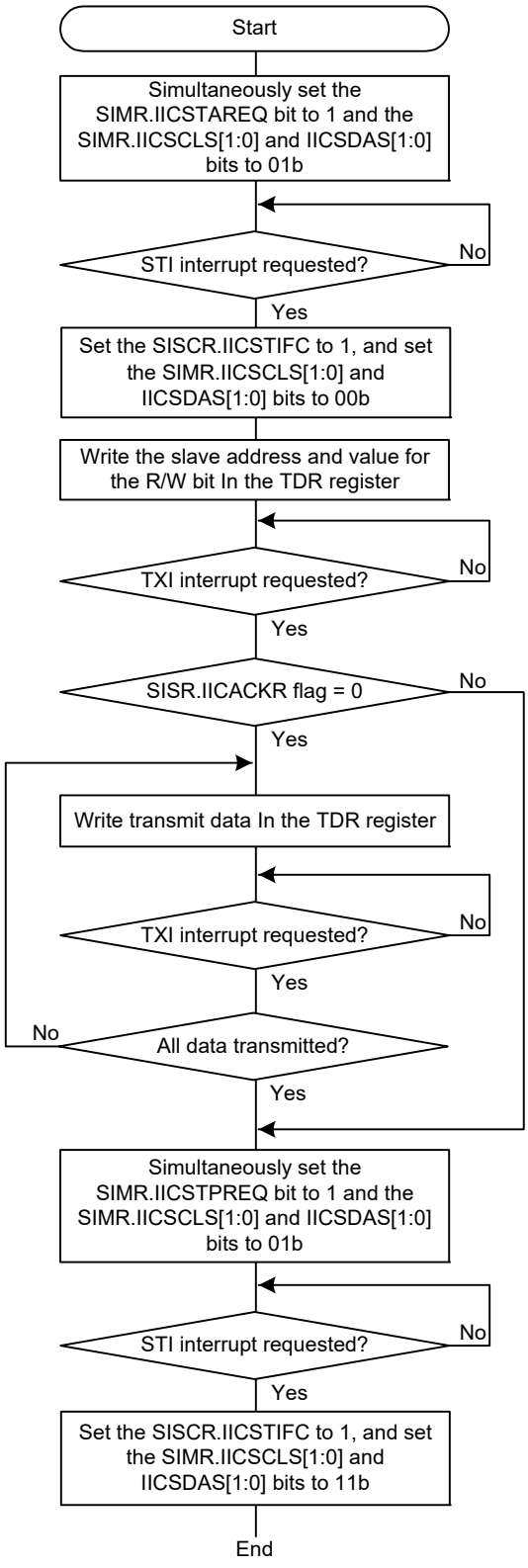
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.79 Example 2 of Operations for Master Transmission in Simple I²C Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use (SIMR.IICINTM Bit = 0))



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] [3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

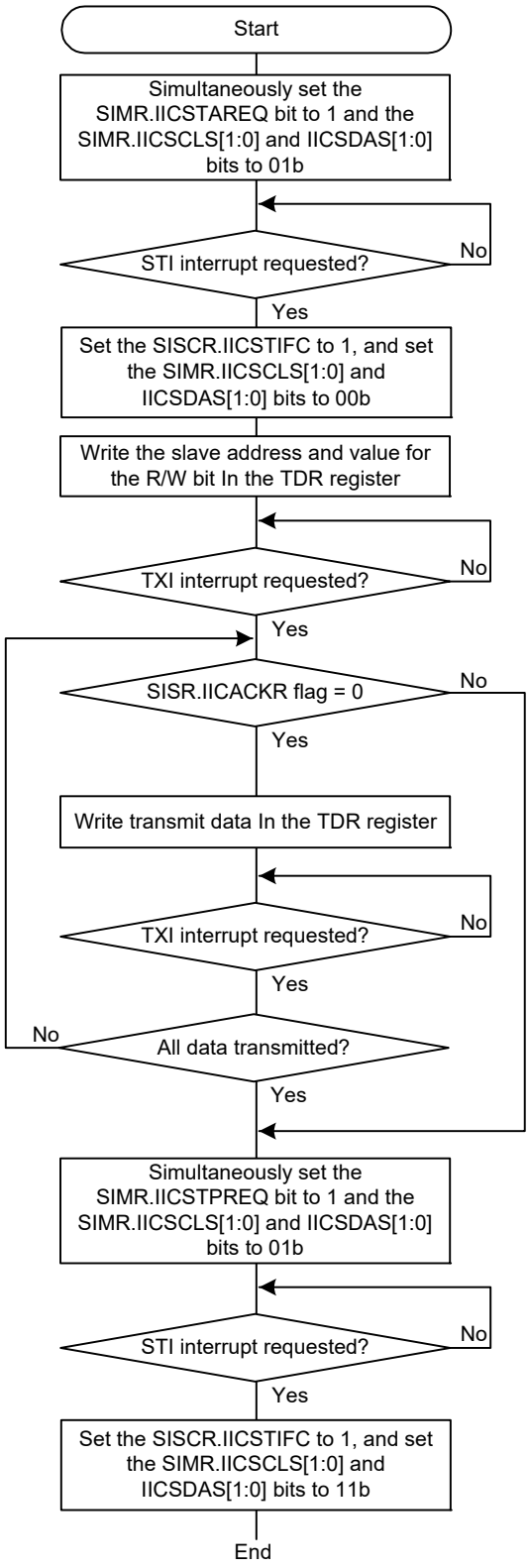
If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] [4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR register. Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to TDR register.
But it cannot be confirmed status of ACK or NACK in this case.

[5] [5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 32.80 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK by Address Transmission Only)



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. To checking whether ACK or NACK in every transmission, you cannot use DMA or DTC.

[5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 32.81 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)

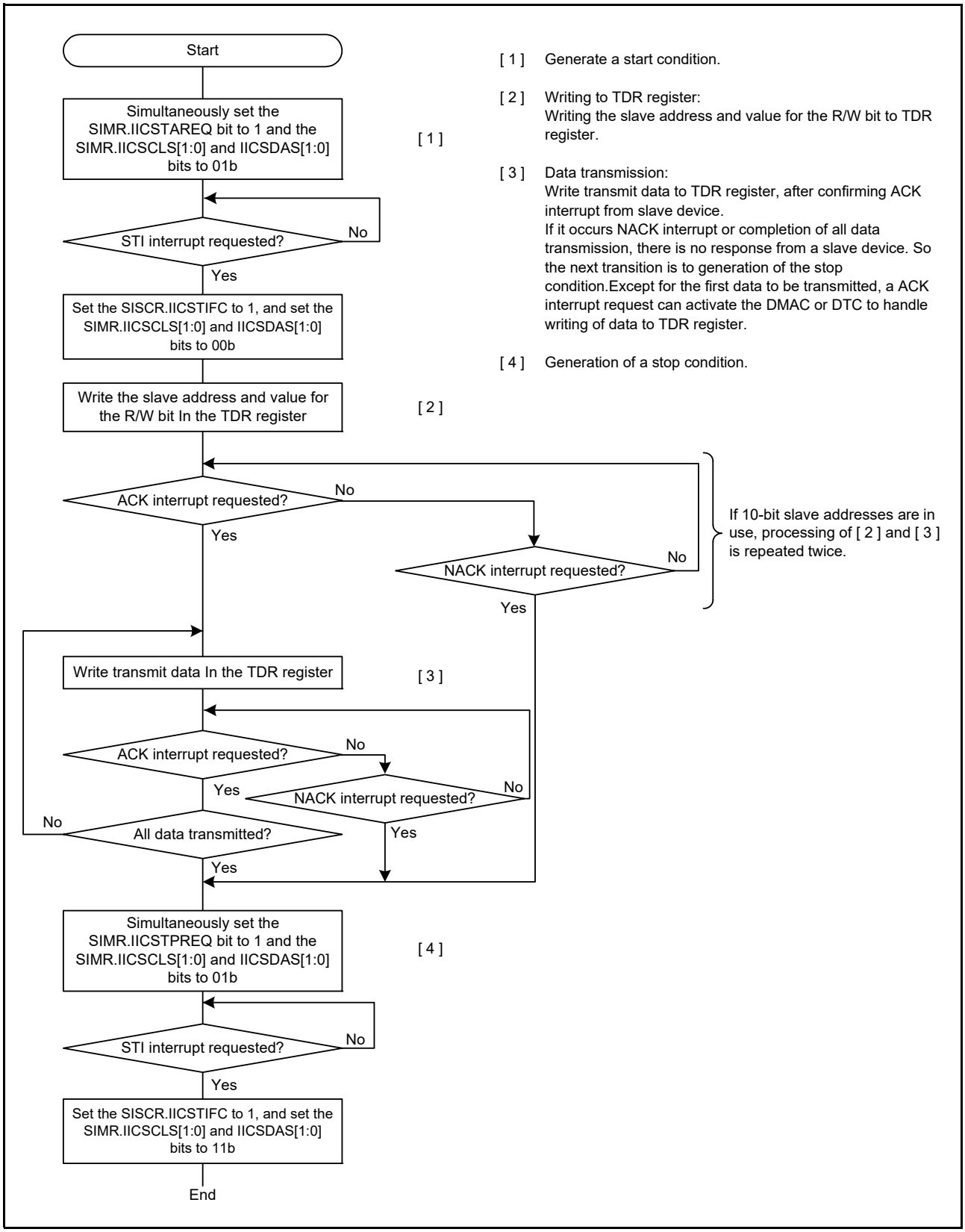
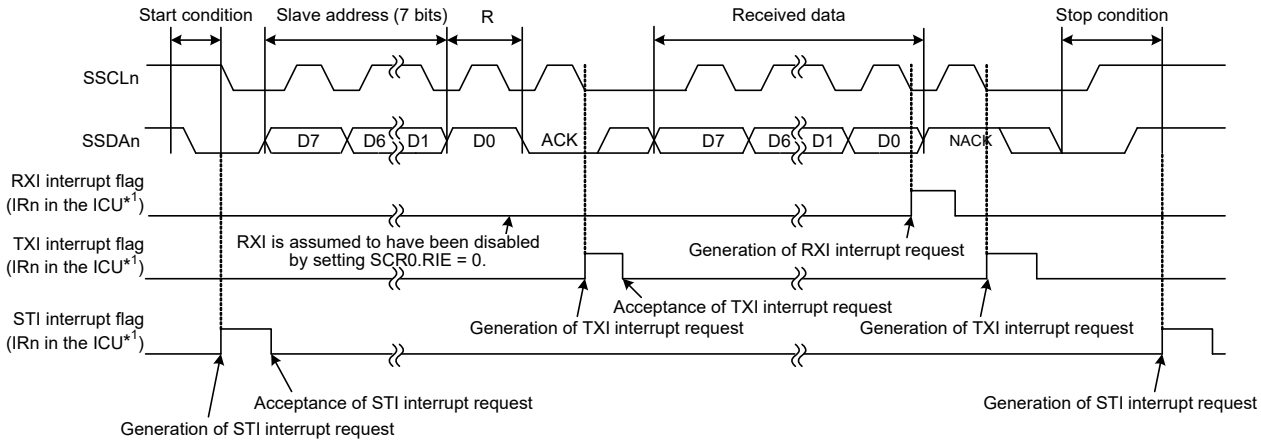


Figure 32.82 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (when SIMR.IICINTM Bit is 0)

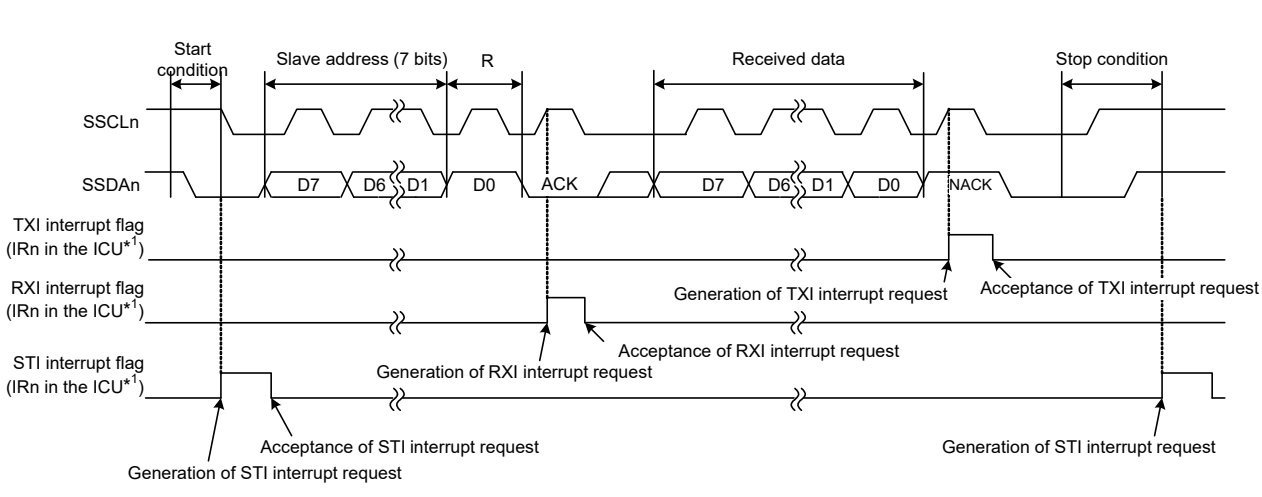
32.9.6 Master Reception (Simple I²C Mode)

Figure 32.83 and Figure 32.84 show example of operations in simple I²C mode master reception. Figure 32.85 and Figure 32.86 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.83 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.84 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))

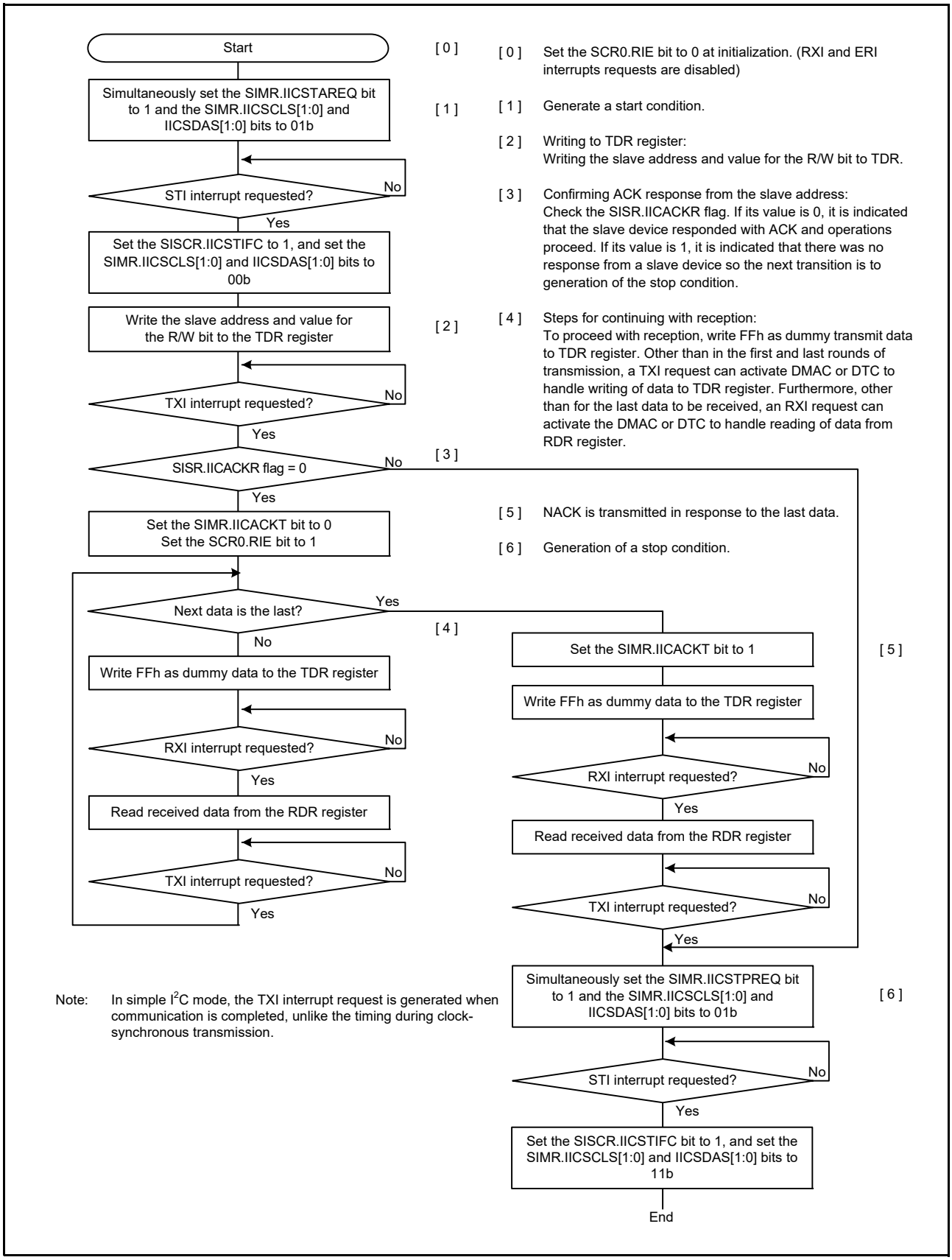


Figure 32.85 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 1, and Transmission Interrupts and Reception Interrupts in Use)

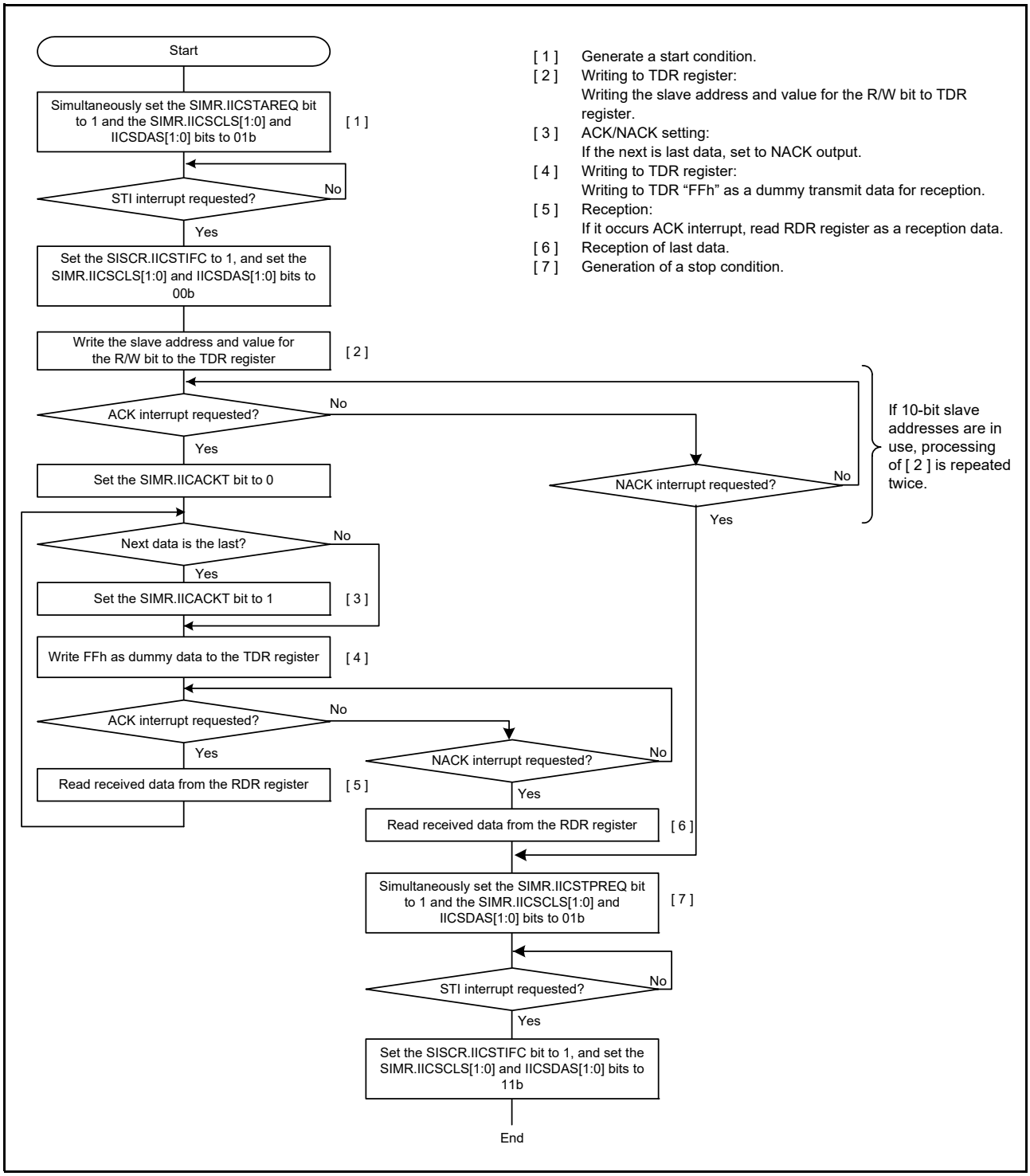


Figure 32.86 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 0, and ACK Interrupts and NACK Interrupts are in Use)

32.10 Operation in Clock Synchronous Mode

Figure 32.87 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.

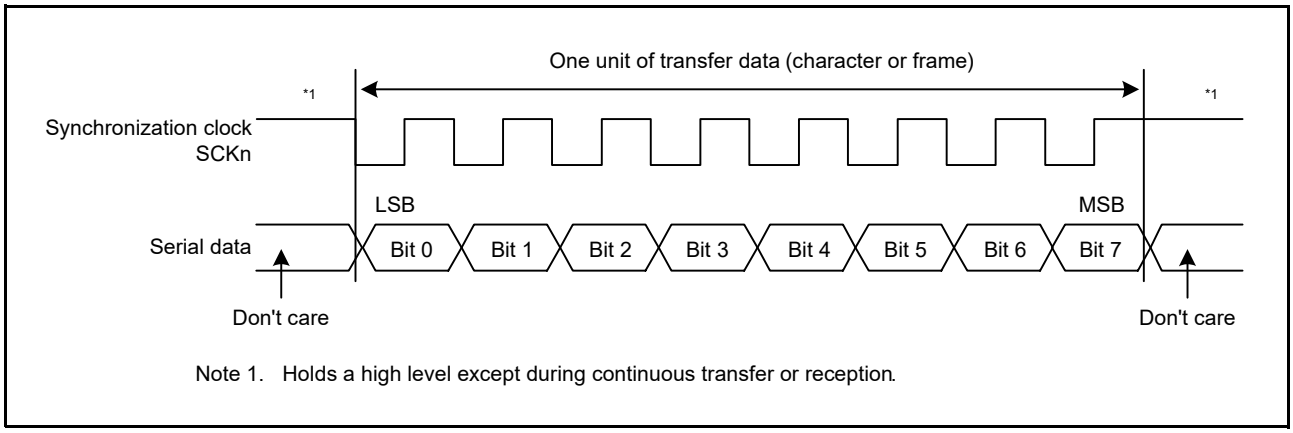


Figure 32.87 Data Format in Clock Synchronous Serial Communications (LSB First, CPHA Bit = 1, CPOL Bit = 1)

32.10.1 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.

32.10.2 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.
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”. 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.

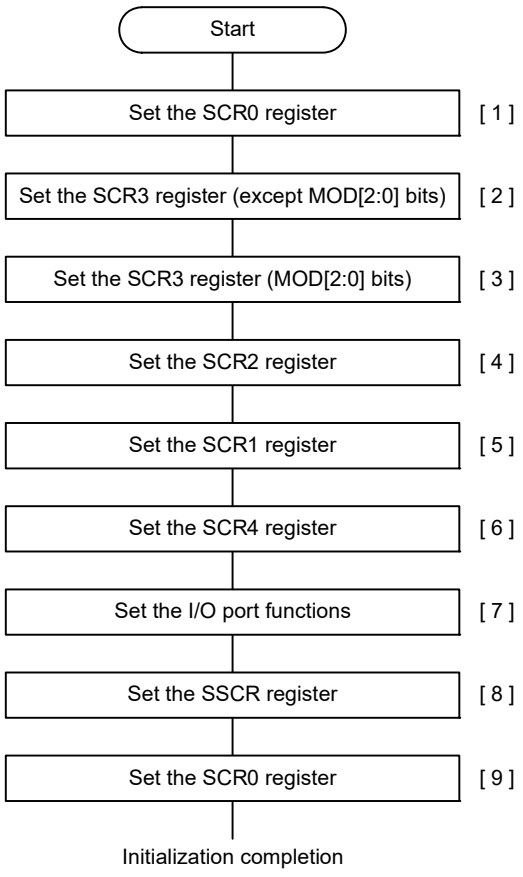
[Conditions for low-level output]
When the following conditions are all satisfied

- 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.

32.10.3 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 32.88.
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. 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 SCR3 except of communication mode.
Transmission/reception format
Clock setting
Leave unused bits (CHR[1:0], STOP, RXDESEL, MP, DEEN, GM, BLK) at their initial values.[*2]

[3] Set communication mode (MOD[2:0] bits = 010b)[*2].

[4] Select clock, set bit rate[*3]. Leave unused bits (SCR2.BCP[2:0], ABCS, ABCSE, BRME, and MDDR[7:0]) at their initial values.

[5] Set up the half-duplex communication, the loop-back function, communication pin status, and the CTS/RTS function.

[6] Set up the adjust sampling timing function. Leave unused bits (SCR4.CMPD[8:0] bits) at their initial values.

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Write 1 to the following bits to clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC

[9] 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.

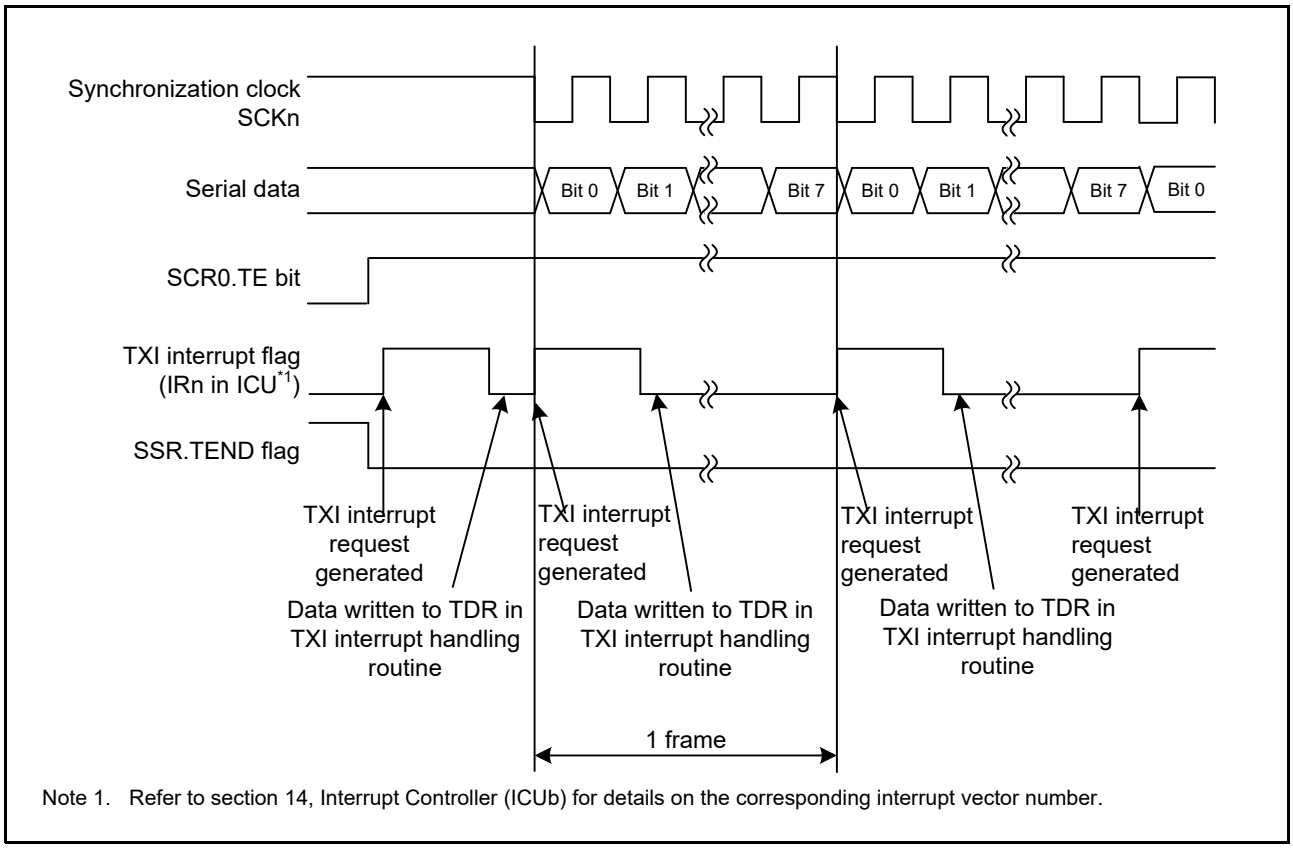
Figure 32.88 Example of RSCI Initialization Flowchart (Clock Synchronous Mode)

32.10.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 32.89 to Figure 32.91 show an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the RSCI operates as described below.

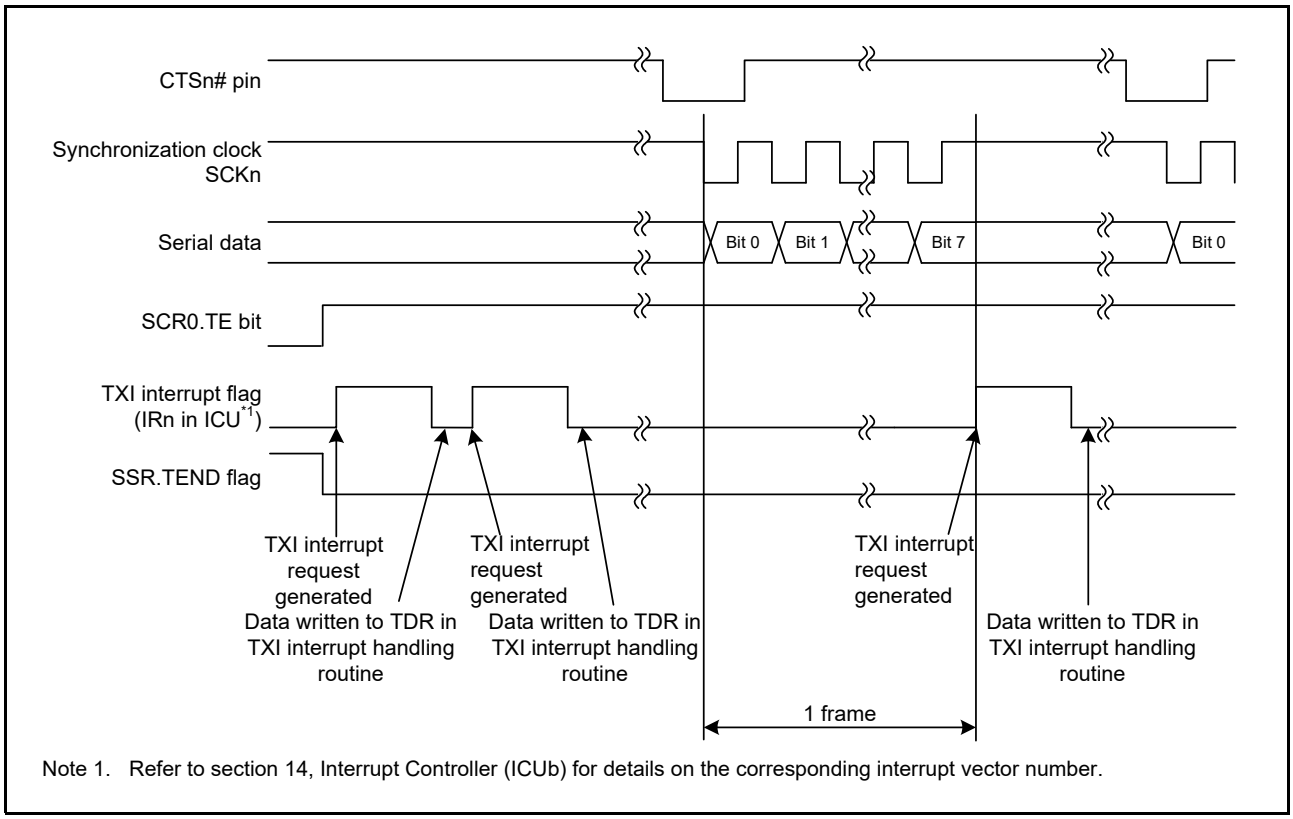
1. When data is written to the TDR register in the TXI interrupt routine, the RSCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the SCR0.TIE bit and the SCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
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 32.92 shows a sample flowchart of serial data transmission.



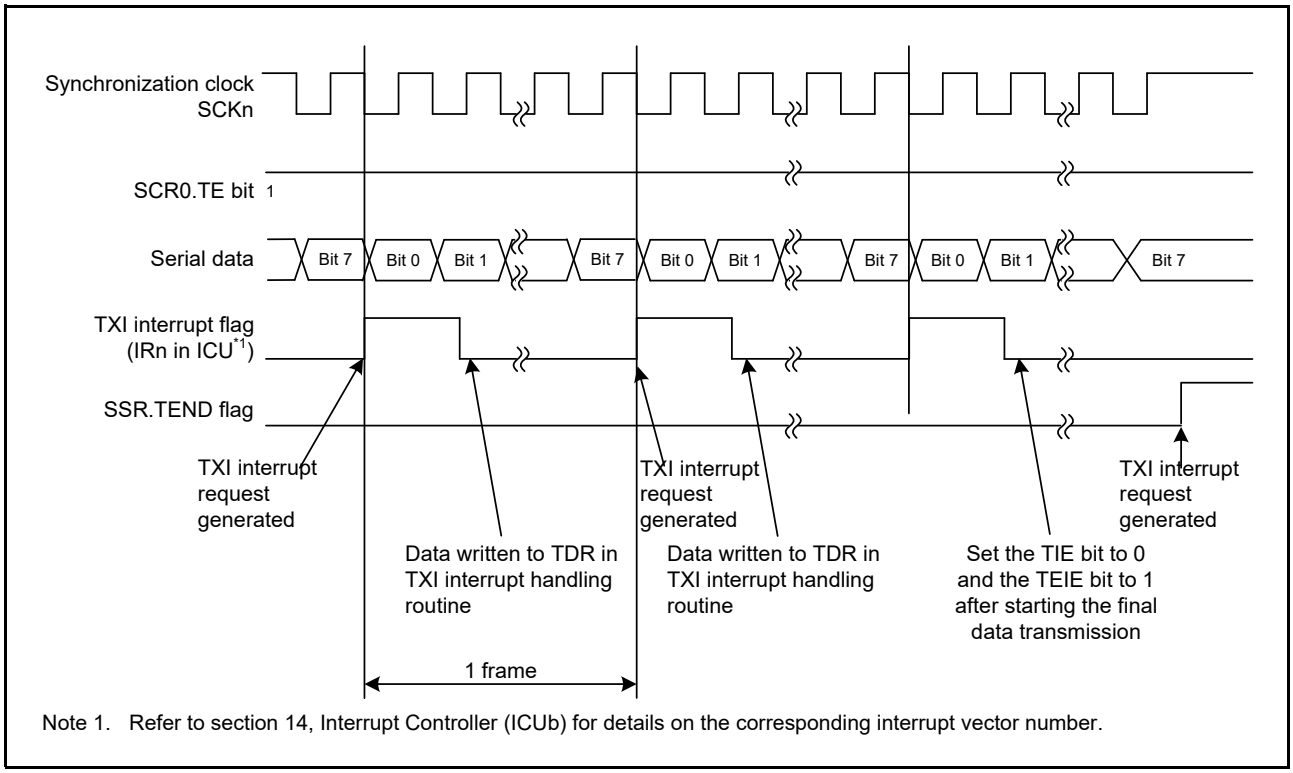
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.89 Serial Transmission Example in Clock Synchronous Mode (1) (CTS Function Not Used/ Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



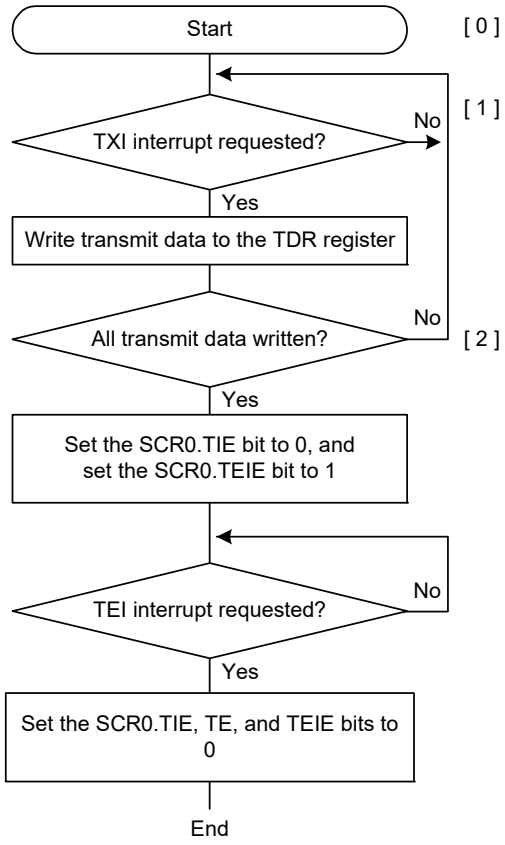
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.90 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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.91 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.

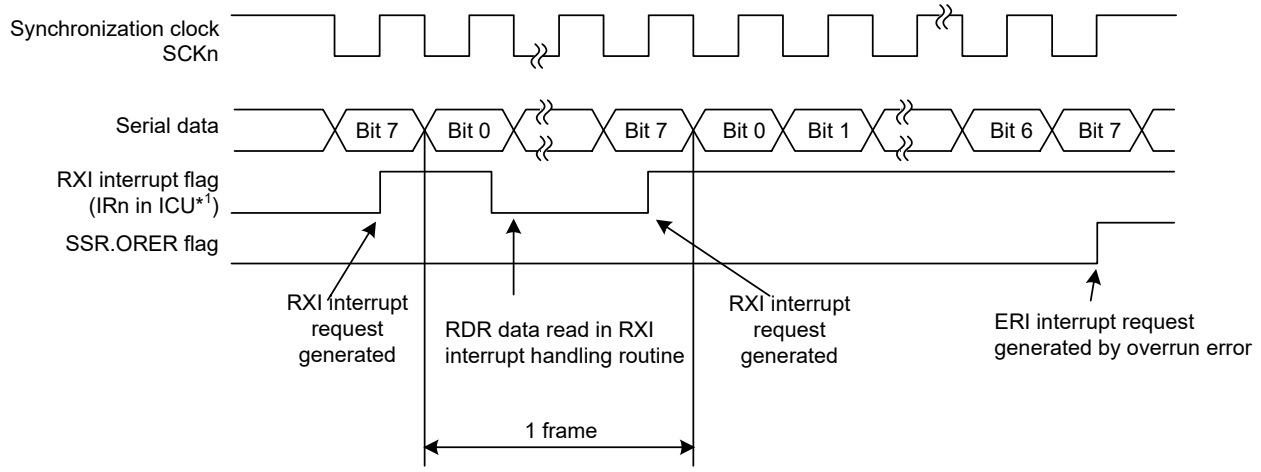
Note: When the external clock is in use (the value of the SCR3.CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 32.92 Example Flowchart of Serial Transmission in Clock Synchronous Mode

32.10.5 Serial Data Reception (Clock Synchronous Mode)

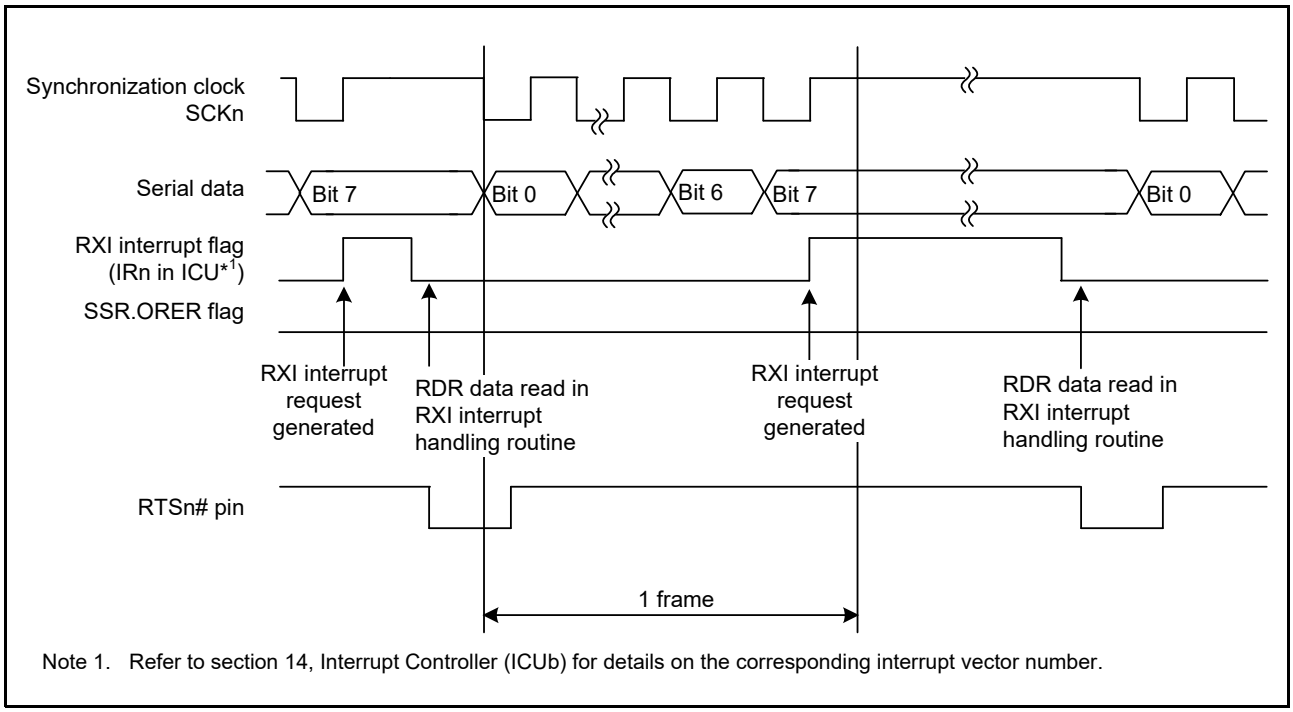
Figure 32.93 and Figure 32.94 show operation examples of serial data reception in clock synchronous mode.
The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.93 Example of Operation for Serial Reception in Clock Synchronous Mode (1)
(When RTS Function is Not Used/CPHA Bit = 1, CPOL Bit = 1)

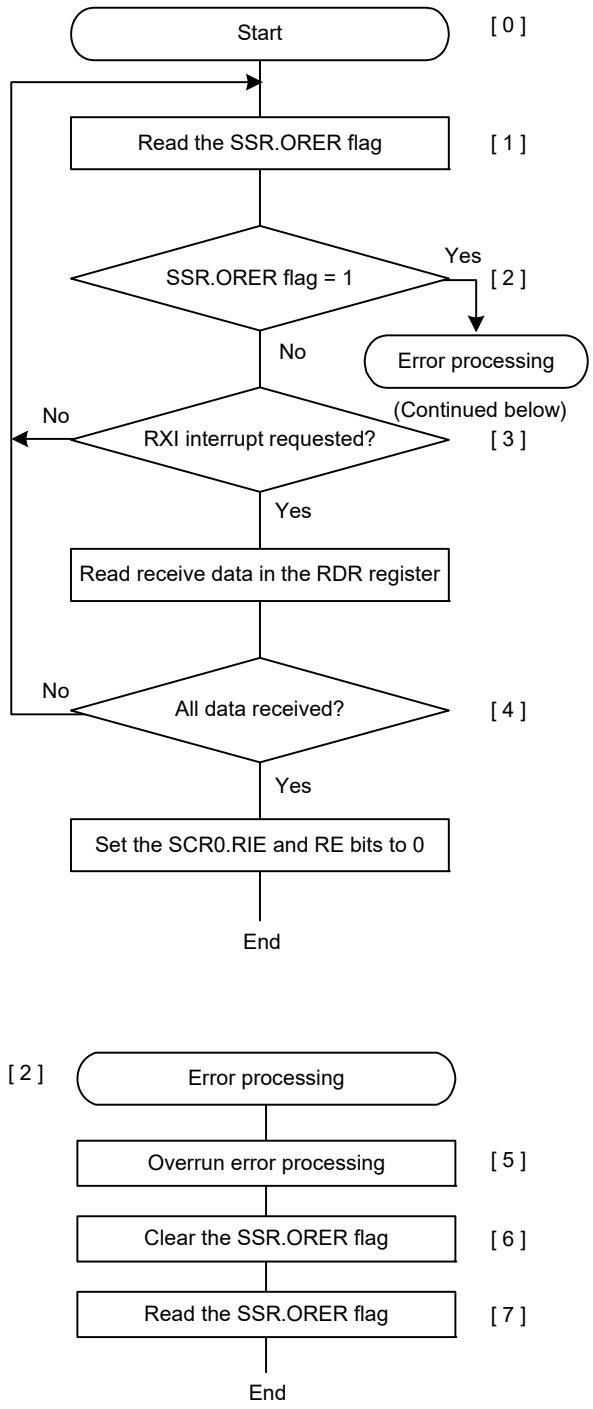


Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 32.94 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 32.95 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 32.95 Example Flowchart of Serial Reception in Clock Synchronous Mode

32.10.6 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.

Figure 32.96 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.

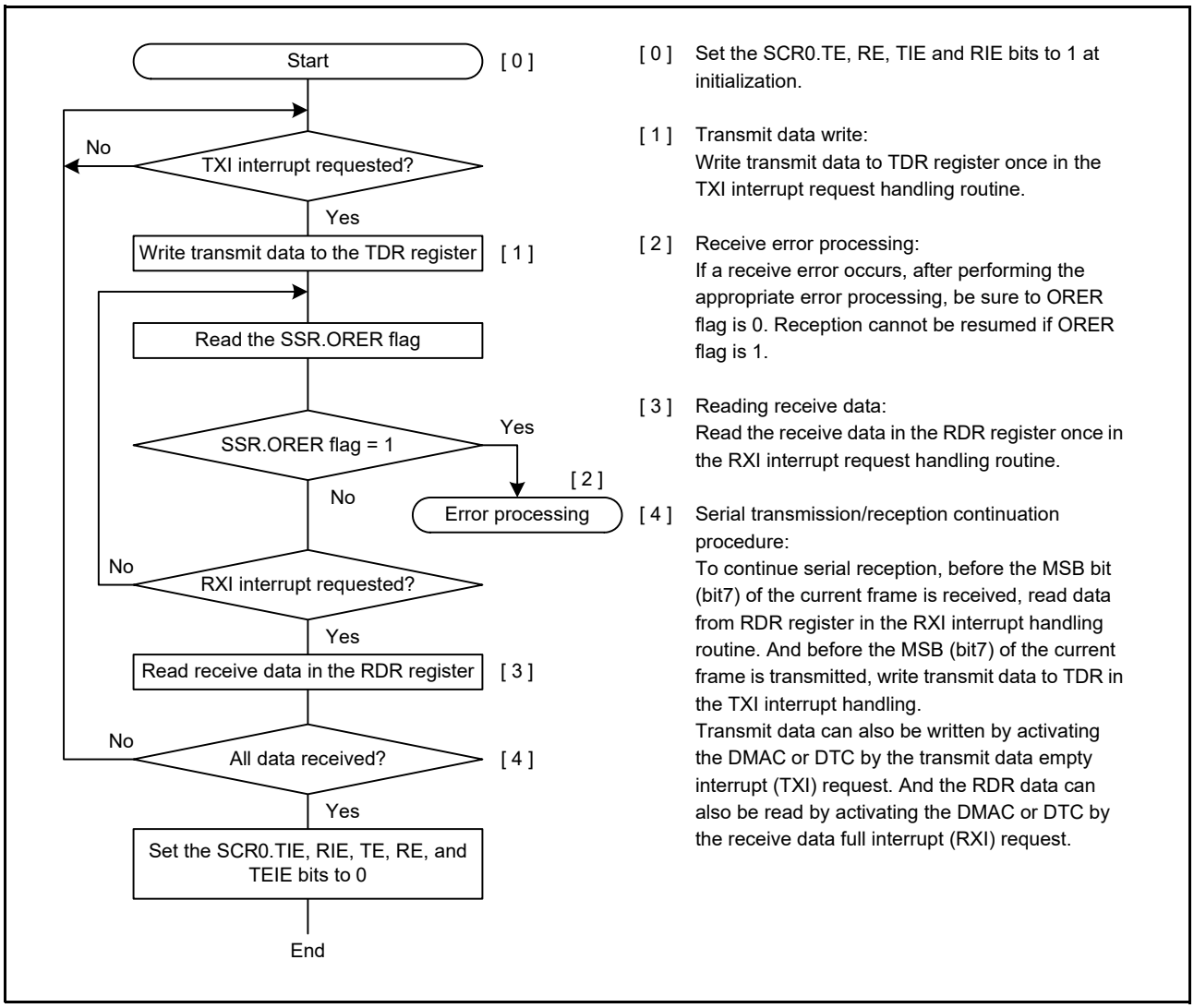


Figure 32.96 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

32.10.7 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 PCLK 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.

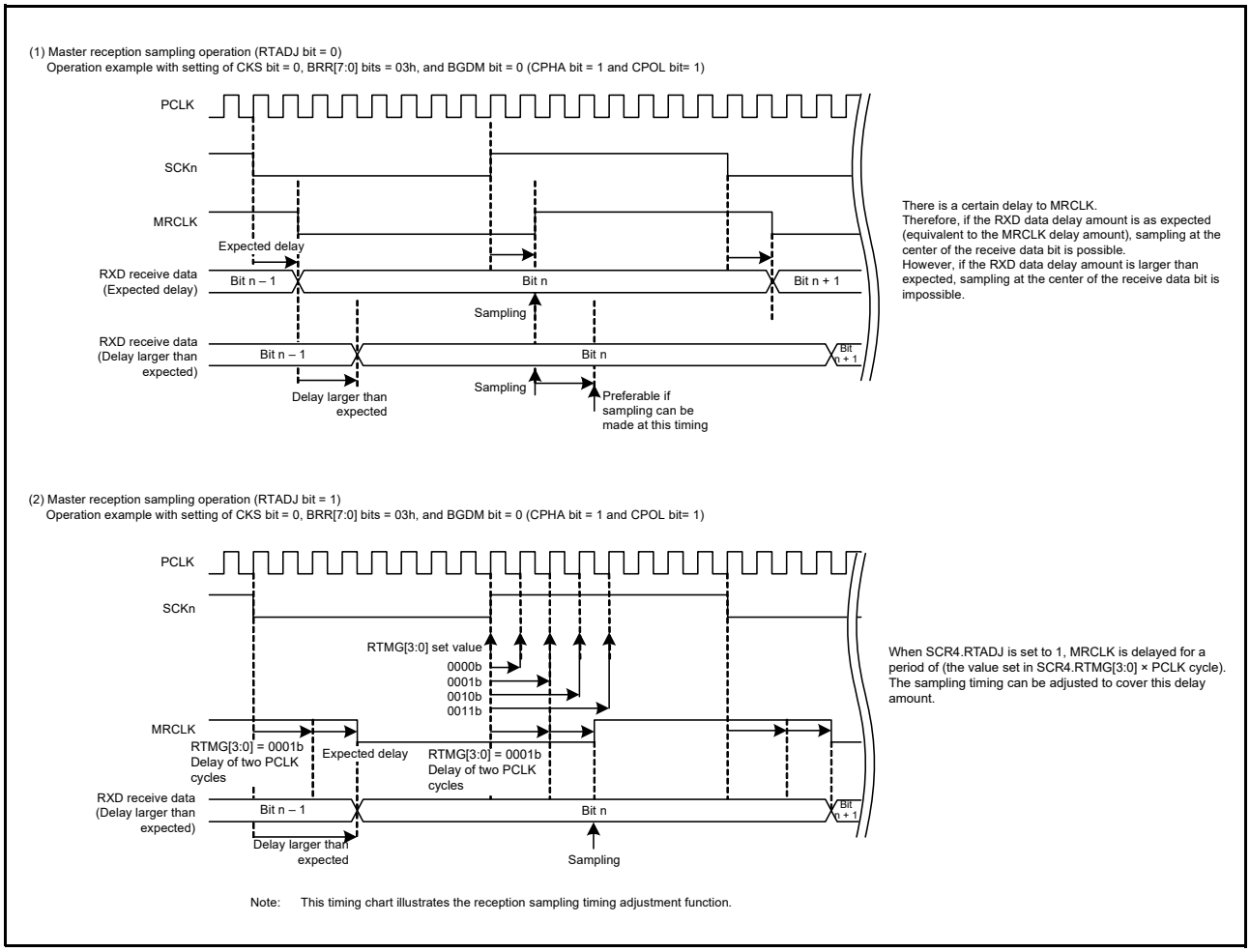


Figure 32.97 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master) and Simple SPI Mode (Master)

32.11 Operation in Simple SPI Mode

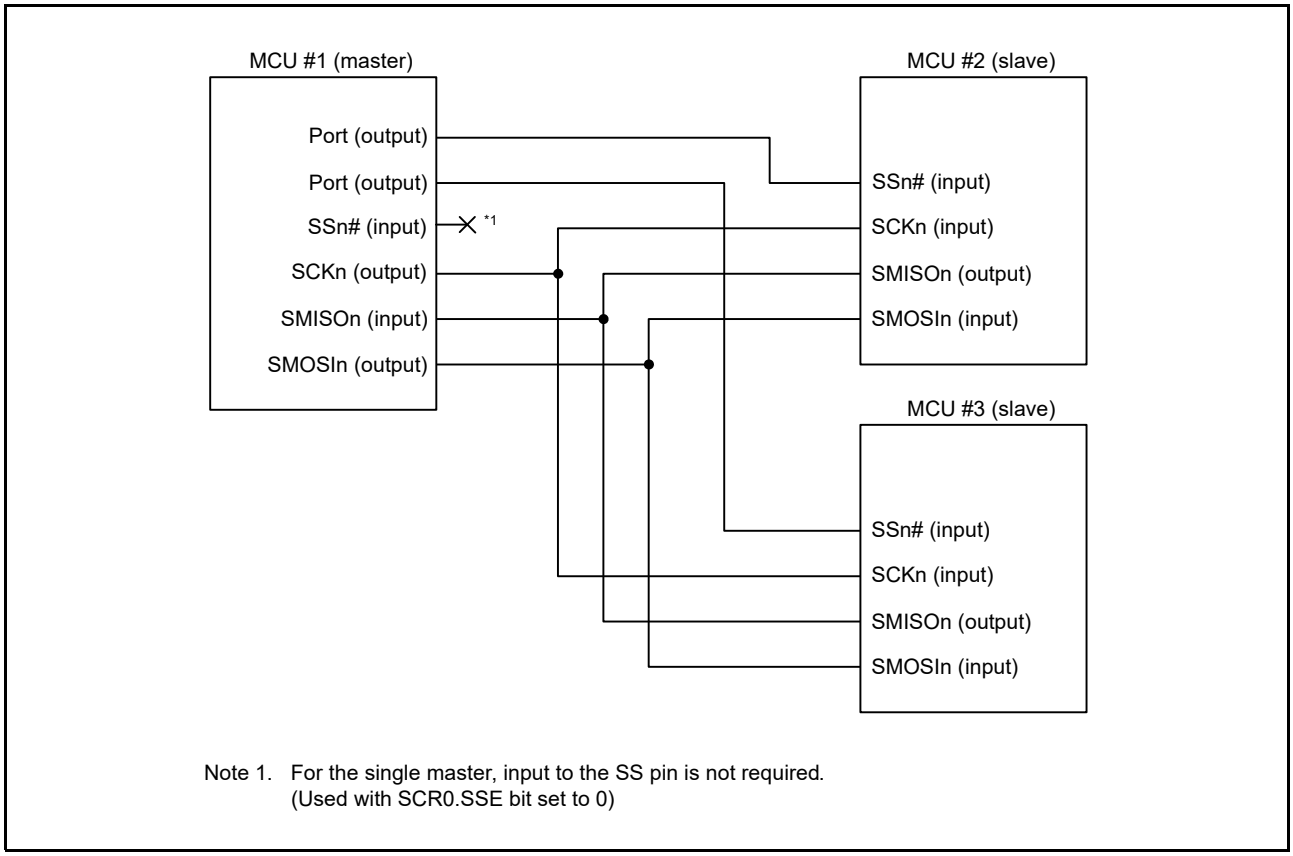
As an extended function, the RSCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

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 32.98 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 32.98 Example of Connections via a Simple SPI Mode

32.11.1 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 32.36 lists the states of pins according to the mode and the level on the SSn# pin.

Table 32.36 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*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).

32.11.2 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. In multi-master mode, read this flag bit to perform the error processing. If a mode fault 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.

32.11.3 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 PCLK cycles.

32.11.4 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 32.99 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

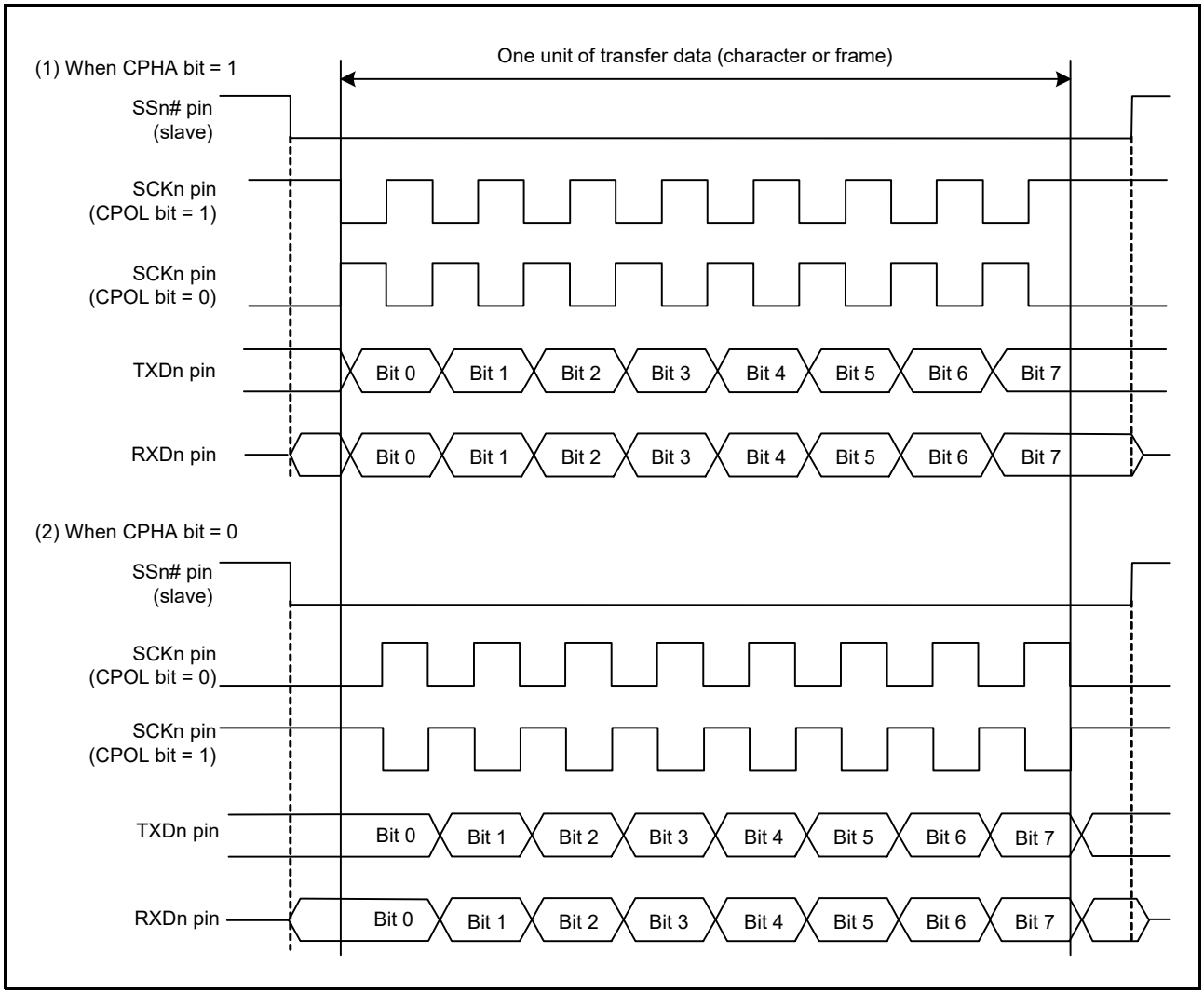


Figure 32.99 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

32.11.5 RSCI Initialization (Simple SPI Mode)

The RSCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 32.88, 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.

32.11.6 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 will occur if the SSn# pin goes low. Therefore, make sure that no mode fault has occurred before starting communication, and start communication, and make sure that no mode fault has occurred even after communication ends. If a mode fault 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.

32.11.7 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 32.10.7, Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

32.12 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 32.100 shows an example where the PCLK 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.

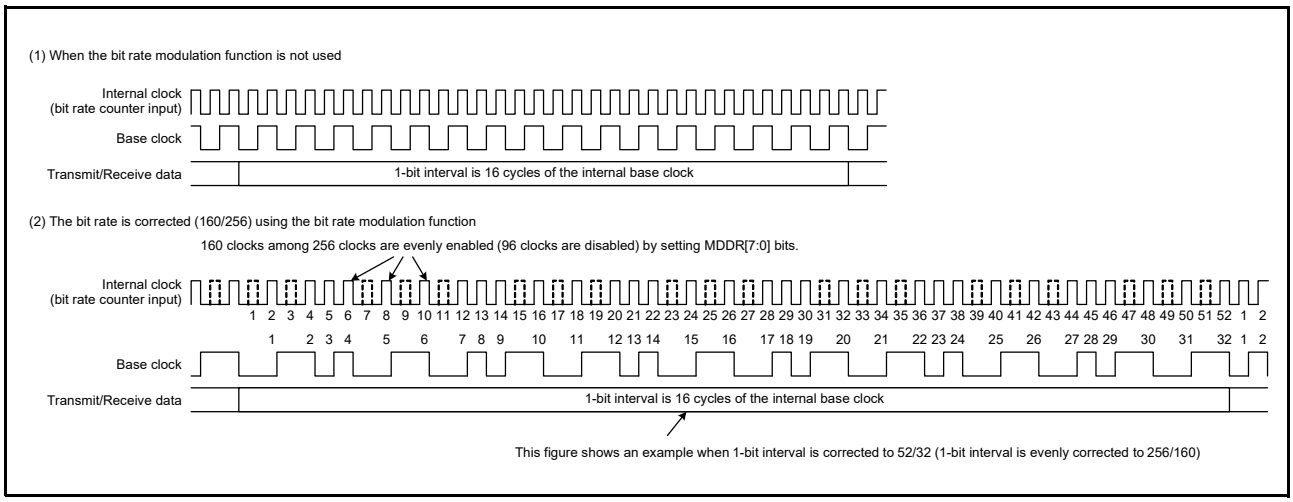


Figure 32.100 Example of Base Clock when Using the Bit Rate Modulation Function

32.13 Noise Cancellation Function

Figure 32.101 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.

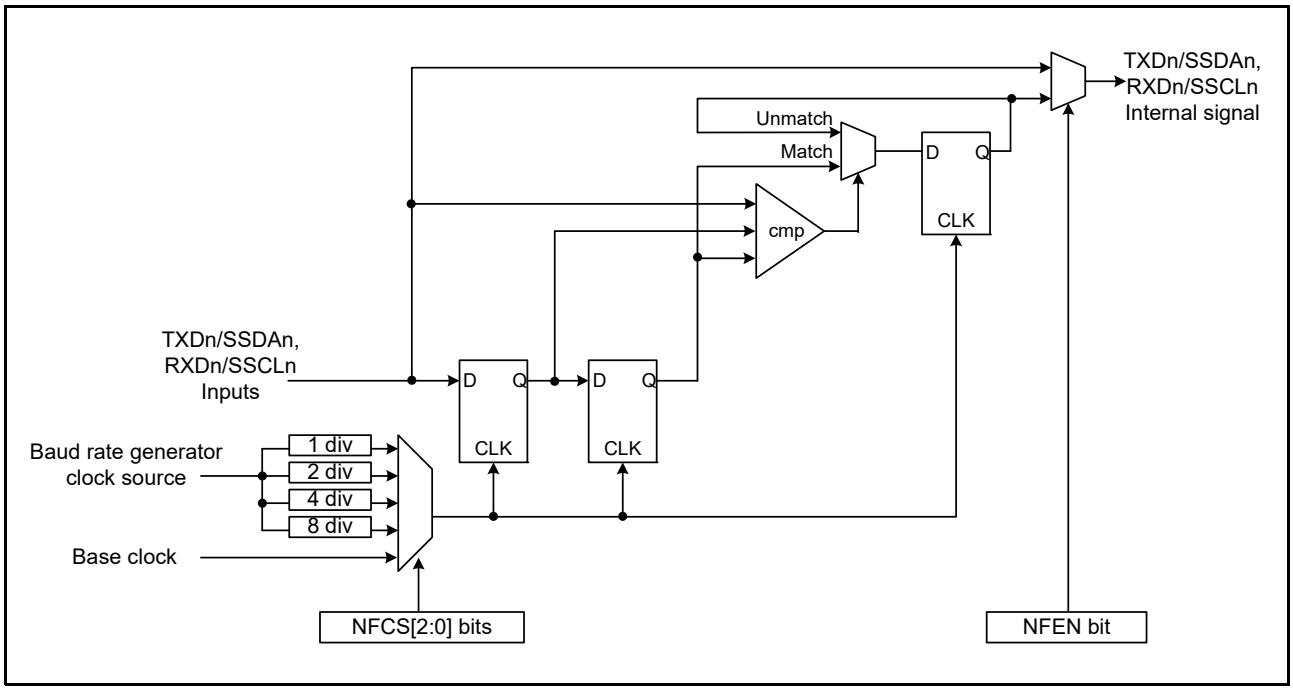


Figure 32.101 Block Diagram of Digital Noise Filter

32.14 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 (1/6, 1/8, or 1/16 bit time). For details, refer to section 32.2.12, 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.

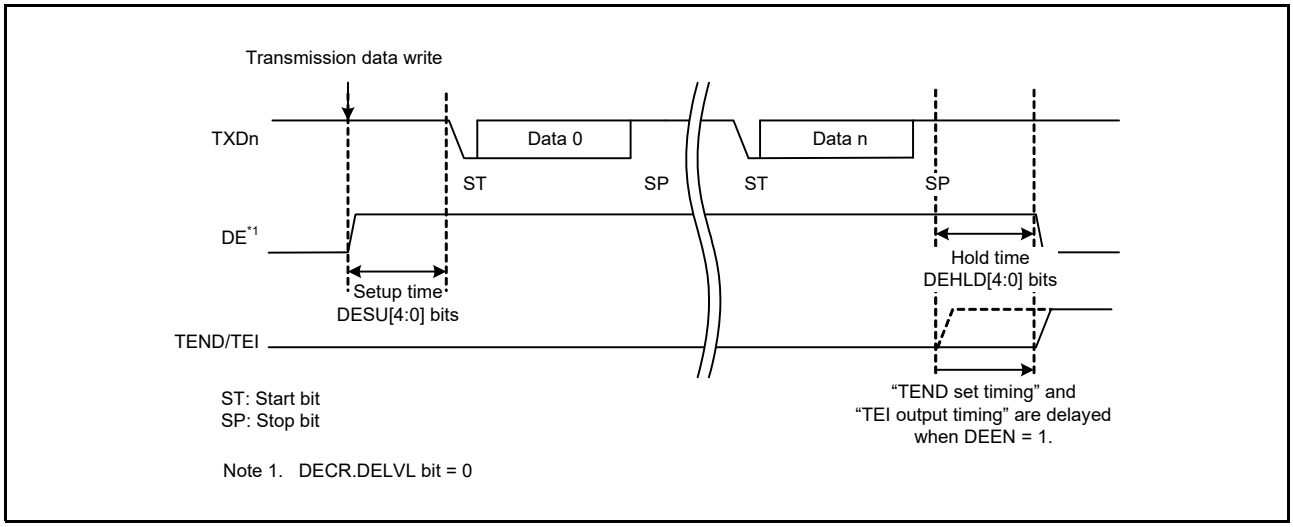


Figure 32.102 Image Waveform for RS-485 Driver Control DE Signal Output

32.15 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 32.37 shows the relationship between the TINV and LOOP bit settings and the received data.

Table 32.37 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 32.103 shows the configuration of the shift register input/output path in loopback mode.

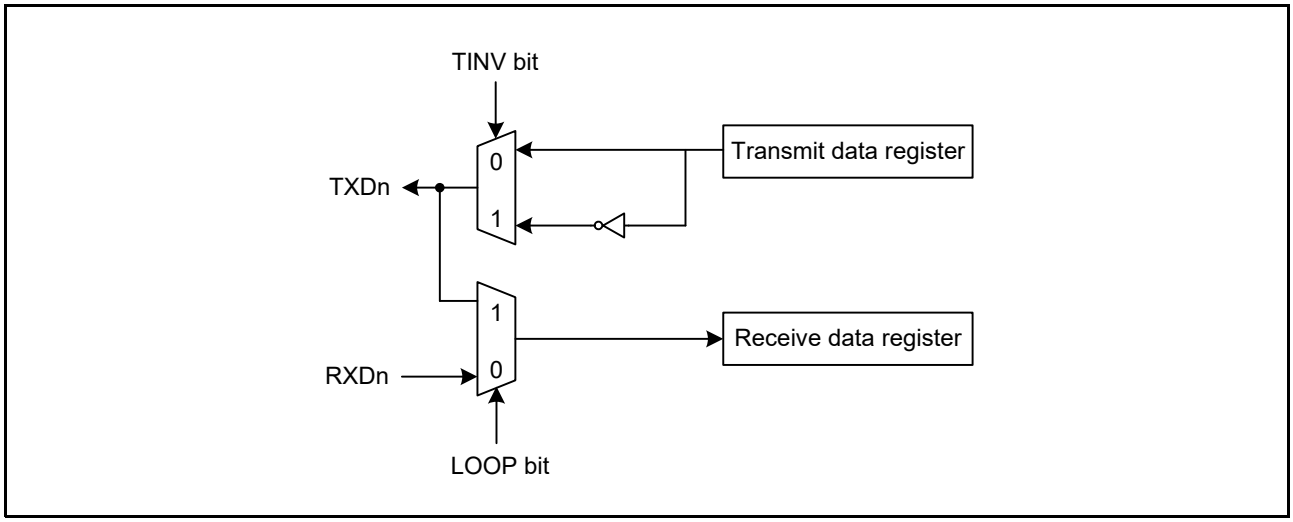


Figure 32.103 Shift Register Input/Output Configuration Image in Loopback Mode

32.16 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.

32.17 Interrupt Signal

Table 32.38 lists RSCI interrupt signals.
The interrupt explanation corresponding to each operation mode is described in sections 32.17.2 to 32.17.5. Also, TXI and RXI have an interrupt buffer function. Refer to section 32.17.1, 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 (DMACA) for how to set DTC or DMAC.

Table 32.38 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	PCLK	
RXI	Simple I²C: Reception end interrupt Other mode: Receive data full interrupt	Pulse	1 cycle	Low	PCLK	
TXI	Simple I²C and smart card interface: Transmit end interrupt Other mode: Transmit data empty interrupt, Break Field transmission completion	Pulse	1 cycle	Low	PCLK	
TEI	Simple I²C: Completion of generation of a start, restart, or stop condition (STI) Other mode: Transmit end interrupt	Level	—	Low	PCLK	
AED	Active edge detection interrupt	Pulse	1 cycle	Low	PCLK	
BFD	Break Field detection interrupt	Level	—	Low	PCLK	Only when extended serial mode

32.17.1 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.

32.17.2 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.

Table 32.39 lists interrupt sources in asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode.

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.

Table 32.39 RSCI Interrupt Sources

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

32.17.3 Interrupt in Smart Card Interface Mode

Table 32.40 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 32.40 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 (DMACA).

32.17.4 Interrupts in Simple I²C Mode

Table 32.41 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 32.41 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.

32.17.5 Interrupts in Extended Serial Mode

Table 32.42 lists interrupt sources in extended serial mode.

Table 32.42 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 32.66), 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).

32.18 Usage Notes

32.18.1 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.

32.18.2 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 32.104 shows a sample flowchart for transition to software standby mode during transmission. Figure 32.105 and Figure 32.106 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 32.107 shows a sample flowchart for reception to software standby mode during reception.

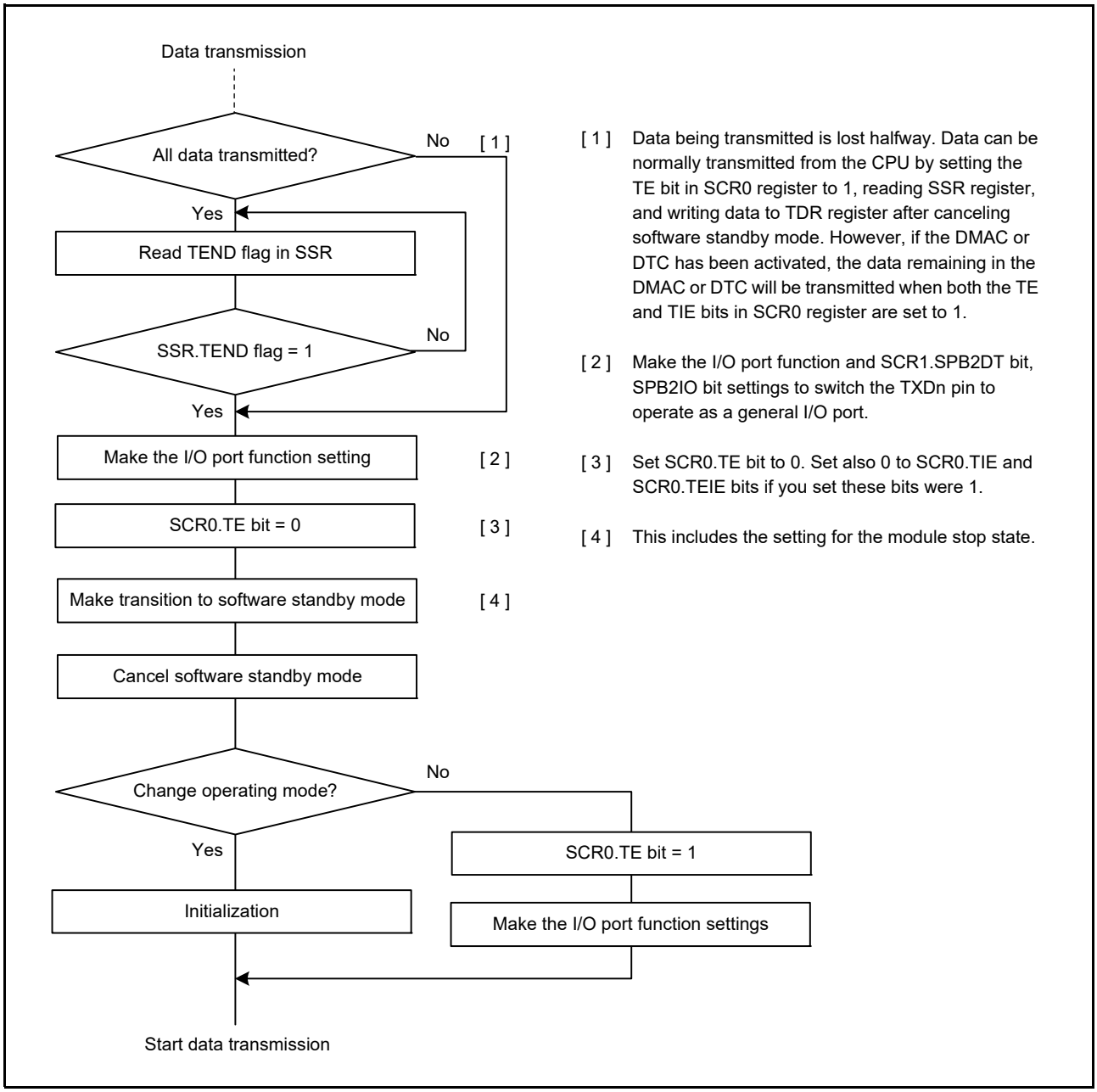


Figure 32.104 Example of Flowchart for Transition to Software Standby Mode during Transmission

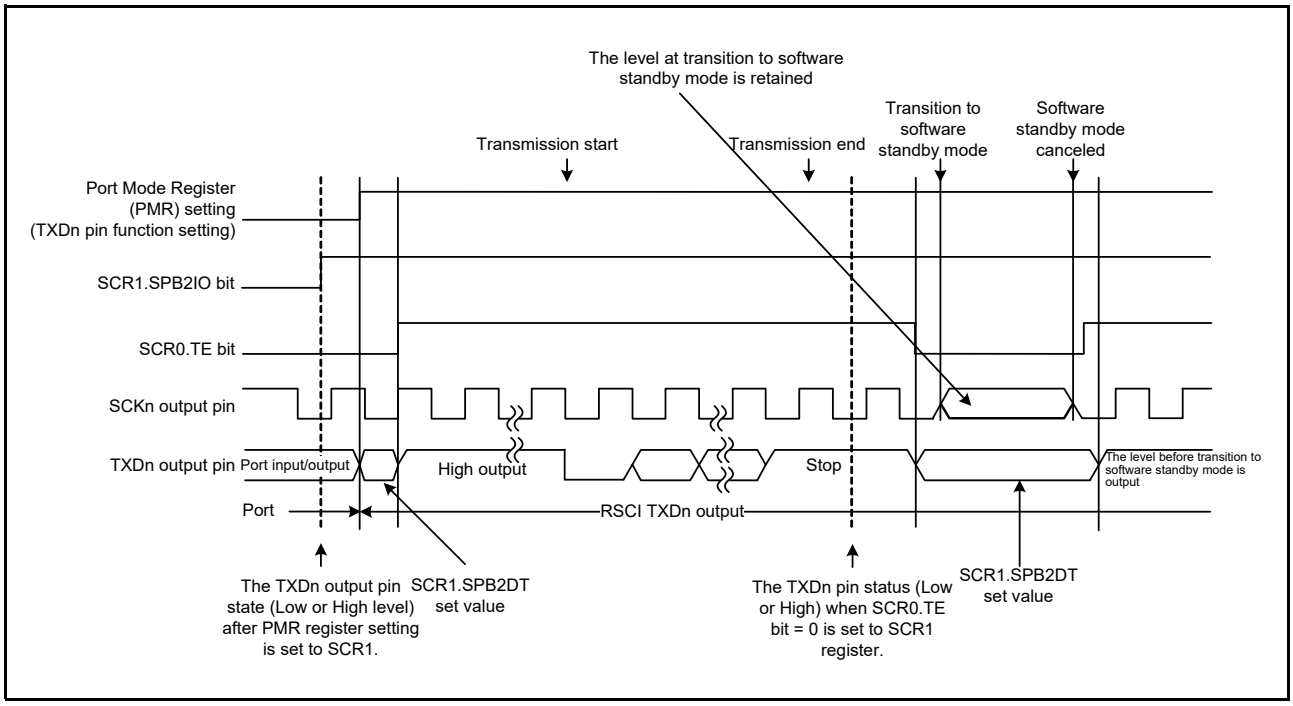


Figure 32.105 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

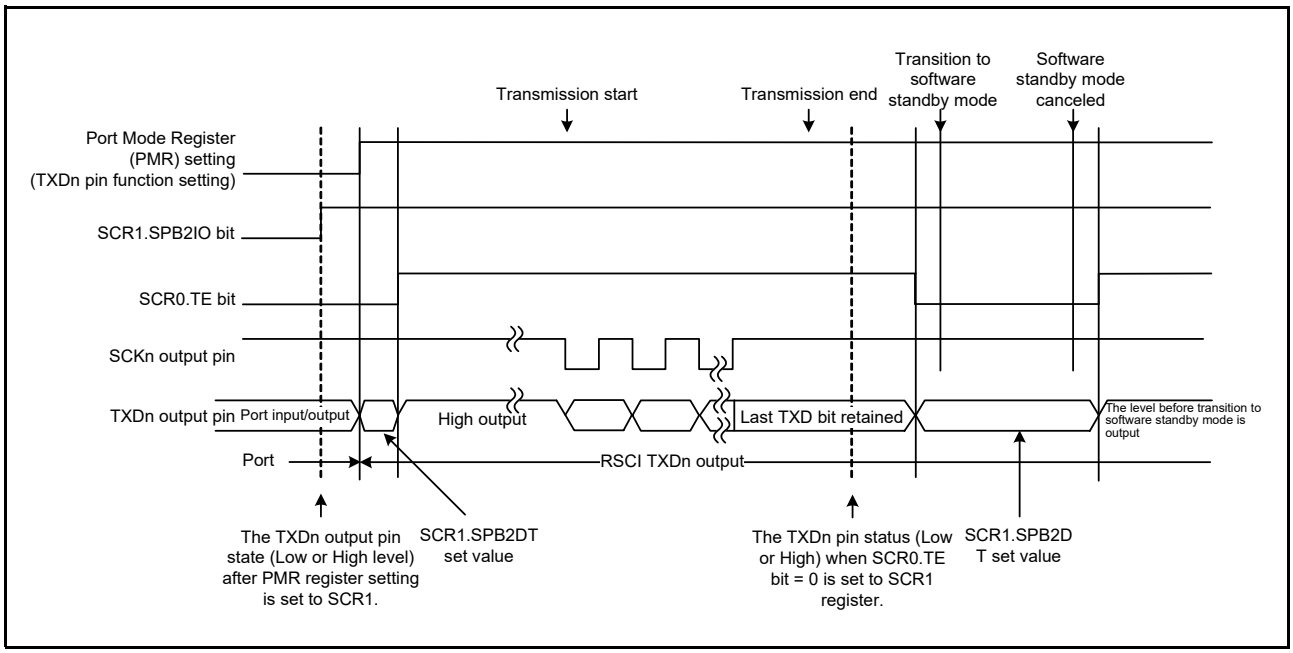


Figure 32.106 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

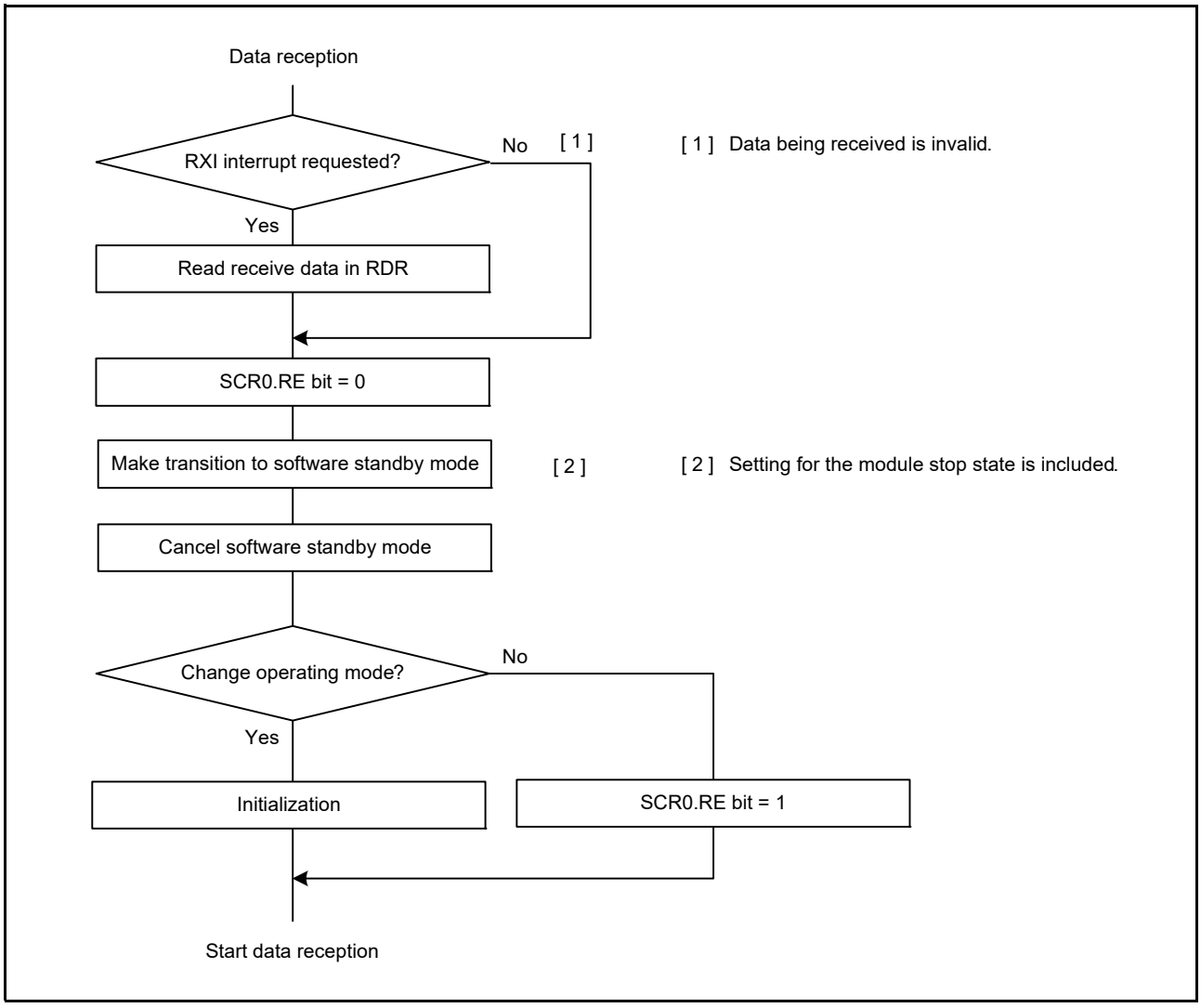


Figure 32.107 Example of Flowchart for Transition to Software Standby Mode during Reception

32.18.3 Break Detection and Processing

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.

32.18.4 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.

32.18.5 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).

32.18.6 Writing Data to the TDR Register

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.

32.18.7 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.

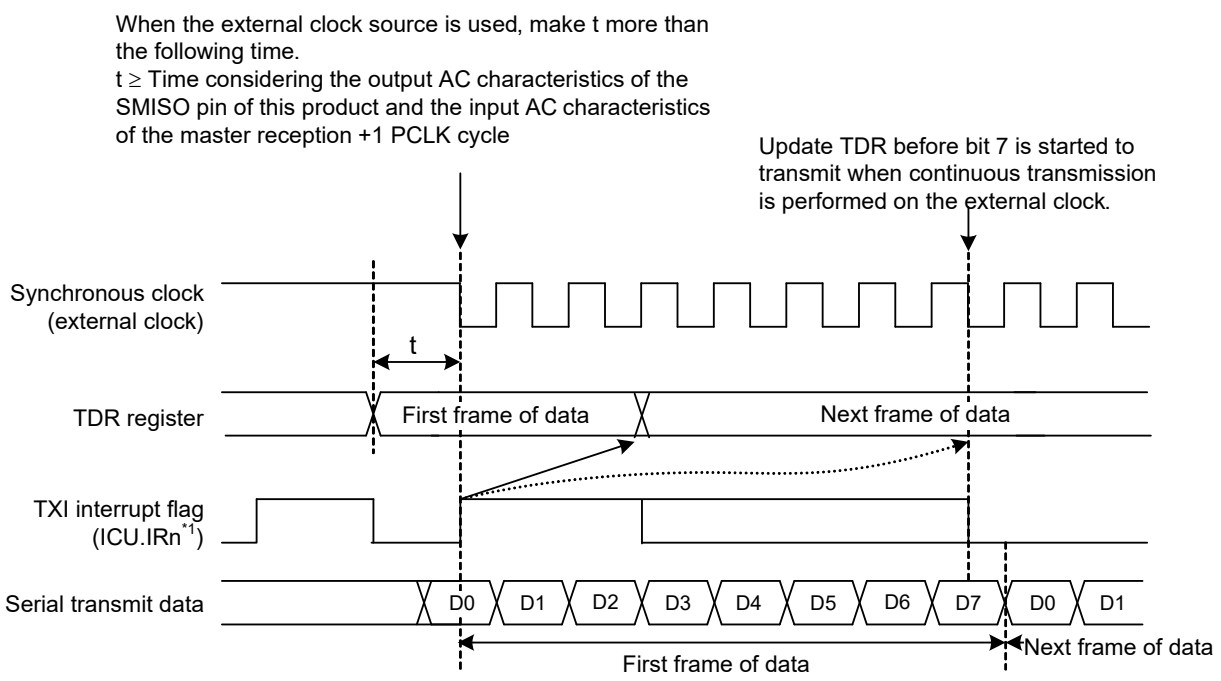
(1) Start of Transmission

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 32.108)
Time considering the output AC characteristics of the SMISO pin of this product and the input AC characteristics of the master reception + 1 PCLK cycle.

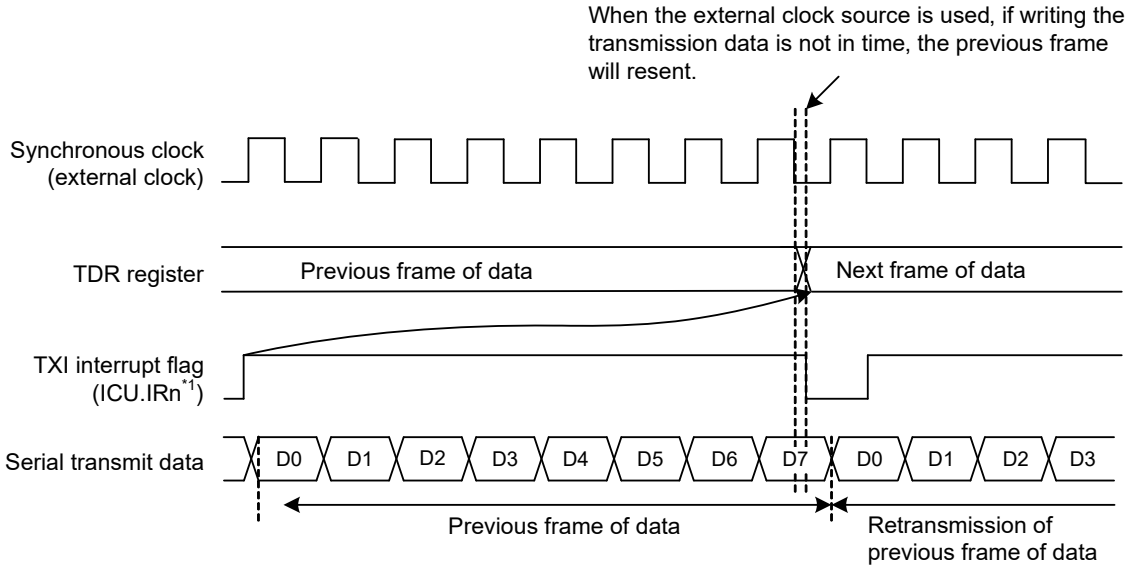
(2) Continuous Transmission

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 32.108)

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 (ICUb) for details on the corresponding interrupt vector number.

Figure 32.108 Restrictions on Use of External Clock in Clock Synchronous Transmission

32.18.8 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.

32.18.9 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 (ICUb).

- 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.

32.18.10 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 32.109. 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 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.

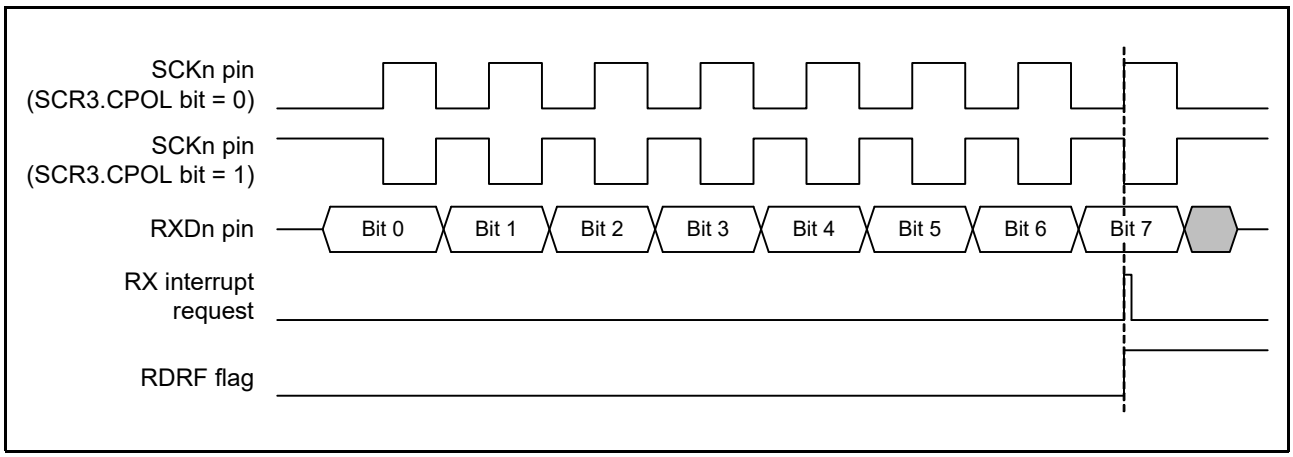


Figure 32.109 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- It takes “1 PCLK + 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.

32.18.11 Note on Transmit Enable Bit (TE Bit)

In the initial register value, when setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is "TXDn", output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(a) Connect a pull-up or pull-down resistor to the TXDn line.
(b) Set the SCR1 register to determine the level of the TXDn pin when the TE bit is 0.

32.18.12 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

32.18.13 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.

32.18.14 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).

32.18.15 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.

33. I2C-bus Interface (RIICa)

This MCU has a single-channel I2C-bus interface (RIIC0).
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.

33.1 Overview

Table 33.1 lists the specifications of the RIIC, Figure 33.1 shows a block diagram of the RIIC. Table 33.2 lists the I/O pins of the RIIC.

Table 33.1 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 33.1 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

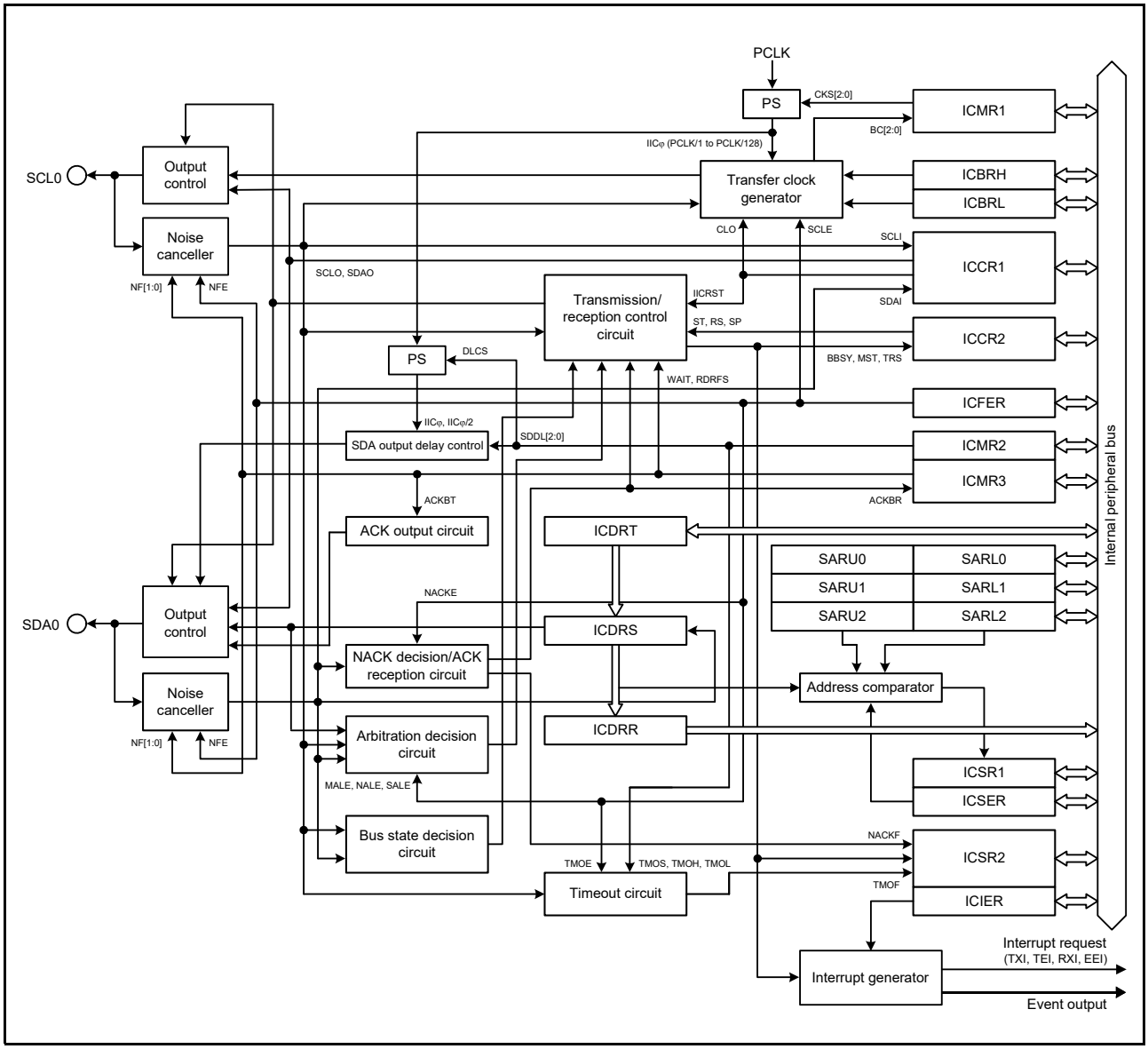


Figure 33.1 RIIC Block Diagram

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 33.2 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

33.2 Register Descriptions

33.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

	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: SDA0 line is low. 1: SDA0 line is high.	R
b1	SCLI	SCL Line Monitor	0: SCL0 line is low. 1: SCL0 line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDA0 pin low. 1: The RIIC has released the SDA0 pin. • Write: 0: The RIIC drives the SDA0 pin low. 1: The RIIC releases the SDA0 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 SCL0 pin low. 1: The RIIC has released the SCL0 pin. • Write: 0: The RIIC drives the SCL0 pin low. 1: The RIIC releases the SCL0 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 SCL0/SDA0 output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 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 SDA0 and SCL0 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 33.11.2, 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 33.3 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 33.14, 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 SCL0 pin and SDA0 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 SCL0 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 33.3 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 SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 33.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 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 SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 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.

33.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

	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 33.10, 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 33.10, 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 33.10, 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 SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been generated.
The RIIC recognizes the SDA0 line changing from low to high while the SCL0 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)

33.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h

	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 SCL0 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 SCL0 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.

33.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h

	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 SCL0 line is low. 1: Count-up is enabled while the SCL0 line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCL0 line is high. 1: Count-up is enabled while the SCL0 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 SCL0 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 33.11.1, 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 SCL0 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 SCL0 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 33.5, 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))

33.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h

	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 SCL0 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 SCL0 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 33.6, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL0 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 SCL0 line low at the falling edge of the eighth SCL.
When the RDRFS bit is 0, the SCL0 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 SCL0 line is held low at the falling edge of the eighth SCL. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCL0 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 SCL0 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.

33.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

	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 33.11.1, 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 33.8.2, 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 SCL0 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.

33.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

	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 33.7.3, 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.

33.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h

	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.

33.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

	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

33.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h

	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 SCL0 line state remains unchanged for a certain period.
[Setting condition]

- When the SCL0 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 SDA0 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 SDA0 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 SDA0 line is driven low while the internal SDA output is high (the SDA0 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 SDA0 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 SDA0 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 SDA0 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 33.4 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 SDA0 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

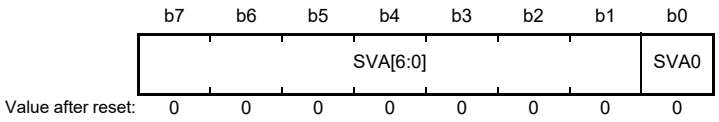
[Clearing conditions]

- 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.

33.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh



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.

33.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh

	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.

33.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low 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 33.8, 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))

33.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h

	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 + SCL0 line rise time [tr] + SCL0 line fall time [tf]}
Duty cycle = {SCL0 line rise time [tr]*2 + (ICBRH + 1) / IICφ} / {SCL0 line fall time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCL0 line rise time [tr] and SCL0 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 33.5 lists examples of ICBRH/ICBRL settings.

Table 33.5 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		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)
100	011b	14 (EEh)	17 (F1h)	011b	15 (EFh)	18 (F2h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL0 line rise time (tr): 100 kbps or less (Sm): 1000 ns, 400 kbps or less (Fm): 300 ns
SCL0 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 SCL0 signal, refer to the I2C-bus specification from NXP Semiconductors.

33.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h

	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.

33.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h

	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.

33.2.17 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 SDA0 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.

33.3 Operation

33.3.1 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 33.2 shows the I2C-bus format, and Figure 33.3 shows the I2C-bus timing.

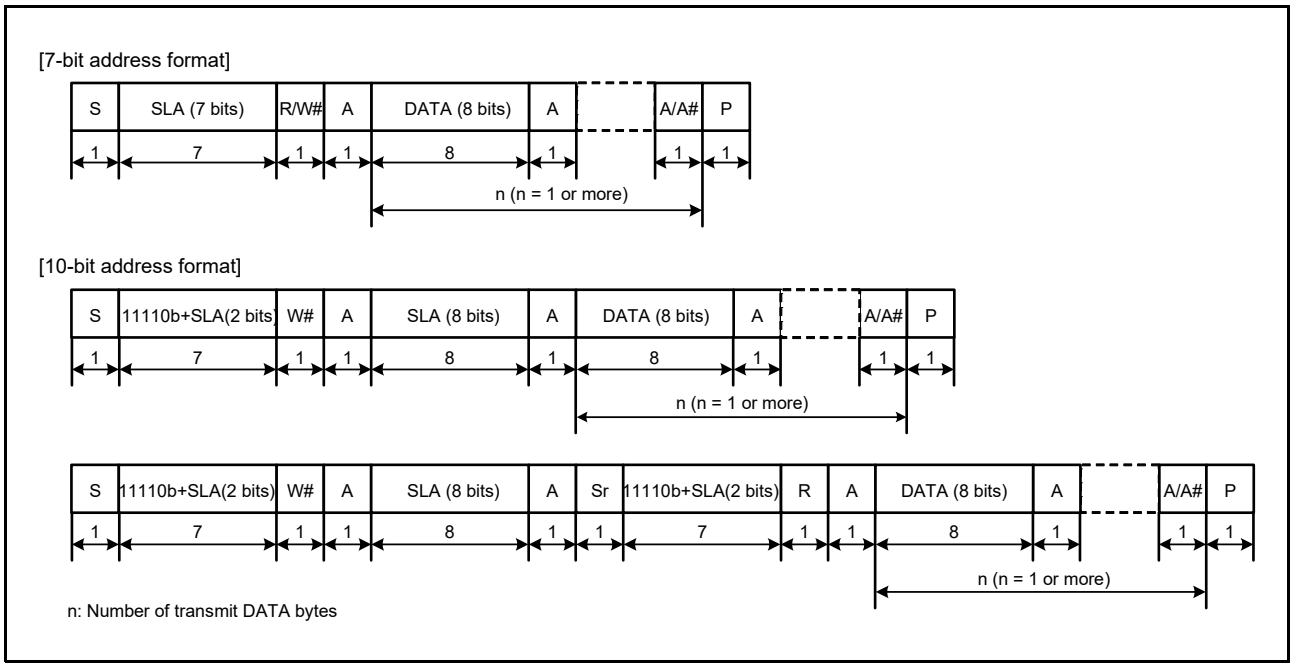


Figure 33.2 I2C-bus Format

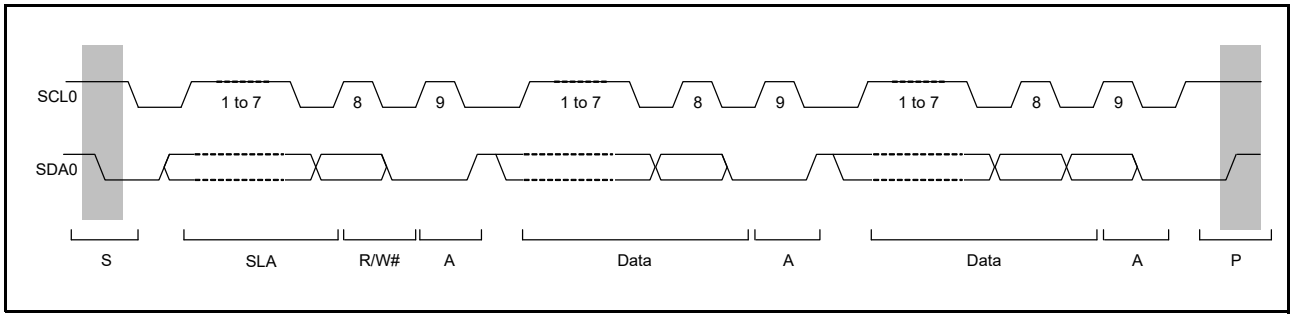


Figure 33.3 I2C-bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA0 line low from high while the SCL0 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 SDA0 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 SDA0 line high.

Sr: Restart condition. The master device drives the SDA0 line low from high after the setup time has elapsed with the SCL0 line high.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA0 line high from low while the SCL0 line is high.

33.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 33.4.
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 (SCL0 and SDA0 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 33.4). 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.

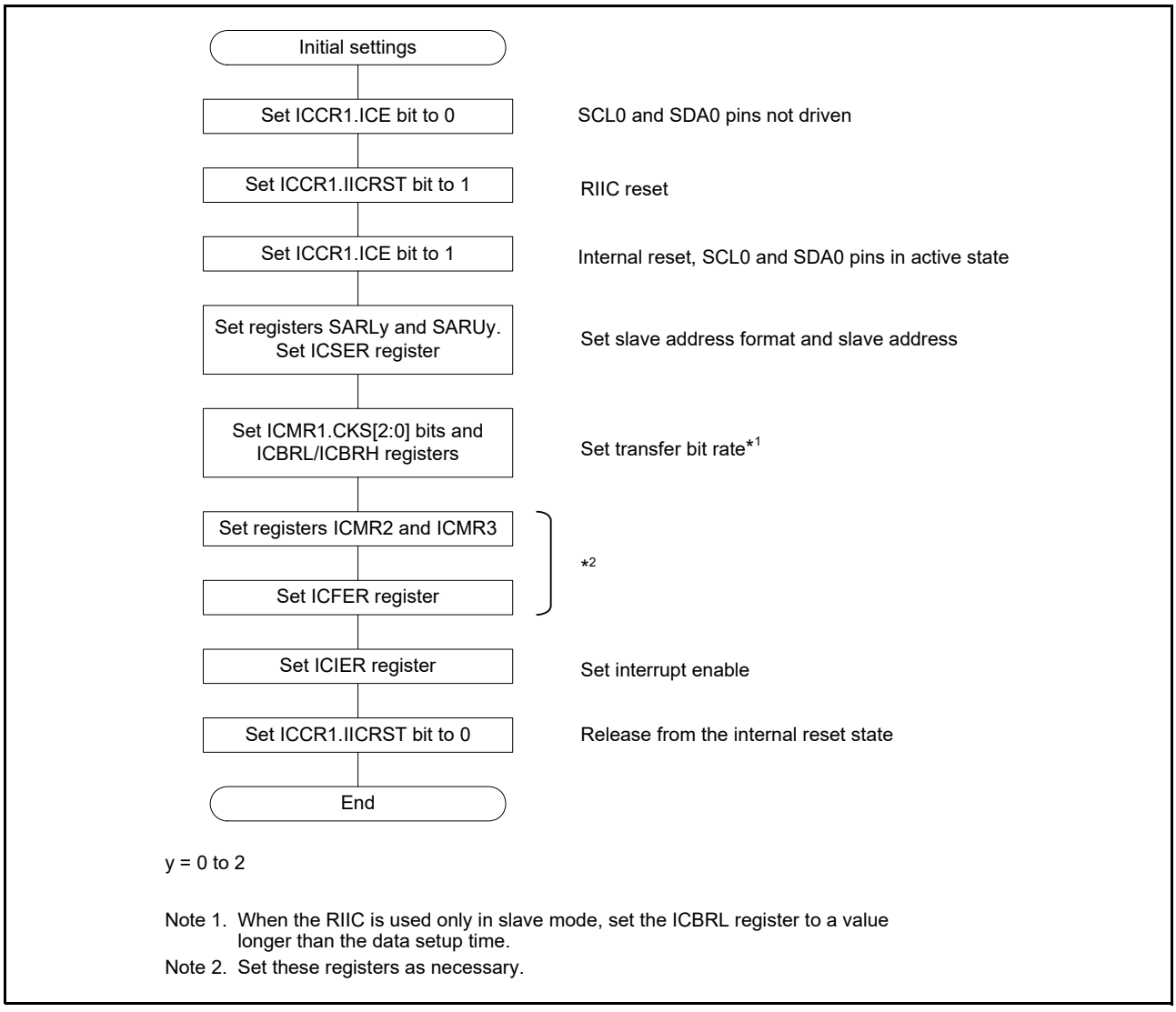


Figure 33.4 Example of RIIC Initialization Flowchart

33.3.3 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 33.5 shows an example of usage of master transmission and Figure 33.6 to Figure 33.8 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 33.3.2, 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 SDA0 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 SCL0 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.

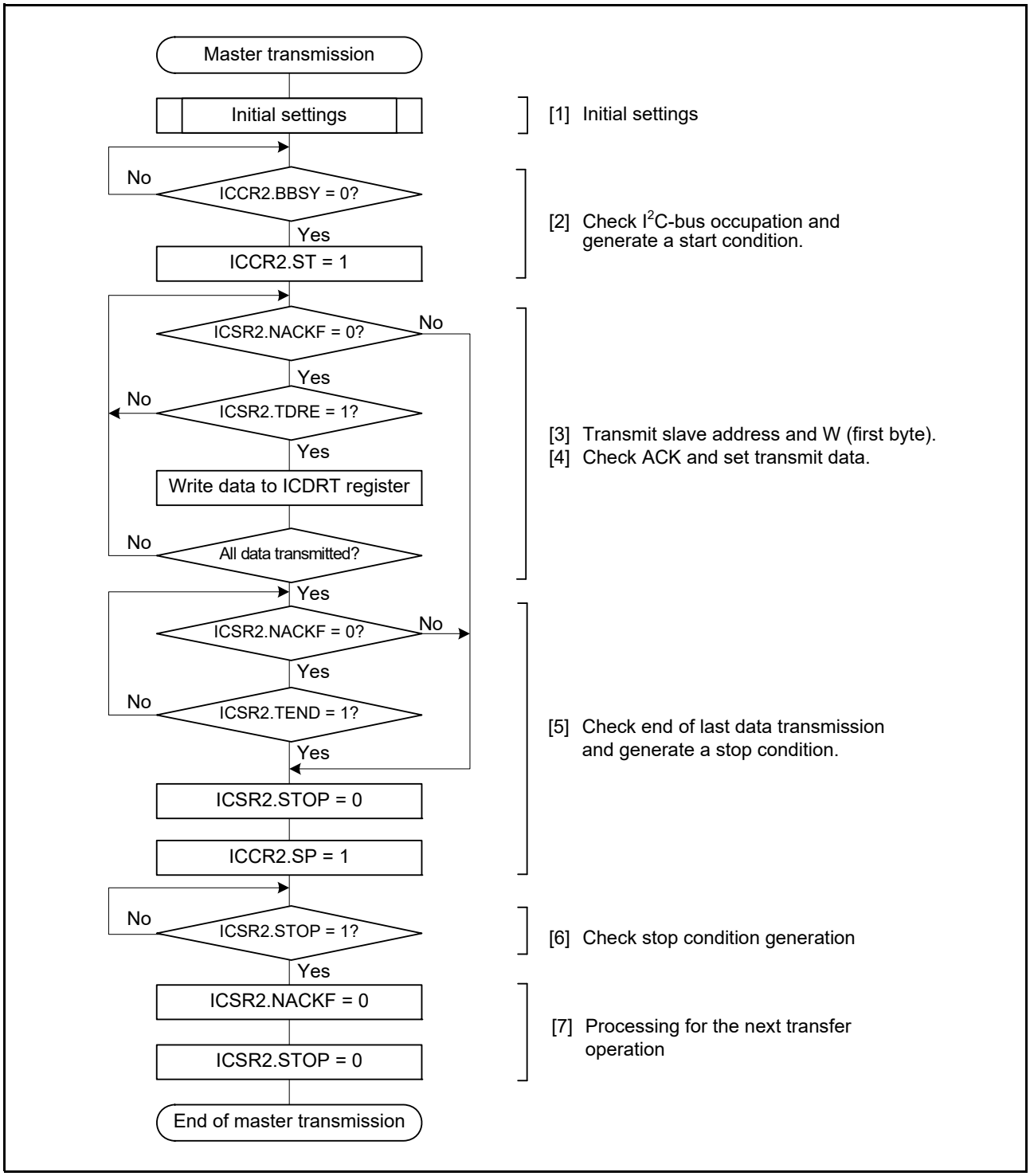


Figure 33.5 Example of Master Transmission Flowchart

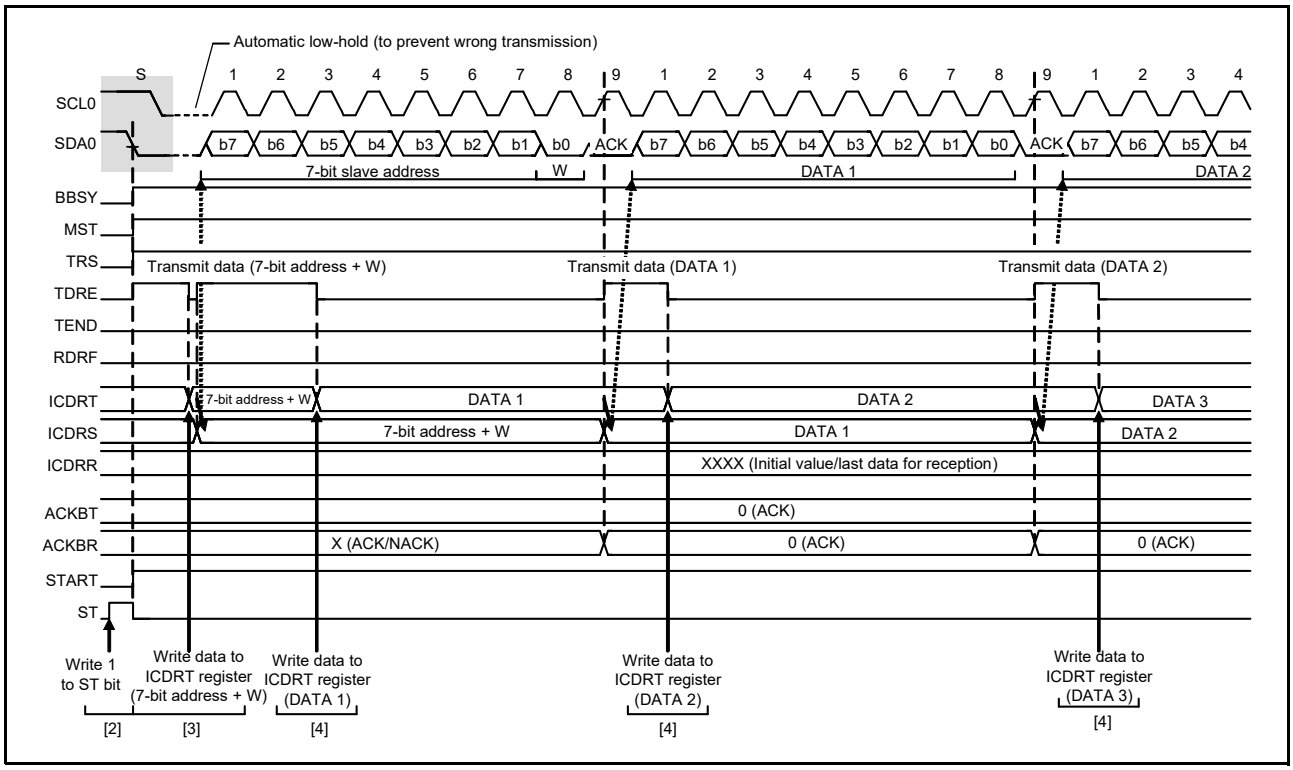


Figure 33.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

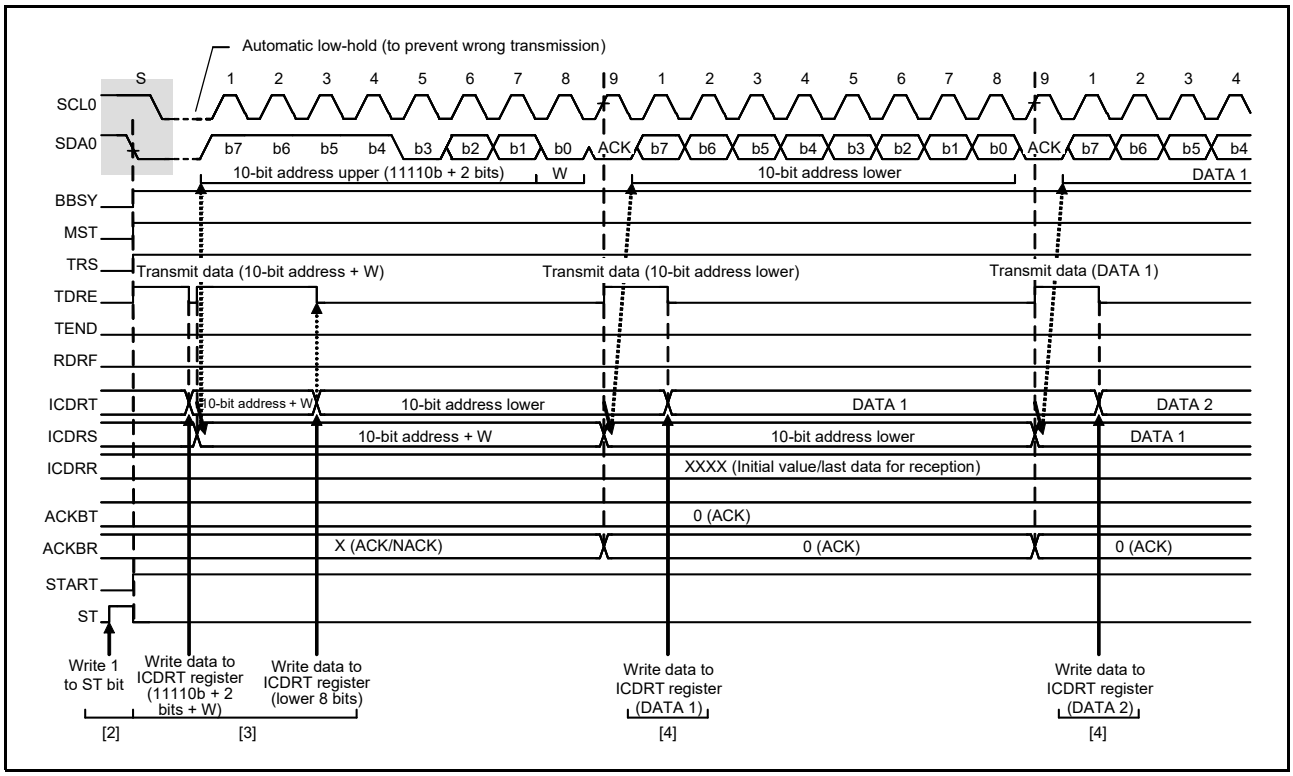


Figure 33.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

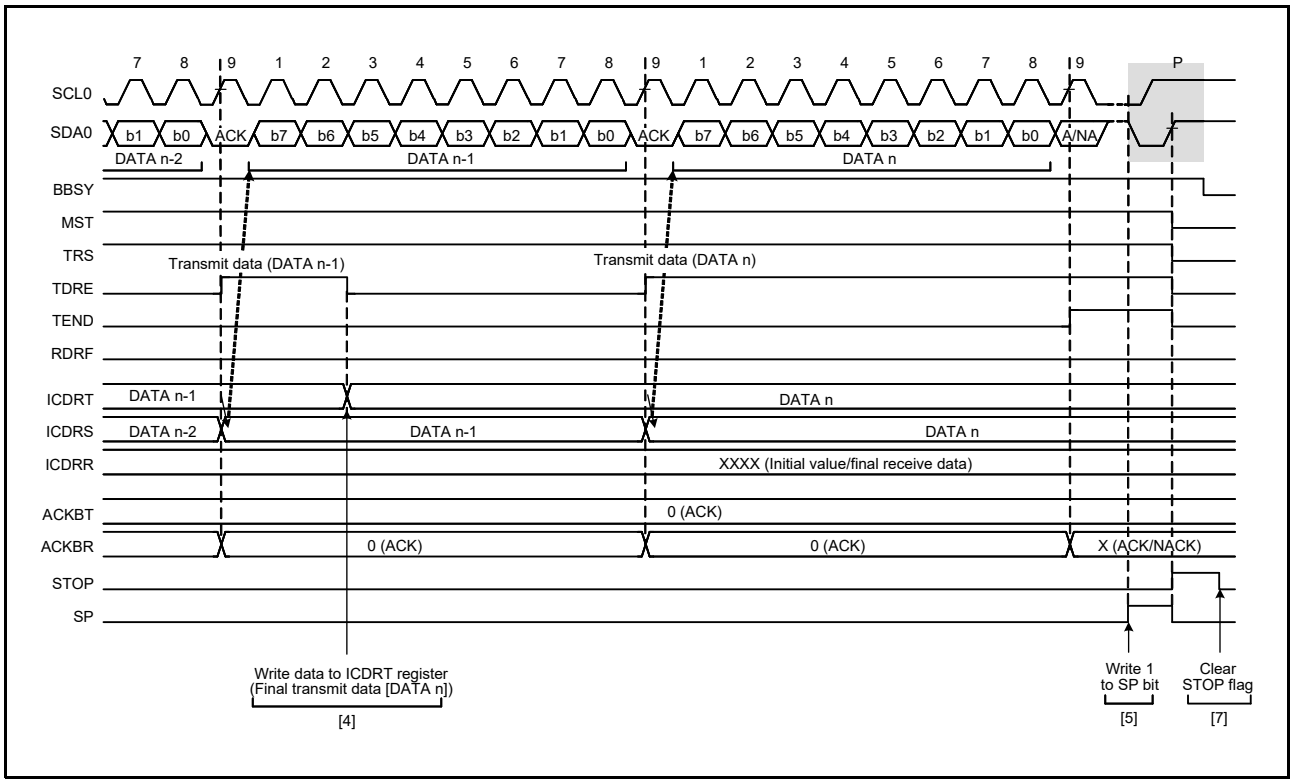


Figure 33.8 Master Transmit Operation Timing (3)

33.3.4 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 33.9 and Figure 33.10 show examples of usage of master reception (7-bit address format) and Figure 33.11 to Figure 33.13 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 33.3.2, 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 SDA0 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 SCL0 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 SCL0 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.

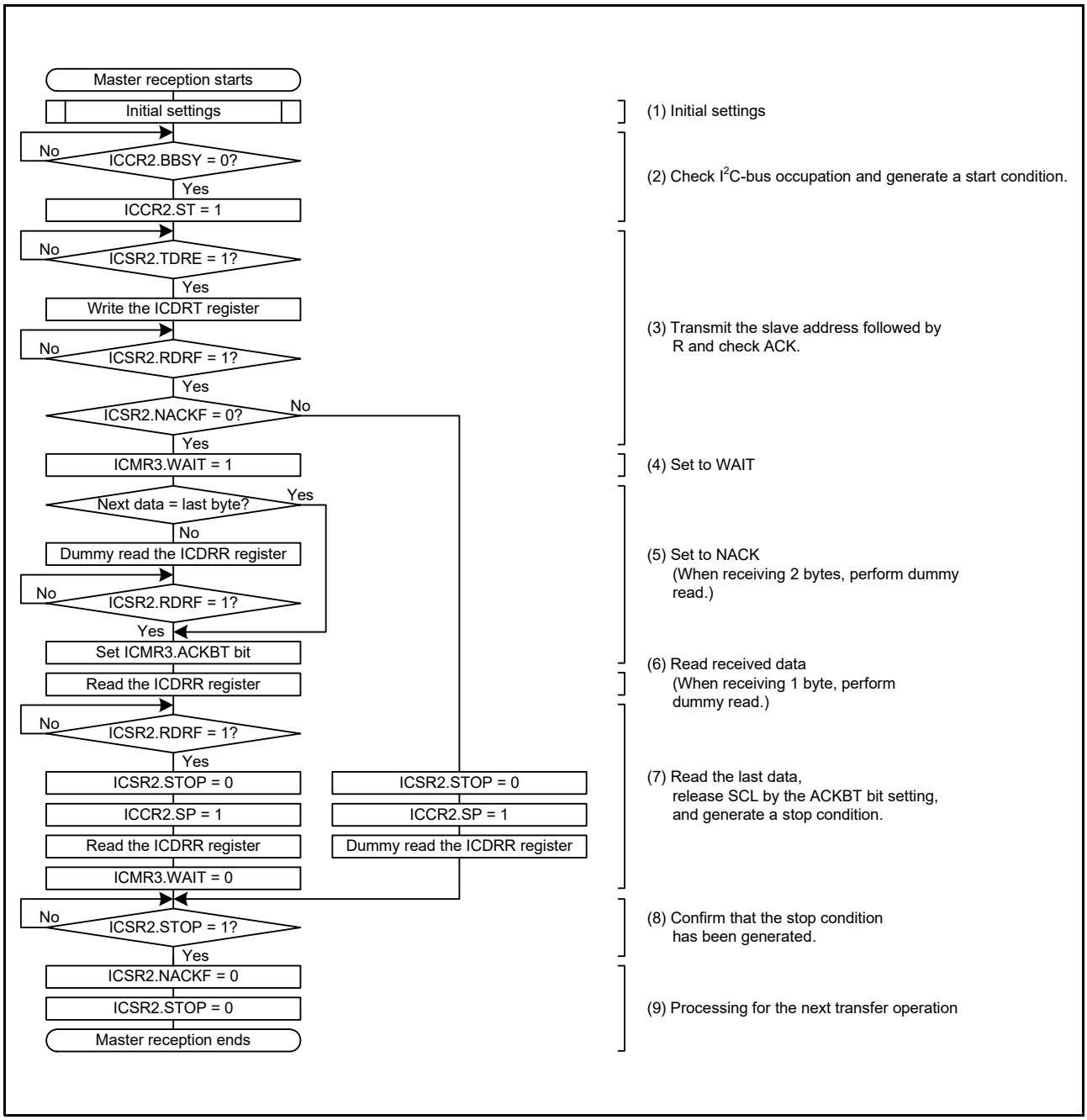


Figure 33.9 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

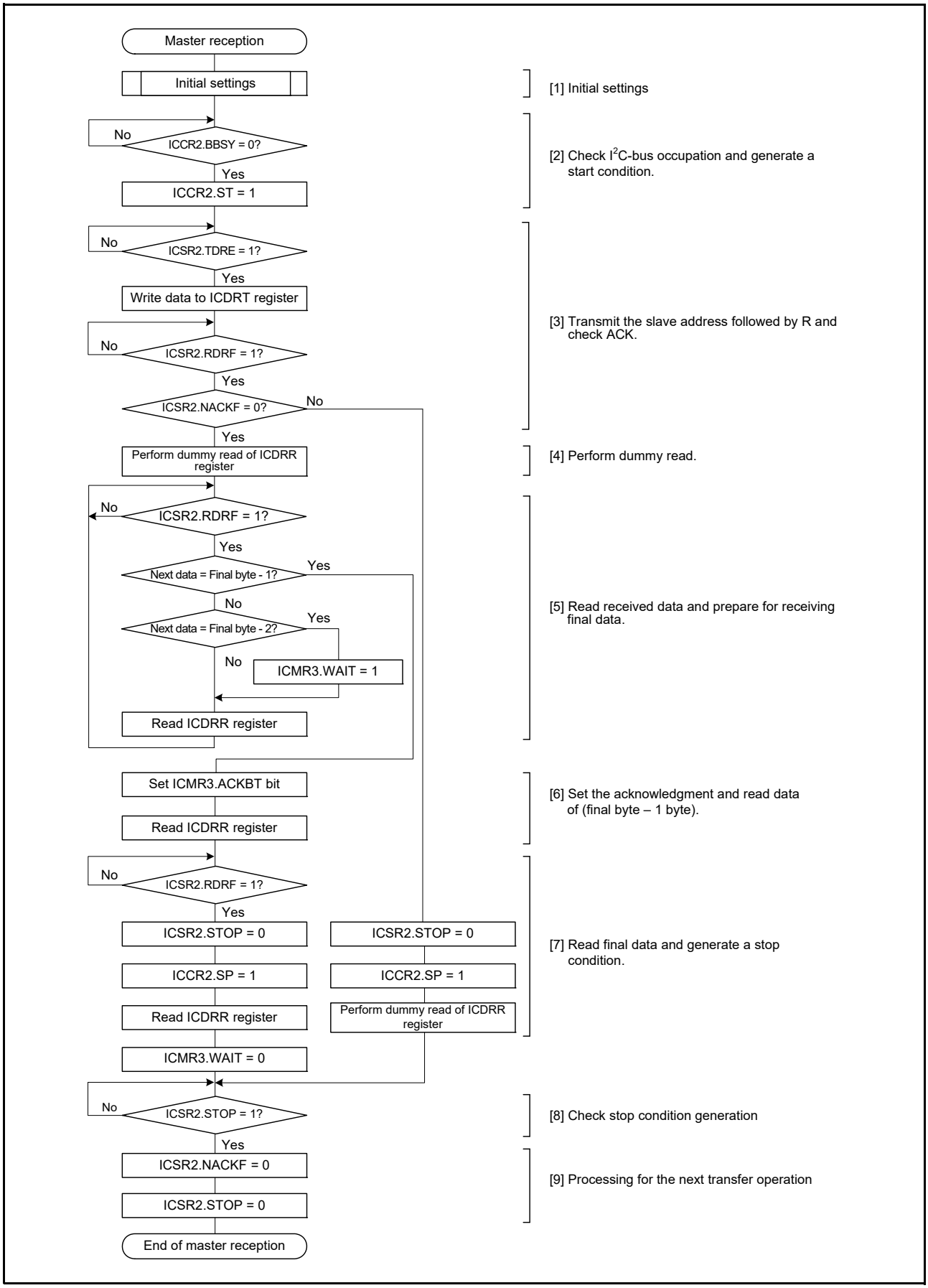


Figure 33.10 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

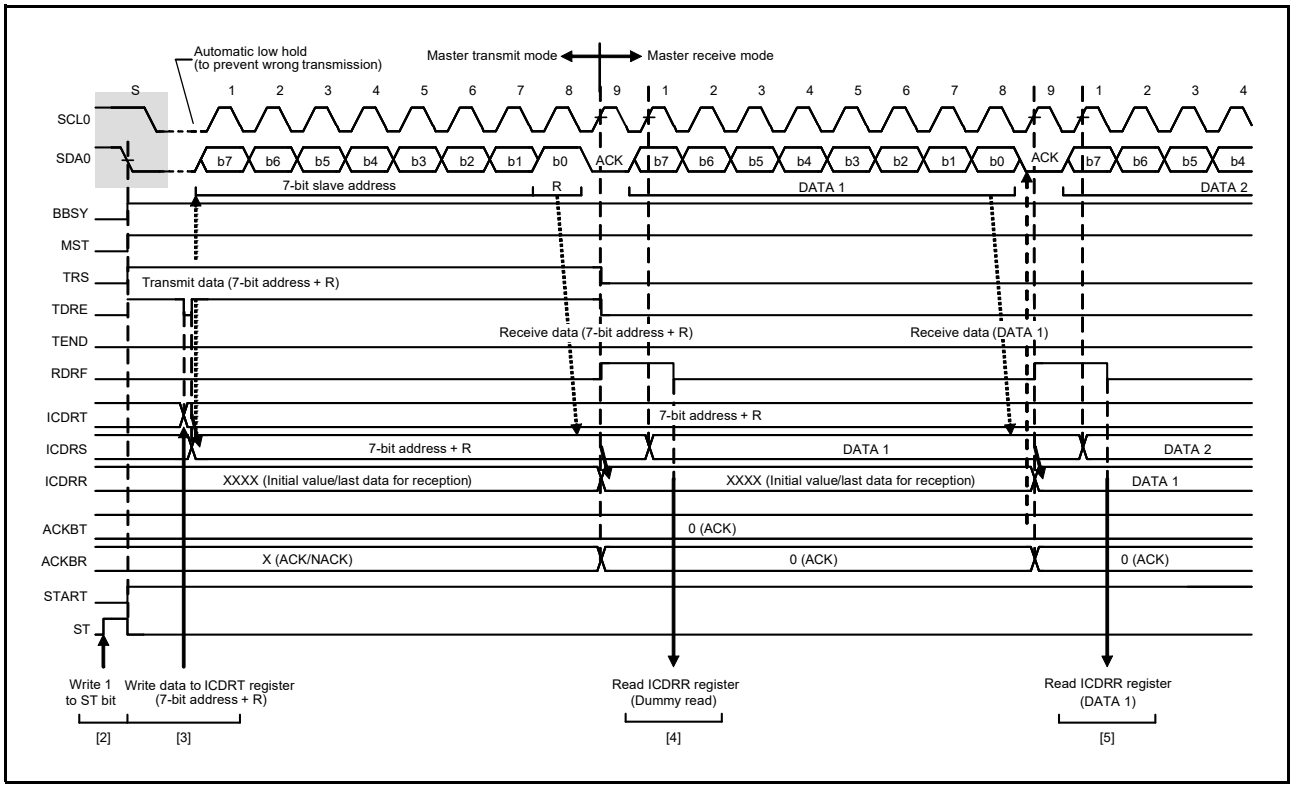


Figure 33.11 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

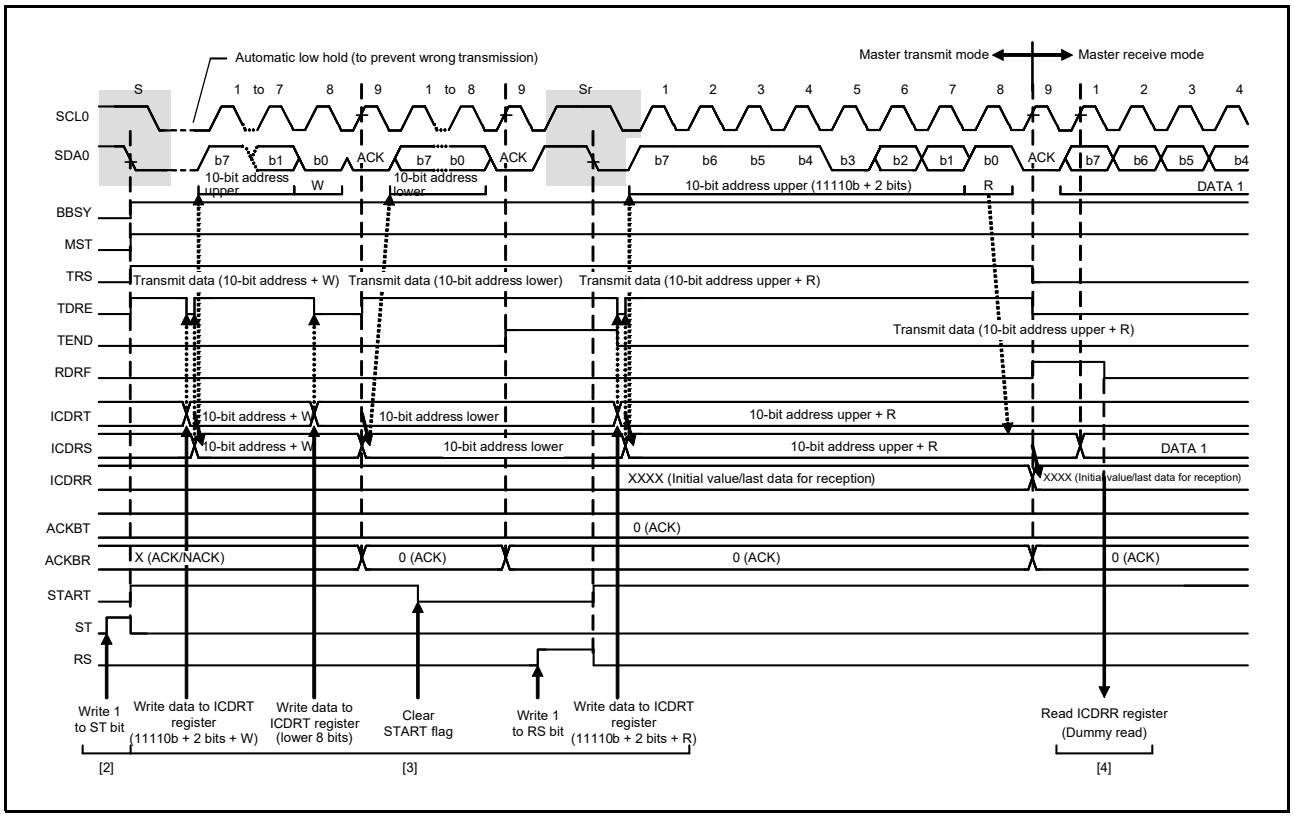


Figure 33.12 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

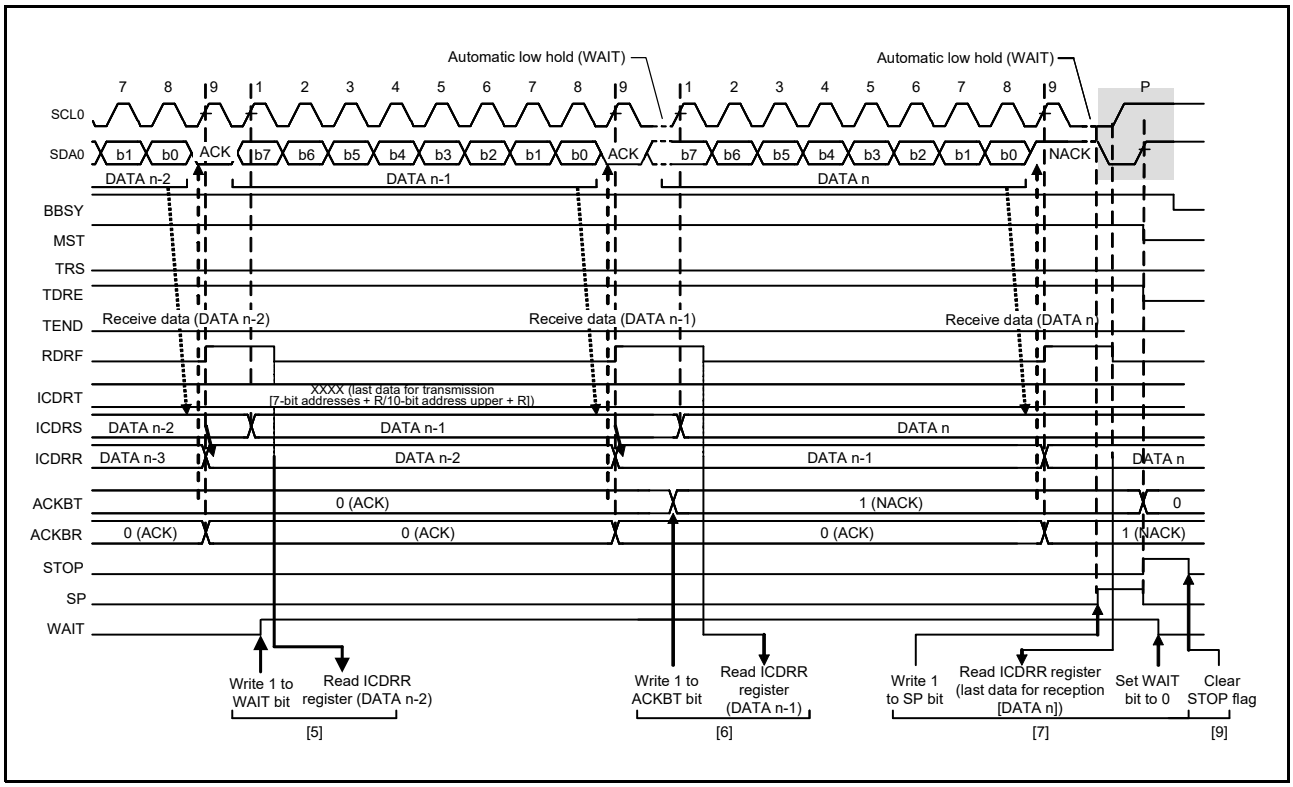


Figure 33.13 Master Receive Operation Timing (3) (When RDRFS bit is 0)

33.3.5 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 33.14 shows an example of usage of slave transmission and Figure 33.15 and Figure 33.16 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 33.3.2, 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 SCL0 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 SCL0 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.

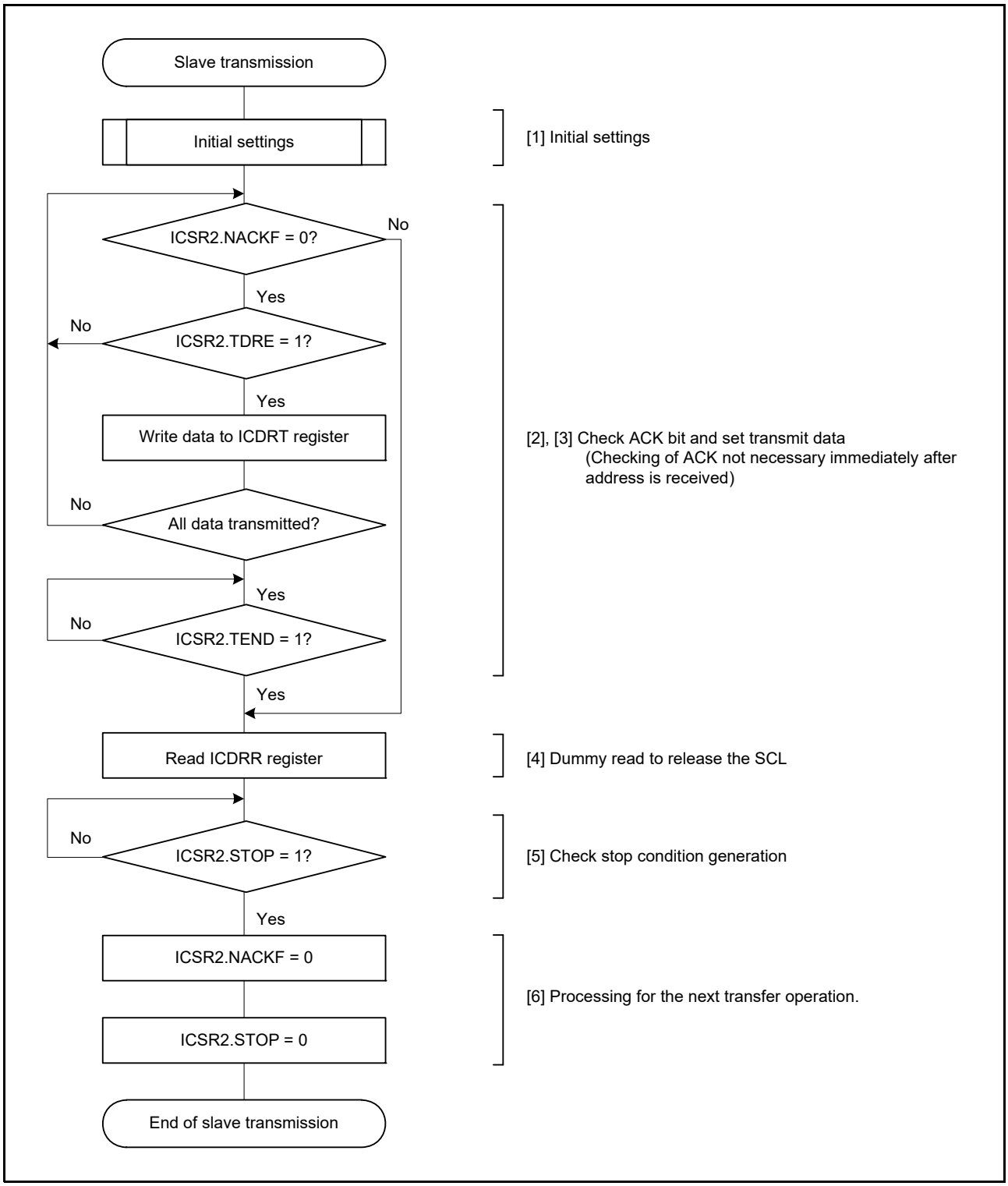


Figure 33.14 Example of Slave Transmission Flowchart

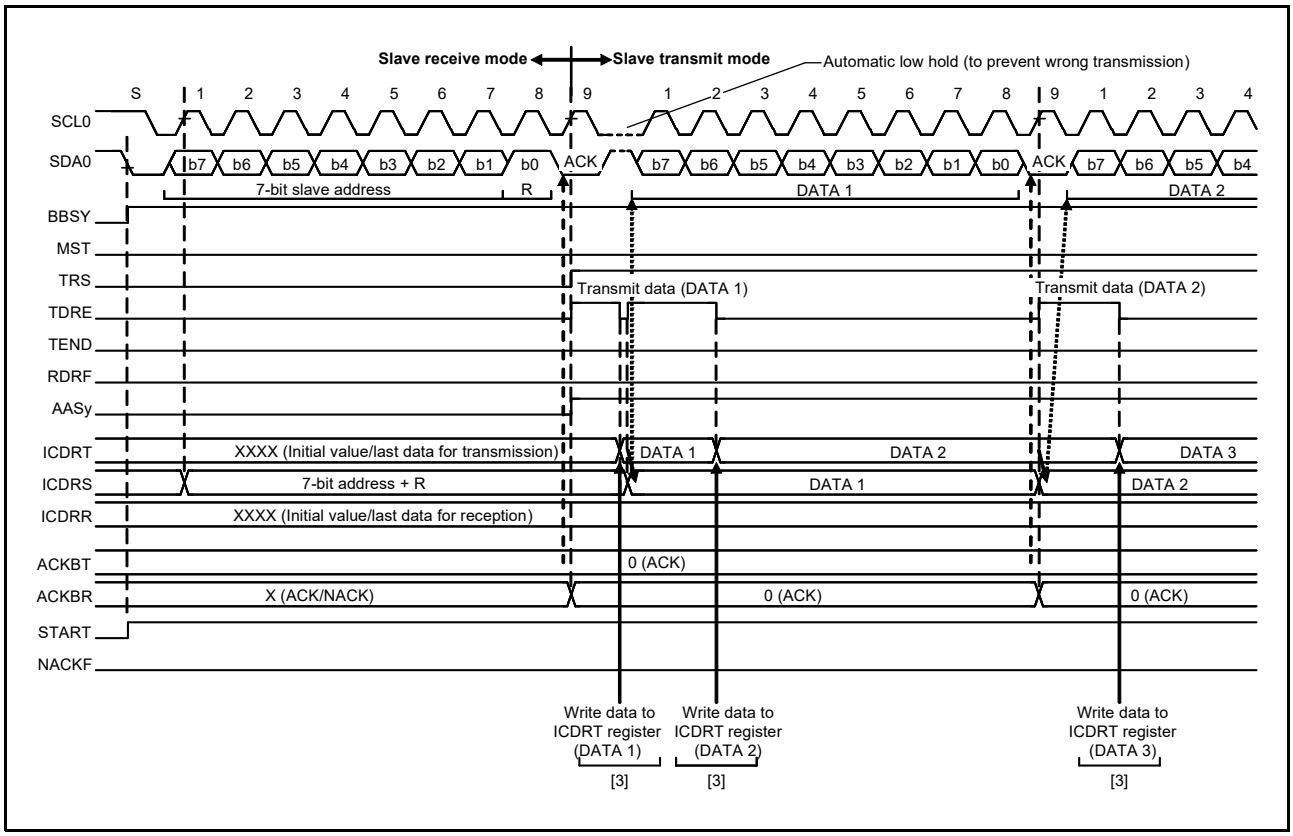


Figure 33.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

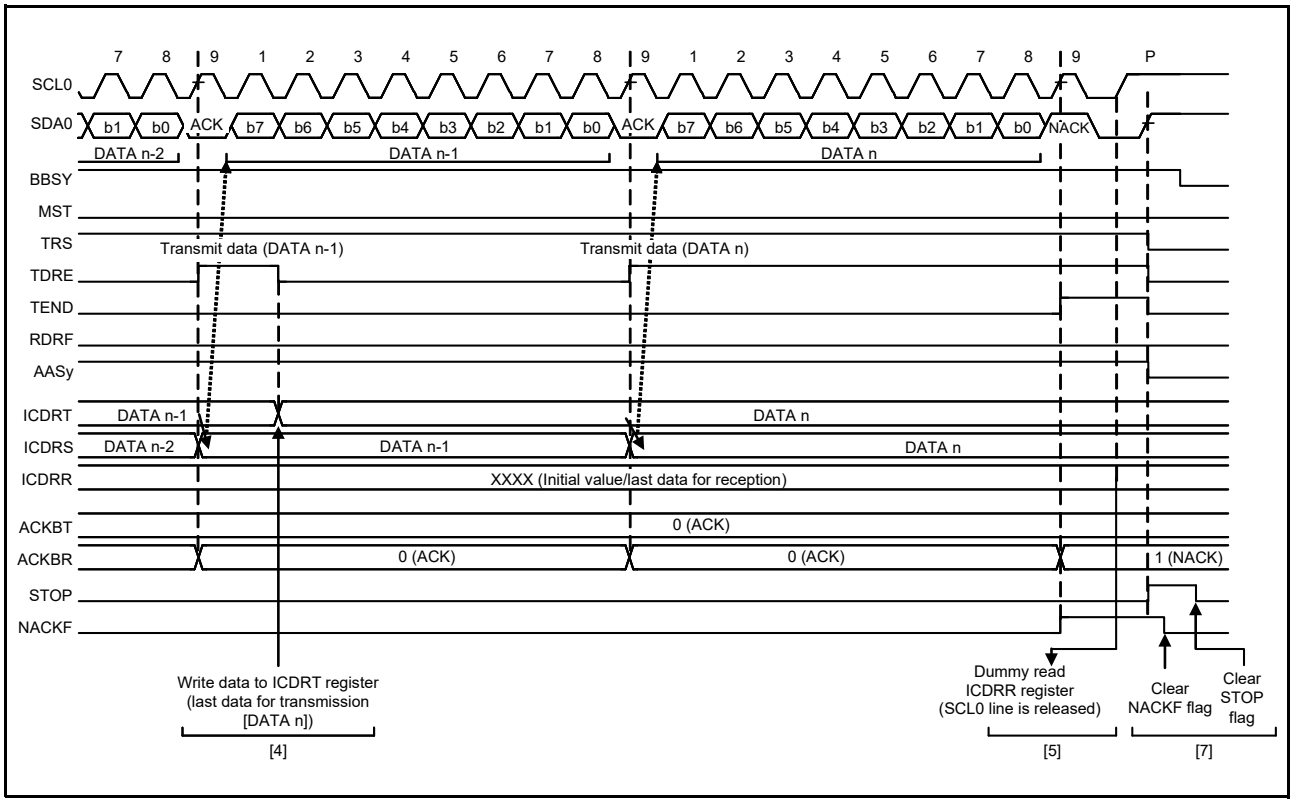


Figure 33.16 Slave Transmit Operation Timing (2)

33.3.6 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 33.17 shows an example of usage of slave reception and Figure 33.18 and Figure 33.19 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 33.3.2, 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 SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCL0 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.

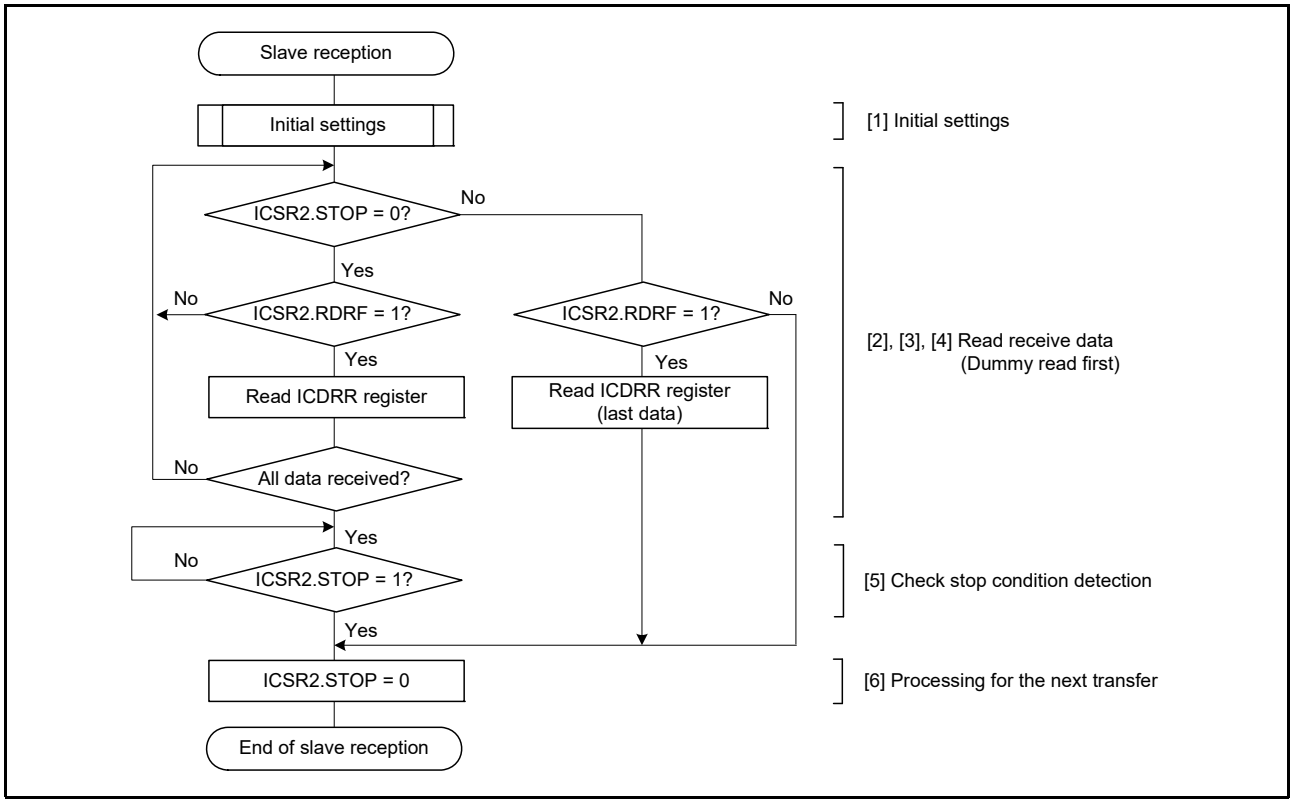


Figure 33.17 Example of Slave Reception Flowchart

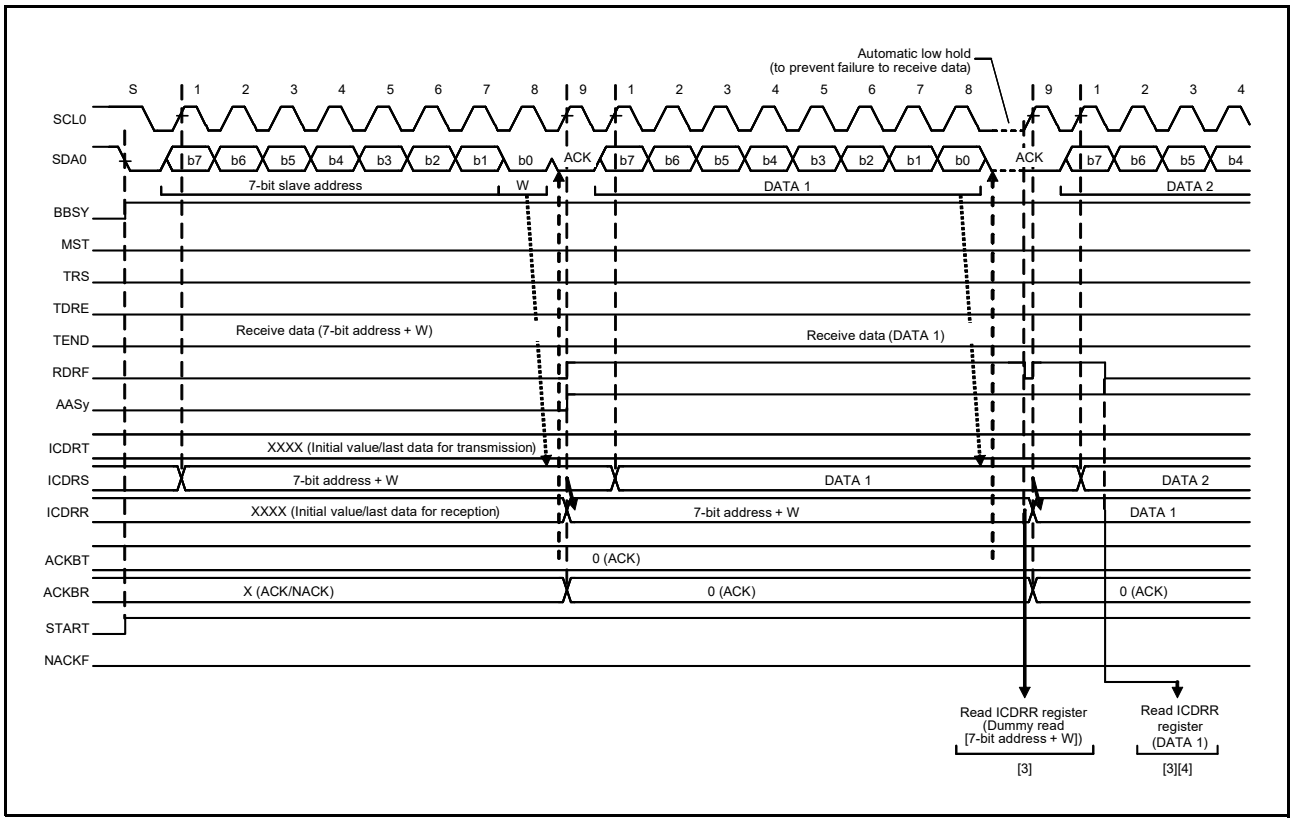


Figure 33.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

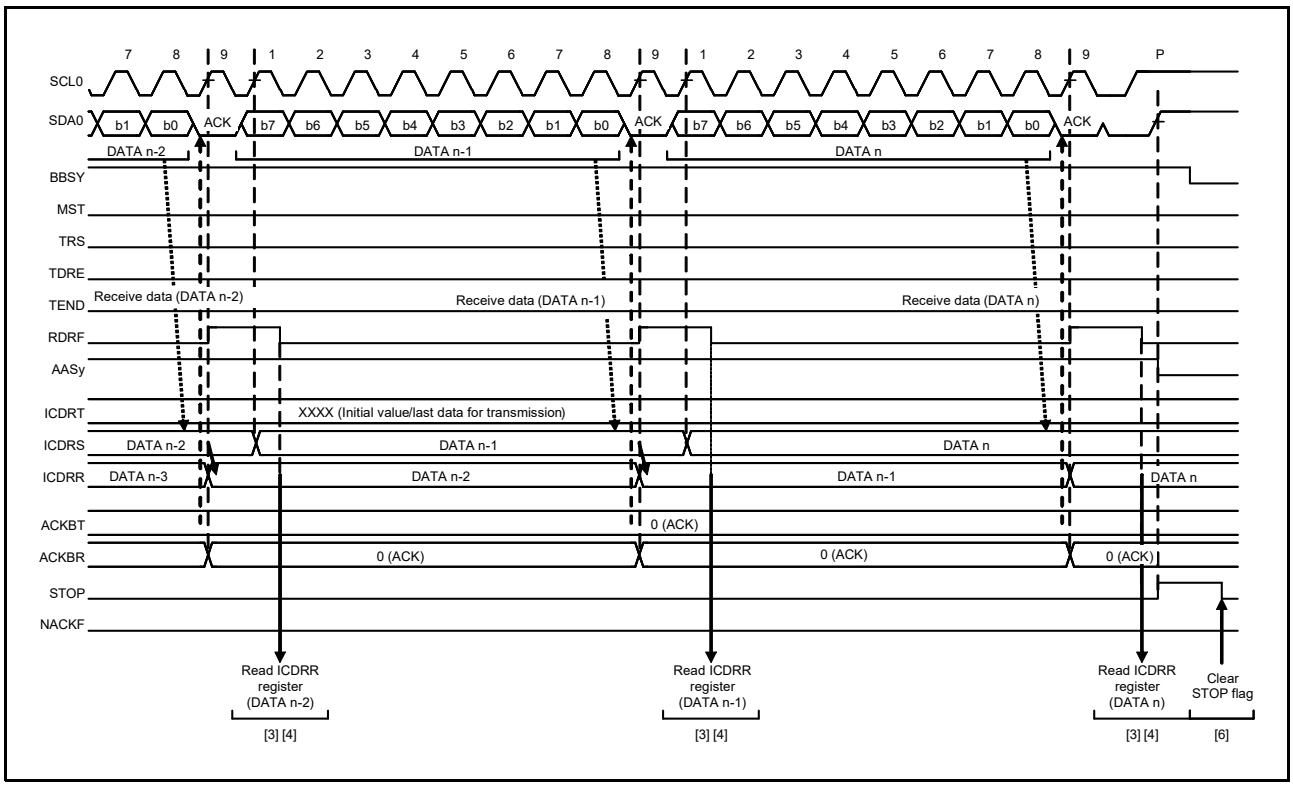


Figure 33.19 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

33.4 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 SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low period specified in the ICBRL register, and then stops driving the SCL0 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 SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCL0 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 SCL0 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 SCL0 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 SCL0 line has been released. When the RIIC finishes outputting the low period of the SCL, the SCL0 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.

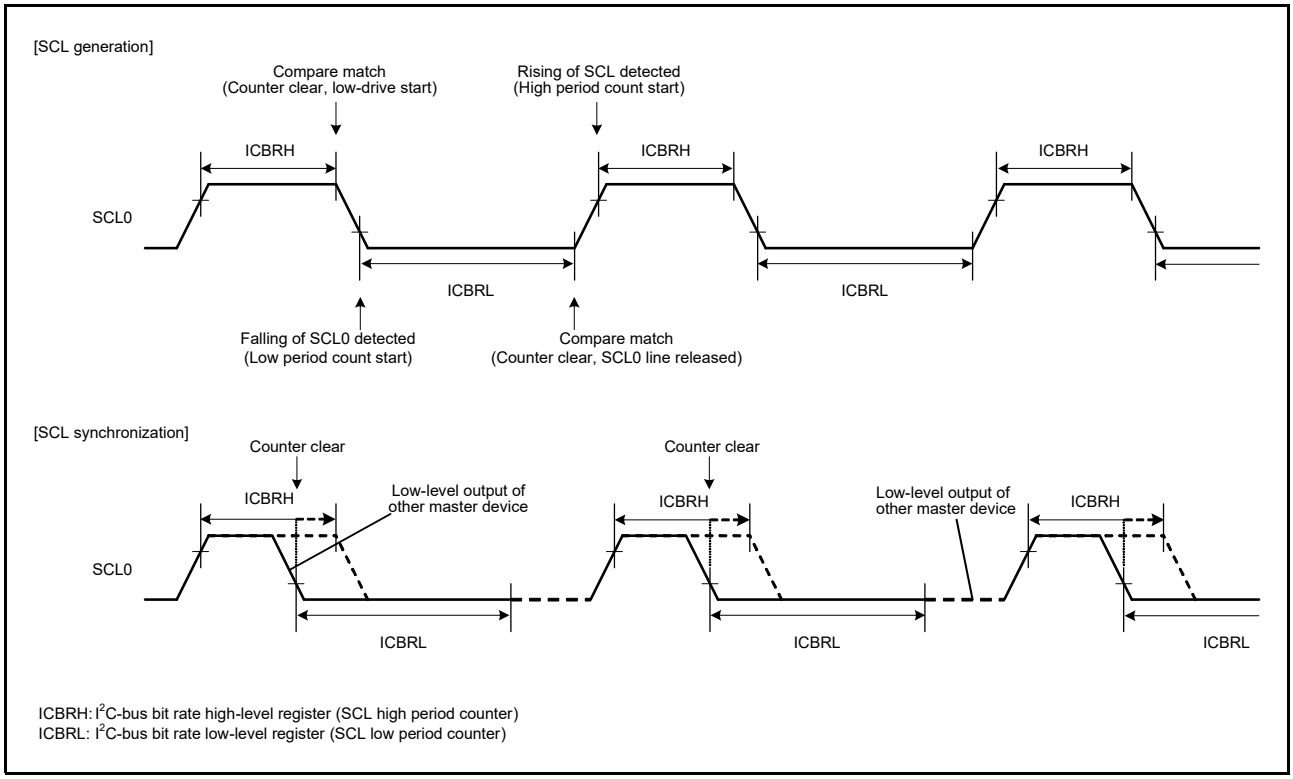


Figure 33.20 Generation and Synchronization of the SCL Signal from the RIIC

33.5 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).

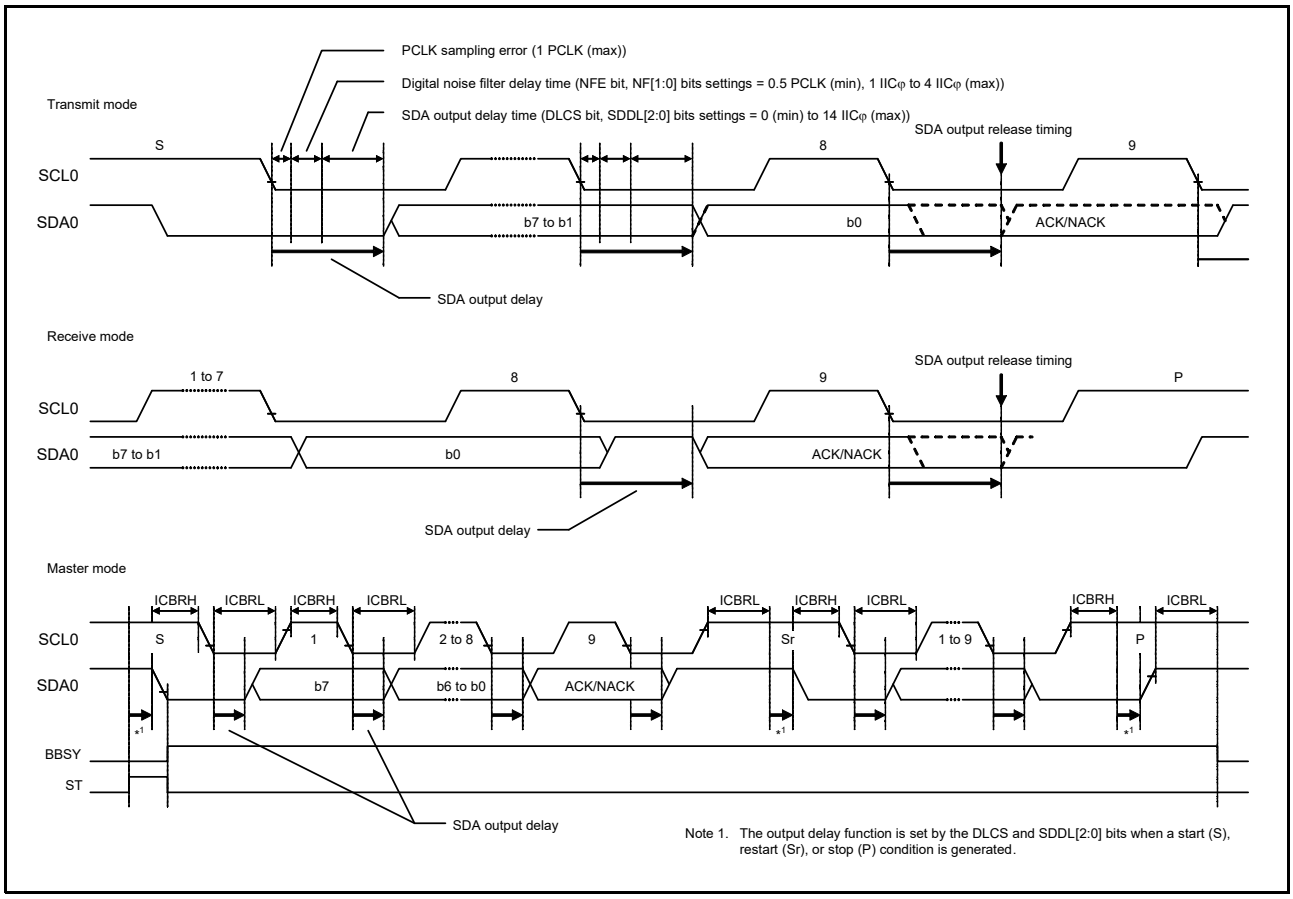


Figure 33.21 SDA Output Delay Function

33.6 Digital Noise Filters

The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through digital noise filters. Figure 33.22 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 SCL0 pin (or SDA0 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.

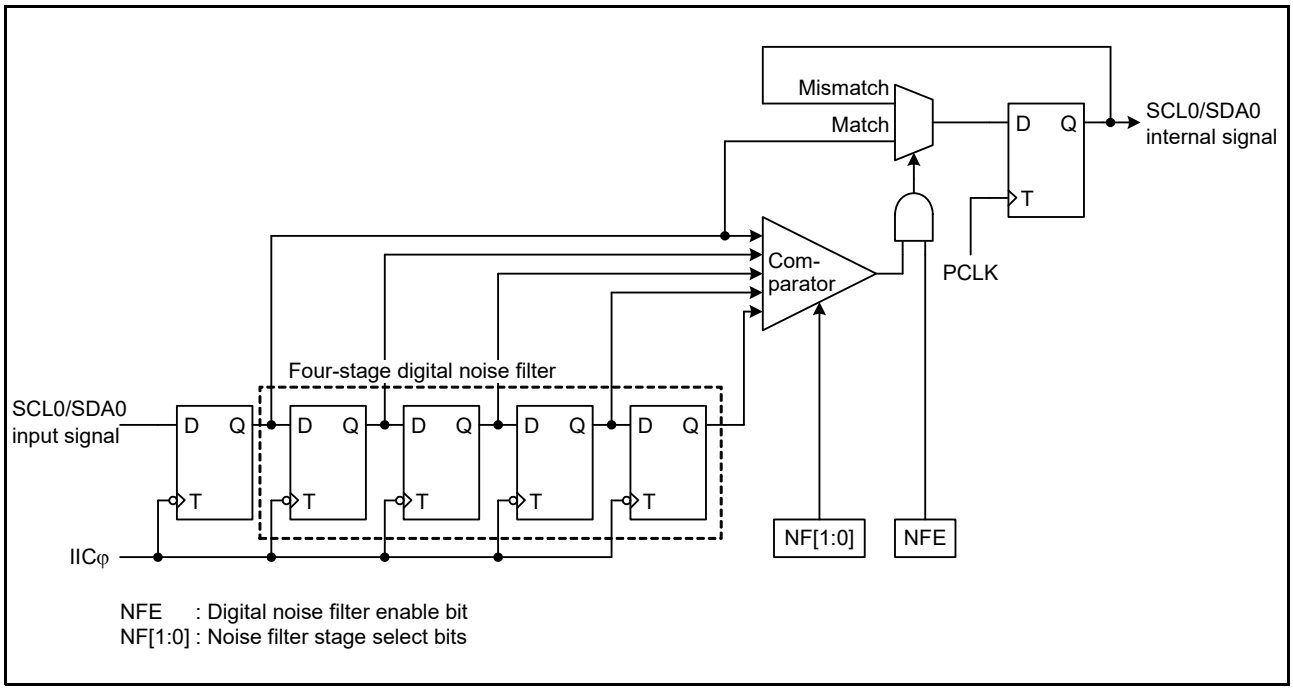


Figure 33.22 Block Diagram of the Digital Noise Filter

33.7 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.

33.7.1 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 33.23 to Figure 33.25 show the AASy flag set timing in three cases.

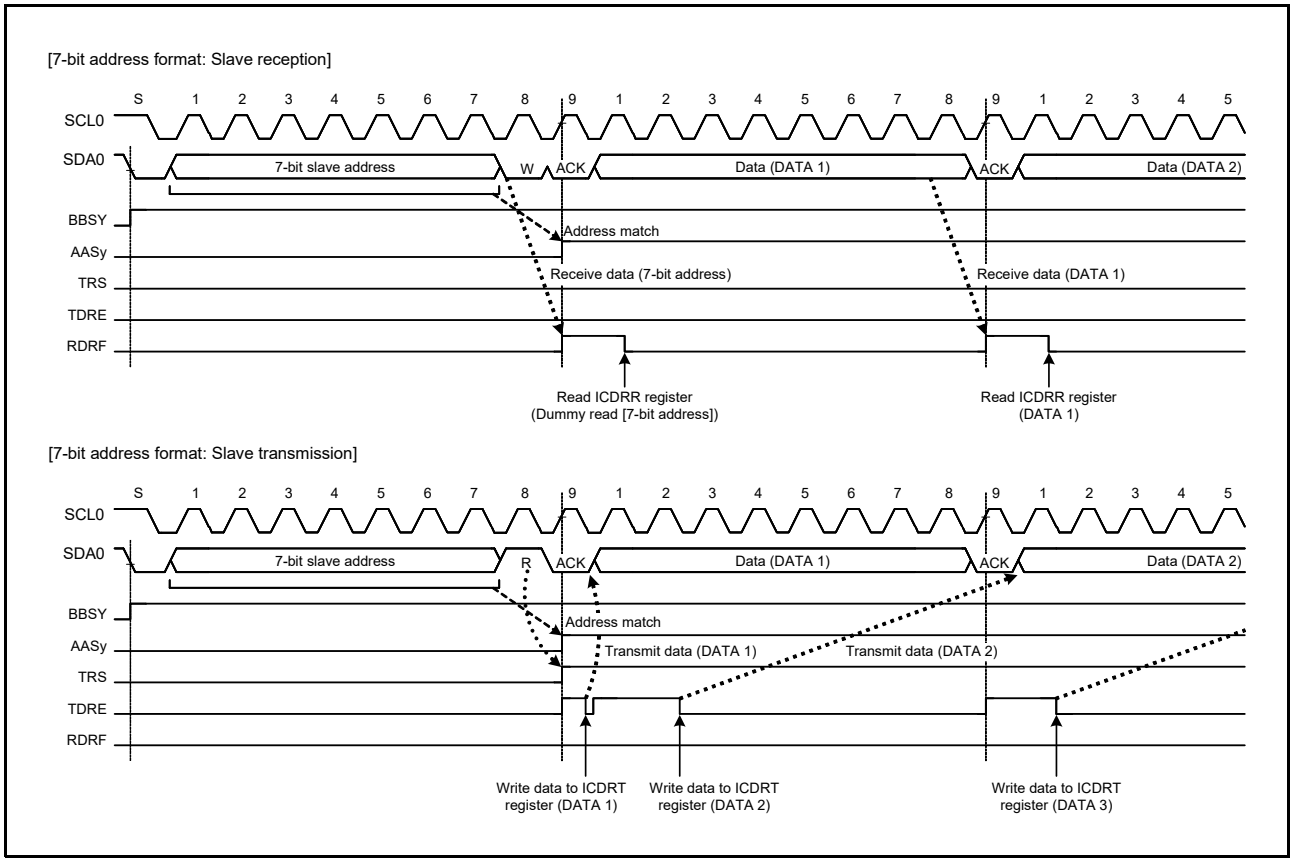


Figure 33.23 AASy Flag Set Timing with 7-Bit Address Format Selected

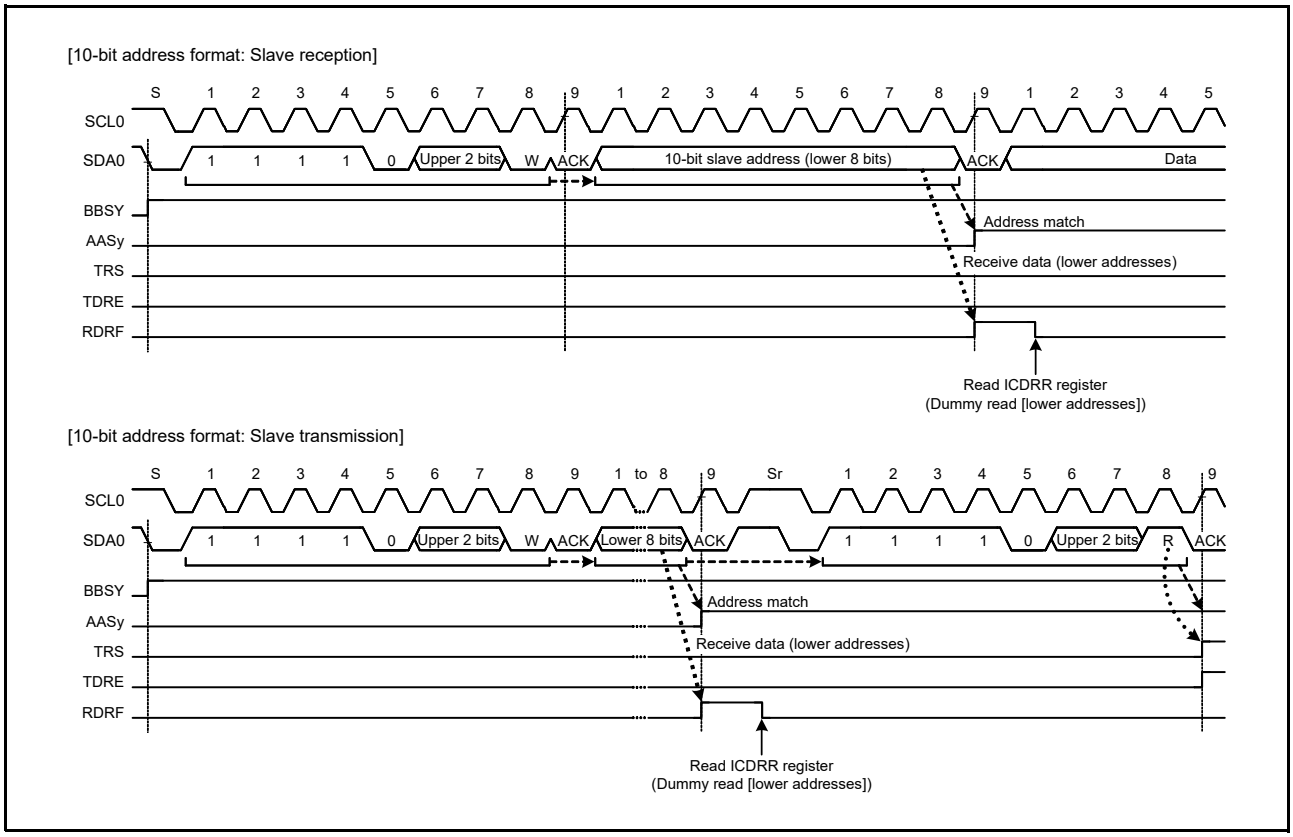


Figure 33.24 AASy Flag Set Timing with 10-Bit Address Format Selected

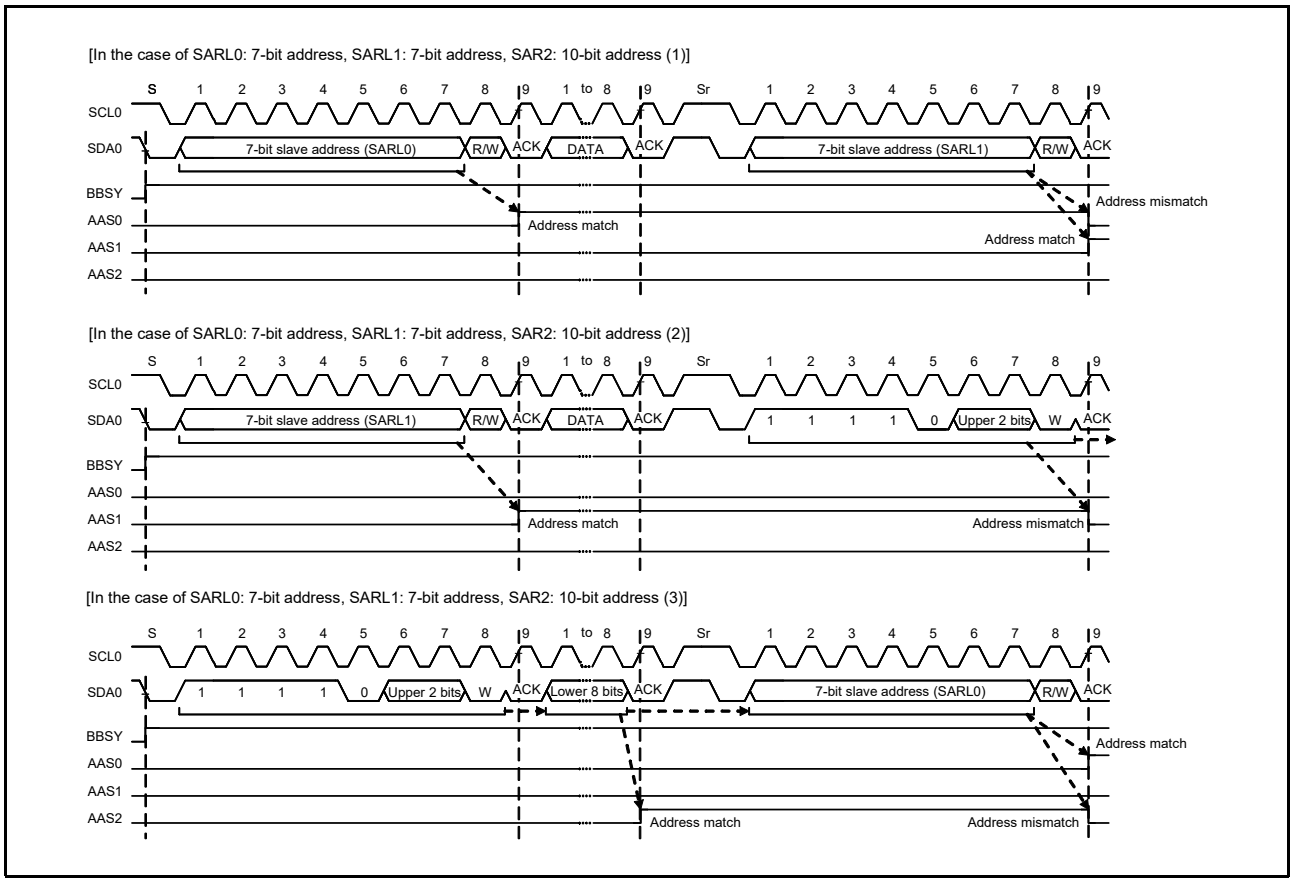


Figure 33.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

33.7.2 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.

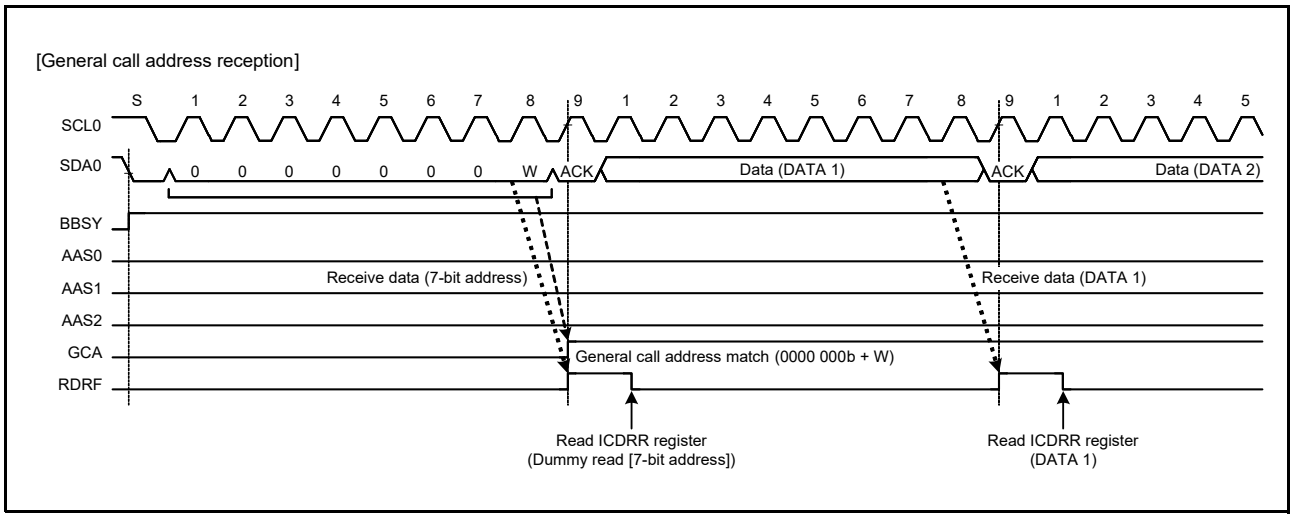


Figure 33.26 Timing of GCA Flag Setting during Reception of General Call Address

33.7.3 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.

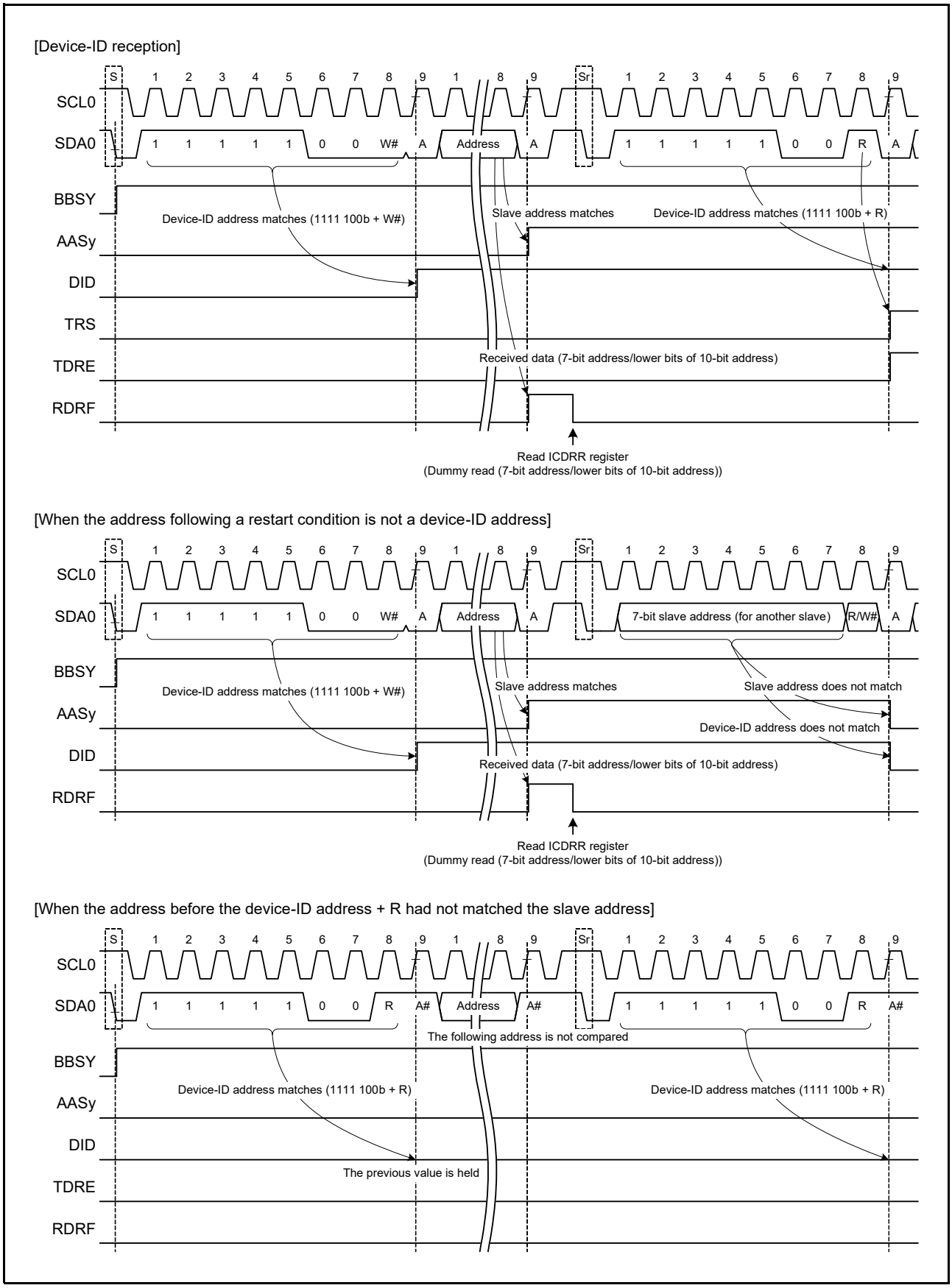


Figure 33.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

33.7.4 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.

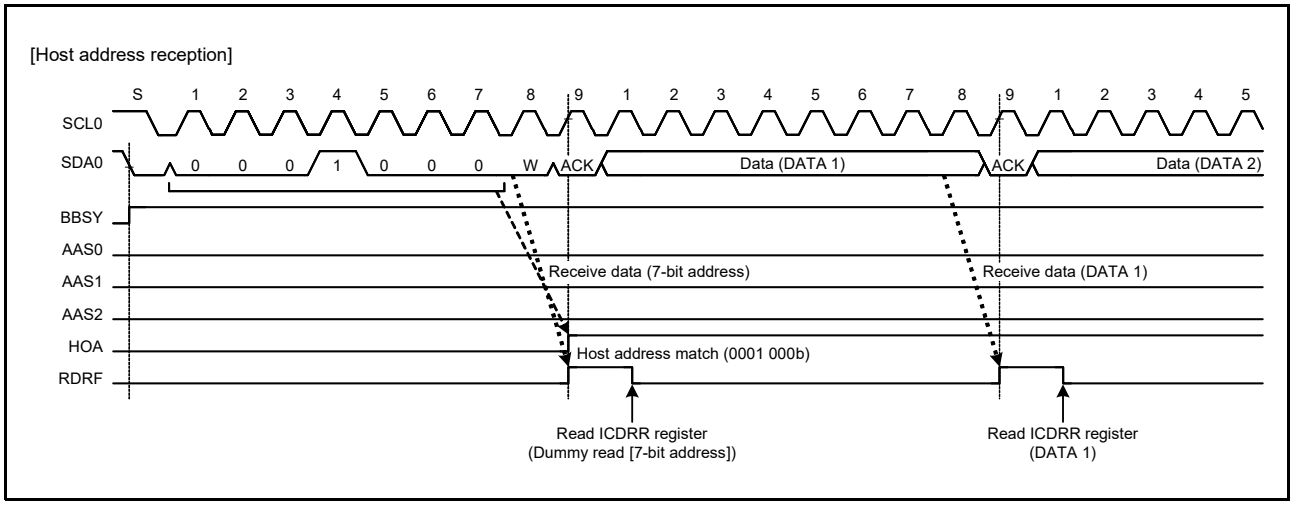


Figure 33.28 HOA Flag Set Timing during Reception of Host Address

33.8 Automatic Low-Hold Function for SCL

33.8.1 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 SCL0 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

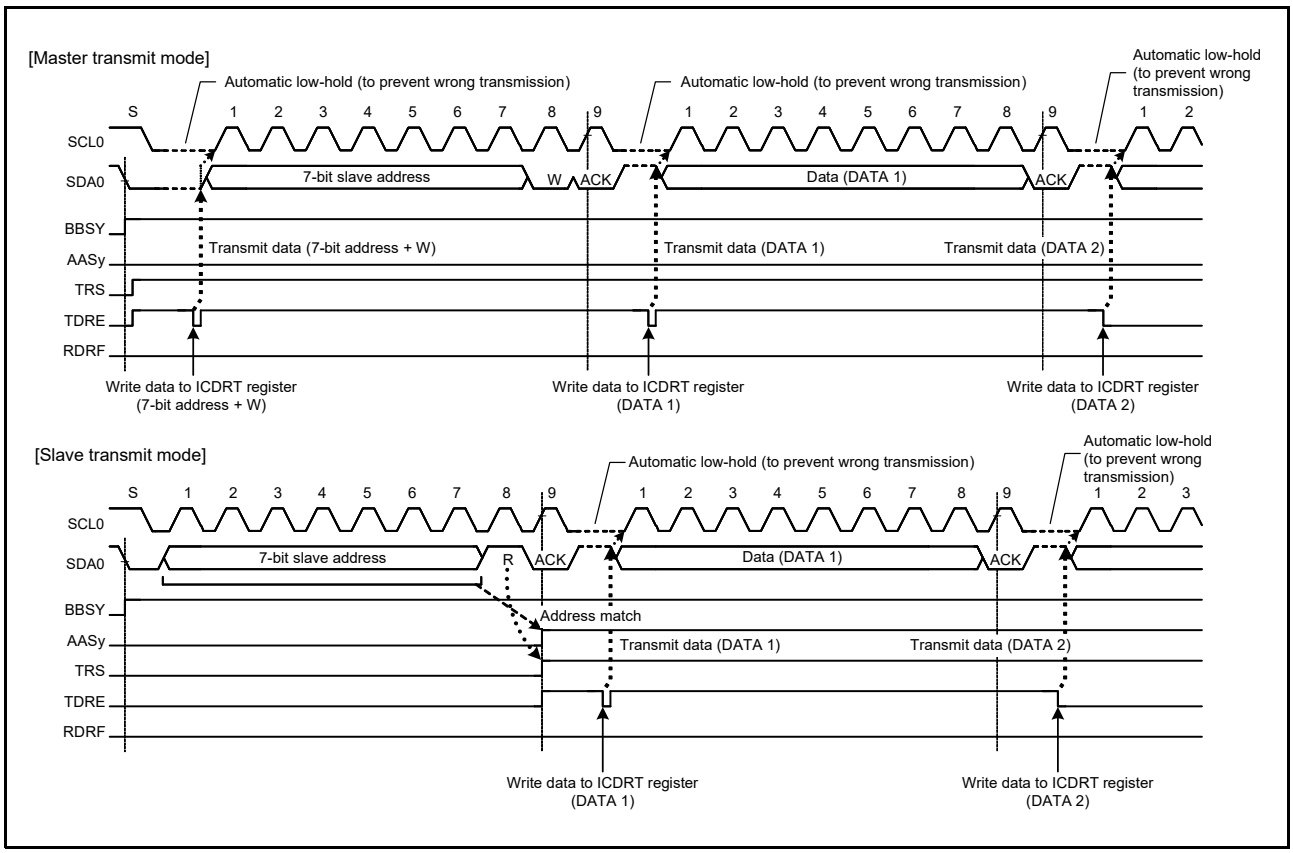


Figure 33.29 Automatic Low-Hold Operation in Transmit Mode

33.8.2 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 SDA0 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.

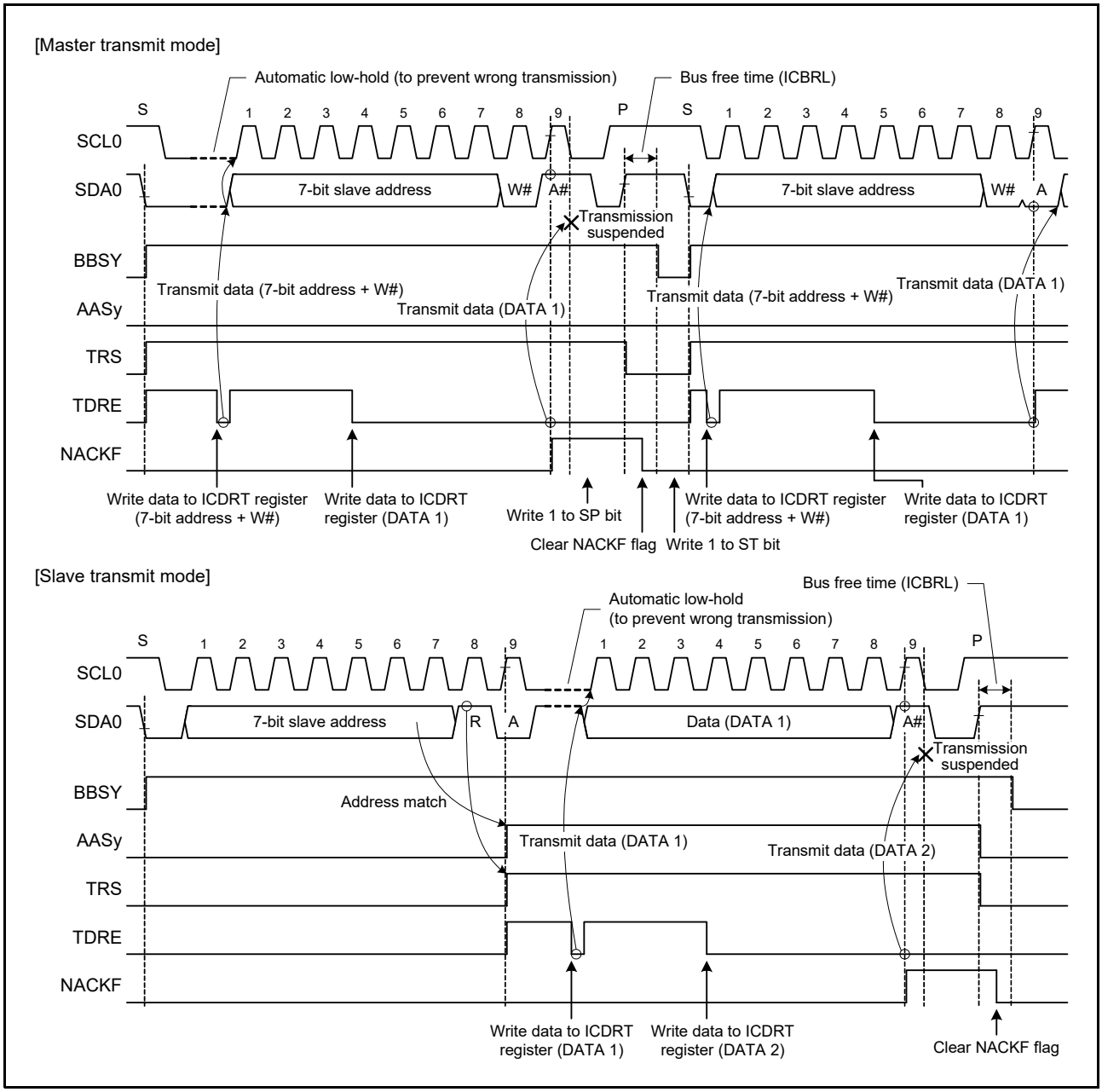


Figure 33.30 Suspension of Data Transmission When NACK is Received (NACKE = 1)

33.8.3 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 SCL0 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 SCL0 line low when a mismatch with its own slave address occurs after a stop condition is generated.
Sections in which the SCL0 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 SCL0 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 SCL0 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.

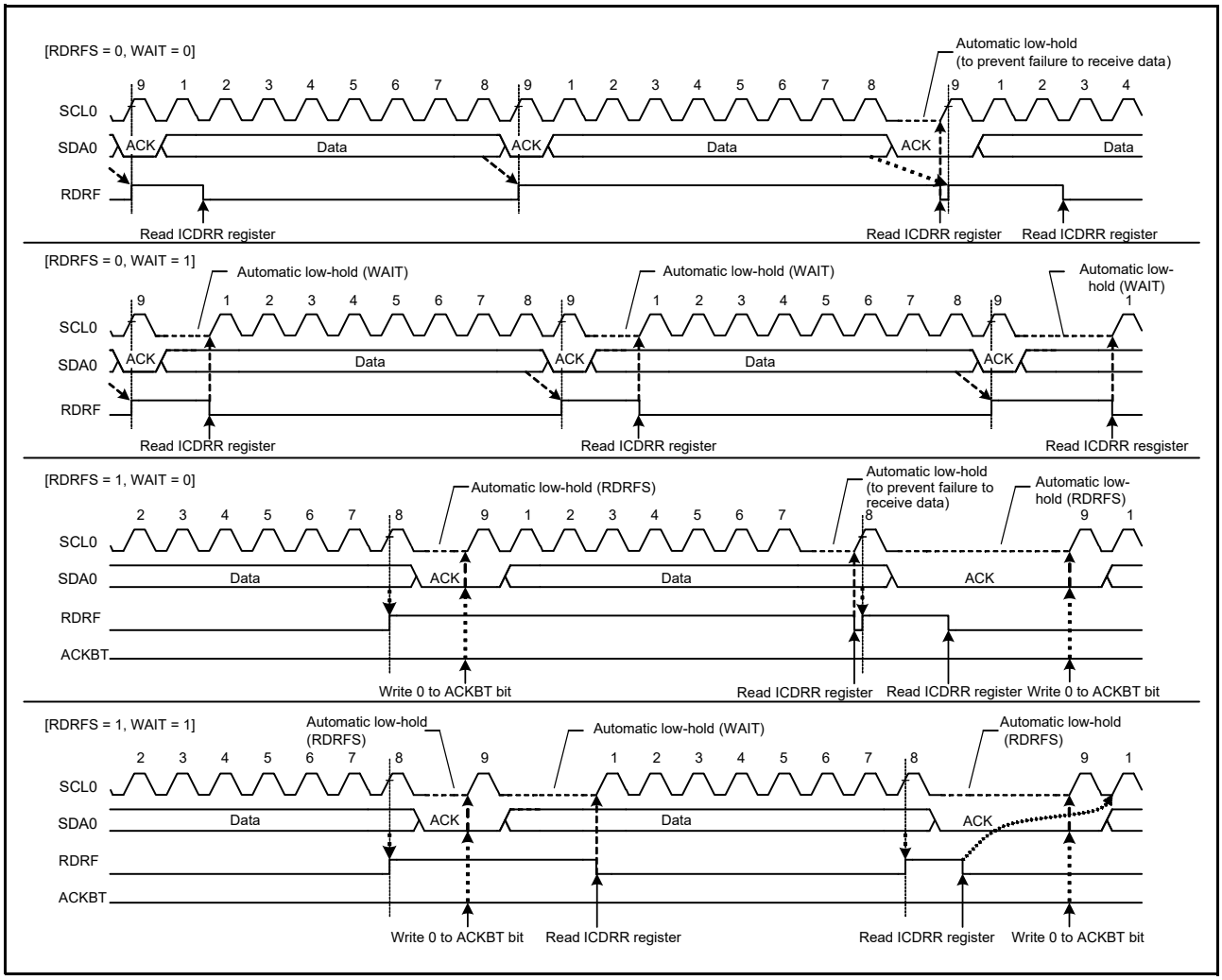


Figure 33.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

33.9 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.

33.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to generate a start condition. However, if the SDA0 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 SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 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 SDA0 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 SDA0 line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

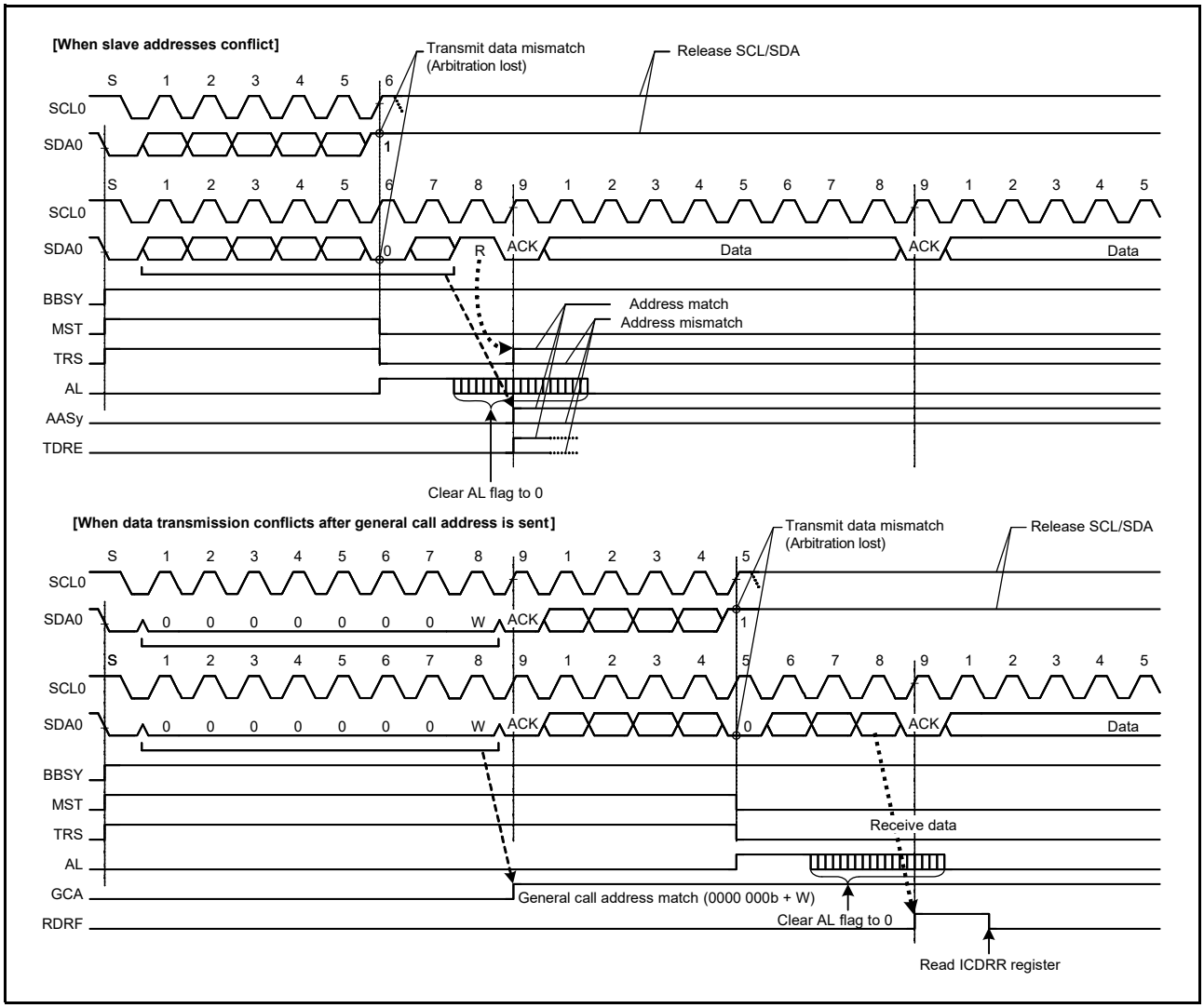


Figure 33.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

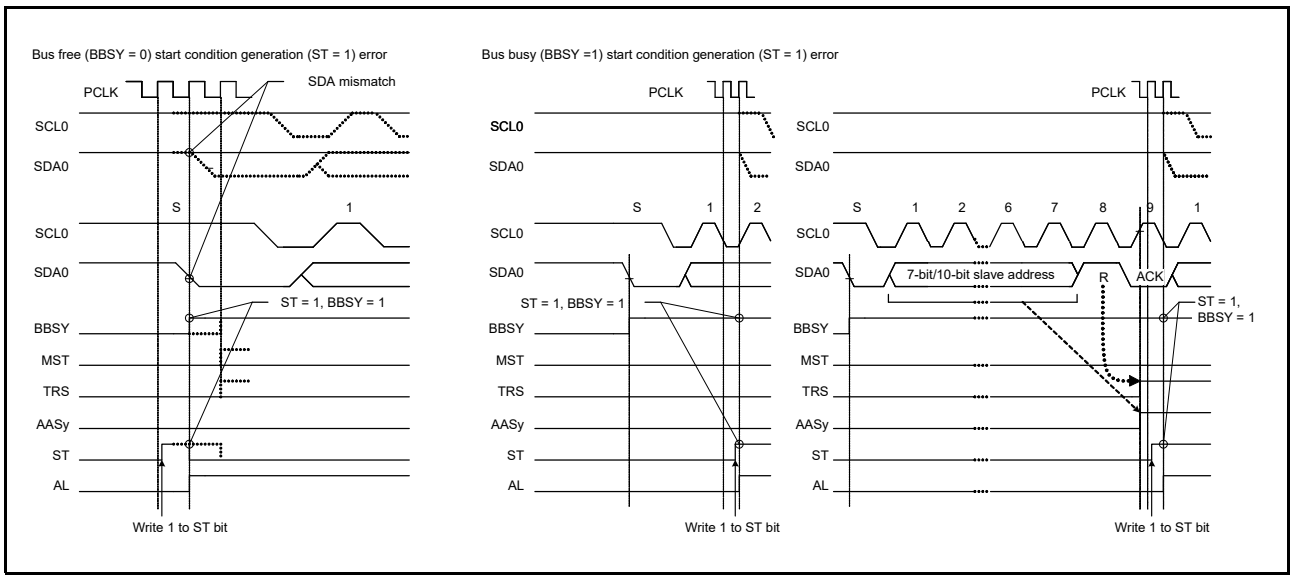


Figure 33.33 Arbitration-Lost When a Start Condition is Generated (MALE = 1)

33.9.2 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 SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 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 33.34 shows an example of NACK transmission arbitration-lost detection.

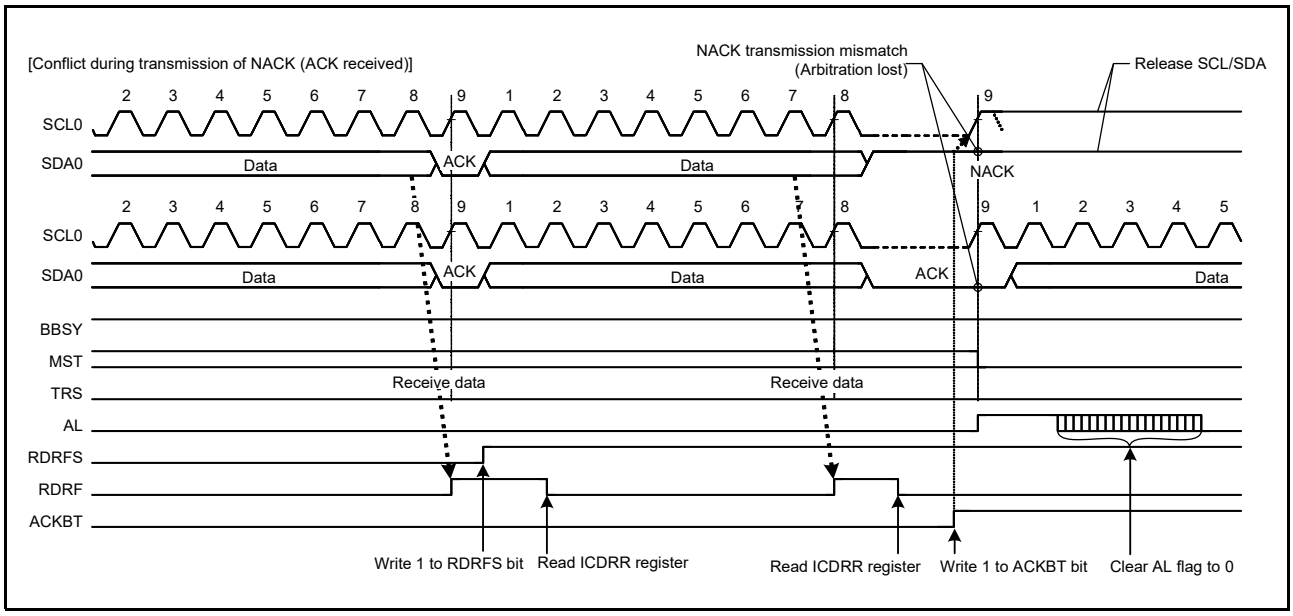


Figure 33.34 Example of NACK transmission arbitration-lost Detection (NALE = 1)

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 SDA0 line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

33.9.3 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 SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state and the low is detected on the SDA0 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 SDA0 line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

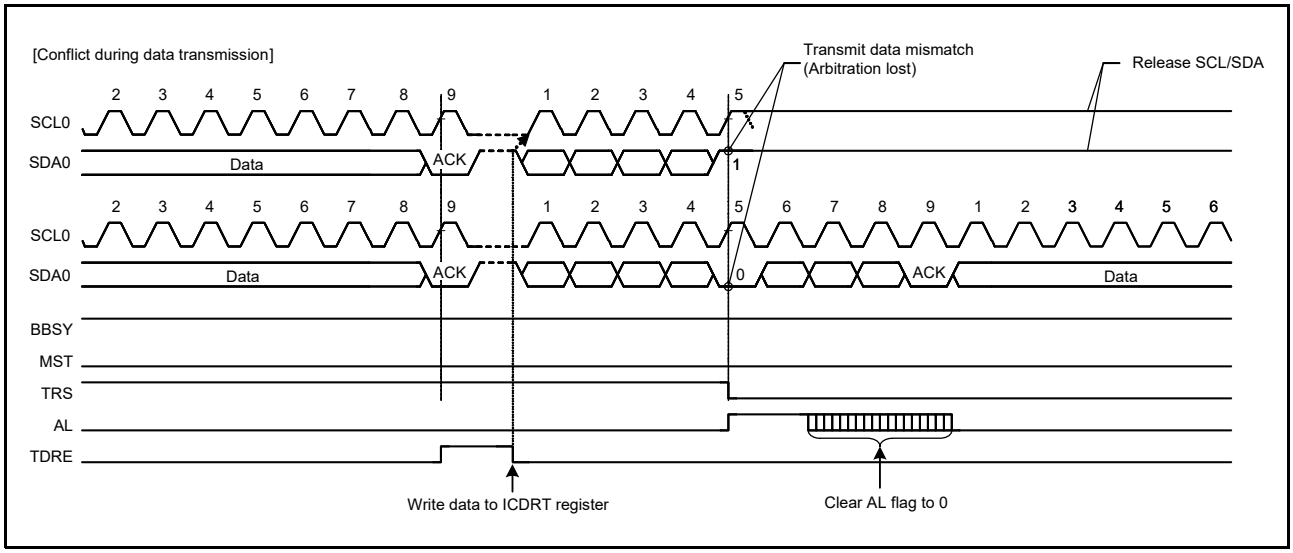


Figure 33.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

33.10 Start Condition/Restart Condition/Stop Condition Generating Function

33.10.1 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 SDA0 line low (high to low).
(2) Secure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCL0 line low (high to low).
(4) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

33.10.2 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 SDA0 line.
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDA0 line low (high to low).
(6) Secure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCL0 line low (high to low).
(8) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

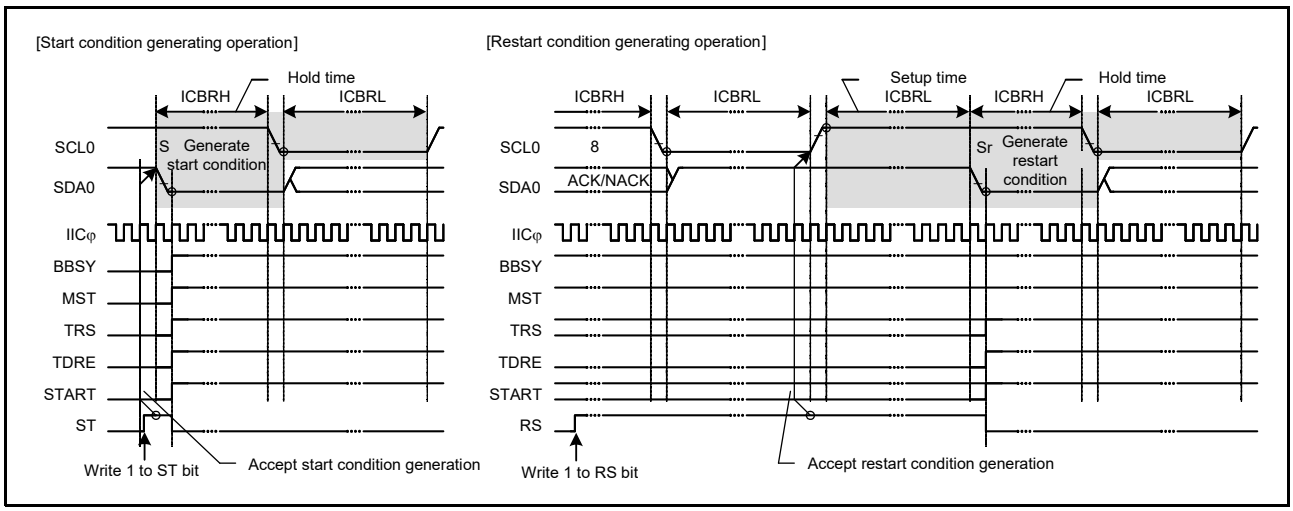


Figure 33.36 Start Condition/Restart Condition Generation Timing (ST and RS Bits)

33.10.3 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 SDA0 line low (high to low).
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRH register and the stop condition setup time.
(5) Release the SDA0 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).

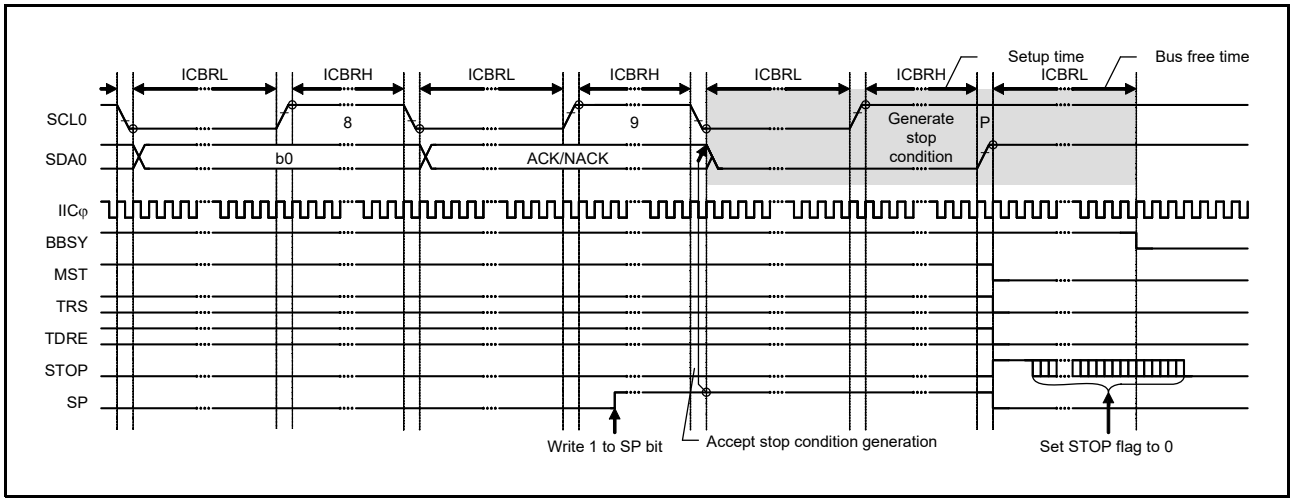


Figure 33.37 Stop Condition Generation Timing (SP Bit)

33.11 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 SCL0 line and/or SDA0 line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 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 SCL0 or SDA0 lines.

33.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.
The timeout function monitors the SCL0 line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 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 SCL0 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 SCL0 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.

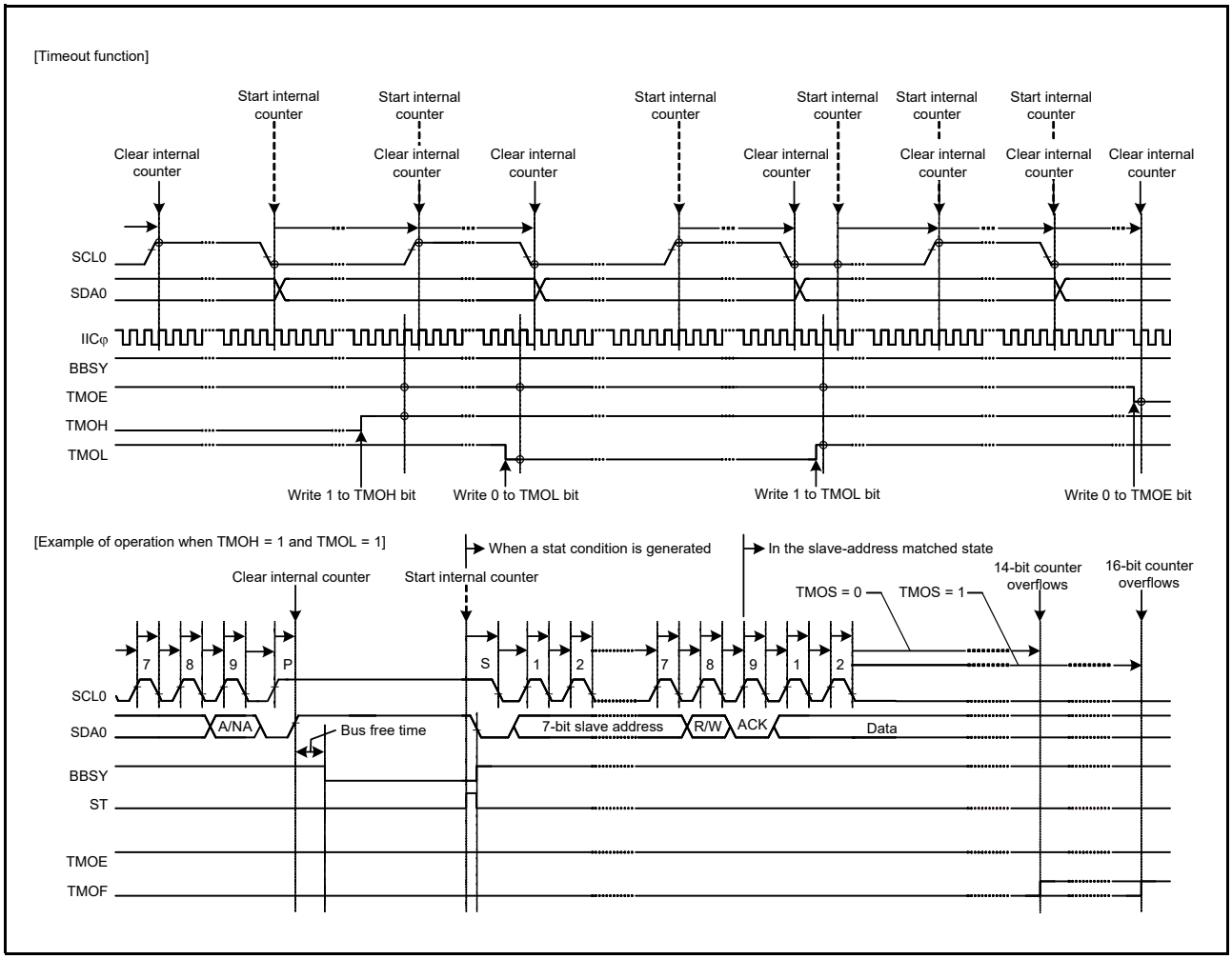


Figure 33.38 Timeout Function

33.11.2 Additional SCL Output Function

In master mode, the RIIC module has a facility for the output of additional SCL to release the SDA0 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 SDA0 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 SDA0 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 SCL0 pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCL0 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 SDA0 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 SDA0 line from being held low, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDA0 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 SCL0 line low

Figure 33.39 shows the operation timing of the additional SCL output function (CLO bit).

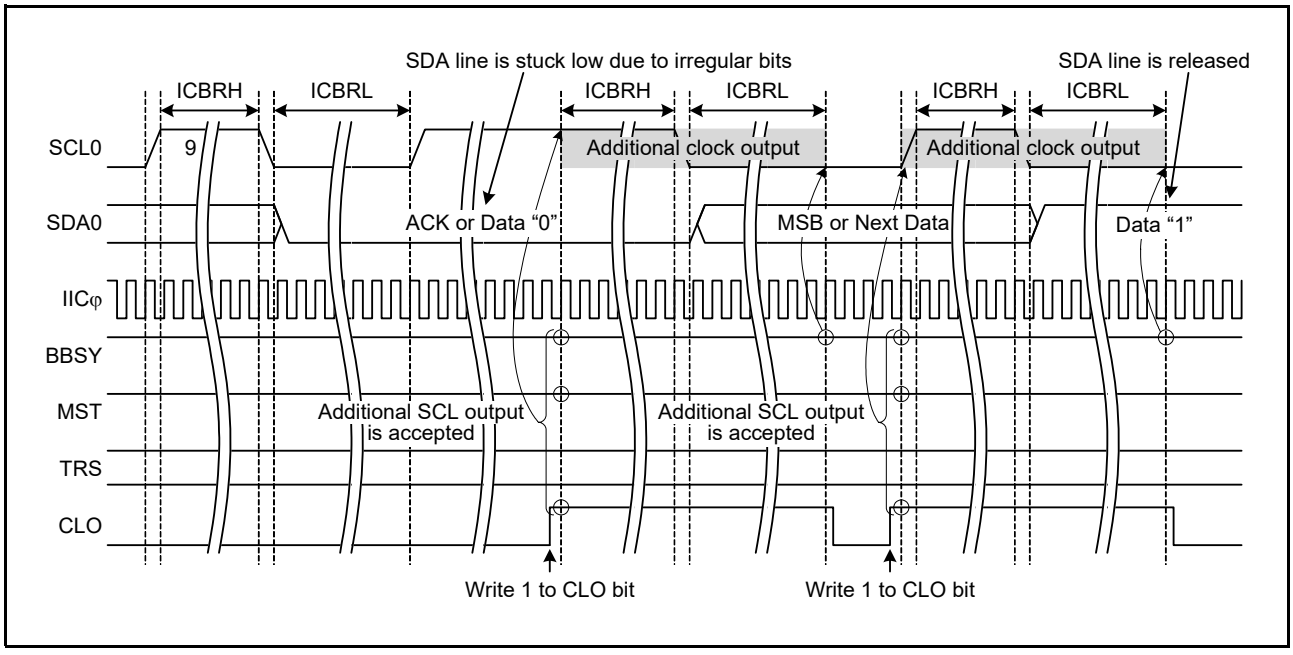


Figure 33.39 Additional SCL Output Function (CLO Bit)

33.11.3 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 SCL0 and SDA0 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 33.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.

33.12 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.

33.12.1 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 GPTW 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 GPTW 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 SCL0 and SDA0 pins of the bus and makes the SCL0/SDA0 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 GPTW 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 GPTW 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).

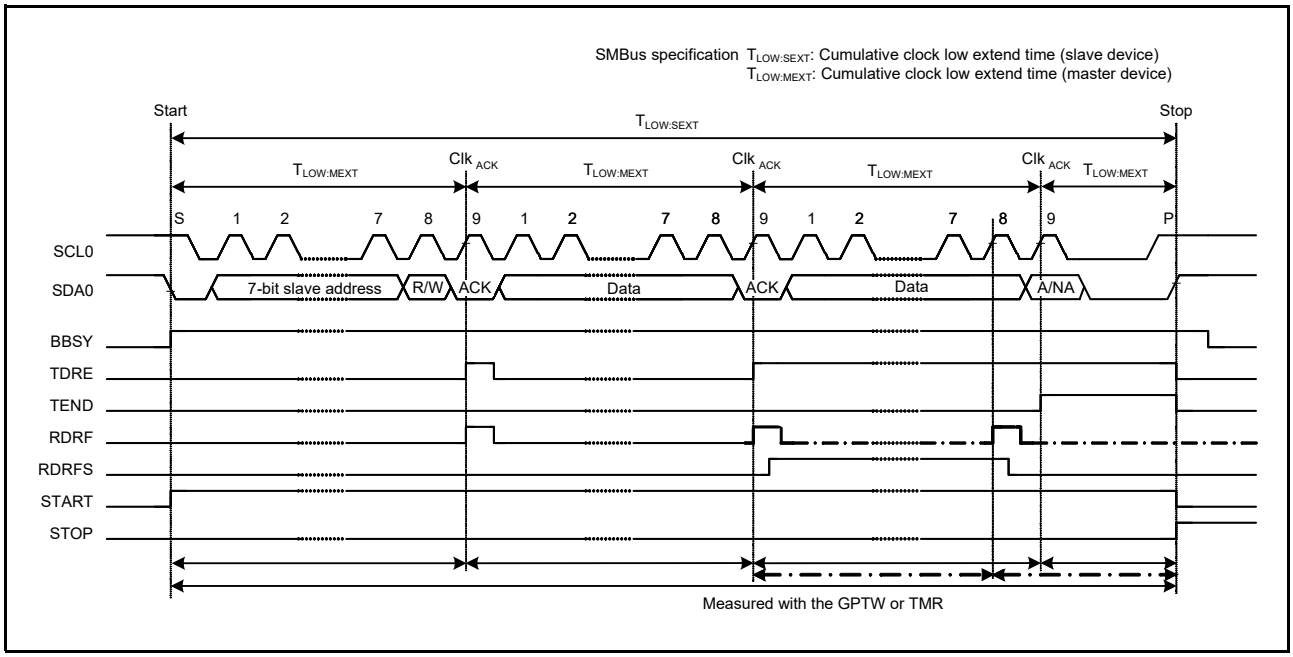


Figure 33.40 SMBus Timeout Measurement

33.12.2 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 36, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock pulse.

33.12.3 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.

33.13 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 33.6 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 33.6 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.

33.13.1 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.

33.14 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 33.7 lists the reset states of registers and functions when a reset is applied or a condition is detected.

Table 33.7 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.

33.15 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

33.15.1 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 33.6.

33.16 Usage Notes

33.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB). 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 register B, refer to section 11, Low Power Consumption.

33.16.2 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.

34. CAN FD Module (CANFD)

34.1 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 34.1 lists the specifications of the CAN FD module, and Figure 34.1 shows a block diagram of the CAN FD module.

Table 34.1 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 5 Mbps*1, *2
Operating frequency*3	Register block: up to 32 MHz (PCLKB) Message buffer RAM: up to 64 MHz (PCLKA)
Operating clock for data link layer (DLL clock)	Up to 32 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 bit rate for communications depends on the board design and external environment. Determine it following sufficient evaluation.
Note 3. 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.

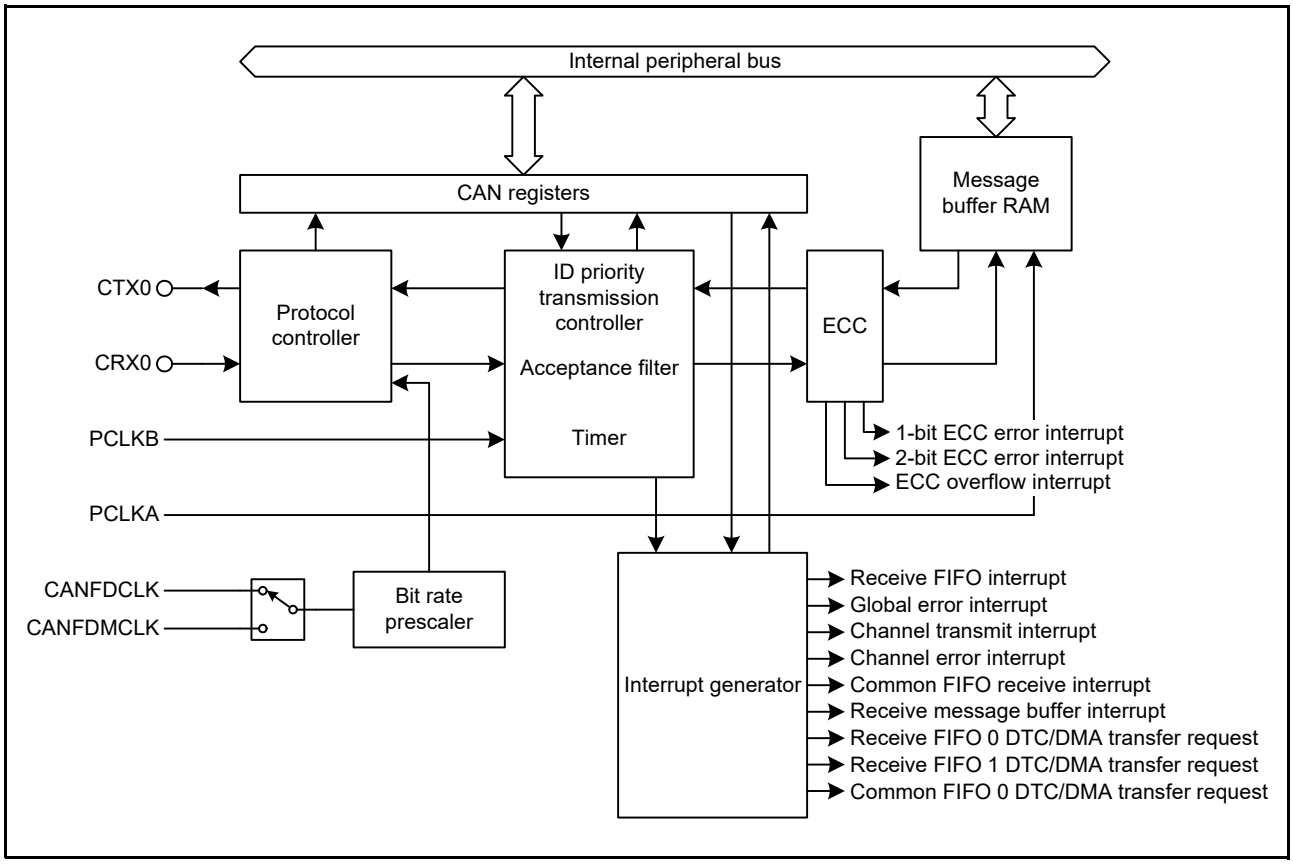


Figure 34.1 CAN FD Module Block Diagram

- 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 34.2 lists the pin configuration of the CAN FD module.

Table 34.2 CAN FD Module Pin Configuration

Pin Name	I/O	Function
CRX0	Input	Receive data input
CTX0	Output	Transmit data output

34.2 Register Descriptions

34.2.1 Nominal Bit Rate Configuration Register (NBCR)

Address(es): CANFD0.NBCR 000A 8000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSEG2[6:0]							TSEG1[7:0]								SJW[6]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SJW[5:0]						BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	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 34.4.1.2, 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.

34.2.2 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*5	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.
In CH_SLEEP mode, values other than the SLPRQ bit cannot be changed.

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 34.3.2, Channel Modes.
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.
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.
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.

34.2.3 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.

34.2.4 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.

34.2.5 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 34.4.1.2, 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.

34.2.6 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 34.4.1.5, 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 34.3 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

34.2.7 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.

34.2.8 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 34.4.1.5, 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.

34.2.9 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.

34.2.10 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.

34.2.11 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 34.3.1, 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.

34.2.12 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_OPERATION 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 by resetting the MCU. This flag is automatically set to 0 when the CANFD module completed RAM initialization.

34.2.13 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.

34.2.14 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.

34.2.15 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.

34.2.16 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.

34.2.17 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.

34.2.18 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 34.2.60, 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 34.5.8, 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.

34.2.19 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.

34.2.20 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 DMBE bit to 0.

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.

34.2.21 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.

34.2.22 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.

34.2.23 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.

34.2.24 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 34.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

34.2.25 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.

34.2.26 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).

34.2.27 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 34.6, 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 34.6.2.1, 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.

34.2.28 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.

34.2.29 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).

34.2.30 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.

34.2.31 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.

34.2.32 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.

34.2.33 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.

34.2.34 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.

34.2.35 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.

34.2.36 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.

34.2.37 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.

34.2.38 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.

34.2.39 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.

34.2.40 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.

34.2.41 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.

34.2.42 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 34.10, 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.

34.2.43 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.

34.2.44 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.

34.2.45 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.

34.2.46 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.

34.2.47 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.

34.2.48 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.

34.2.49 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.

34.2.50 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.

34.2.51 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).

34.2.52 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 34.9.2.1, 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.

34.2.53 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 34.9.2.1, 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.

34.2.54 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.

34.2.55 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 34.9.2, 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.

34.2.56 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.

34.2.57 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.

34.2.58 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.

34.2.59 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 34.10, 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.

34.2.60 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 34.4 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 34.5 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

34.2.61 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 34.33 for an overview of this configuration. For more information on the number and types of message buffers, refer to section 34.6, FIFO Buffers and Message Buffer Configuration.

34.2.61.1 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 34.6.

Table 34.6 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

34.2.61.2 Receive Message Buffer n (RMBn) (n = 0 to 31)

The total number of the receive message buffer (RMB) is 32, as shown in Figure 34.33.
The receive message buffer consists of the following registers:

- RMBn.HF0
- RMBn.HF1
- RMBn.HF2
- RMBn.DF0 to RMBn.DF15

Table 34.7 shows the structure of this buffer.

Table 34.7 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

34.2.61.3 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 34.2.60, 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.

34.2.61.4 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.

34.2.61.5 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.

34.2.61.6 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

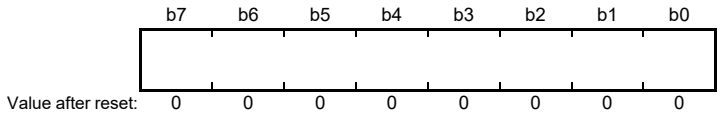
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	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.

34.2.61.7 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.

34.2.61.8 Receive FIFO n (RFBn) (n = 0, 1)

The total number of the receive FIFO (RFB) is two, as shown in Figure 34.33.
The receive FIFO consists of the following registers:

- RFBn.HF0
- RFBn.HF1
- RFBn.HF2
- RFBn.DF0 to RFBn.DF15

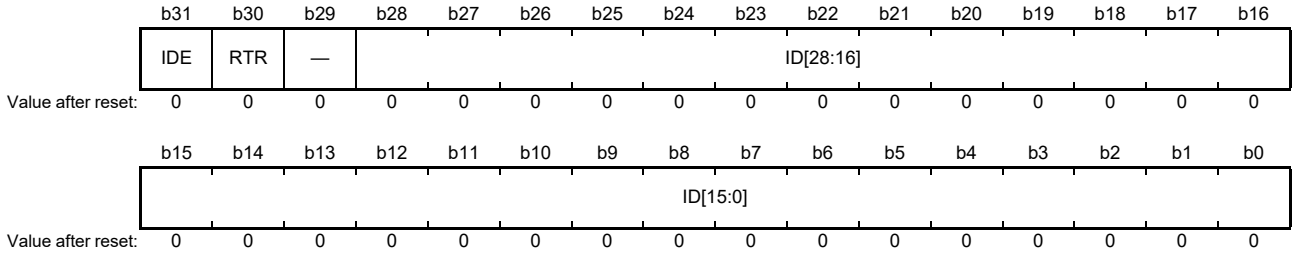
Table 34.8 shows the structure of this buffer.

Table 34.8 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

34.2.61.9 Receive FIFO n Header Filed 0 (RFBn.HF0) (n = 0, 1)

Address(es): CANFD.RFB0.HF0 000A 8520h, CANFD.RFB1.HF0 000A 856Ch



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 34.2.60, 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.

34.2.61.10 Receive FIFO n Header Filed 1 (RFBn.HF1) (n = 0, 1)

Address(es): CANFD.RFB0.HF1 000A 8524h, CANFD.RFB1.HF1 000A 8570h

	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	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.

34.2.61.11 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.

34.2.61.12 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.

34.2.61.13 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.

34.2.61.14 Common FIFO 0 (CFB0)

The total number of the common FIFO (CFB) is one, as shown in Figure 34.33.
The common FIFO consists of the following registers:

- CFB0.HF0
- CFB0.HF1
- CFB0.HF2
- CFB0.DF0 to CFB0.DF15

Table 34.9 shows the structure of this buffer.

Table 34.9 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

34.2.61.15 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 34.2.60, 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.

34.2.61.16 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.

34.2.61.17 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.

34.2.61.18 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.

34.2.61.19 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.

34.2.61.20 Transmit Message Buffer n (TMBn) (n = 0 to 3)

The total number of transmit message buffer (TMB) is four, as shown in Figure 34.33.
The transmit message buffer consists of the following registers:

- TMBn.HF0
- TMBn.HF1
- TMBn.HF2
- TMBn.DF0 to TMBn.DF15

Table 34.10 shows the structure of this buffer.

Table 34.10 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

34.2.61.21 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 34.2.60, 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.

34.2.61.22 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.

34.2.61.23 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.

34.2.61.24 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.

34.2.61.25 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.

34.2.62 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.

34.2.63 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.

34.2.64 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.

34.2.65 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.

34.3 Operating Mode

The operating modes of the CANFD module can be classified into two groups:

- Global modes
- Channel modes.

34.3.1 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 34.2 shows the possible transitions between the Global modes.

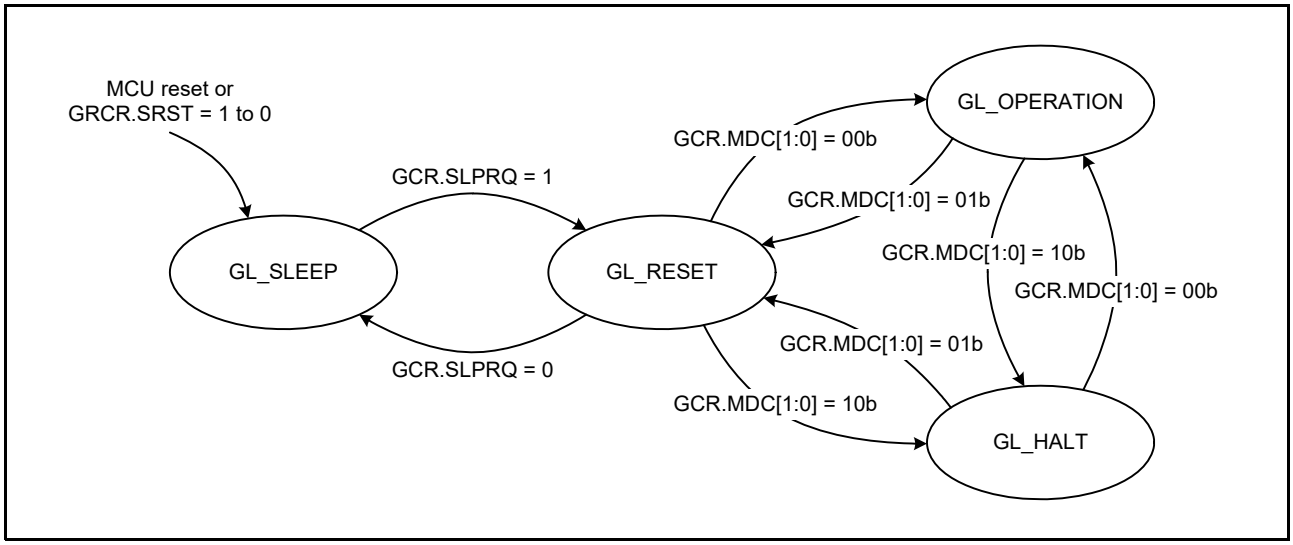


Figure 34.2 Transition between CANFD Global Modes

Change in the Global mode can affect the Channel mode. For details, refer to section 34.3.3, Global Mode and Channel Mode Transition Interactions.

34.3.1.1 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.

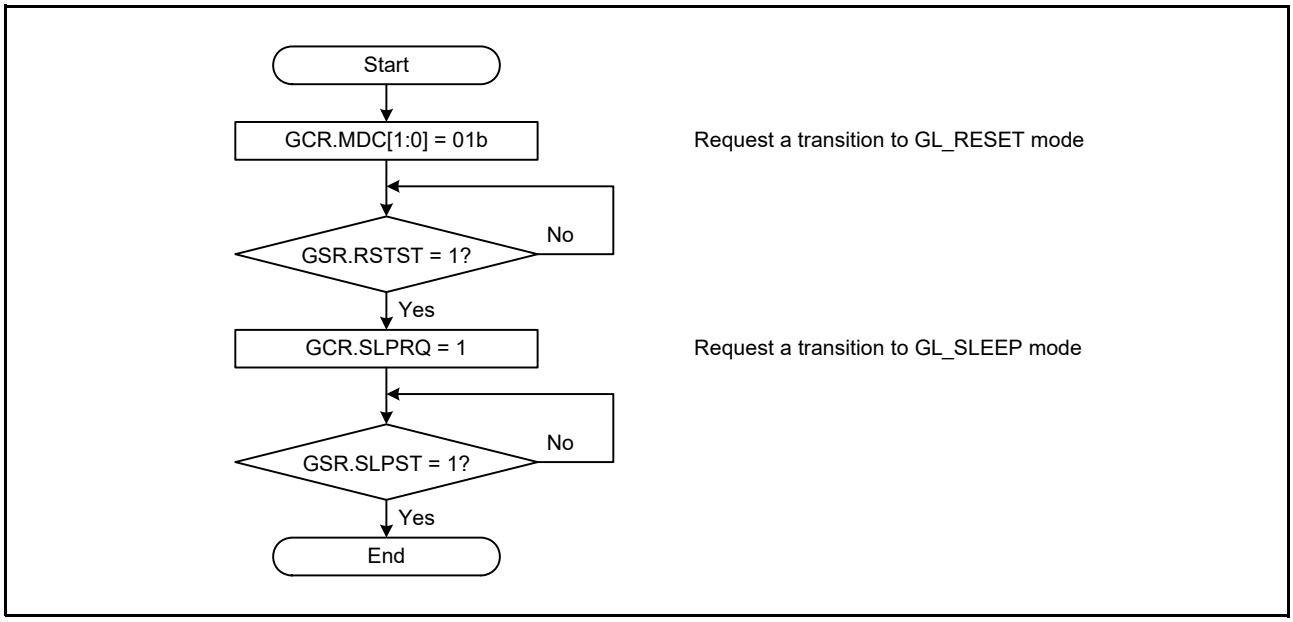


Figure 34.3 Procedure for Entering GL_SLEEP Mode

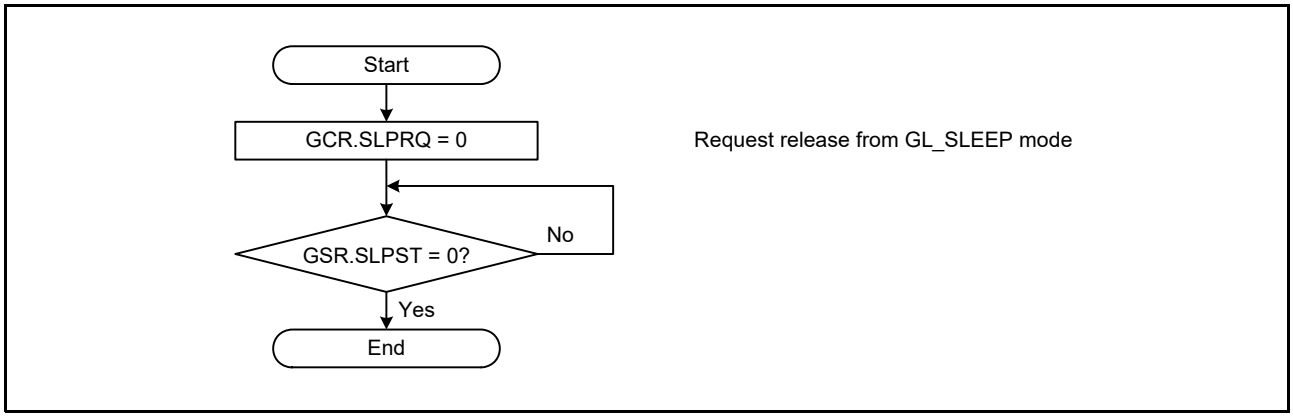


Figure 34.4 Procedure for Exiting GL_SLEEP Mode

34.3.1.2 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.

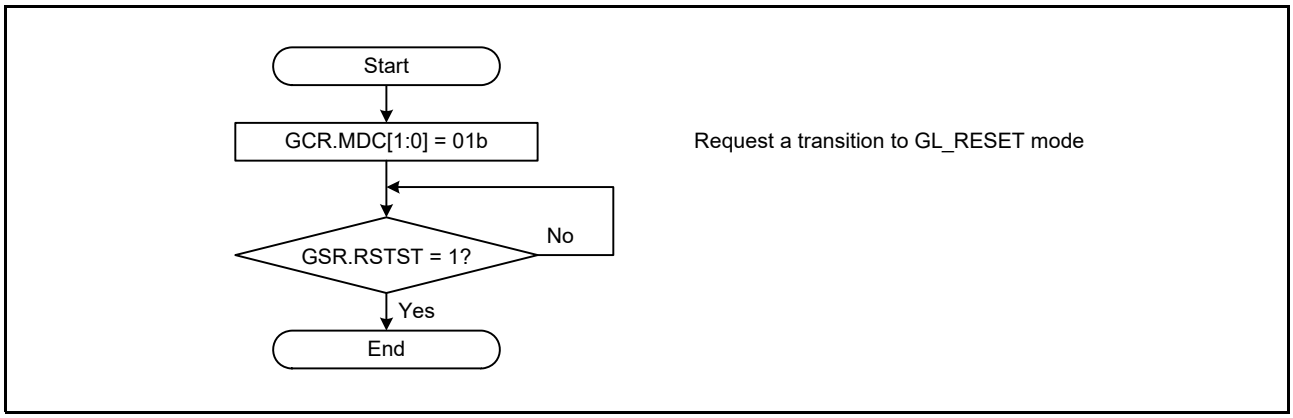


Figure 34.5 Procedure for Entering GL_RESET Mode

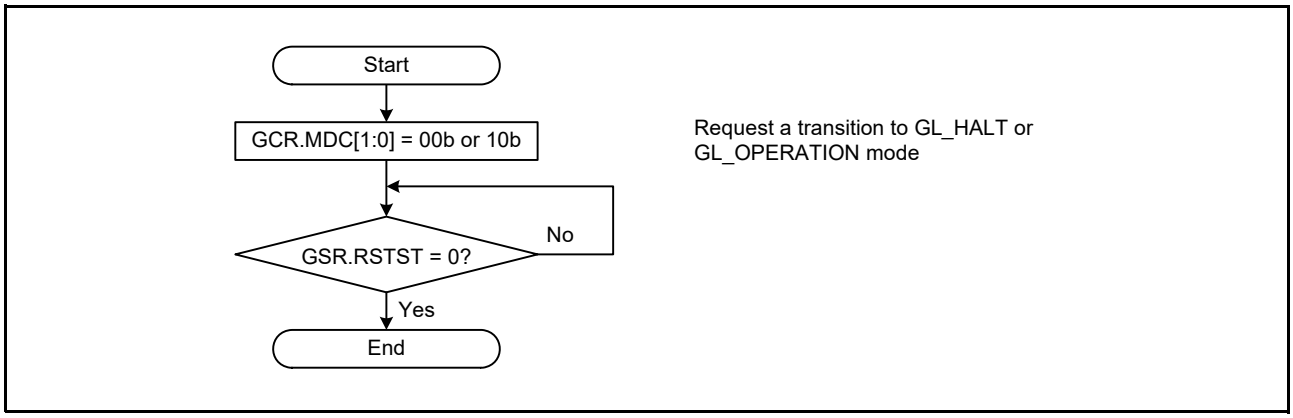


Figure 34.6 Procedure for Exiting GL_RESET Mode

34.3.1.3 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.

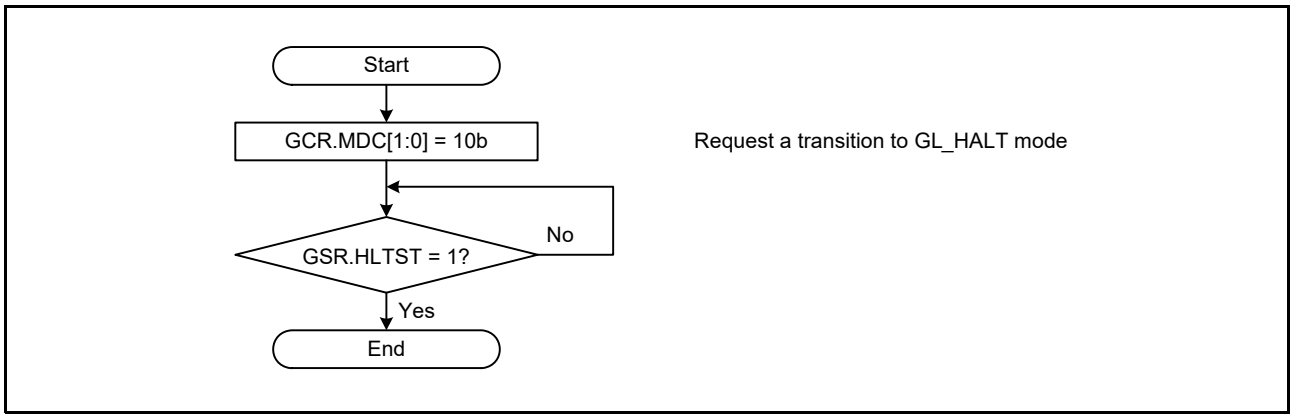


Figure 34.7 Procedure for Entering GL_HALT Mode

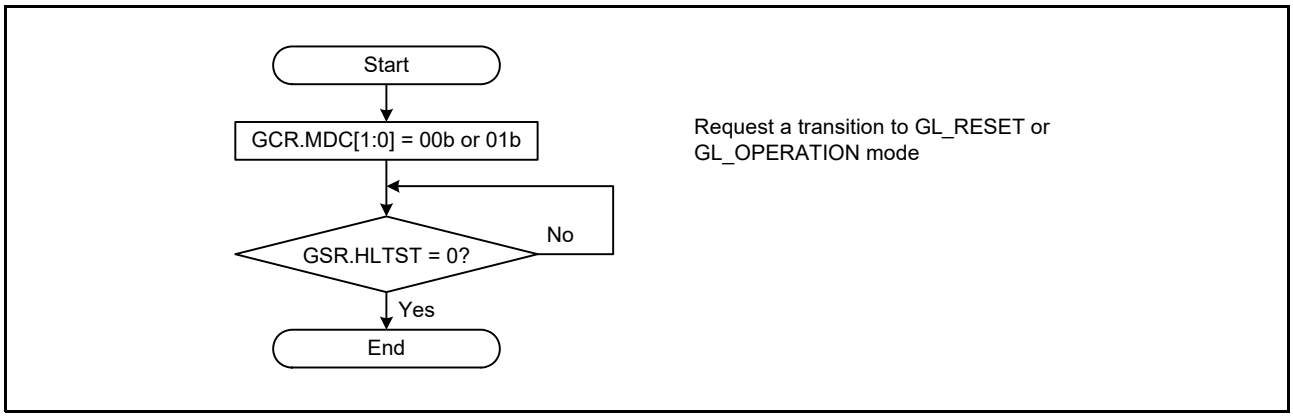


Figure 34.8 Procedure for Exiting GL_HALT Mode

34.3.1.4 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.

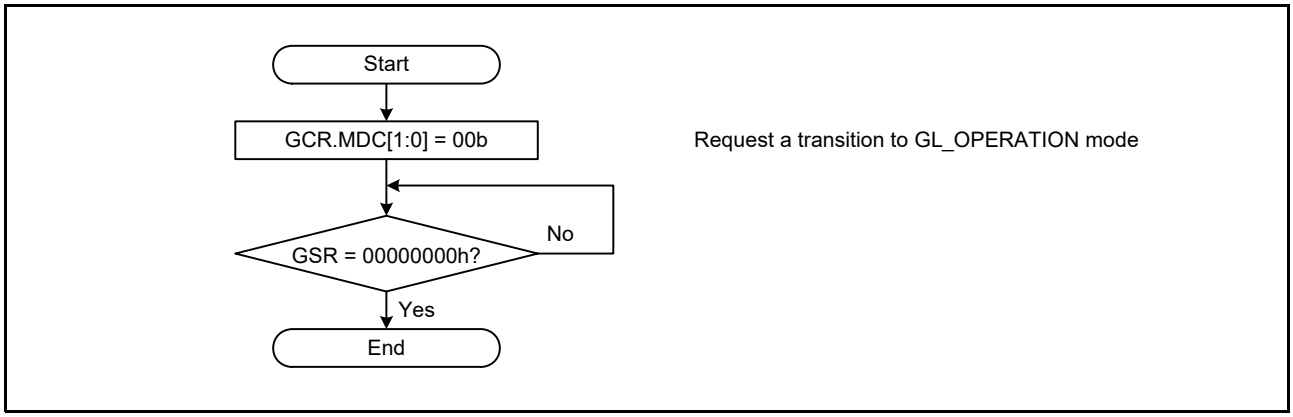


Figure 34.9 Procedure for Entering GL_OPERATION Mode

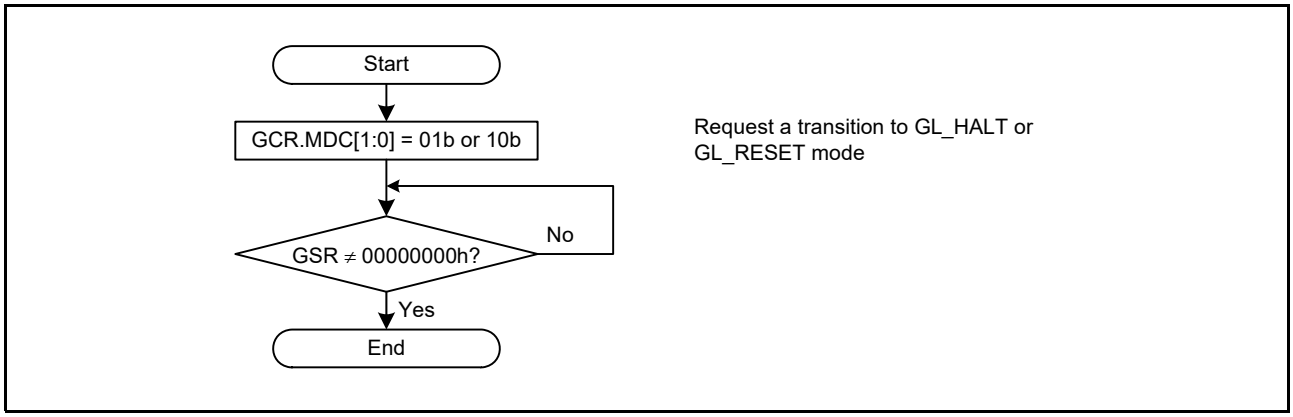


Figure 34.10 Procedure for Exiting GL_OPERATION Mode

34.3.2 Channel Modes

The Channel modes of the CANFD module are:

- CH_RESET
- CH_HALT
- CH_OPERATION
- CH_SLEEP.

Figure 34.11 shows the possible transitions between the channel modes.

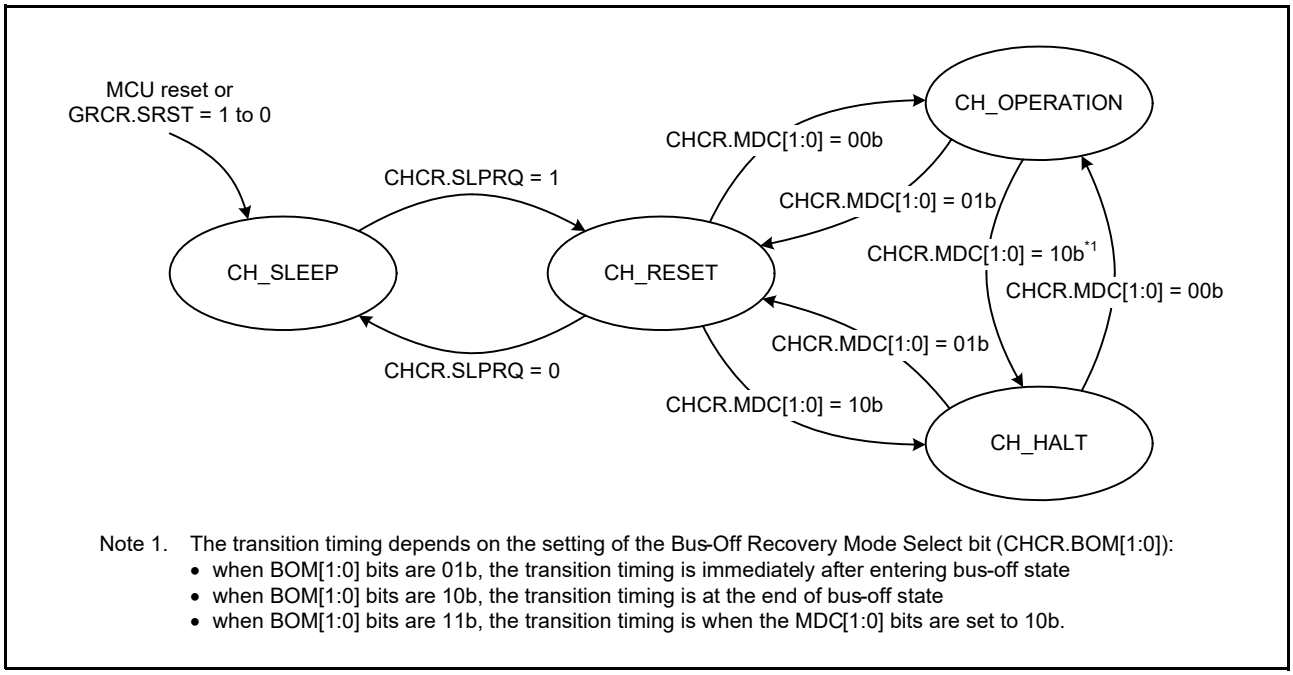


Figure 34.11 Transition between CAN Channel Modes

34.3.2.1 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.

34.3.2.2 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 34.11 for the behavior of transitioning to CH_RESET mode while CAN communication is in progress.

34.3.2.3 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 34.11 for the transition behavior to CH_HALT mode while CAN communication is in progress.

Table 34.11 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.

34.3.2.4 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 34.12):

- 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.

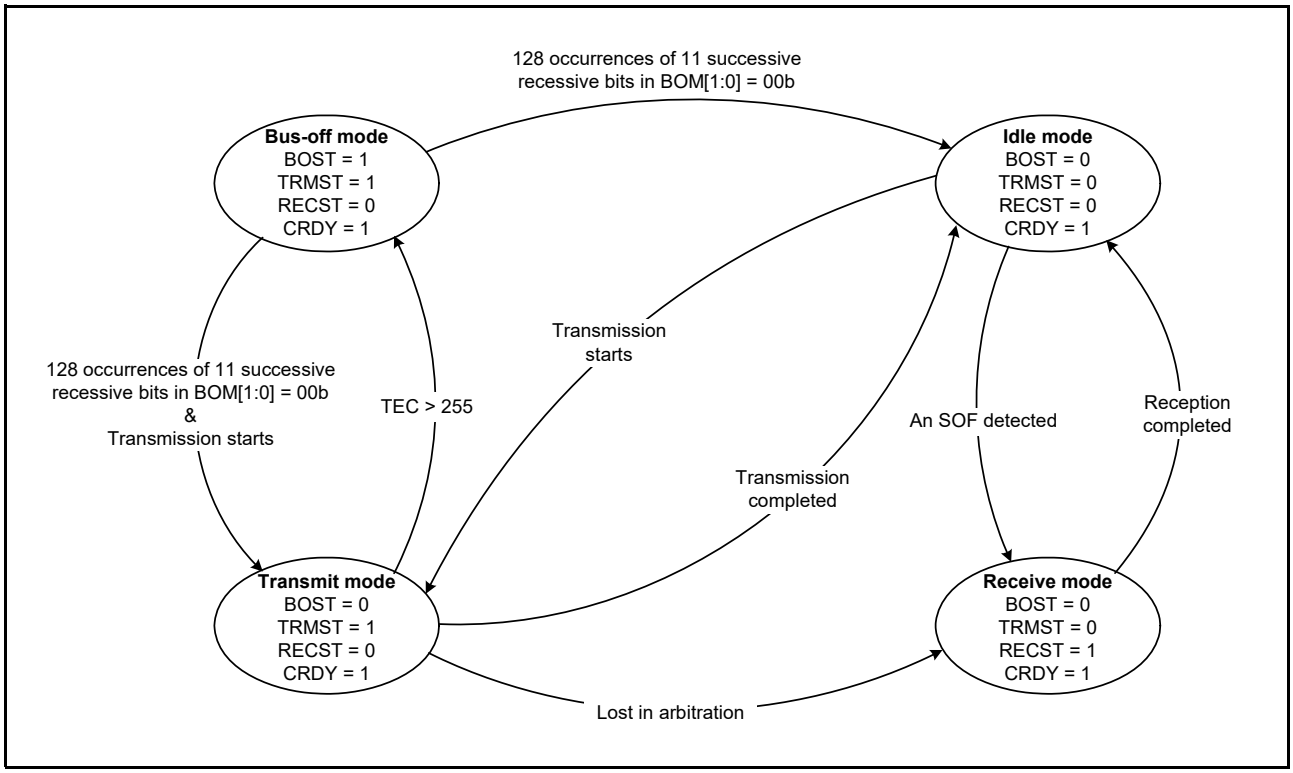


Figure 34.12 Sub-modes of CH_OPERATION Mode (only when CHCR.BOM[1:0] = 00b)

34.3.2.5 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 34.12 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 34.12 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 34.13.

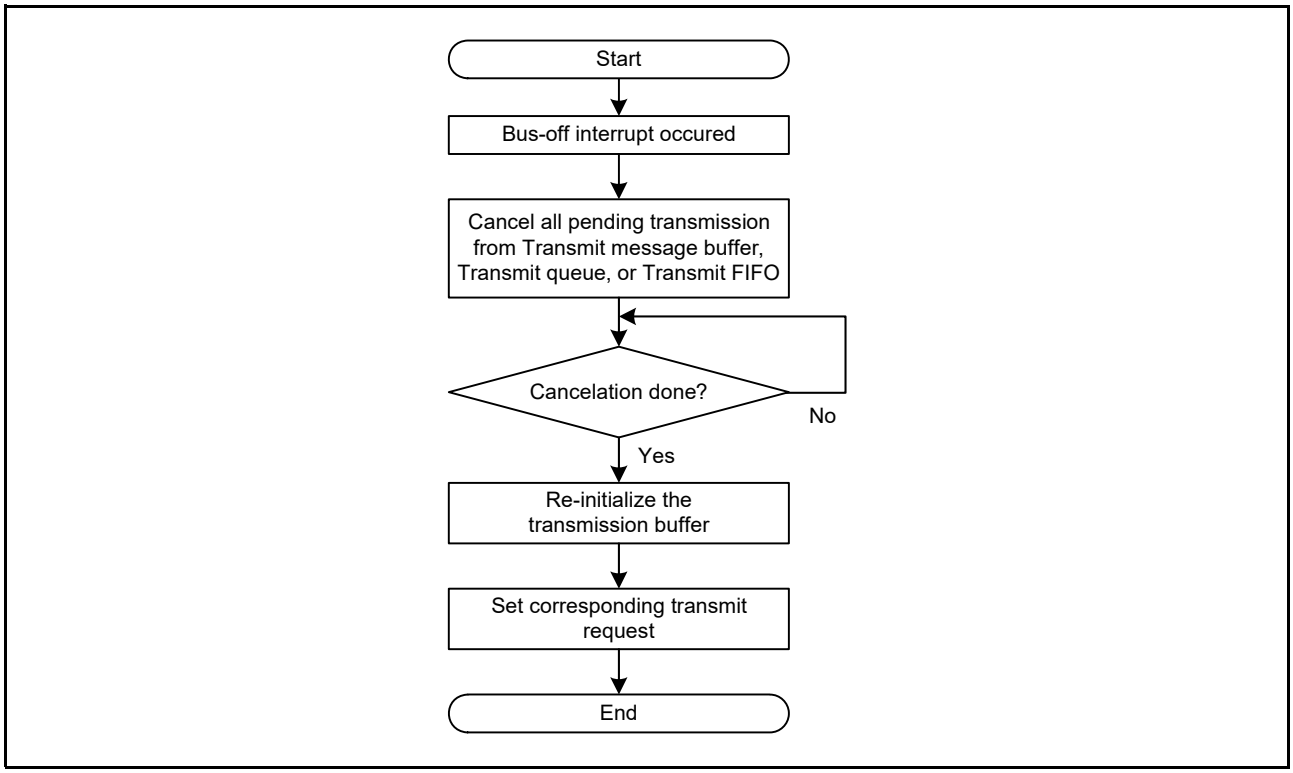


Figure 34.13 Transmission Re-Initialization During Bus-Off

34.3.3 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 34.13.

Table 34.13 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

34.3.3.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

Table 34.14 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.

34.3.3.2 Timing of Channel Mode Change

Table 34.15 shows the transition time for the Channel mode changes.

Table 34.15 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.

34.4 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.

34.4.1 Initialization of CAN Clock, Bit Timing and Bit Rate

34.4.1.1 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 34.16 shows an example of how to set the bit timing to achieve the required sample point settings.

Table 34.16 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

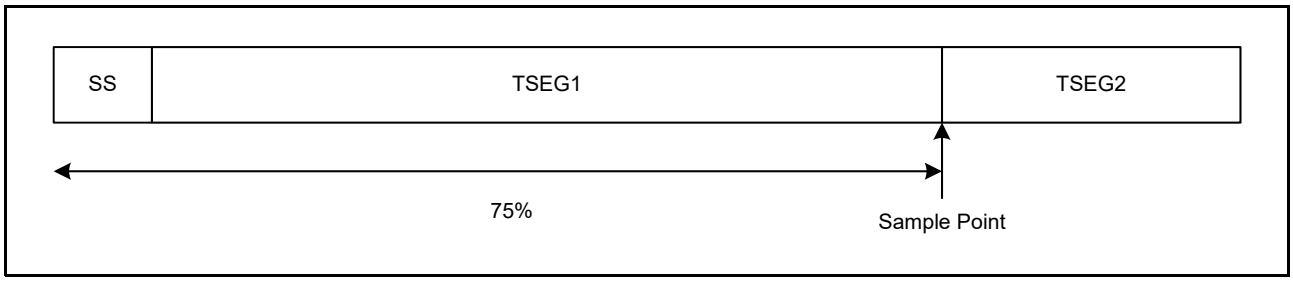


Figure 34.14 The Sample Point (in Case of 75%)

34.4.1.2 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 34.15 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).

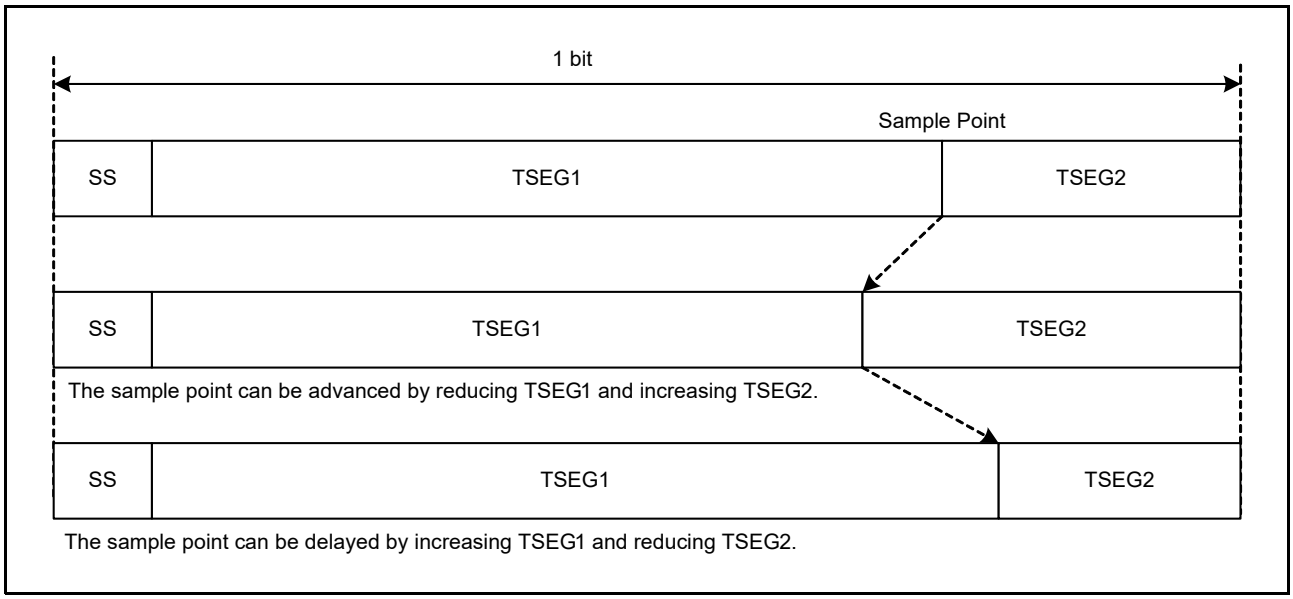


Figure 34.15 Segment Composition of A Bit and The Sample Point

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 34.15 shows an example of a typical sample point.

34.4.1.3 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 34.16 shows a block diagram of the circuit that generates the CAN communication clock.

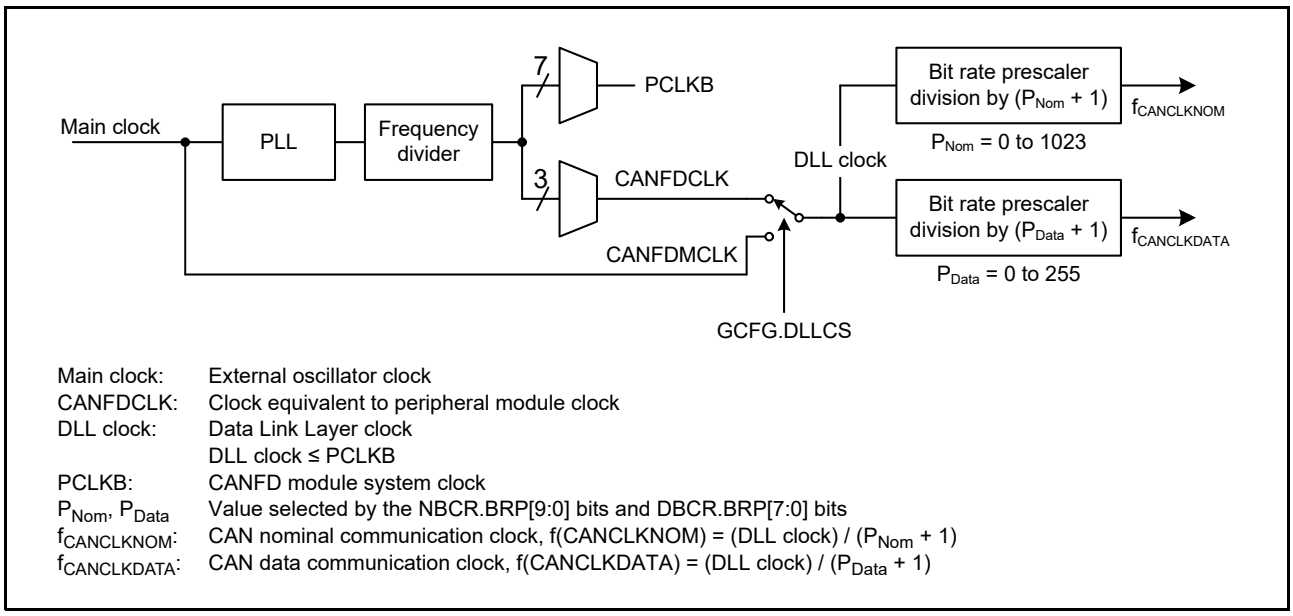


Figure 34.16 Block Diagram of the Circuit that Generates the CAN Communication Clock

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.

$$\text{Bit rate} = \frac{\text{DLL clock frequency}}{\text{Number of Tq per bit} \times (\text{P} + 1)} = \frac{\text{CAN communication clock frequency}}{\text{Number of Tq per bit}}$$

Table 34.17 lists examples of setting the nominal bit rate for Classical CAN frame.

Table 34.17 Nominal Bit Rate Setting Examples for Classical CAN Frame

Bit Rate	DLL Clock Frequency													
	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
1 Mbps	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	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	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	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 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	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 34.18 lists examples of setting the nominal and data bit rate for CAN FD frame.

Table 34.18 Nominal and Data Bit Rate Setting Examples for CAN FD Frame

Bit Rate		DLL Clock Frequency								
		32 MHz			30 MHz			20 MHz		
		No. of Tq			No. of Tq			No. of Tq		
Nominal	Data	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1
1 Mbps	5 Mbps	—	—	—	30 Tq	6 Tq	1	—	—	—
1 Mbps	4 Mbps	32 Tq	8 Tq	1	—	—	—	20 Tq	5 Tq	1
1 Mbps	2 Mbps	32 Tq	16 Tq	1	30 Tq	15 Tq	1	20 Tq	10 Tq	1
500 kbps	2 Mbps	64 Tq	16 Tq	1	60 Tq	15 Tq	1	40 Tq	10 Tq	1

34.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate

Figure 34.17 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.

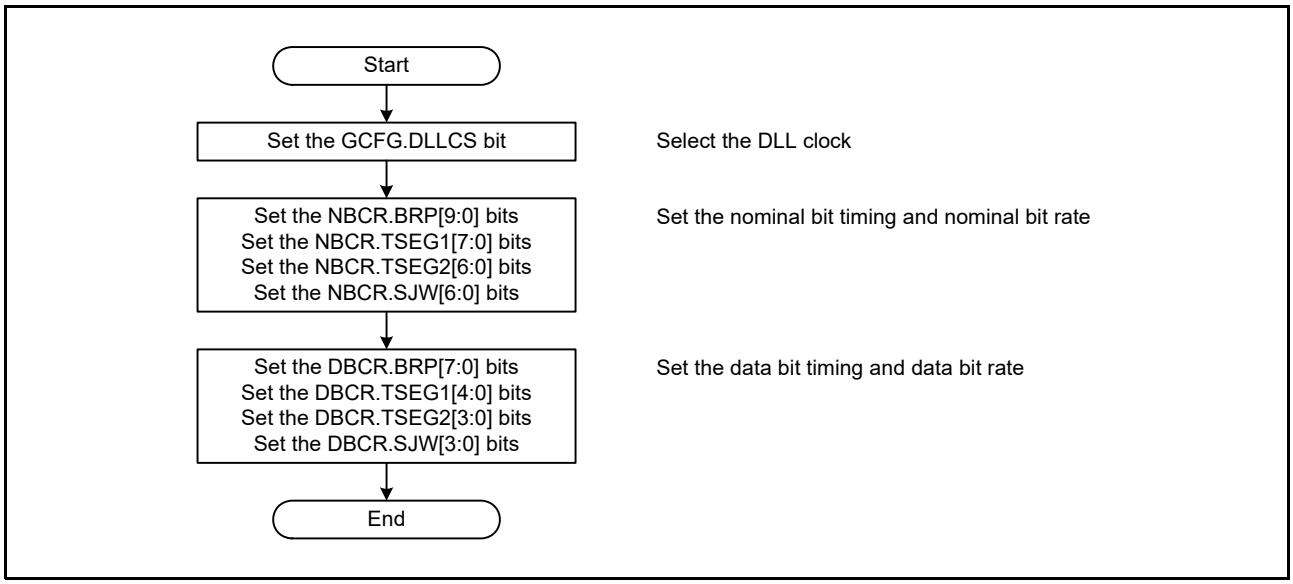


Figure 34.17 Procedure for Setting the Bit Timing and Bit Rate

34.4.1.5 Transceiver Delay Compensation

When a high bit rate is used such as 5 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 34.18.
The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

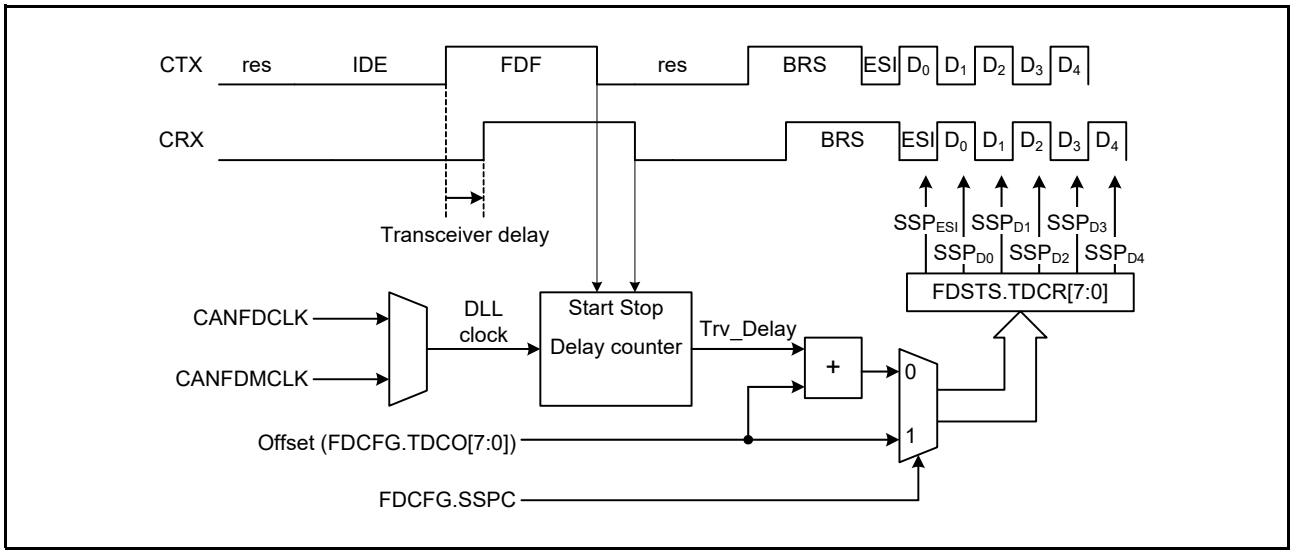


Figure 34.18 Transceiver Delay Compensation

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 34.19 shows the measurement example. Trv_Delay is counted up to 127 on each DLL clock.

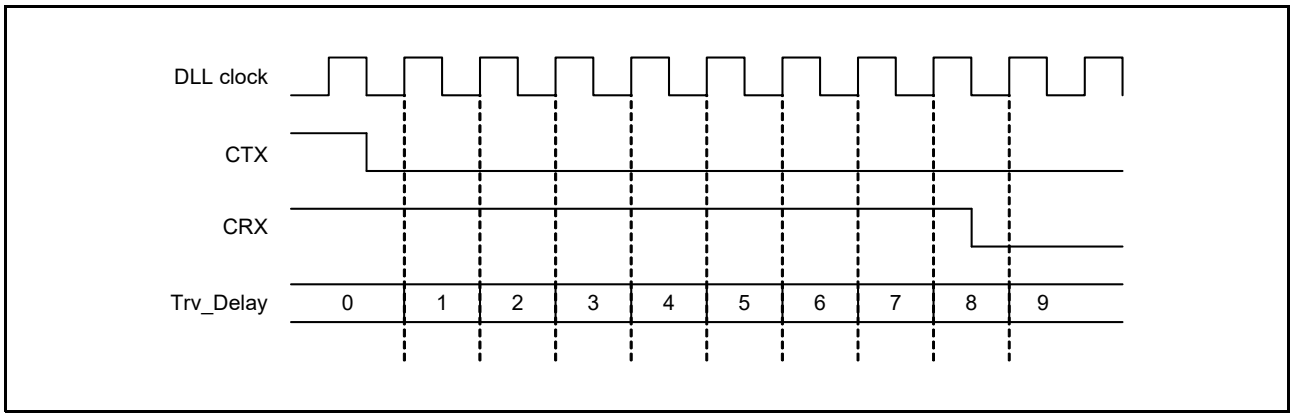


Figure 34.19 Trv_Delay Measurement Example

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 34.20 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.

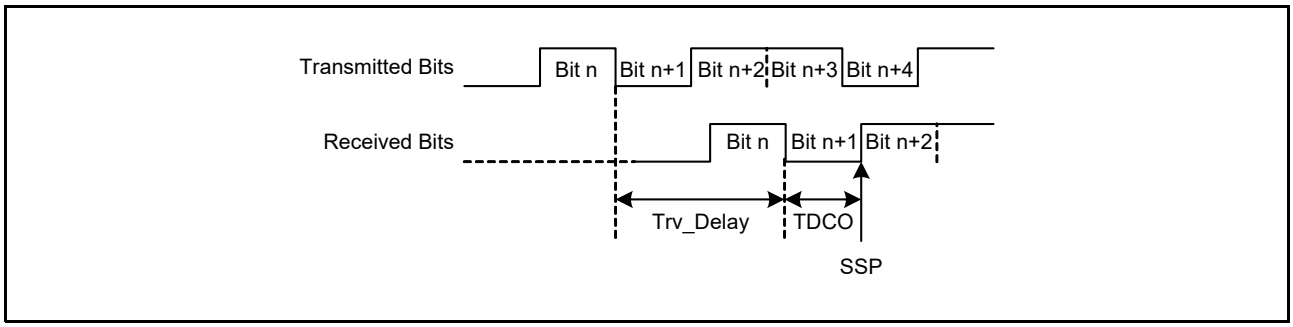


Figure 34.20 Positioning of the Secondary Sample Point (SSP)

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 34.21.
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.

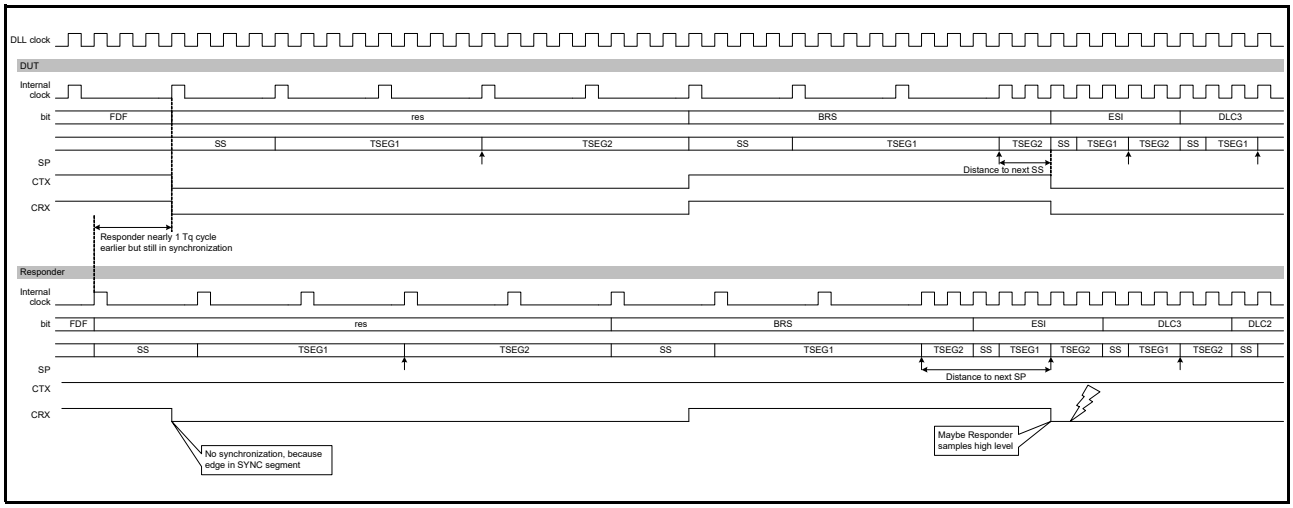


Figure 34.21 Loss of Synchronization Between Two CAN Nodes

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 34.22 shows the read flow to get the measured transceiver delay compensation result.

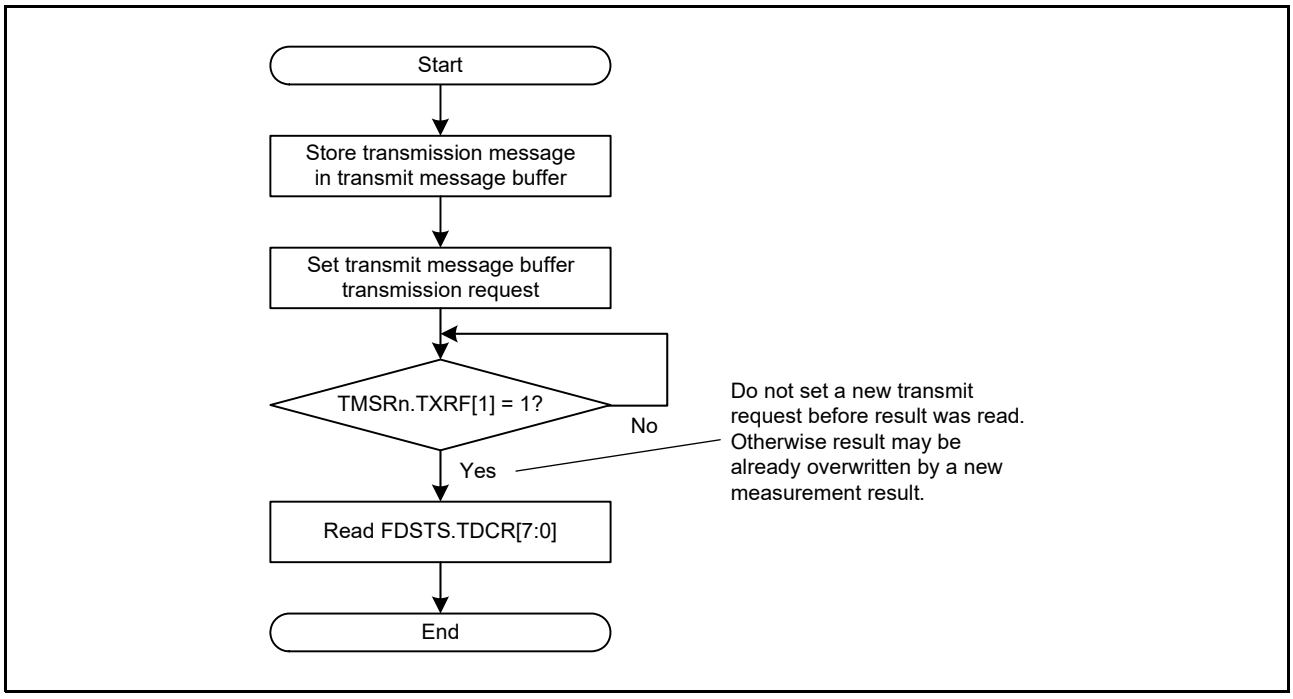


Figure 34.22 Transceiver Delay Compensation Result Read Flow

34.4.2 CANFD Module Configuration after a Reset

After resetting the MCU or applying a software reset using the GRCR.SRST bit, the CANFD module automatically enters GL_SLEEP mode.

To configure the CANFD module, set the GCR.SLPRQ bit to 0 to exit GL_SLEEP mode.

After resetting the MCU, the CANFD module starts RAM initialization when the module-stop state is released and clocks are supplied. At this time, the GSR.RAMST flag is automatically set to 1, indicating that the CANFD module is initializing the RAM.

After RAM initialization is complete, this bit is automatically set to 0.

RAM initialization is necessary to prevent false detection of ECC errors due to undefined data in RAM after MCU reset. Do not access (read or write) any other CANFD registers until the RAM initialization is complete and the RAMST flag is set to 0.

Before entering communication mode, the Acceptance Filter List and message FIFO buffers must be configured. In addition, the CAN channel settings such as CAN bit timing must be configured. To make this configuration, the CAN channel must be released from CH_SLEEP mode and configured for communication in CH_RESET mode (configuration mode).

Figure 34.23 shows the configuration procedure. For details about each step, refer to section 34.5, Filtering Using Acceptance Filter List (AFL), section 34.6, FIFO Buffers and Message Buffer Configuration, section 34.10, Interrupts and DTC/DMA Requests, and section 34.4.1.3, 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.

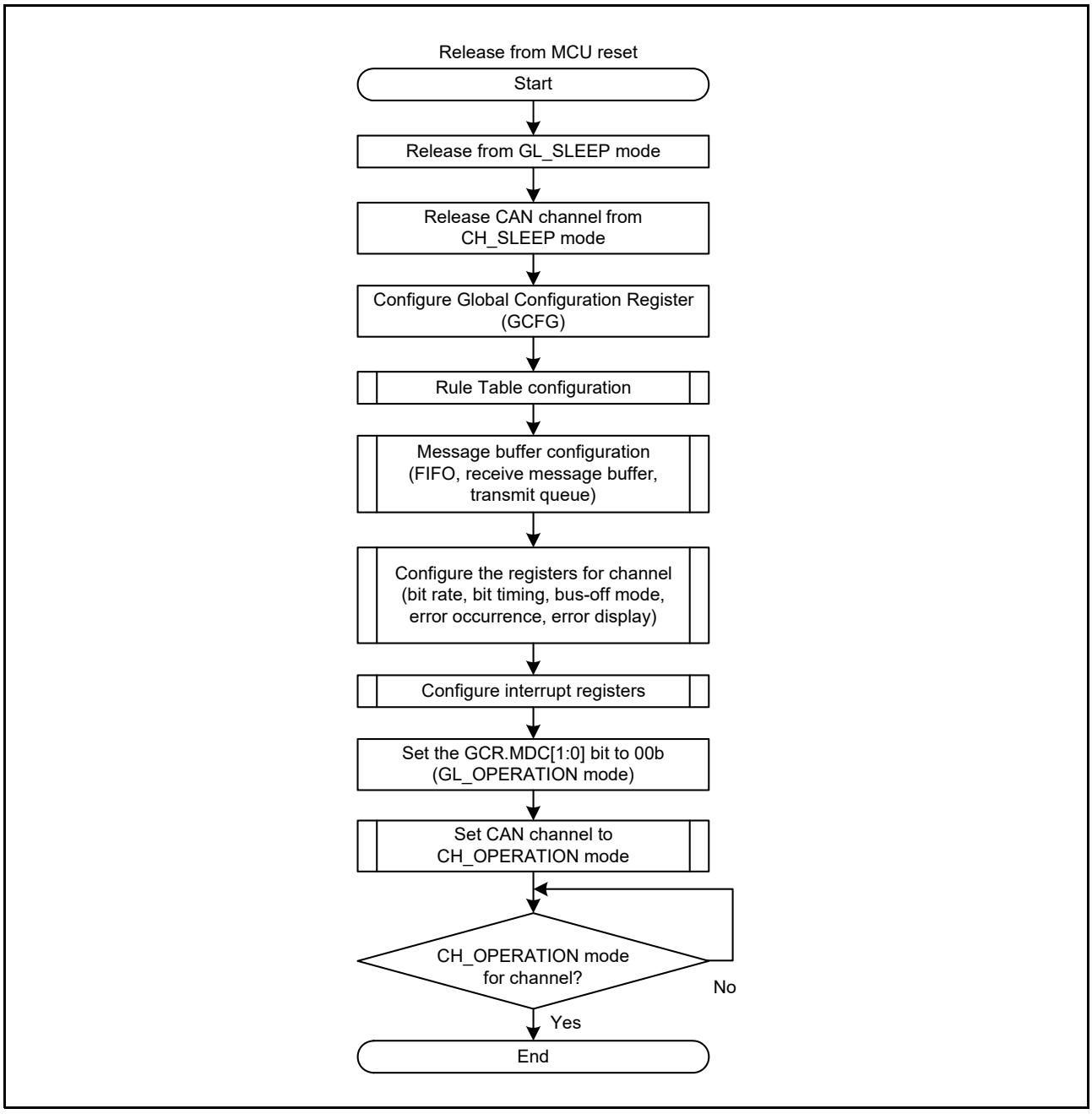


Figure 34.23 Configuration Procedure after MCU Reset

34.5 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.

34.5.1 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.

34.5.2 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.

34.5.3 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.

34.5.4 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.

34.5.5 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 34.24).

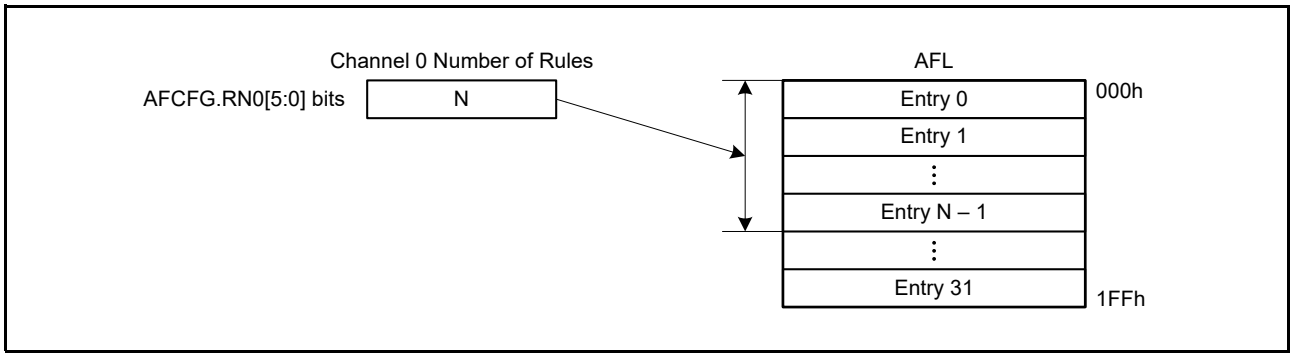


Figure 34.24 Configuration of AFL

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.

34.5.6 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 34.25).
- 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).

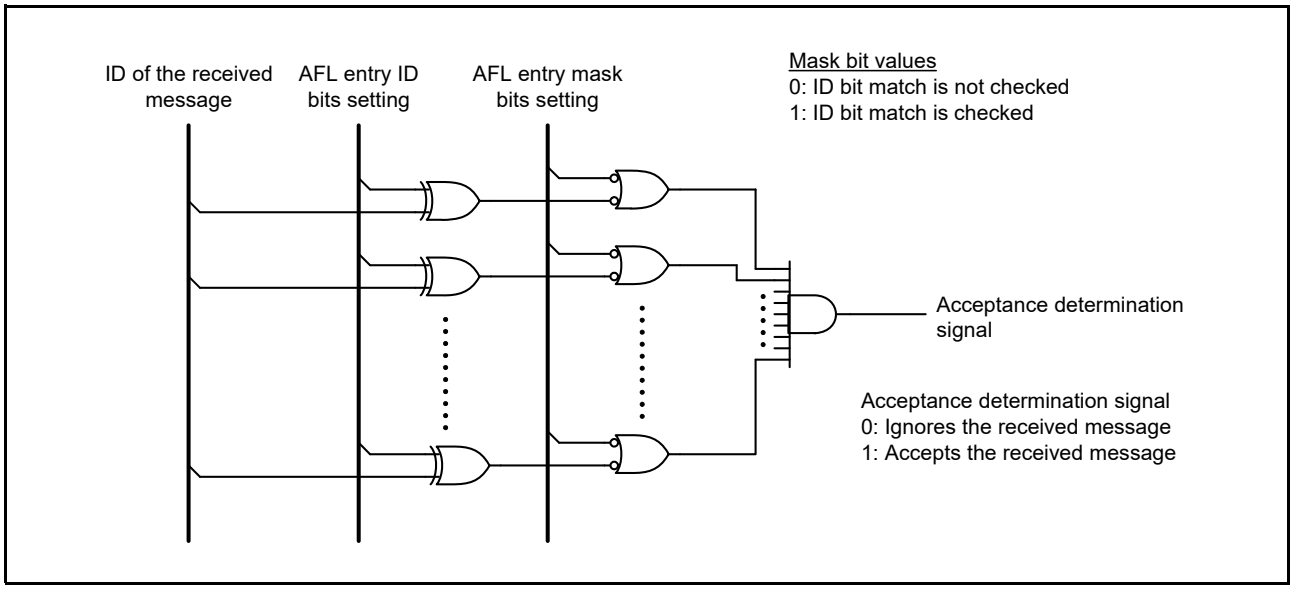


Figure 34.25 Acceptance Filter Function

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 FIFO Destination Enable bit must be set carefully.

34.5.7 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 34.19 shows pages linked to the AFL entries.

Table 34.19 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 34.26). 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.

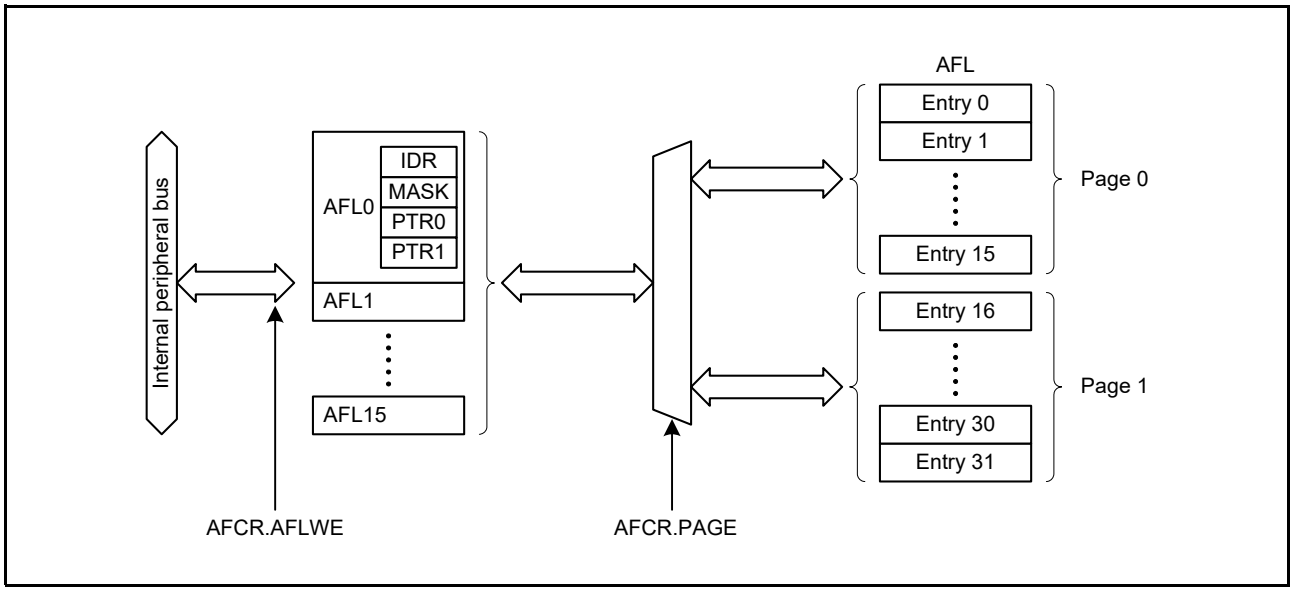


Figure 34.26 AFL Page Access

Follow the flow shown in Figure 34.27 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).

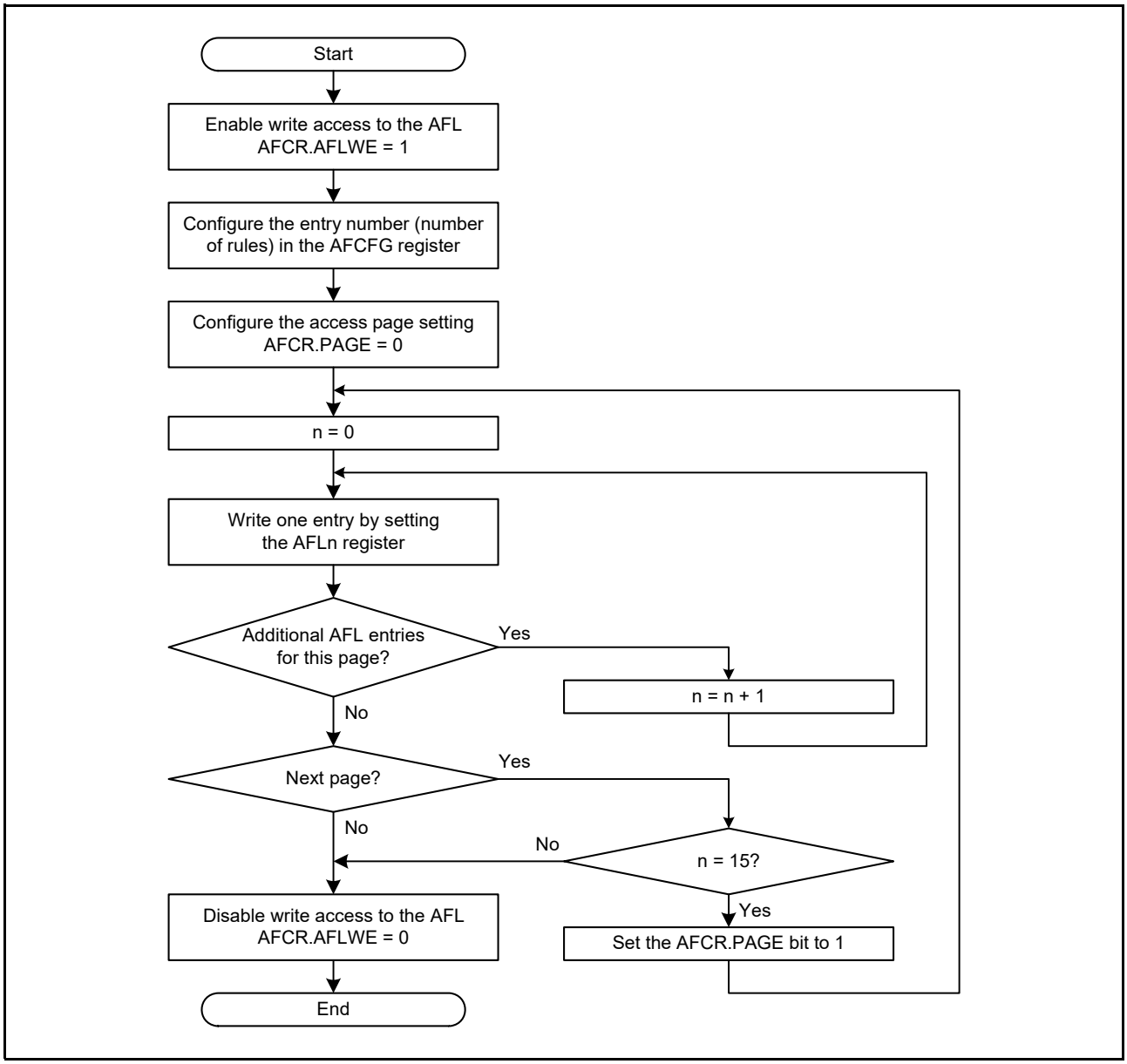


Figure 34.27 AFL Configuration Flow

34.5.8 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 34.20 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 34.20 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.

34.5.9 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.

34.5.10 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 34.28 shows the update flow for an AFL entry.

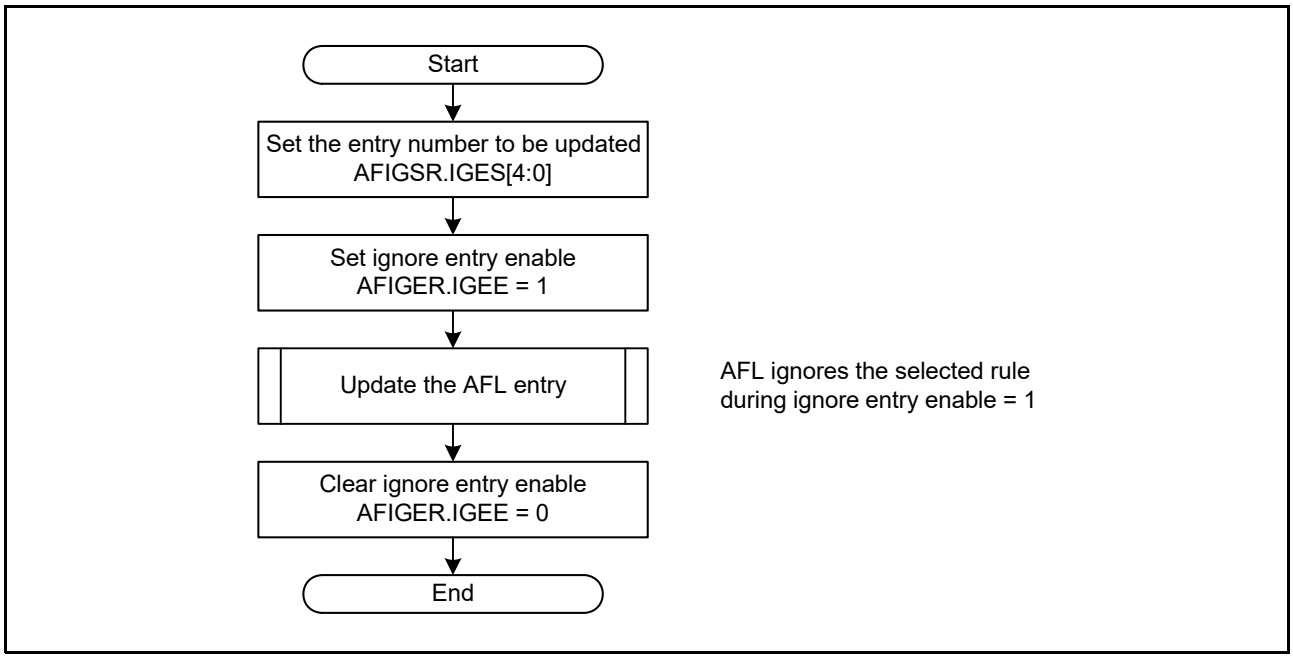


Figure 34.28 Update Flow for An AFL Entry

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.

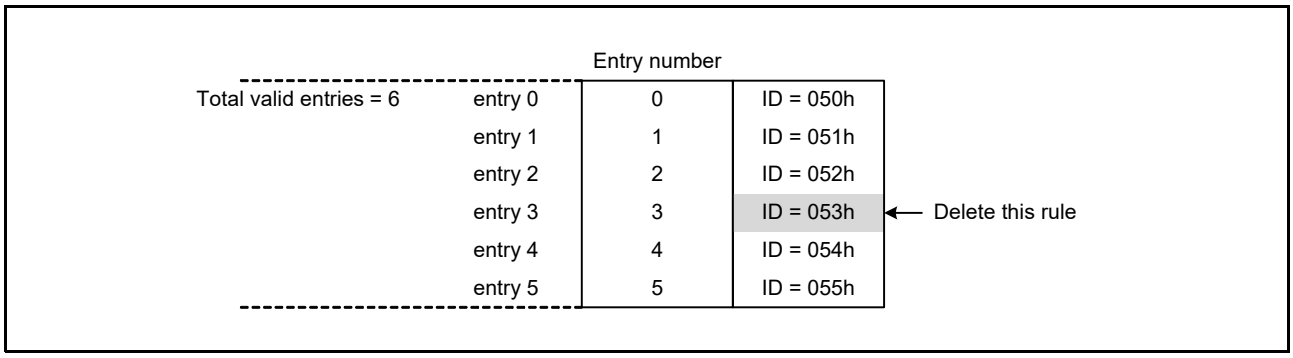


Figure 34.29 Example of Deleting an Entry (Before Deleting Entry 3)

[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.

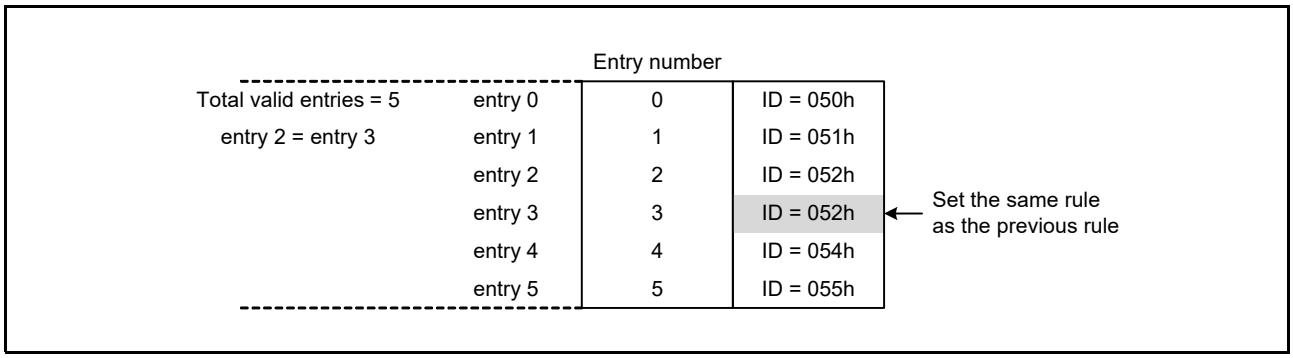


Figure 34.30 Example of Deleting an Entry (After Deleting Entry 3)

(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.

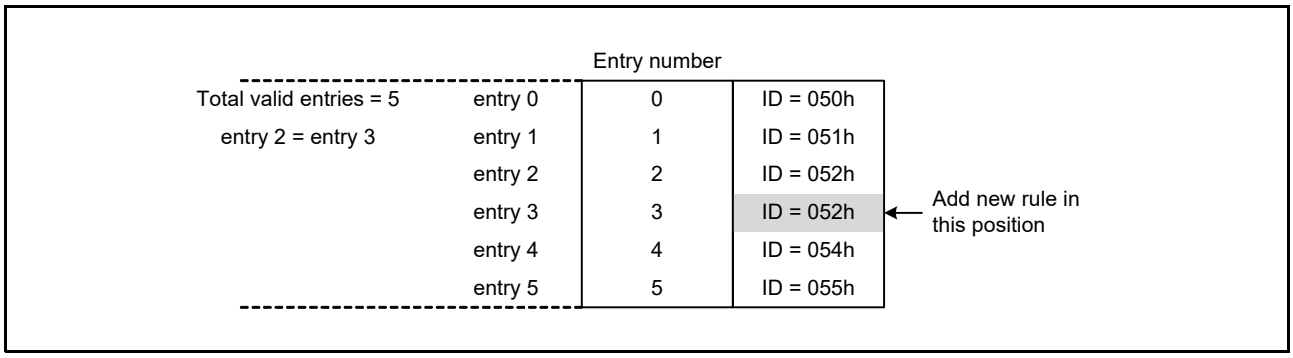


Figure 34.31 Example of Adding an Entry (Before Updating Entry 3)

[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.

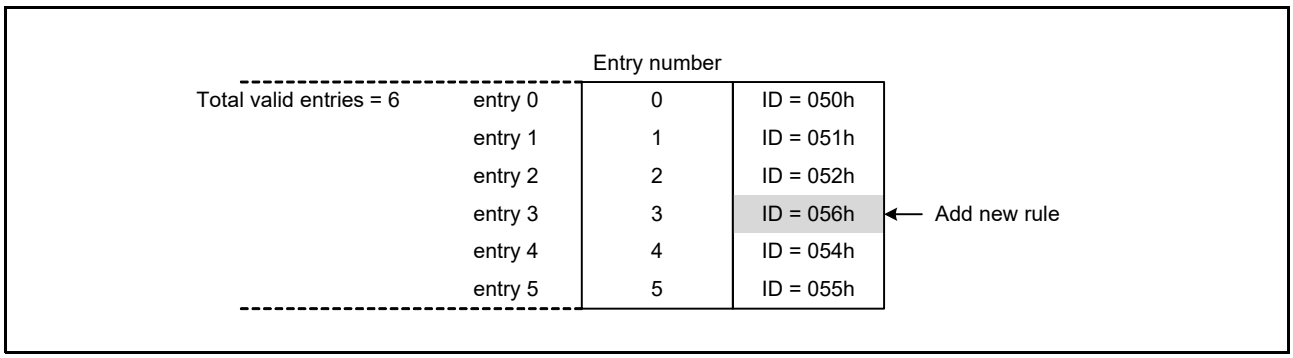


Figure 34.32 Example of Adding an Entry (After Updating Entry 3)

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.

34.6 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 34.33.
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.

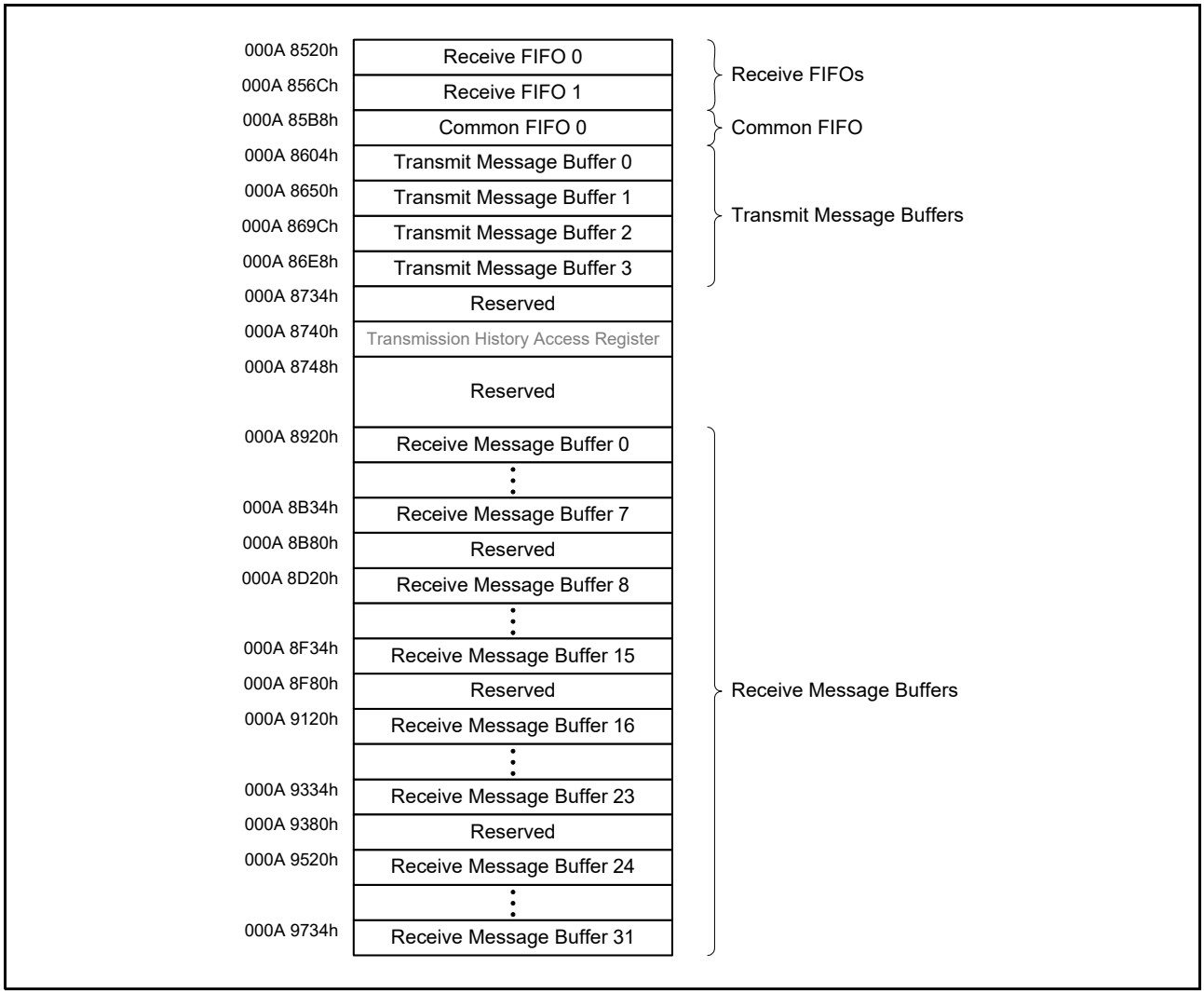


Figure 34.33 Message Buffer Configuration

34.6.1 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.

34.6.1.1 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.

34.6.2 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.

34.6.2.1 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).

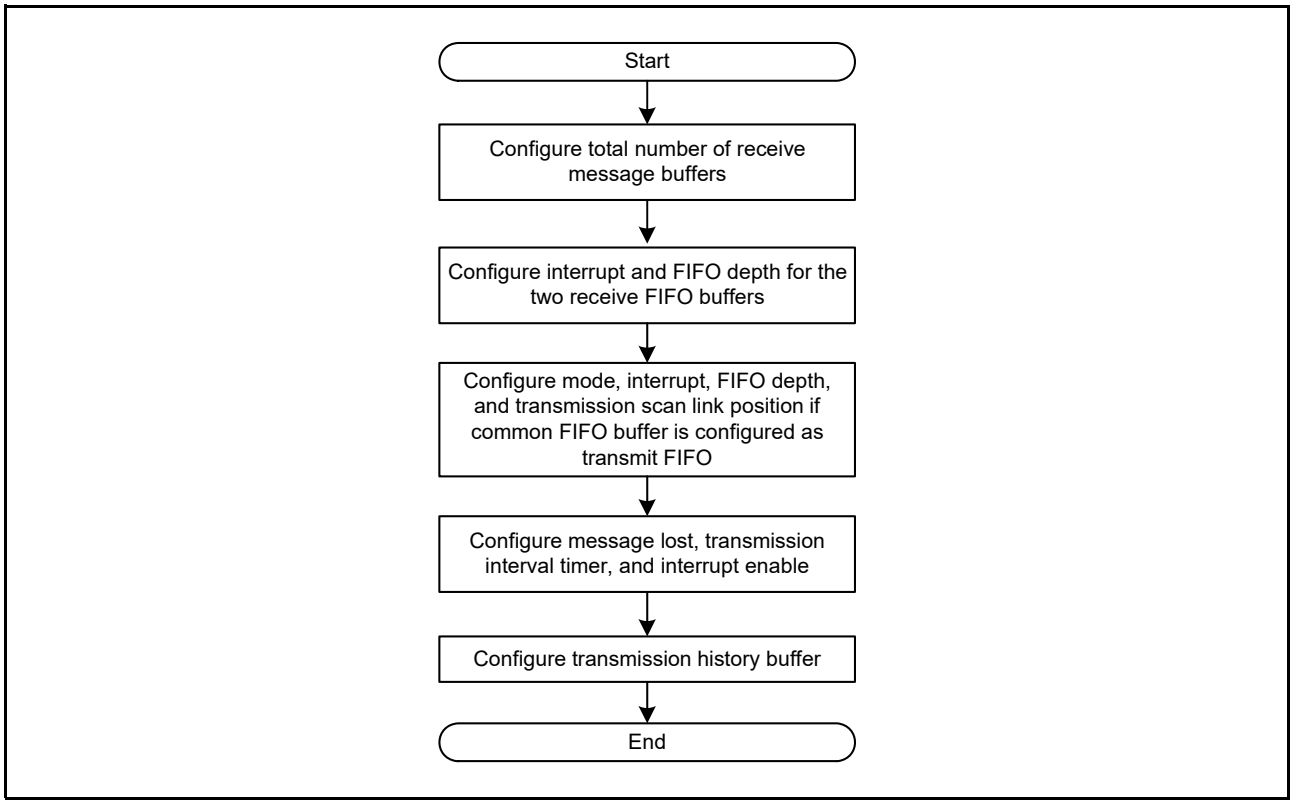


Figure 34.34 FIFO Buffer Configuration Flow in CANFD Module

As shown in Figure 34.34, 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 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 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 transmit 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 34.21, 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 34.21 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							

34.6.2.2 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.

34.7 Reception and Transmission

34.7.1 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

34.7.1.1 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.

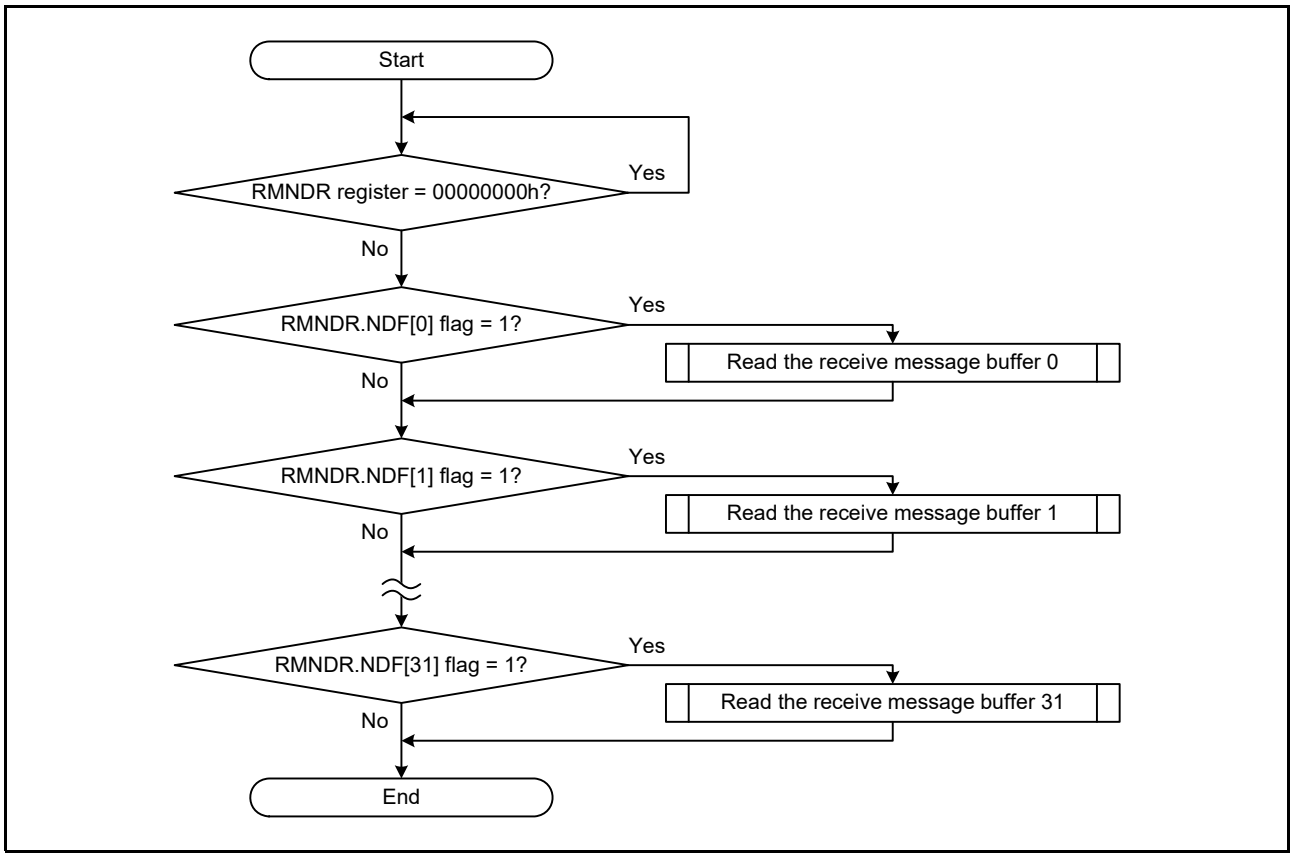


Figure 34.35 Receive Message Buffer Message Access Flow (Example for Polling Case)

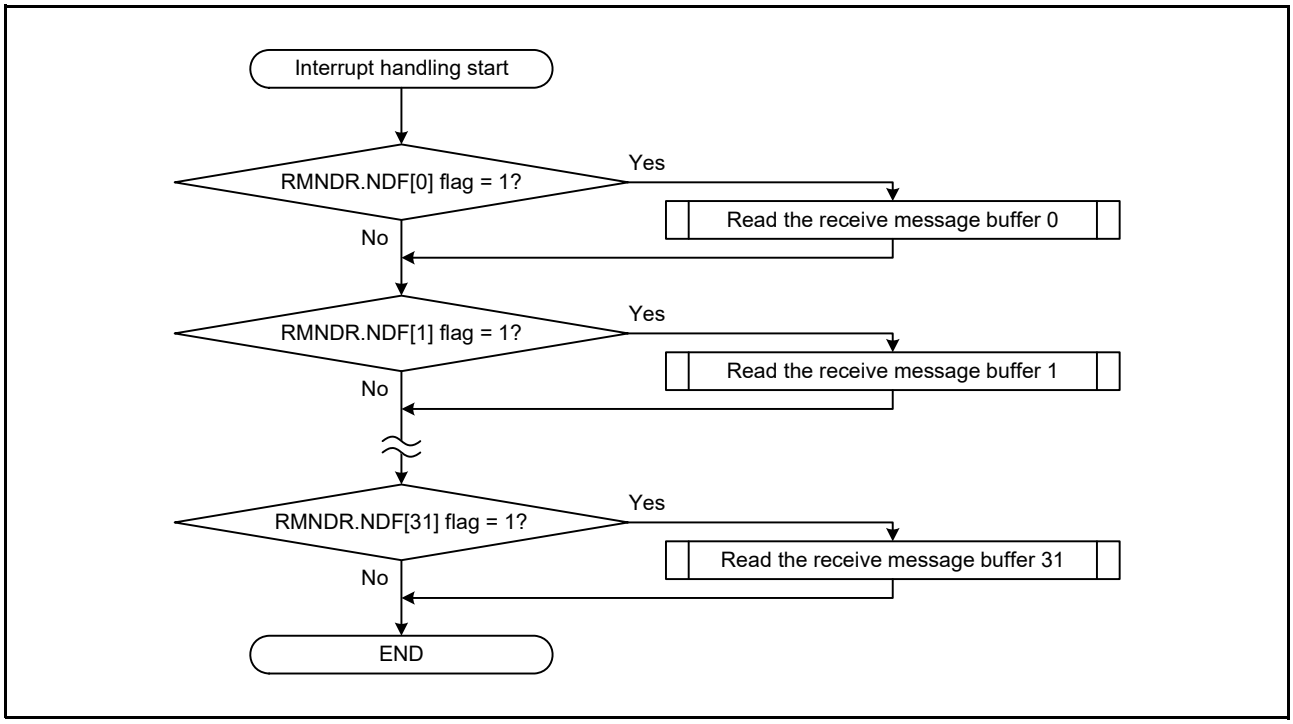


Figure 34.36 Receive Message Buffer Message Access Flow (Example for Interrupt Case)

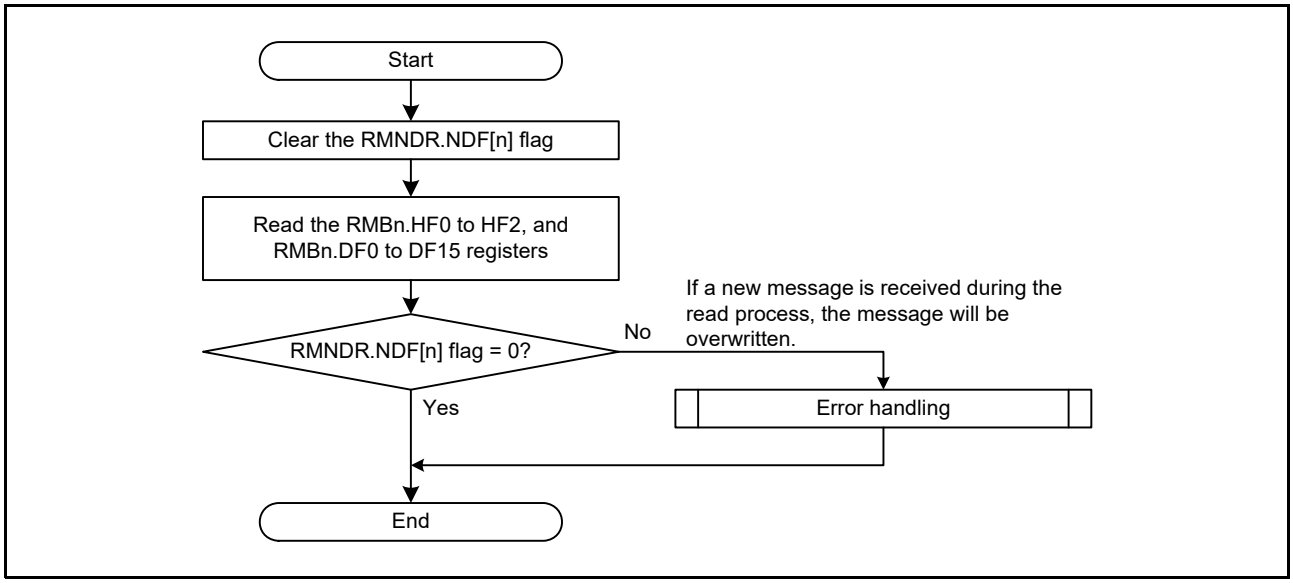


Figure 34.37 Receive Message Buffer n Read Flow

34.7.1.2 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.

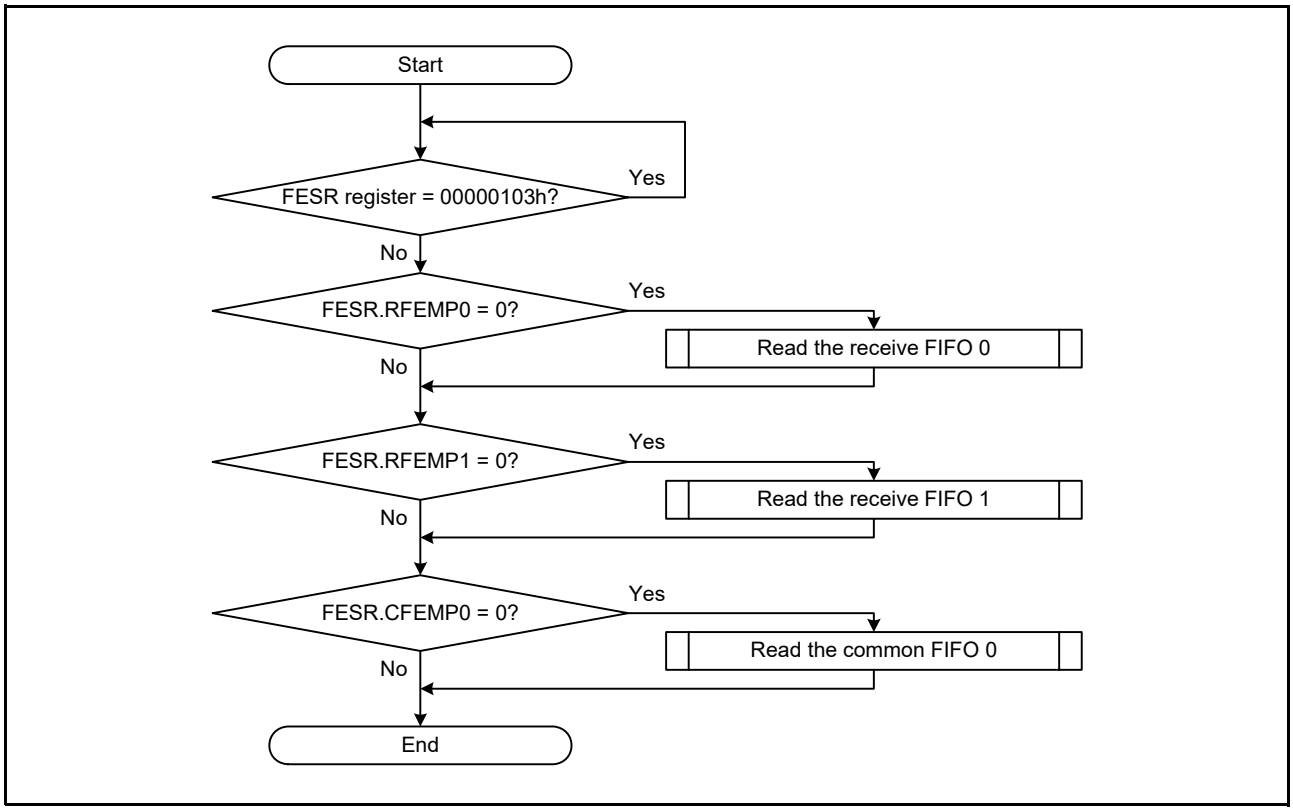


Figure 34.38 FIFO Buffer Message Access Flow (Example for Polling Case)

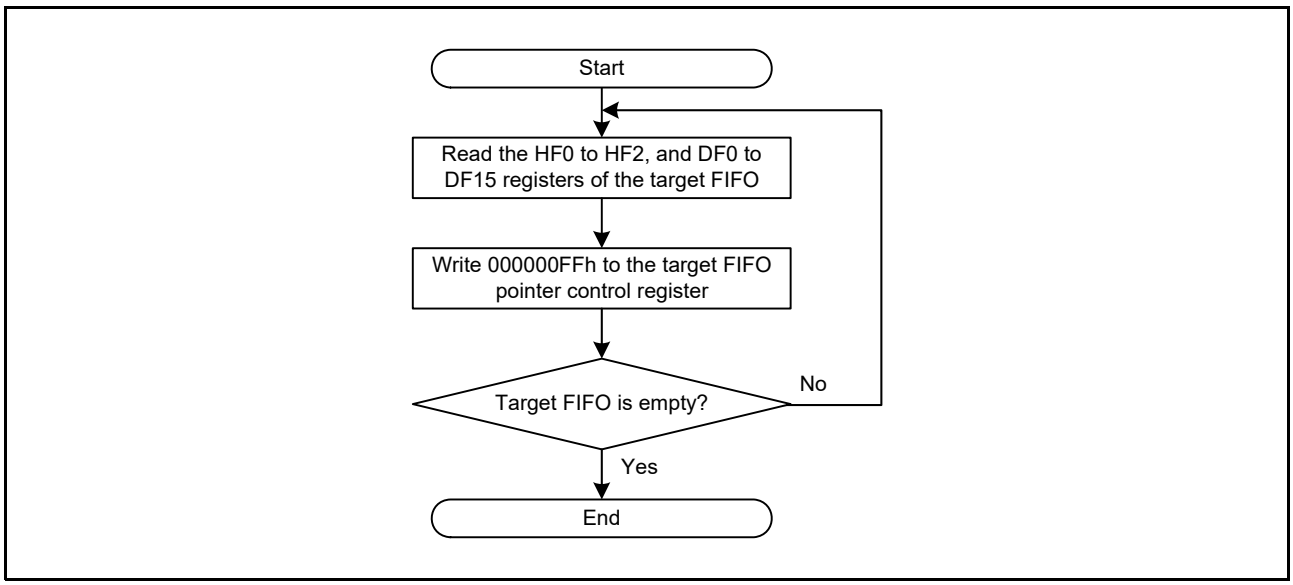


Figure 34.39 Receive FIFO Read Flow (Example for Polling Case)

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.

34.7.1.3 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).

34.7.2 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 34.33):

- 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.

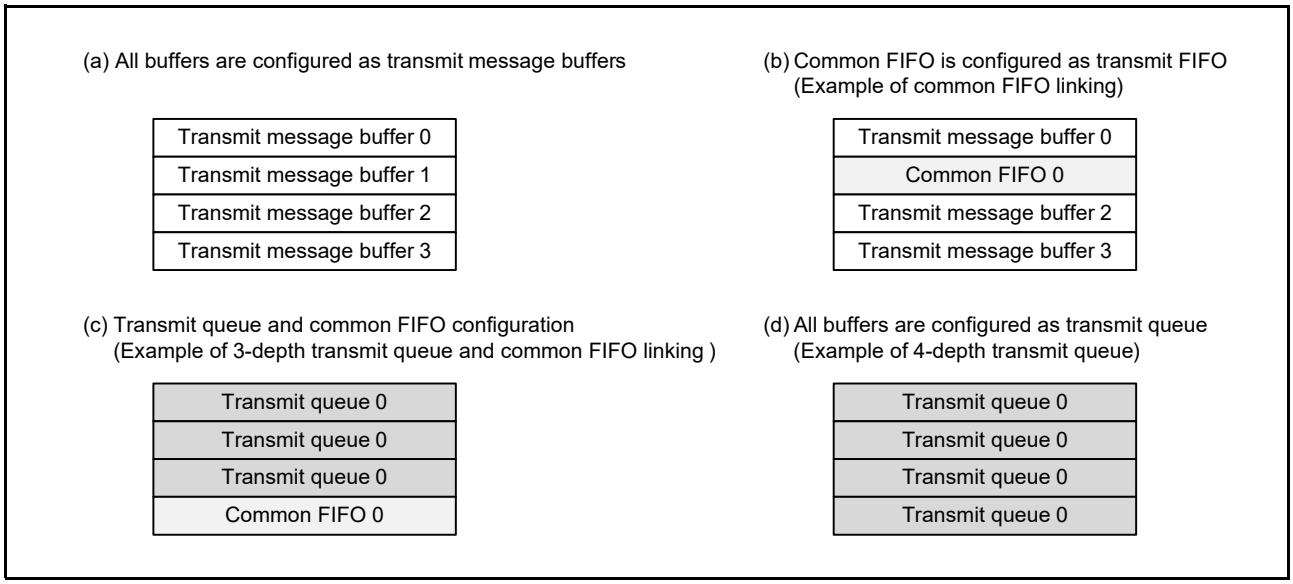


Figure 34.40 Channel Transmit Message Buffer Configuration

34.7.2.1 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 34.41 below shows the transmission configuration flow.

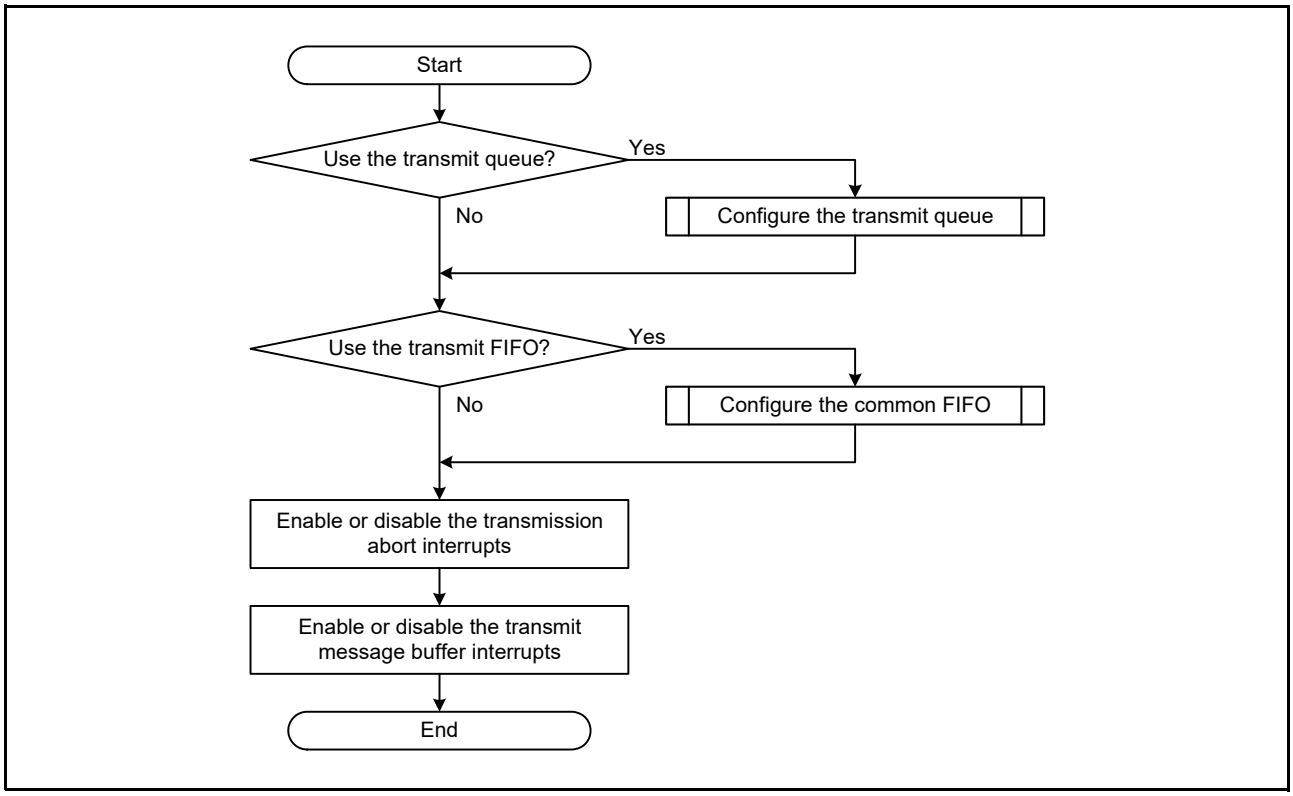


Figure 34.41 Transmission Configuration Flow

34.7.2.2 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 34.42 shows the transmission request procedure from transmit message buffer.

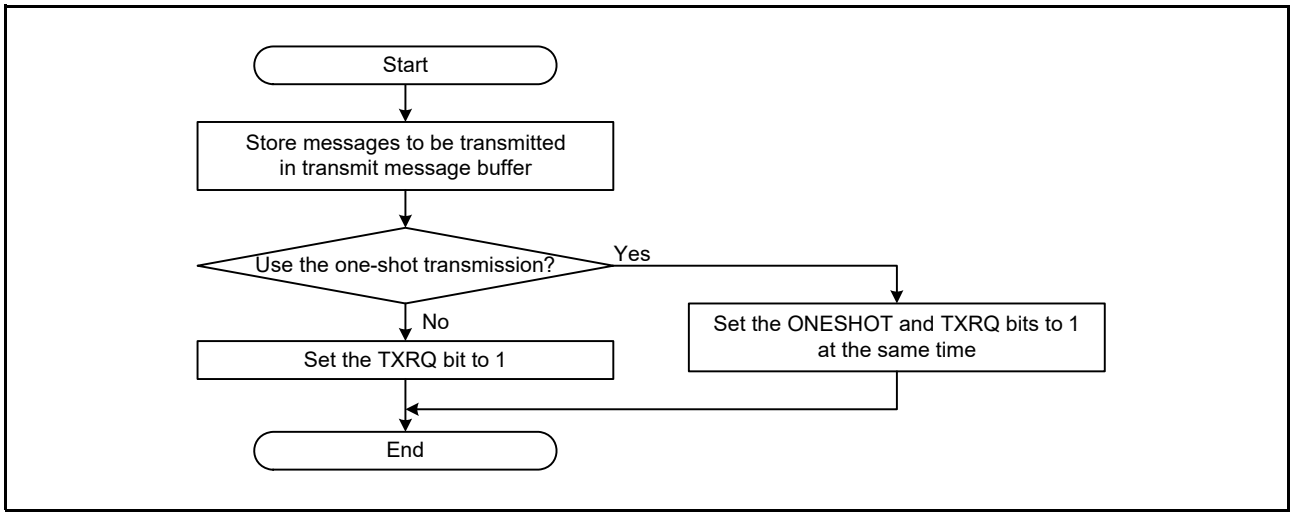


Figure 34.42 Transmission Request Procedure from Transmit Message Buffer

Table 34.22 shows configuration of the TMCRn register.

Table 34.22 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 34.43 shows timings for successful transmission from two message buffers.

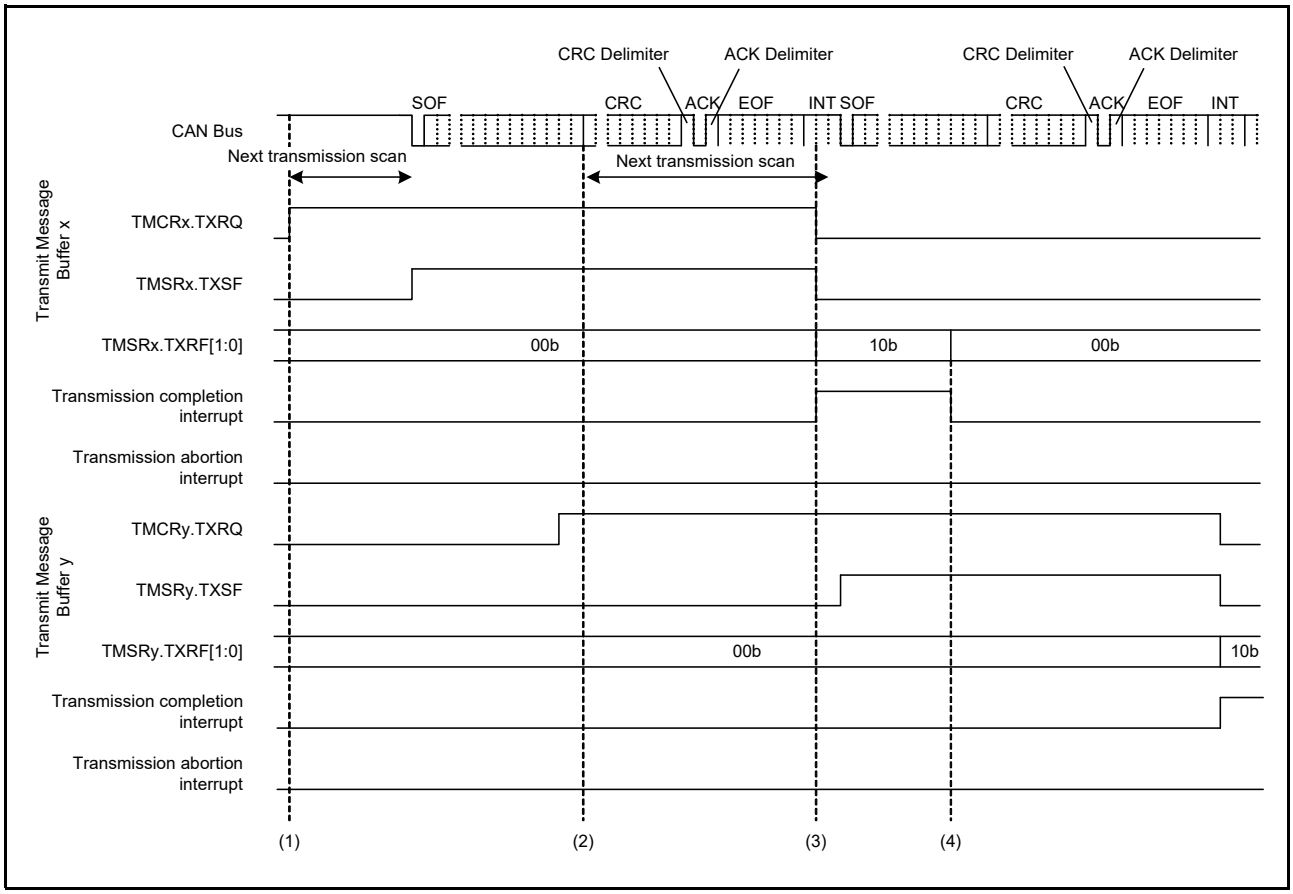


Figure 34.43 Timing of Request and Flag Bits for Successful Transmission

(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 34.44 shows timings for transmission abort for two message buffers.

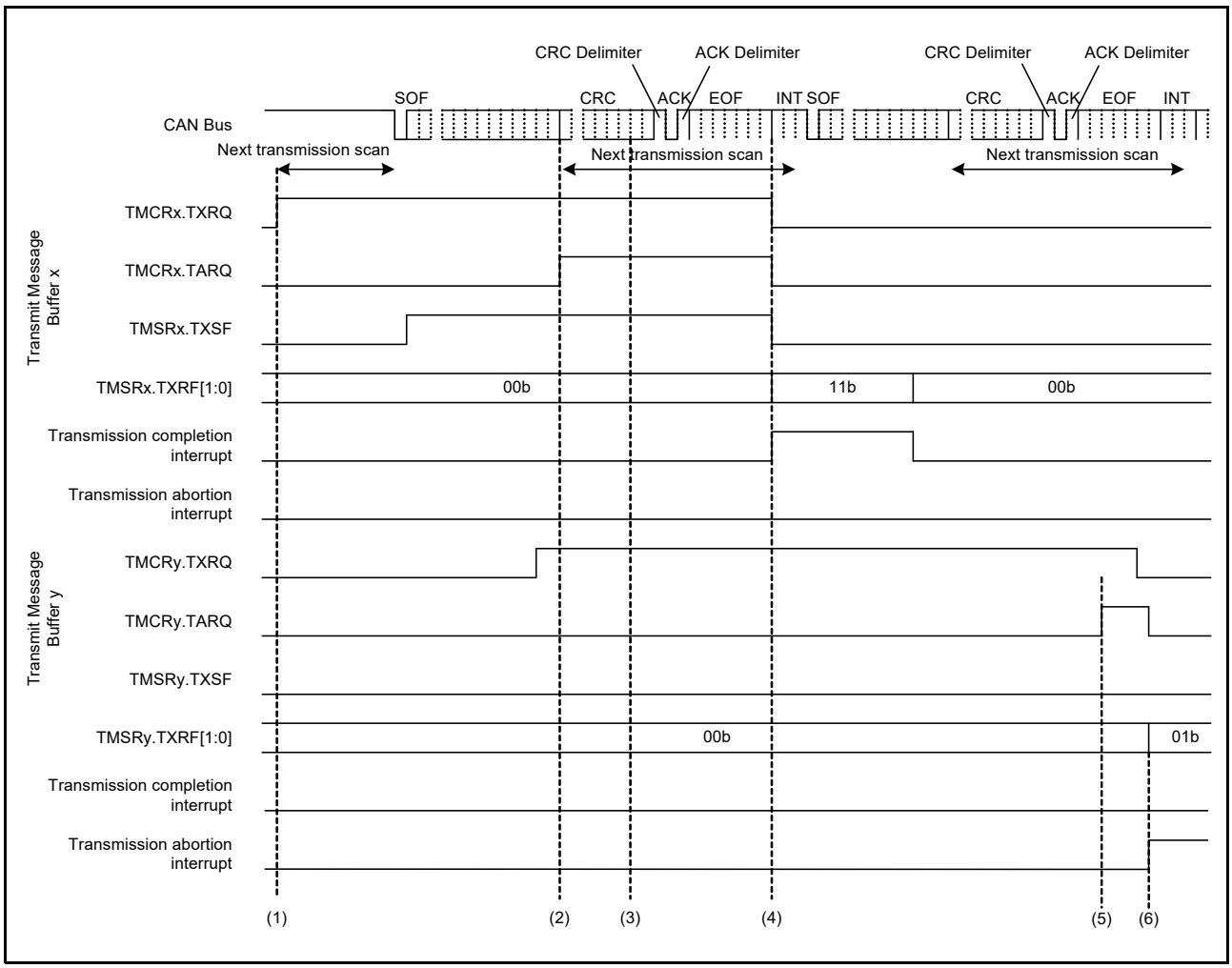


Figure 34.44 Timing of Request and Flag Bits for Transmission Abort

(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.

34.7.2.3 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 34.45.

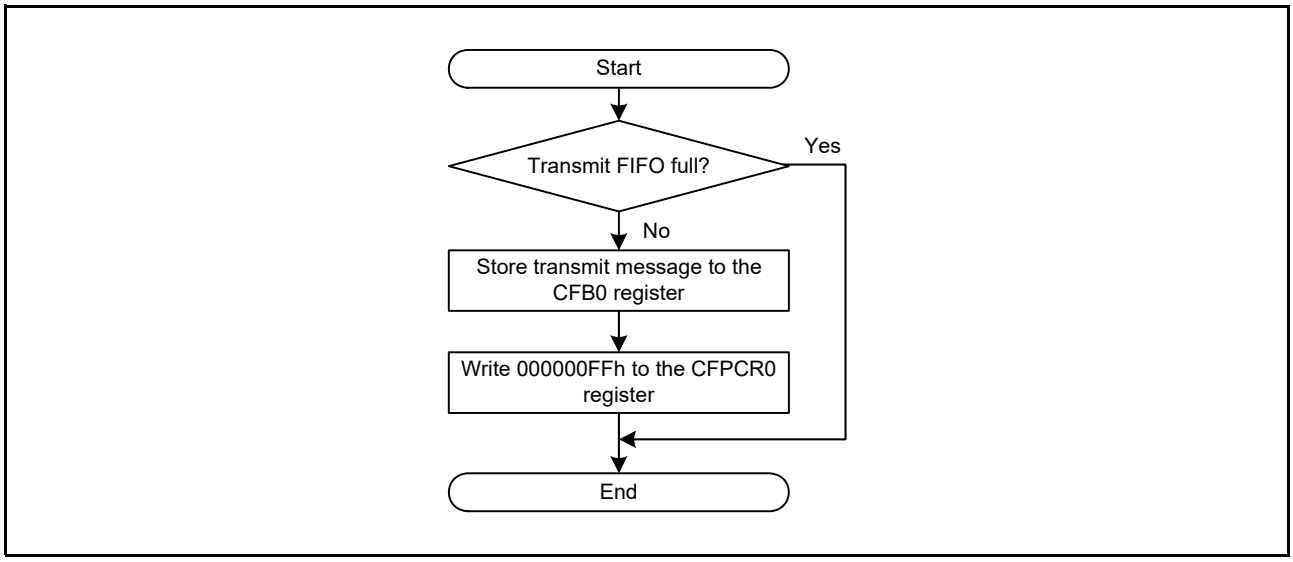


Figure 34.45 Transmit FIFO Transmission Request Procedure

(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 34.23 for configuration values of the GCFG.ITP[15:0] bits to achieve different reference clock periods based on the PCLKB frequency/period.

Table 34.23 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

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 34.46 shows an example timing of the internal processing.

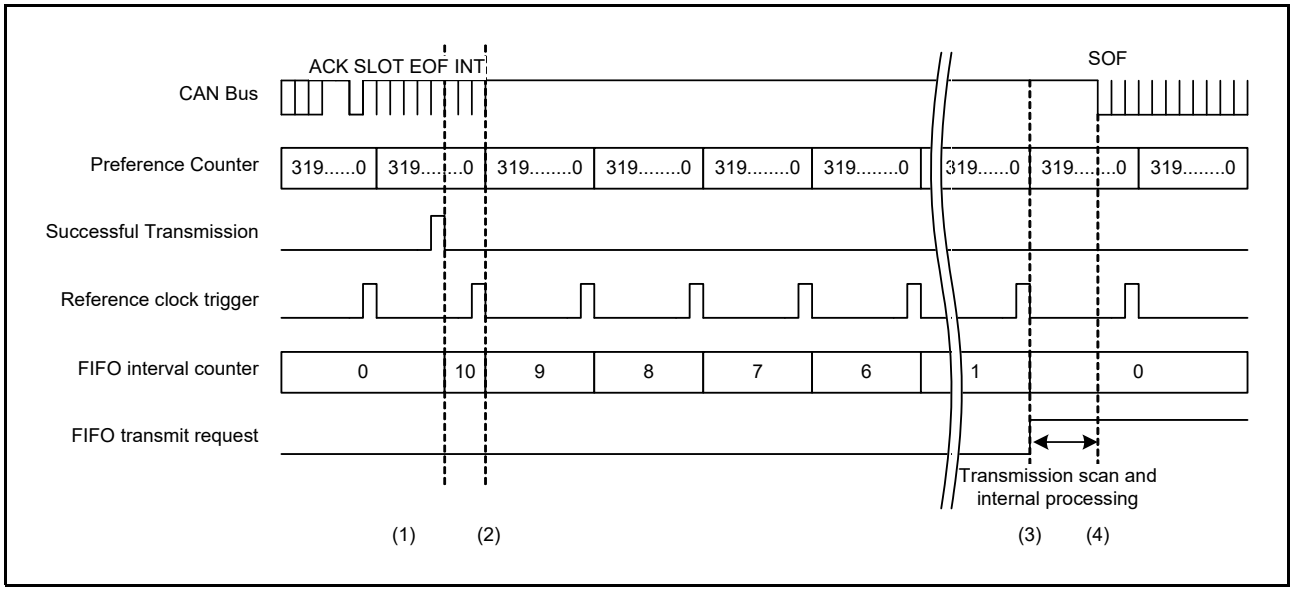


Figure 34.46 Example for Interval Processing Time

The configuration for this timing above is following:

- PCLKB frequency = 32 MHz
- Interval timer prescaler (GCFG.ITP[15:0]) = Divided by 320
- 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 34.46, 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 34.47 shows a block diagram of the FIFO interval timer.

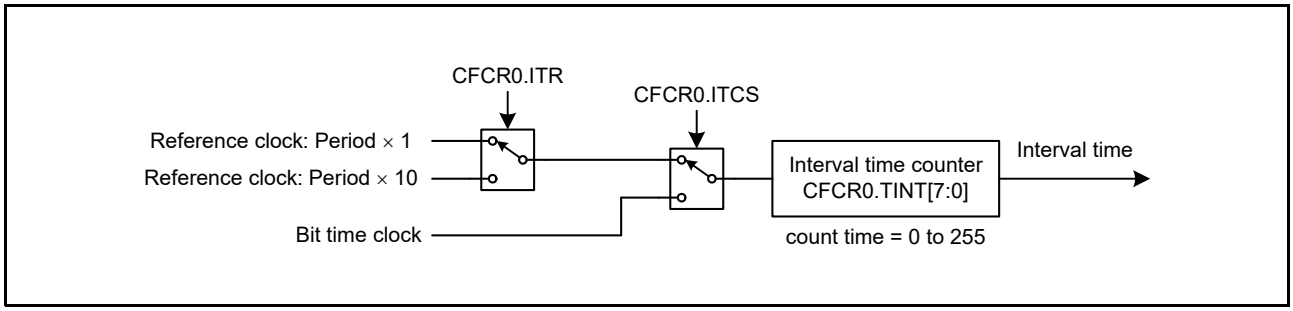


Figure 34.47 Block Diagram of FIFO Interval Timer

34.7.2.4 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 34.48.

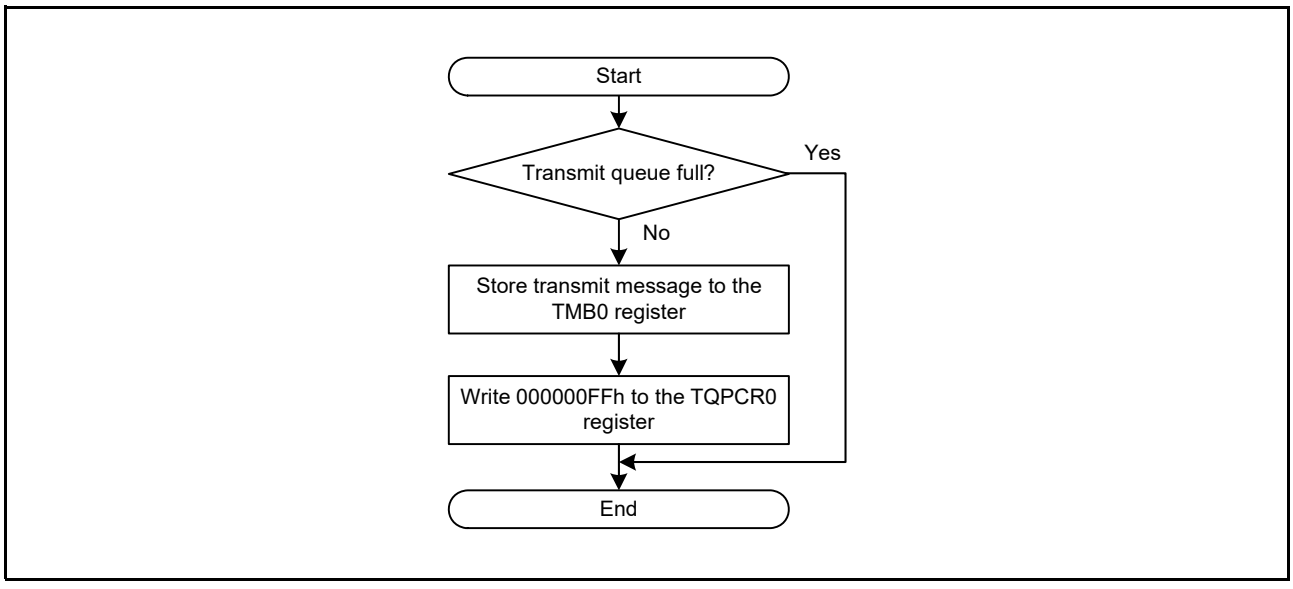


Figure 34.48 Transmit Queue Transmission Request

34.7.2.5 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 34.24.
- 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 34.24 Transmission History Buffer Number Entry

Transmitted Buffer Number (THACR0.BN[1:0])	Transmitted Buffer Type (THACR0.BT[2:0])		
	001b	010b	100b
	Transmit Message Buffer	Transmit FIFO	Transmit Queue
00b	TMB0	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	TMB1		
10b	TMB2		
11b	TMB3		

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 34.49 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.

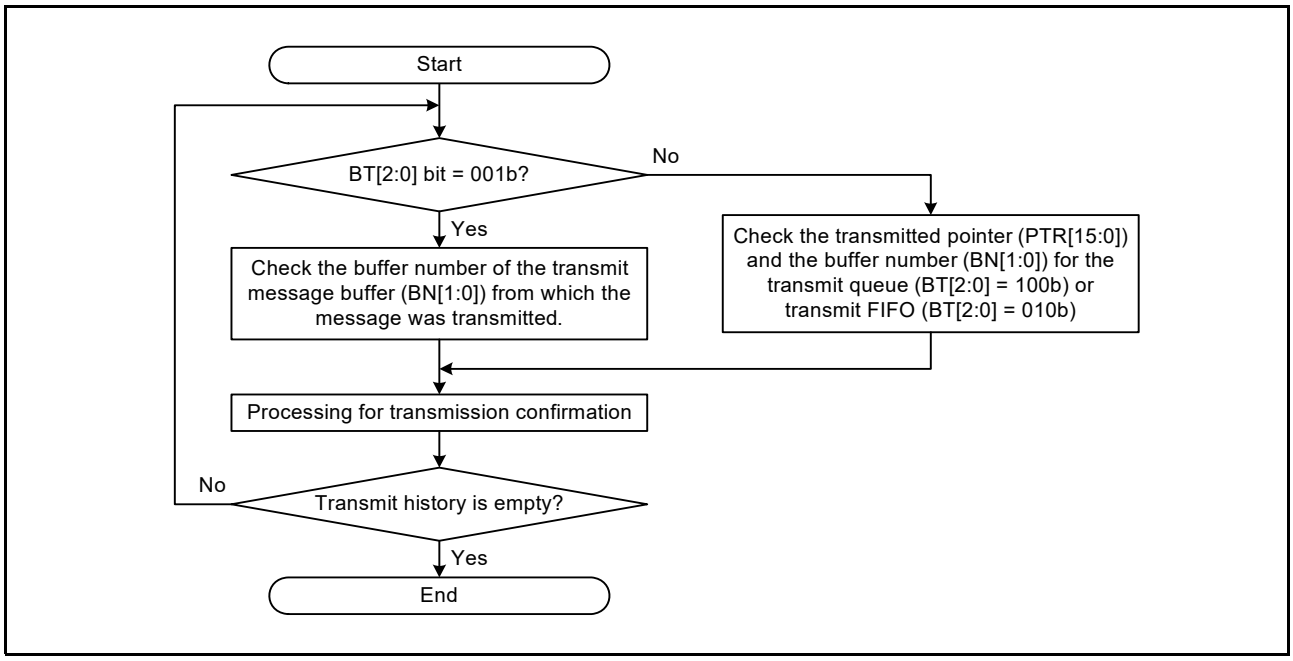


Figure 34.49 Example of Transmission History Processing Flow

34.7.2.6 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.

34.8 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.

34.8.1 ECC Function Setting

Figure 34.50 shows a procedure for ECC function setting.

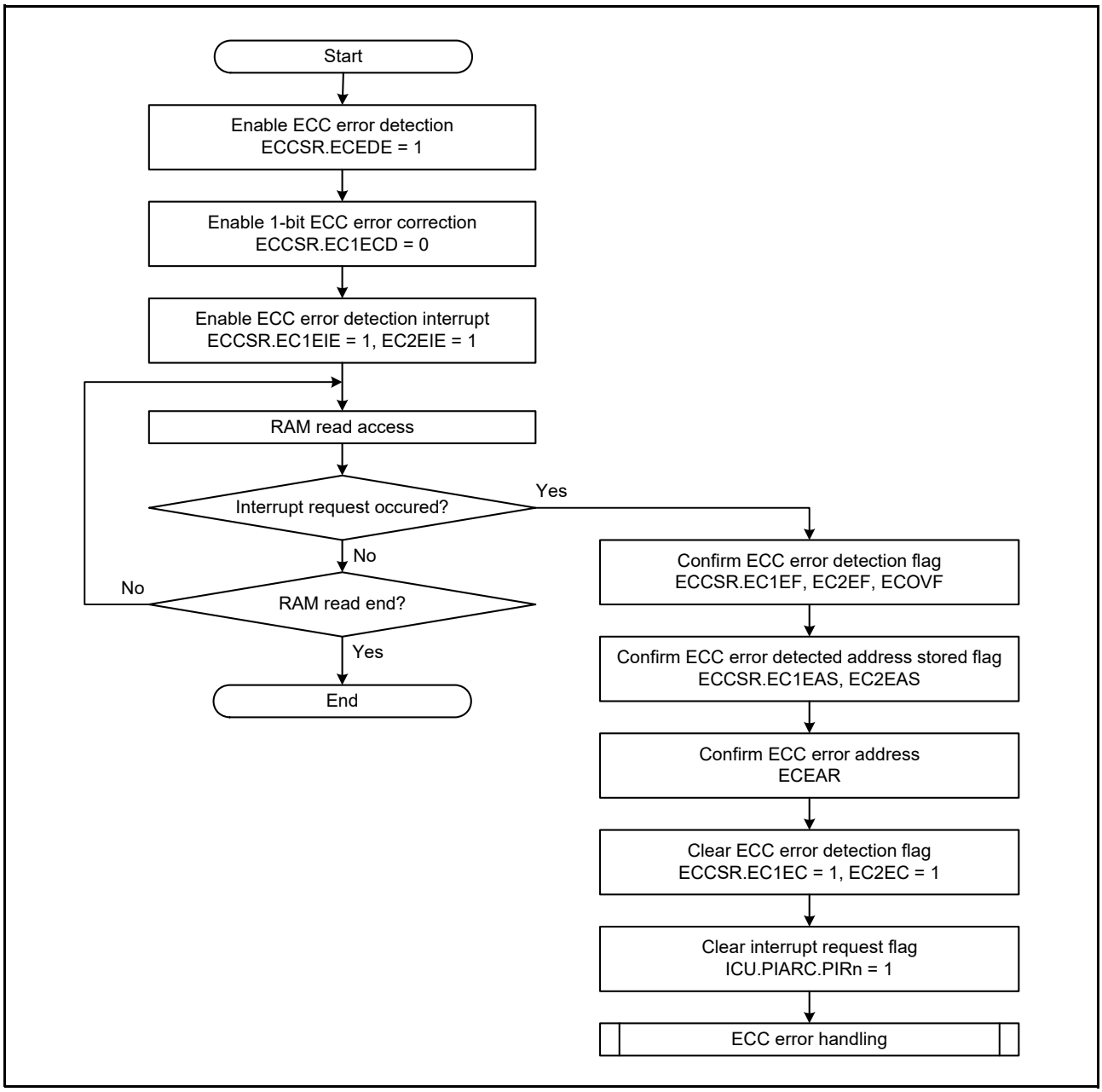


Figure 34.50 Setting Procedure for ECC Function

34.8.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 34.51 shows a procedure for ECC decoder testing.

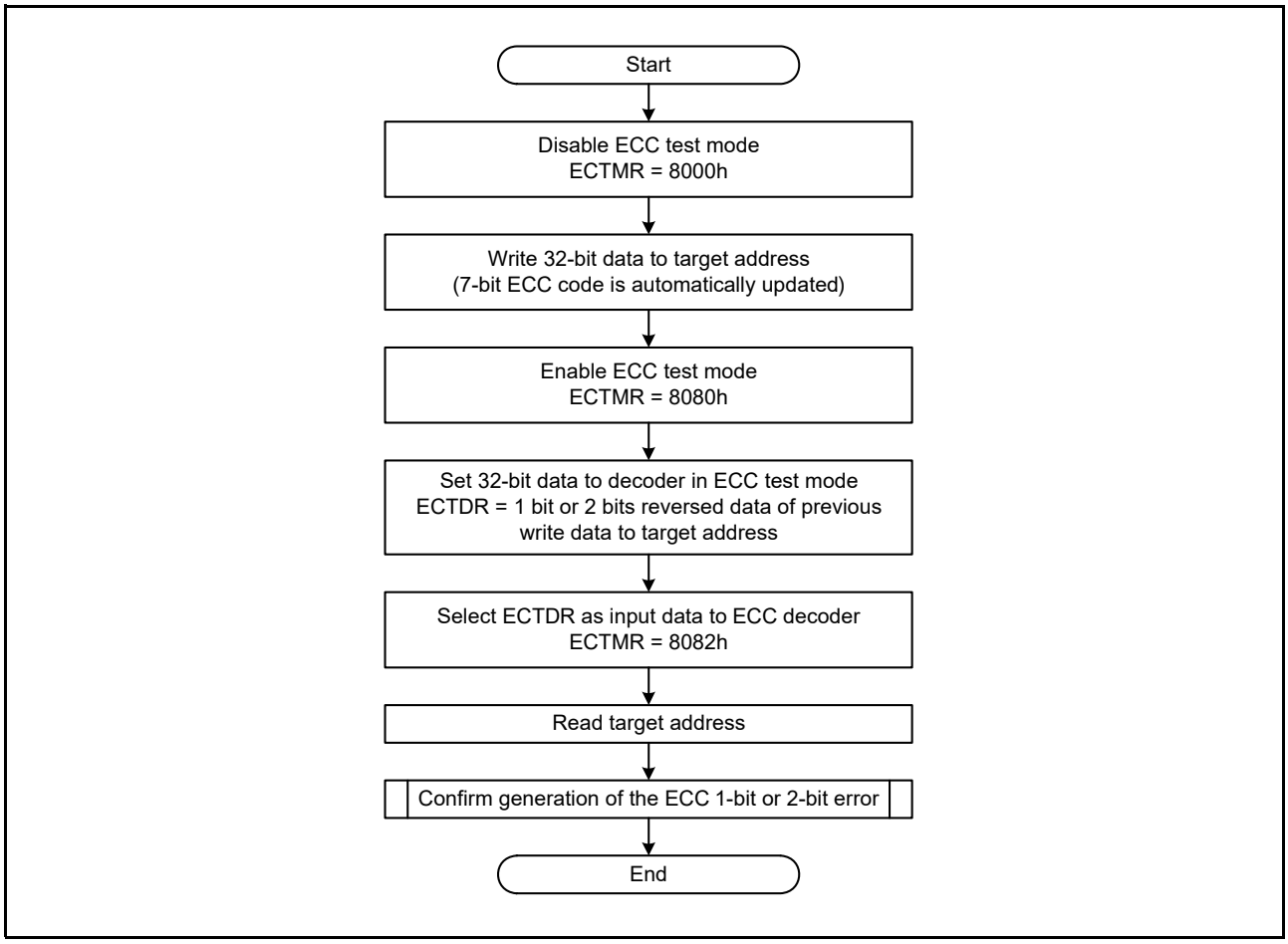


Figure 34.51 Testing Procedure for ECC Decoder

34.9 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

34.9.1 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

34.9.1.1 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.

34.9.1.2 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.

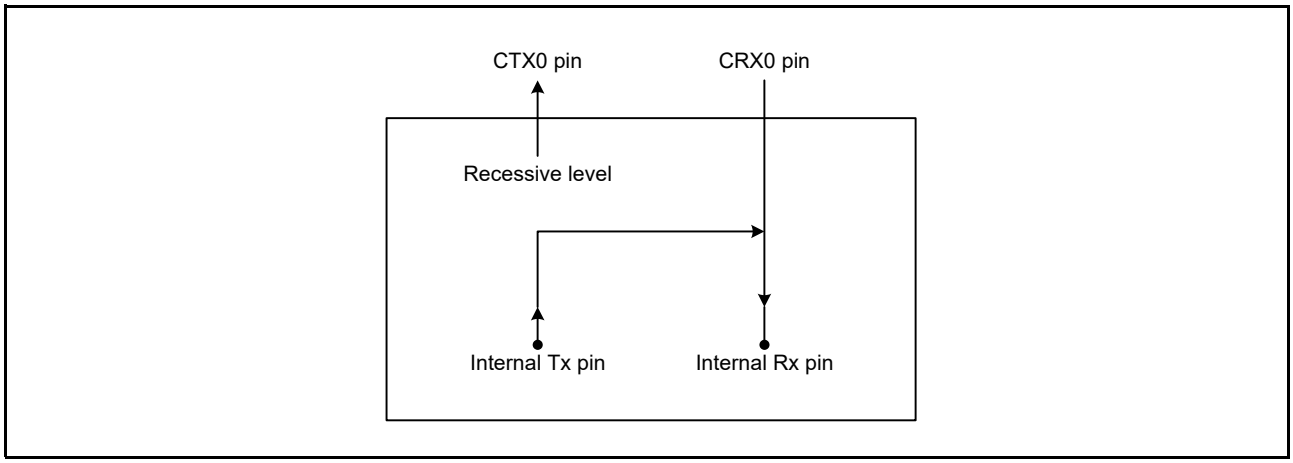


Figure 34.52 Listen-only Mode Configuration

34.9.1.3 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.

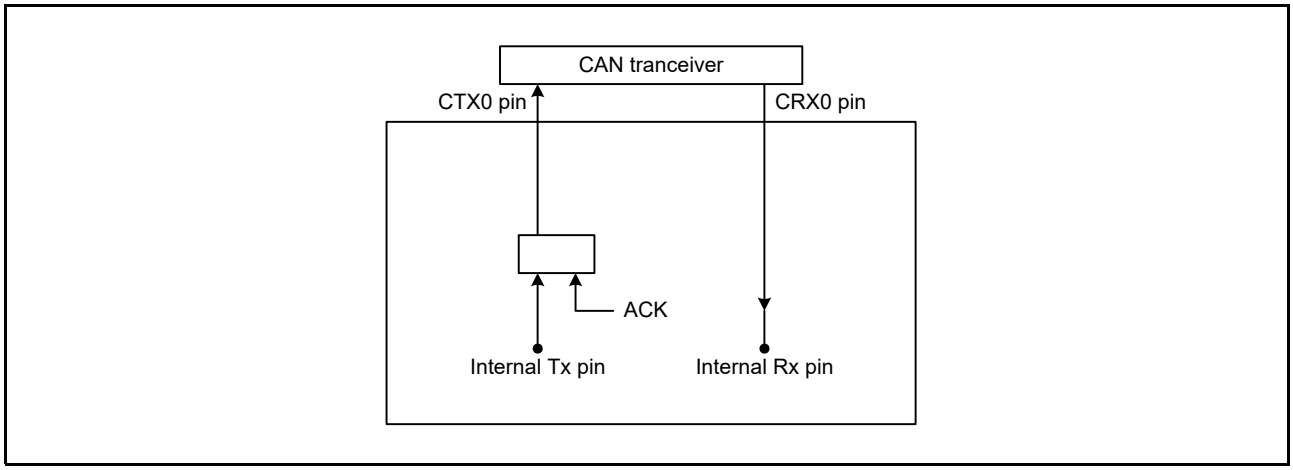


Figure 34.53 Self Test Mode 0 Configuration

34.9.1.4 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.

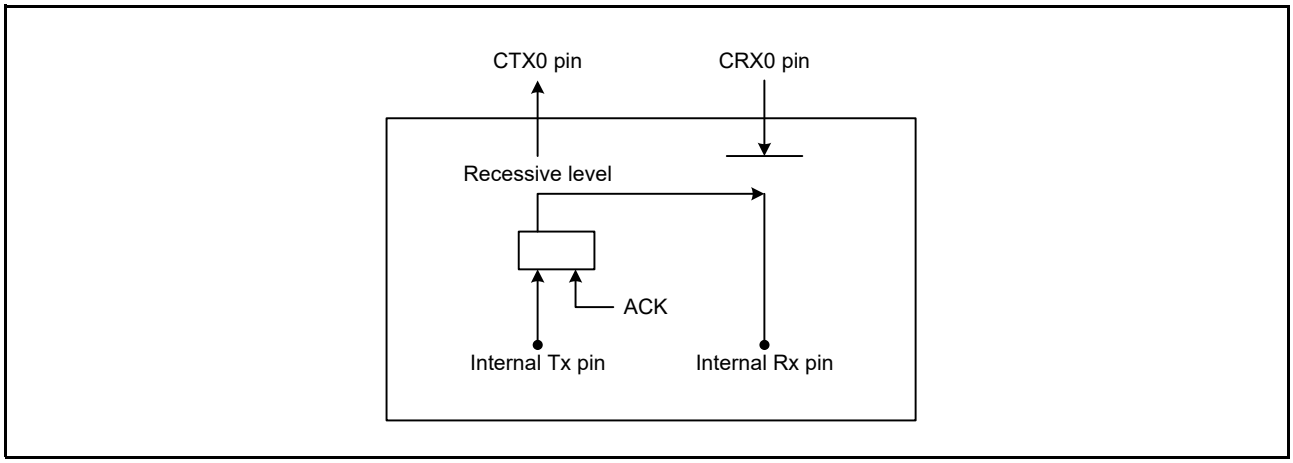


Figure 34.54 Self Test Mode 1 Configuration

34.9.1.5 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.

34.9.2 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 34.25 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.

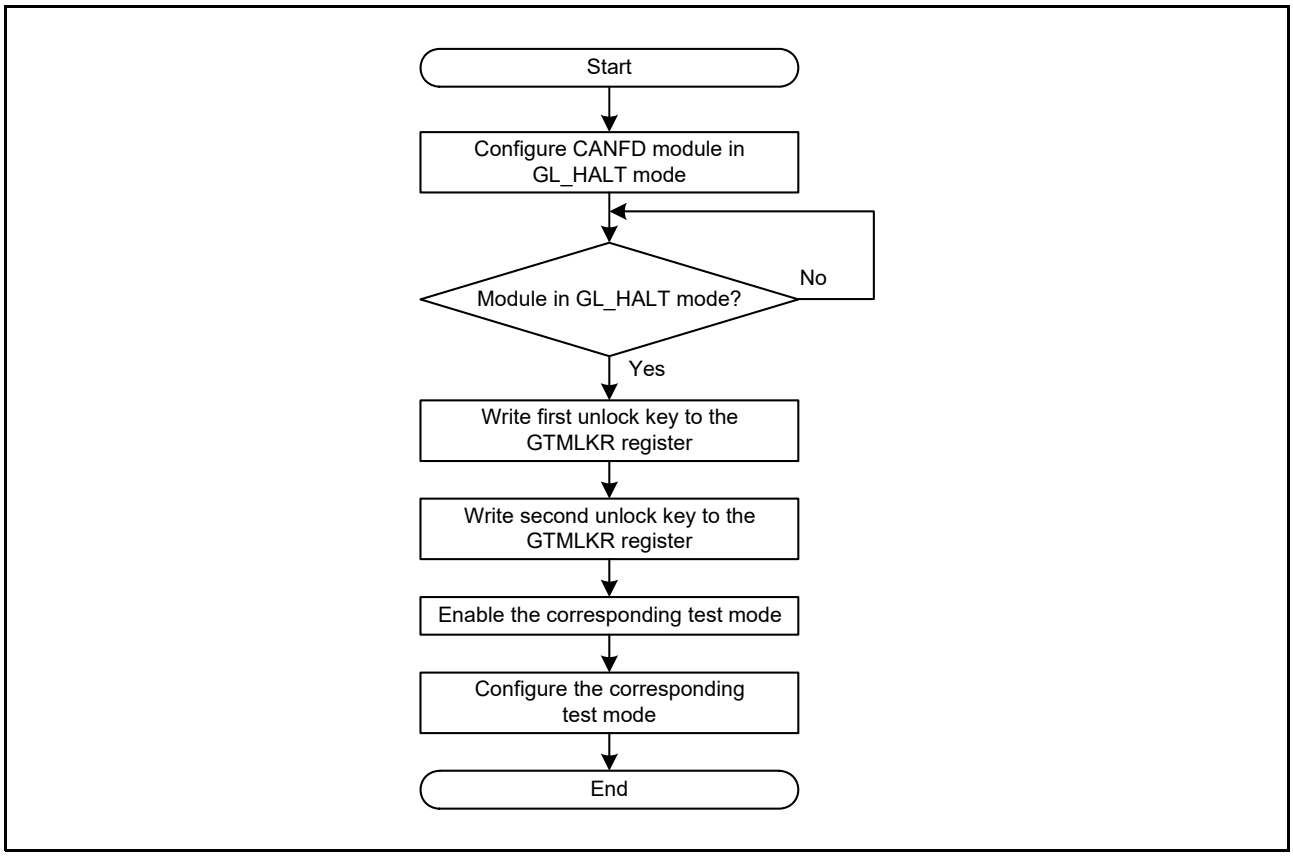


Figure 34.55 Unlock Software Protection Routine

34.9.2.1 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 34.56 shows the structure of the pages in the RAM when performing a RAM test mode.

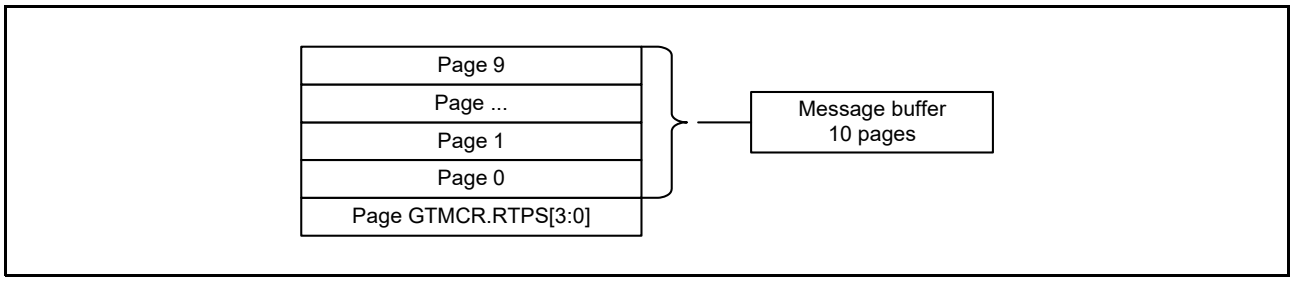


Figure 34.56 RAM Page Structure

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 34.57 below shows the software flow for RAM test mode.

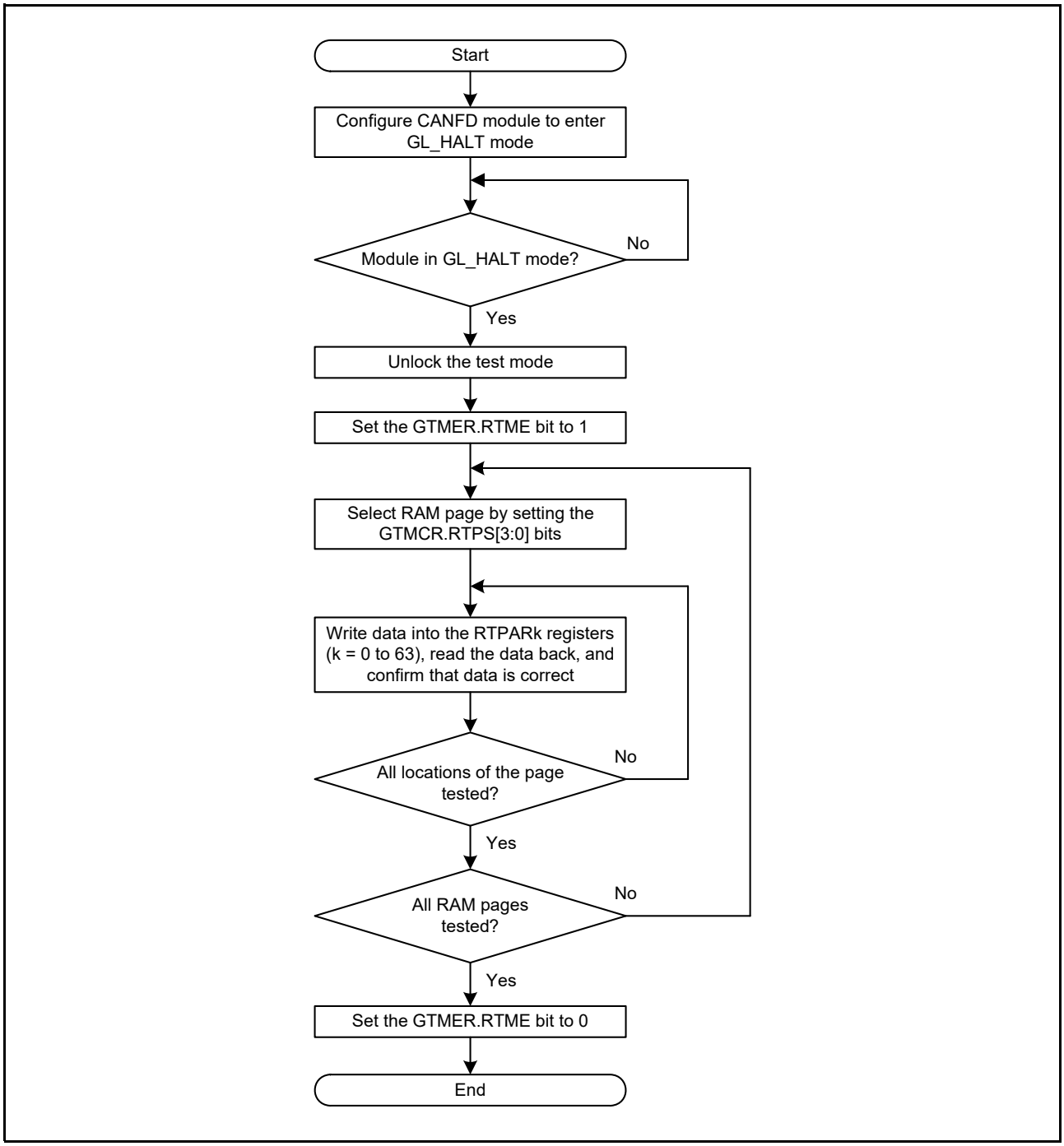


Figure 34.57 RAM Test Mode Software Flow

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.

34.9.2.2 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 CHCR.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.

34.10 Interrupts and DTC/DMA Requests

34.10.1 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 34.26 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 34.26 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.

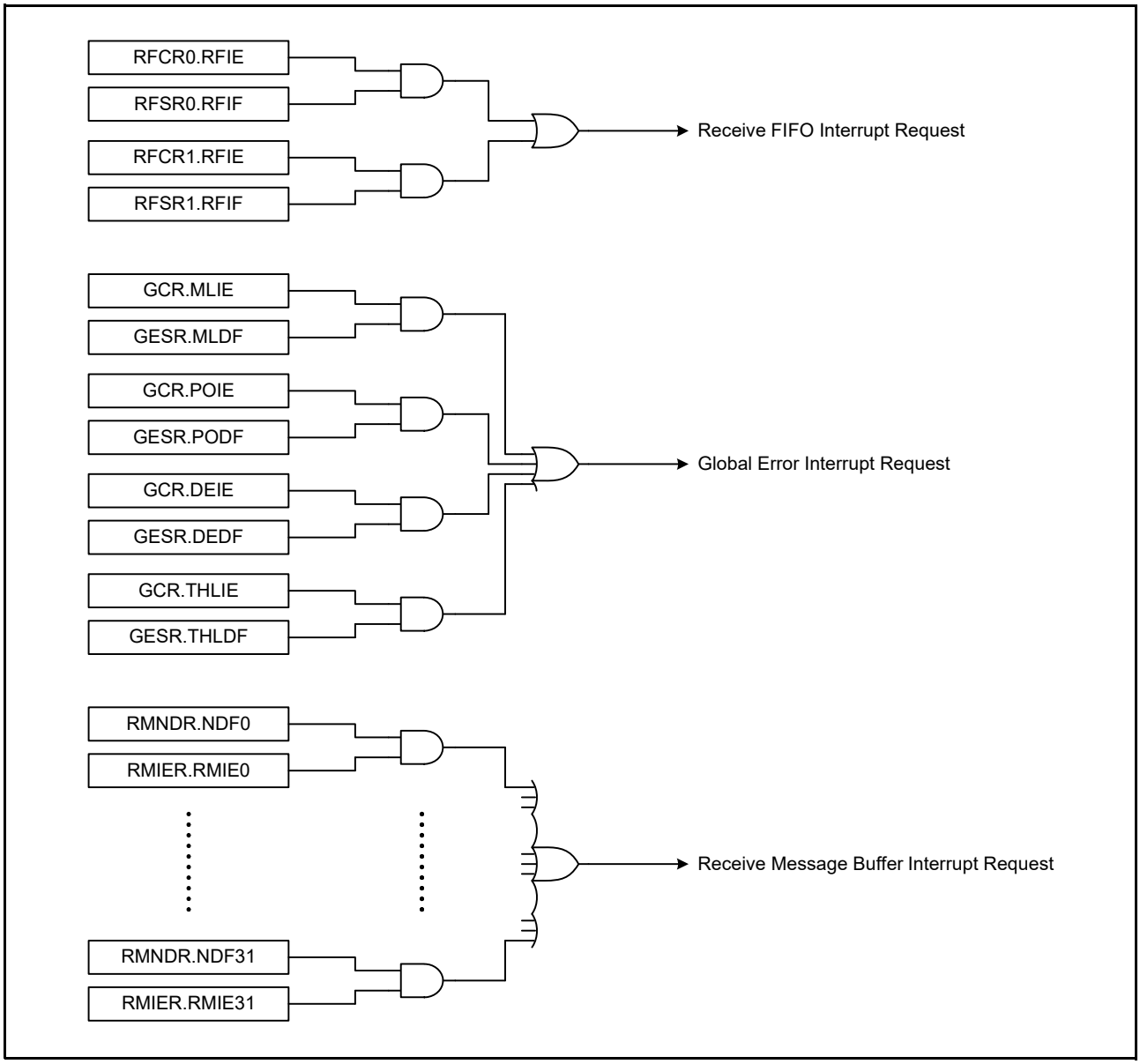


Figure 34.58 Global Interrupt Block Diagram

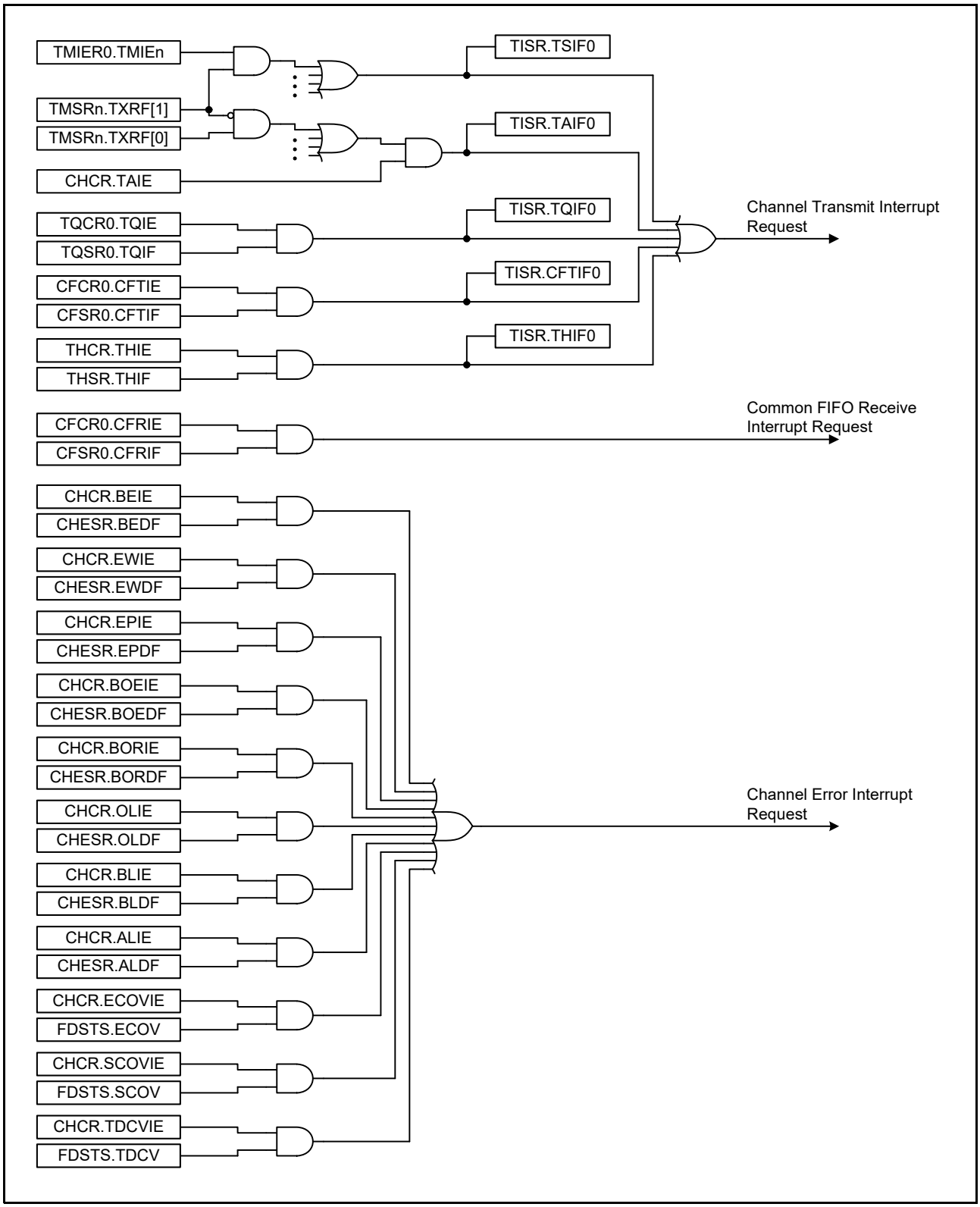


Figure 34.59 Channel Interrupt Block Diagram

34.10.2 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.

34.10.3 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 34.60 shows the DTC/DMA transfer configuration flow.

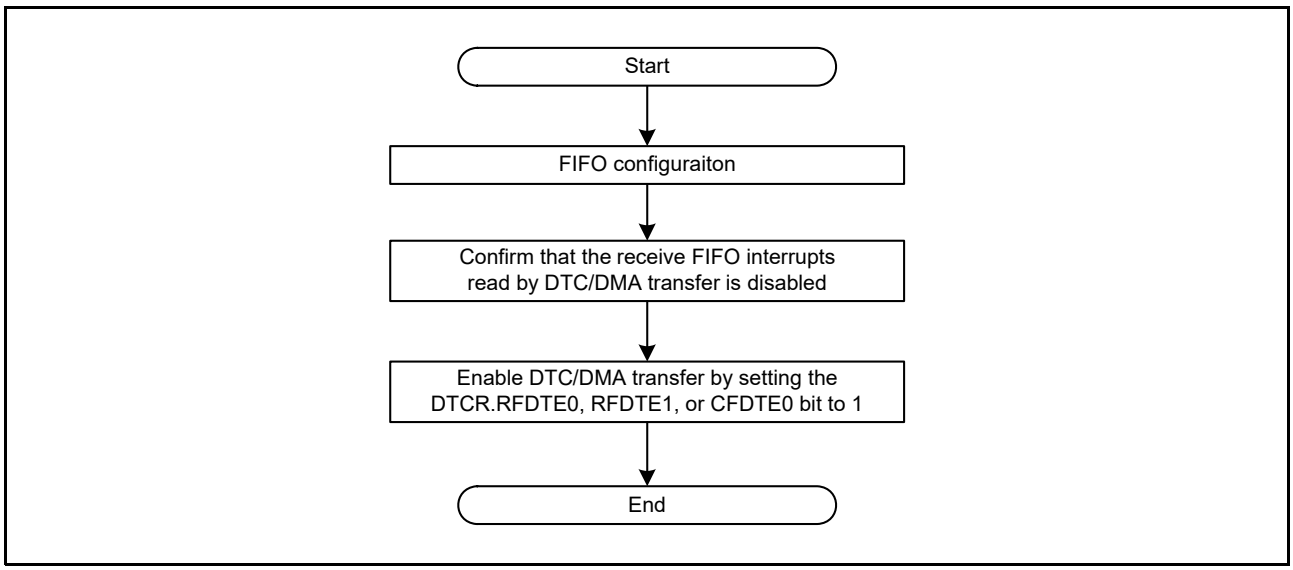


Figure 34.60 DTC/DMA Transfer Enable Flow

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 34.61 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.

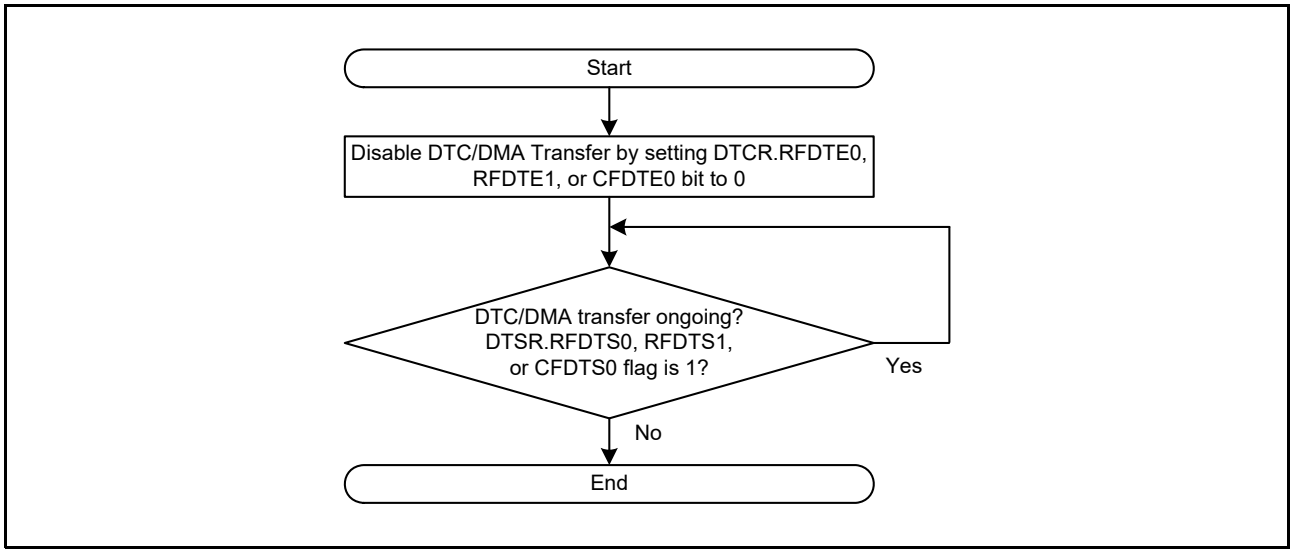


Figure 34.61 DTC/DMA Transfer Disable Flow

34.11 Usage Notes

34.11.1 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.

34.11.2 Notes on the DLL Clock Settings

Set the DLL clock to be used before releasing the module-stop state. For details on the DLL clock settings, refer to section 9, Clock Generation Circuit.
Release the module-stop state after the oscillation of the DLL clock has been stabilized.

34.11.3 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.

35. Serial Peripheral Interface (RSPIc)

In this section, “PCLK” is used to refer to PCLKB.

35.1 Overview

This MCU includes one channel of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex or simplex (transmit-only) synchronous serial communications with multiple processors and peripheral devices.
Table 35.1 lists the specifications of the RSPI, and Figure 35.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

Table 35.1 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) 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.
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

Table 35.1 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
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 Transmit 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.

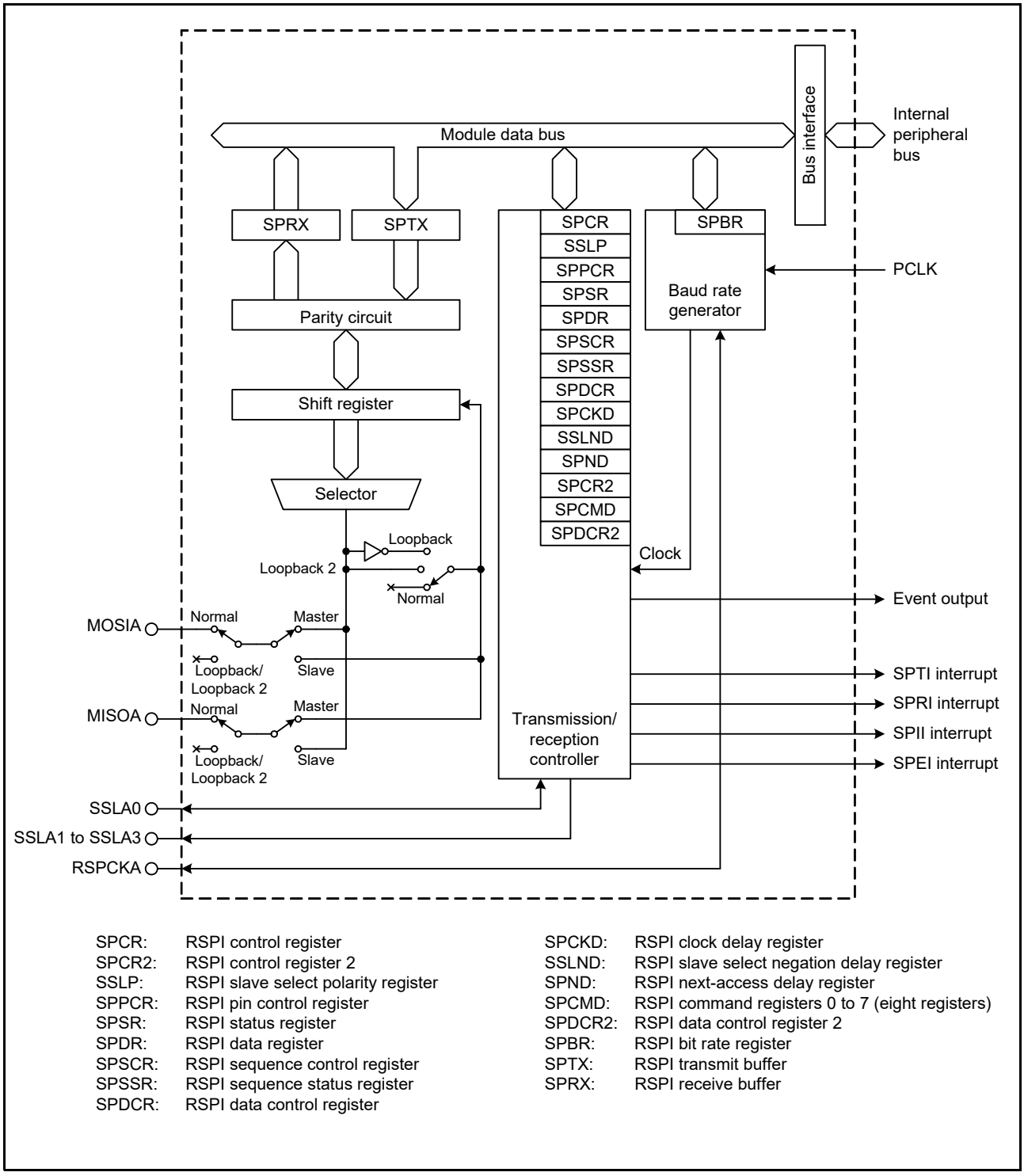


Figure 35.1 RSPI Block Diagram

Table 35.2 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 35.3.2, Controlling RSPI Pins for details.

Table 35.2 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

35.2 Register Descriptions

35.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h

	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 35.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 35.3.9, 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 35.3.2, 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 35.3.9, 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 35.3.9, 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 35.3.10, 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.

35.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h

	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.

35.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h

	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.

35.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SPRF	Receive Buffer Full Flag	0: Receive buffer has no valid data 1: Receive buffer has valid data	R*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 35.3.9.1, Overrun Error.

[Setting condition]

- When the next data reception ends while the SPCR.TXMD bit is 0 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.

[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 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.

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

[Clearing condition]

- When all of the received data are read from the SPDR register.

35.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 0008 8384h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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 0008 8384h

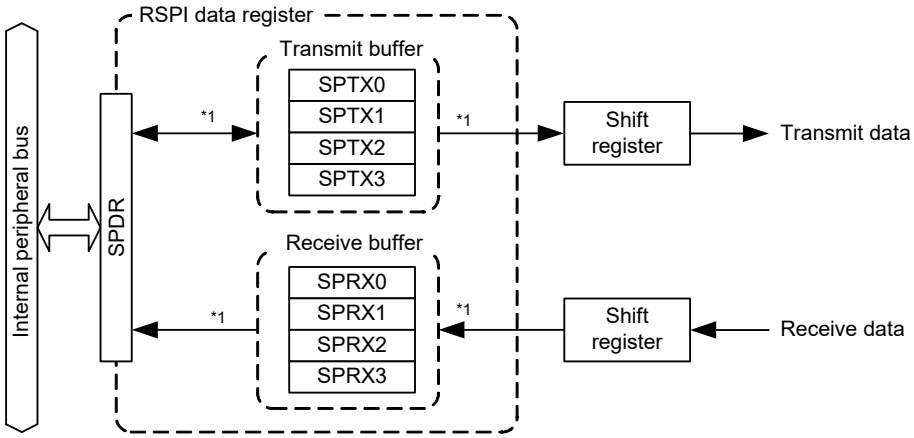
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): RSPI0.SPDR.HH 0008 8384h

	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 35.2 shows the Configuration of the SPDR Register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 35.2 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 35.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

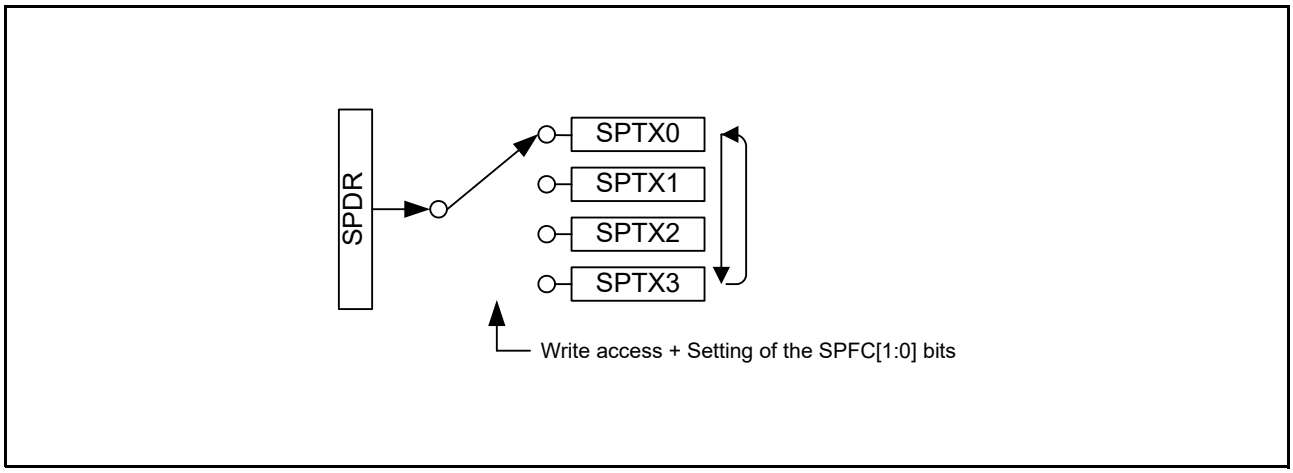


Figure 35.3 Configuration of the SPDR Register (Writing)

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 35.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

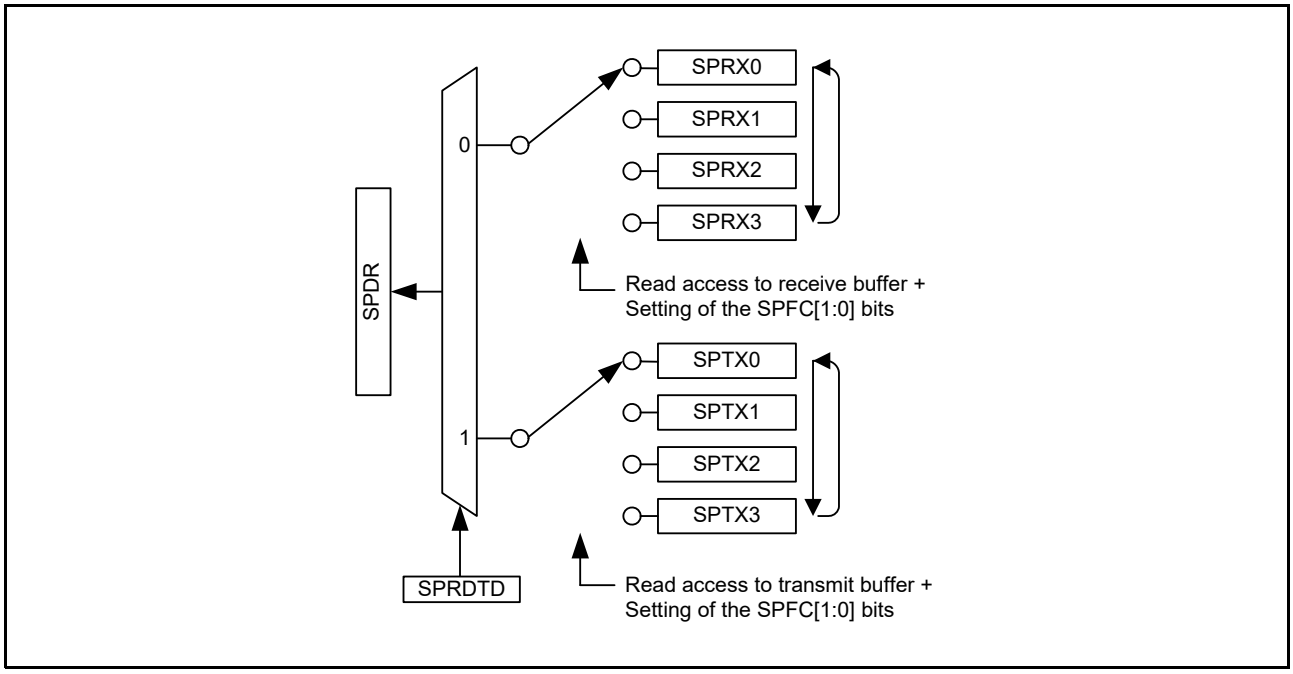


Figure 35.4 Configuration of the SPDR Register (Reading)

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).

35.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h

	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.

35.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h

	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 35.3.11.1, 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 35.3.9, Error Detection. For the RSPI's sequence control, refer to section 35.3.11.1, Master Mode Operation.

35.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah

	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 35.3 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 35.3 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
0	0	2	16.0 Mbps
1	0	4	8.00 Mbps
2	0	6	5.33 Mbps
3	0	8	4.00 Mbps
4	0	10	3.20 Mbps
5	0	12	2.67 Mbps
5	1	24	1.33 Mbps
5	2	48	667 kbps
5	3	96	333 kbps
255	3	4096	7.81 kbps

35.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh

	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 35.4 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 35.4 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 35.2.5, 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.

35.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch

	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.

35.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh

	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.

35.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh

	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.

35.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh

	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 35.3.9.1, Overrun Error.

35.2.14 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh

	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 35.2.8, 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 35.3.11.1, 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.

35.2.15 RSPI Data Control Register 2 (SPDCR2)

Address(es): RSPI0.SPDCR2 0008 83A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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
b7 to b1	—	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 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 35.3.4.3 Byte Swap Transmission and 35.3.4.4 Byte Swap Reception.

35.3 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.

35.3.1 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 35.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 35.5 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				
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	Not supported	Not supported	Supported	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.

35.3.2 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 35.6.

Table 35.6 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

35.3.3 RSPI System Configuration Examples

35.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 35.5 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.

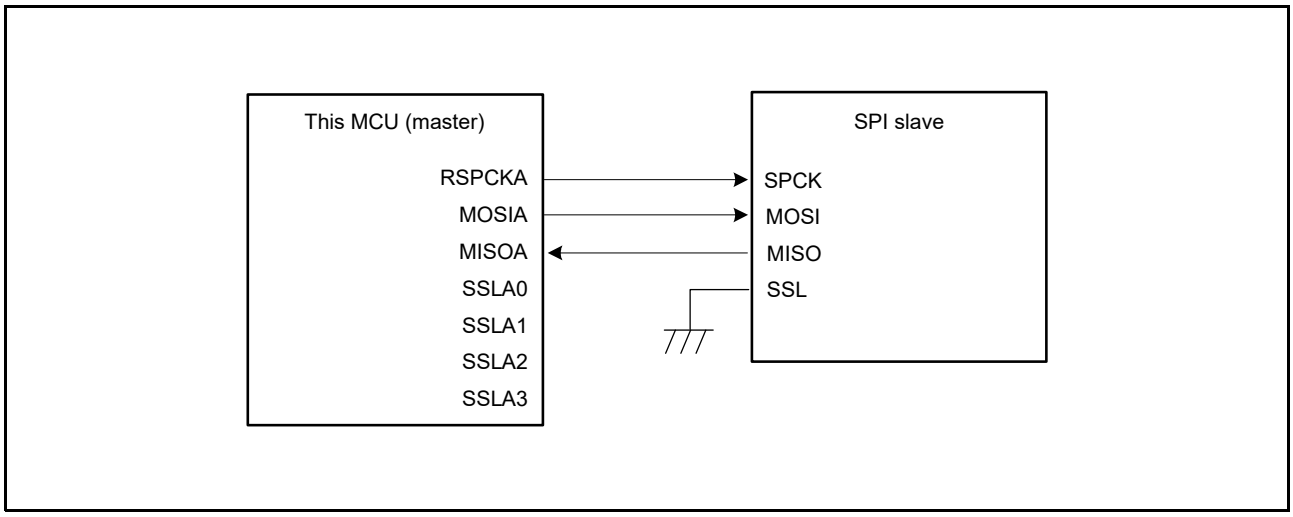


Figure 35.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

35.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 35.6 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 35.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

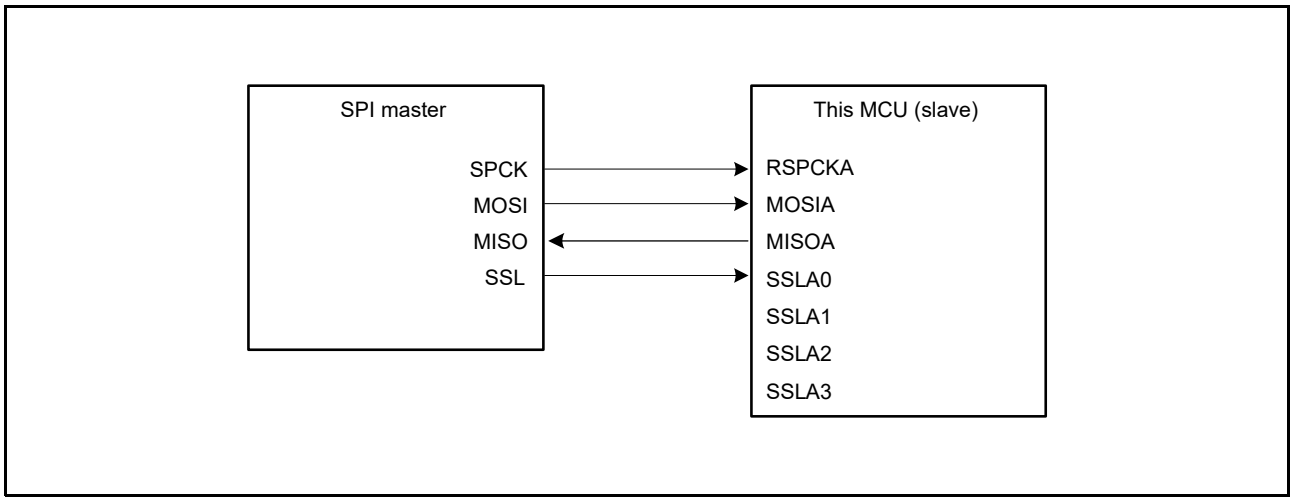


Figure 35.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

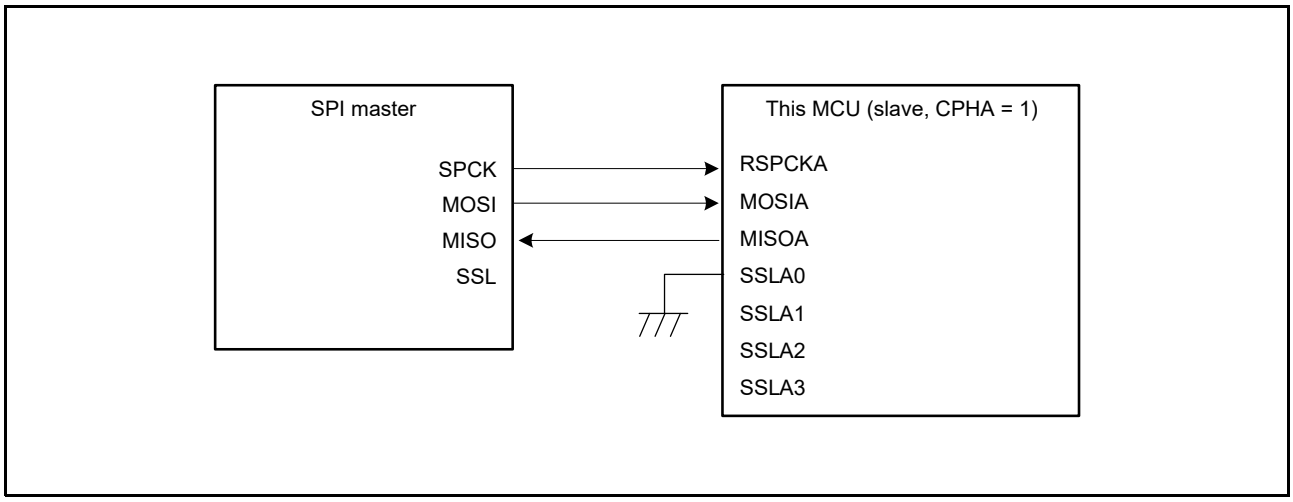


Figure 35.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

35.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 35.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 35.8, 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.

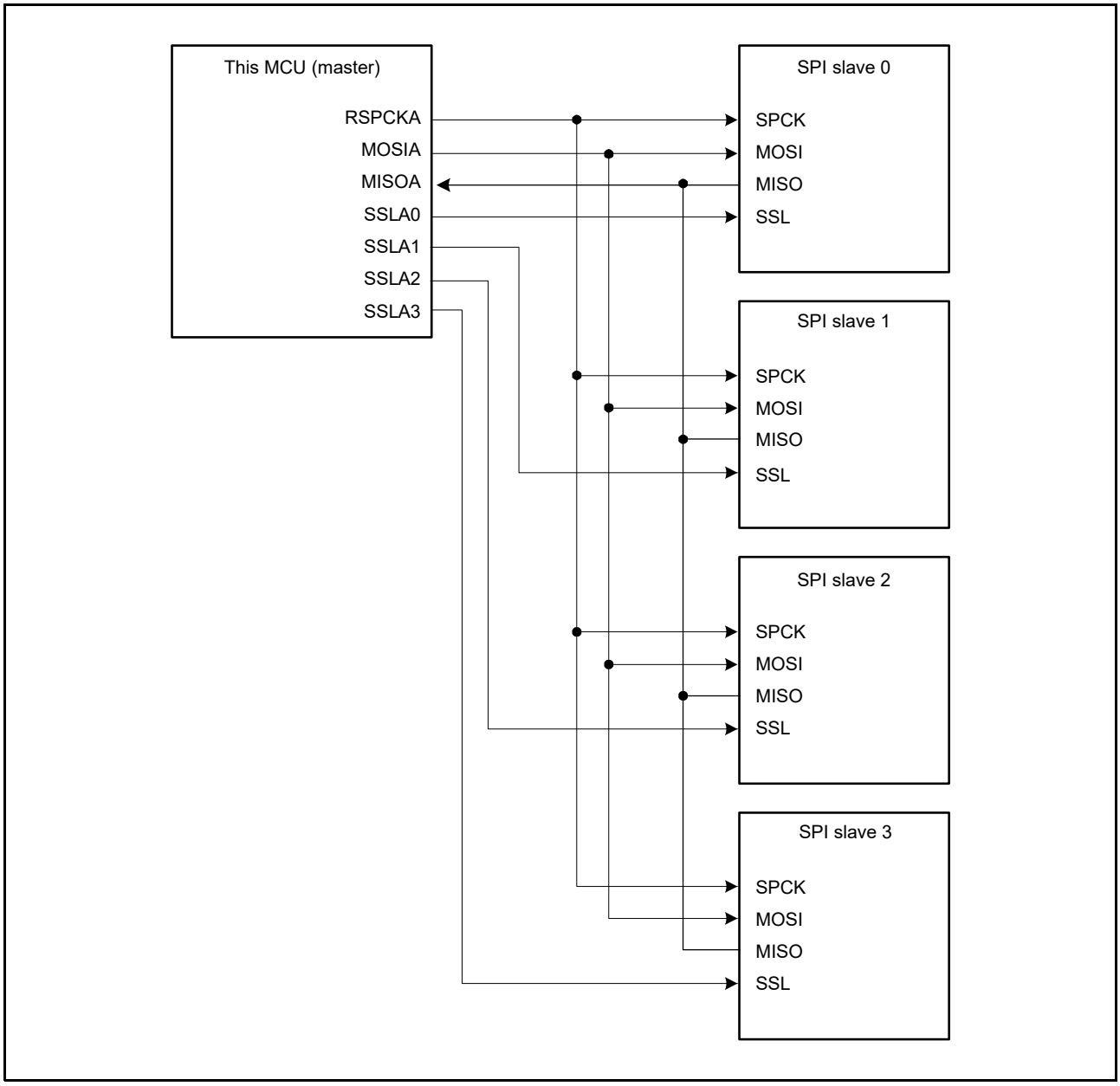


Figure 35.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

35.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 35.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 35.9, 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.

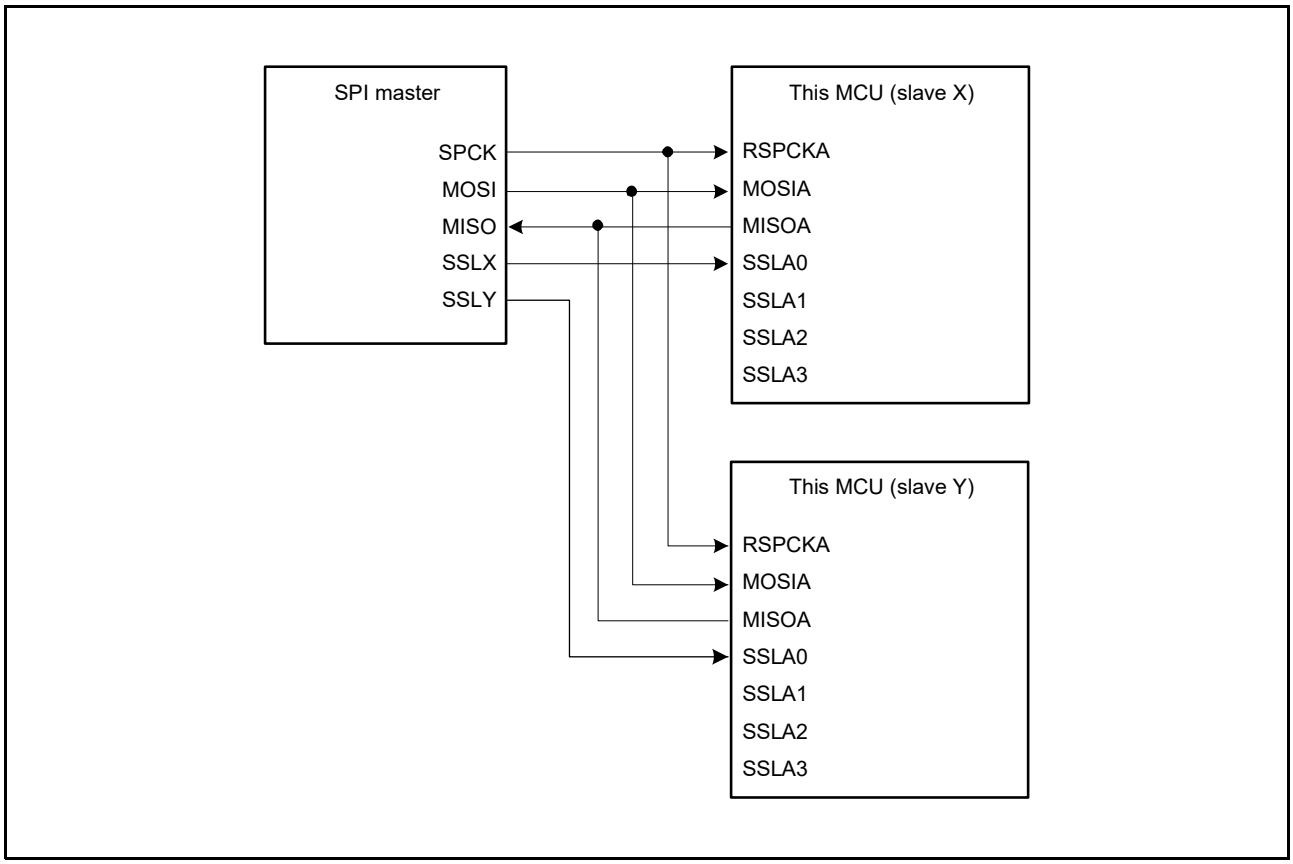


Figure 35.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

35.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 35.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 35.10, 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.

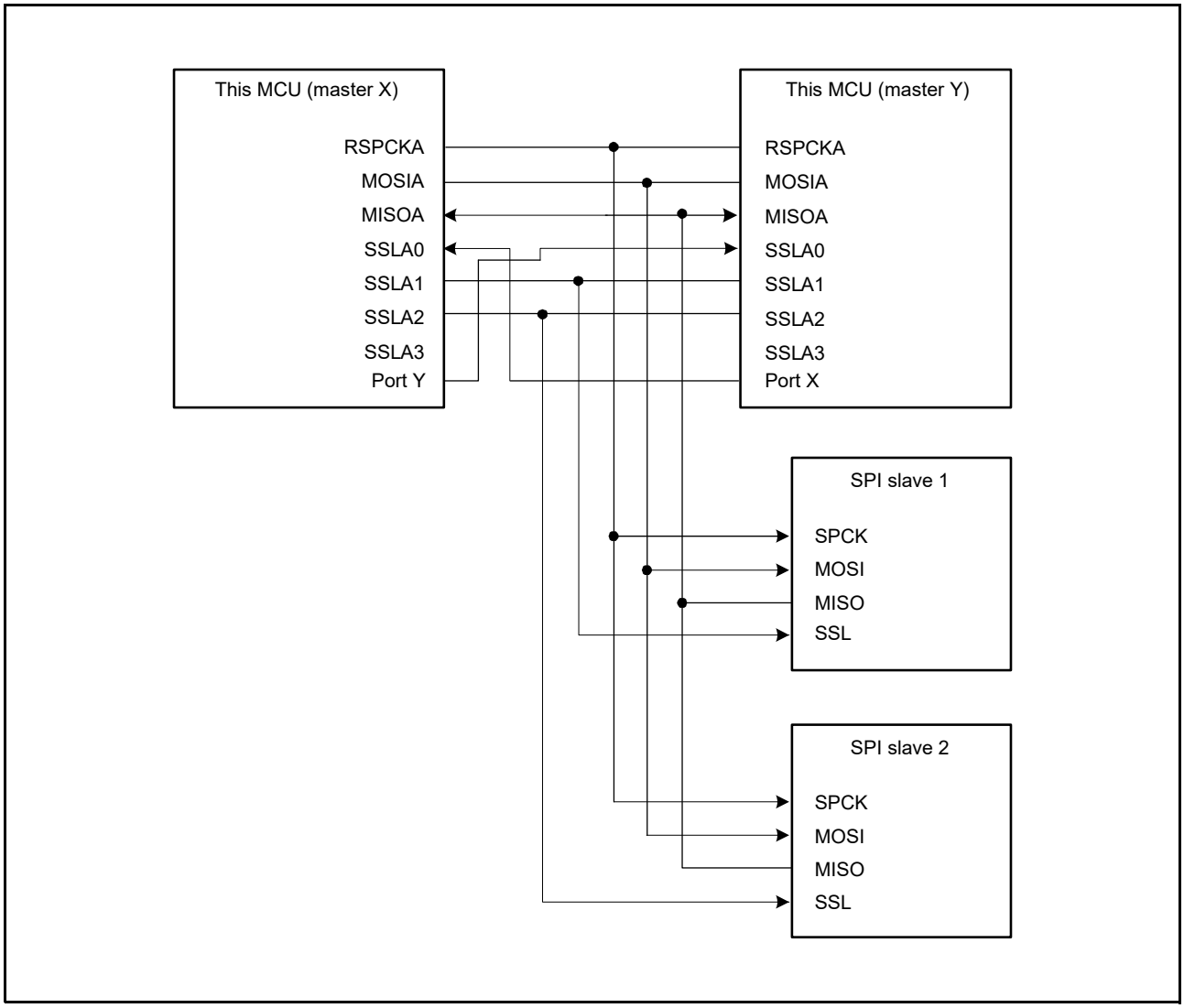


Figure 35.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

35.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 35.11 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.

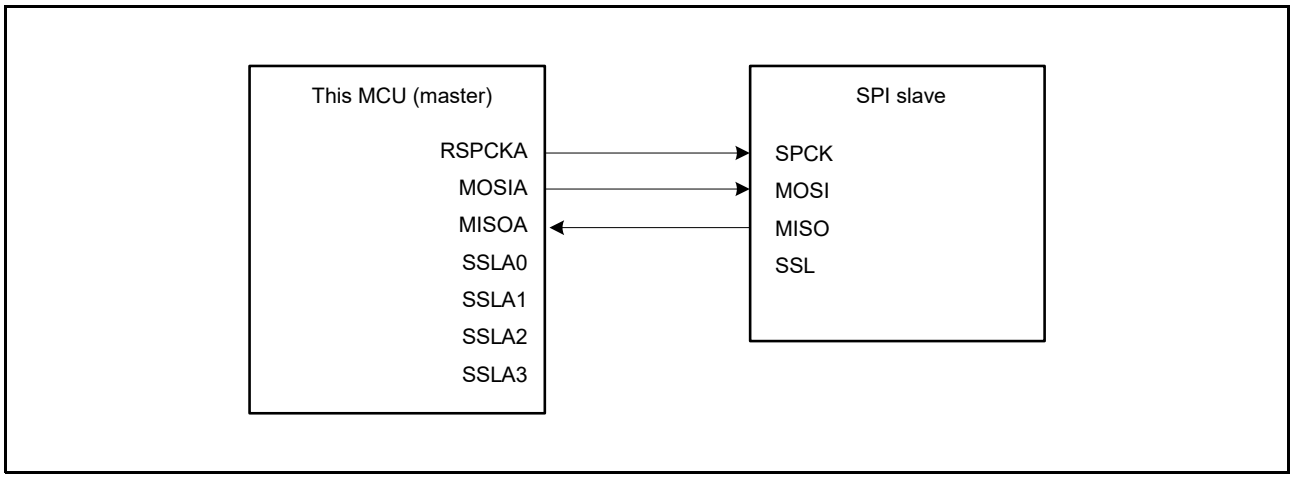


Figure 35.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

35.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 35.12 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.

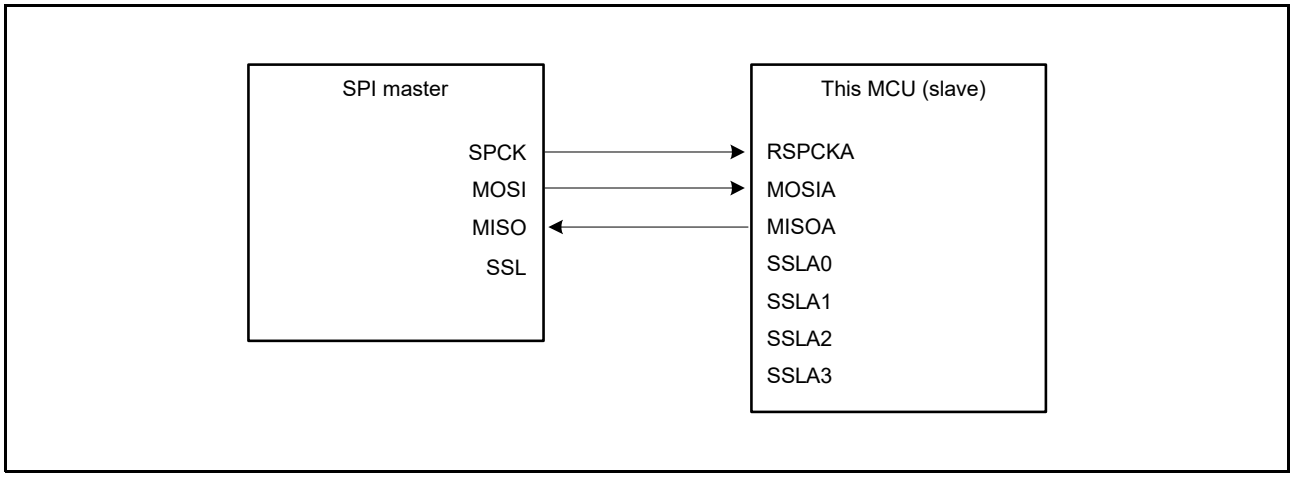


Figure 35.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

35.3.4 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.

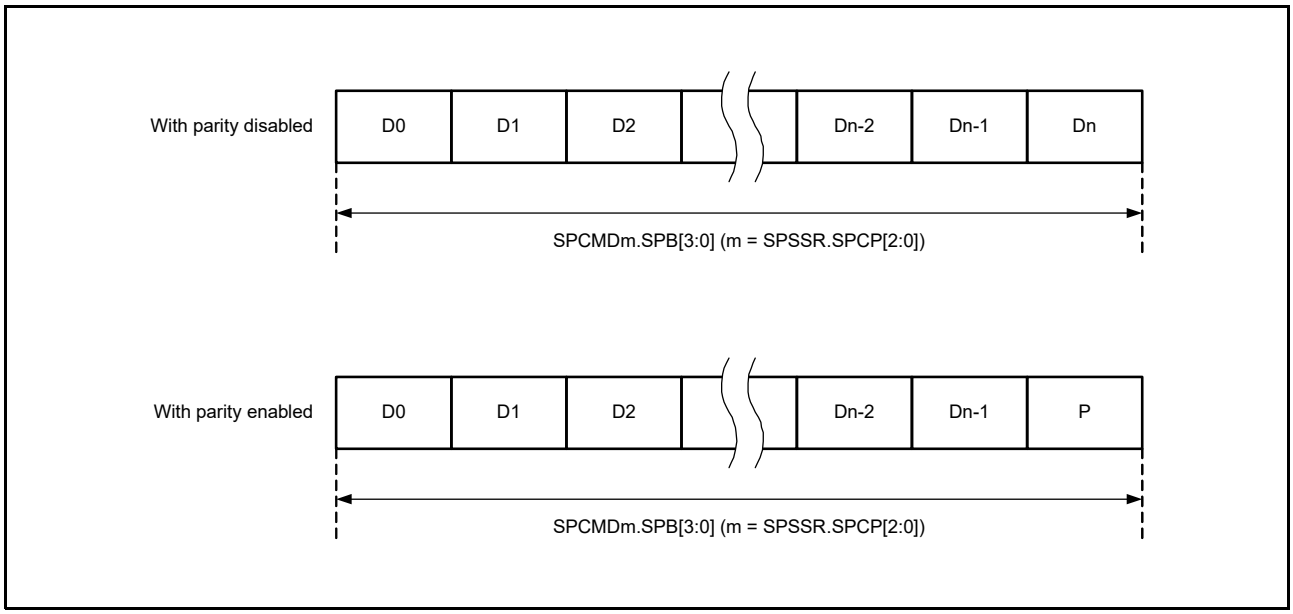


Figure 35.13 Outline of the Data Format (with Parity Disabled/Enabled)

35.3.4.1 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 35.14 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.

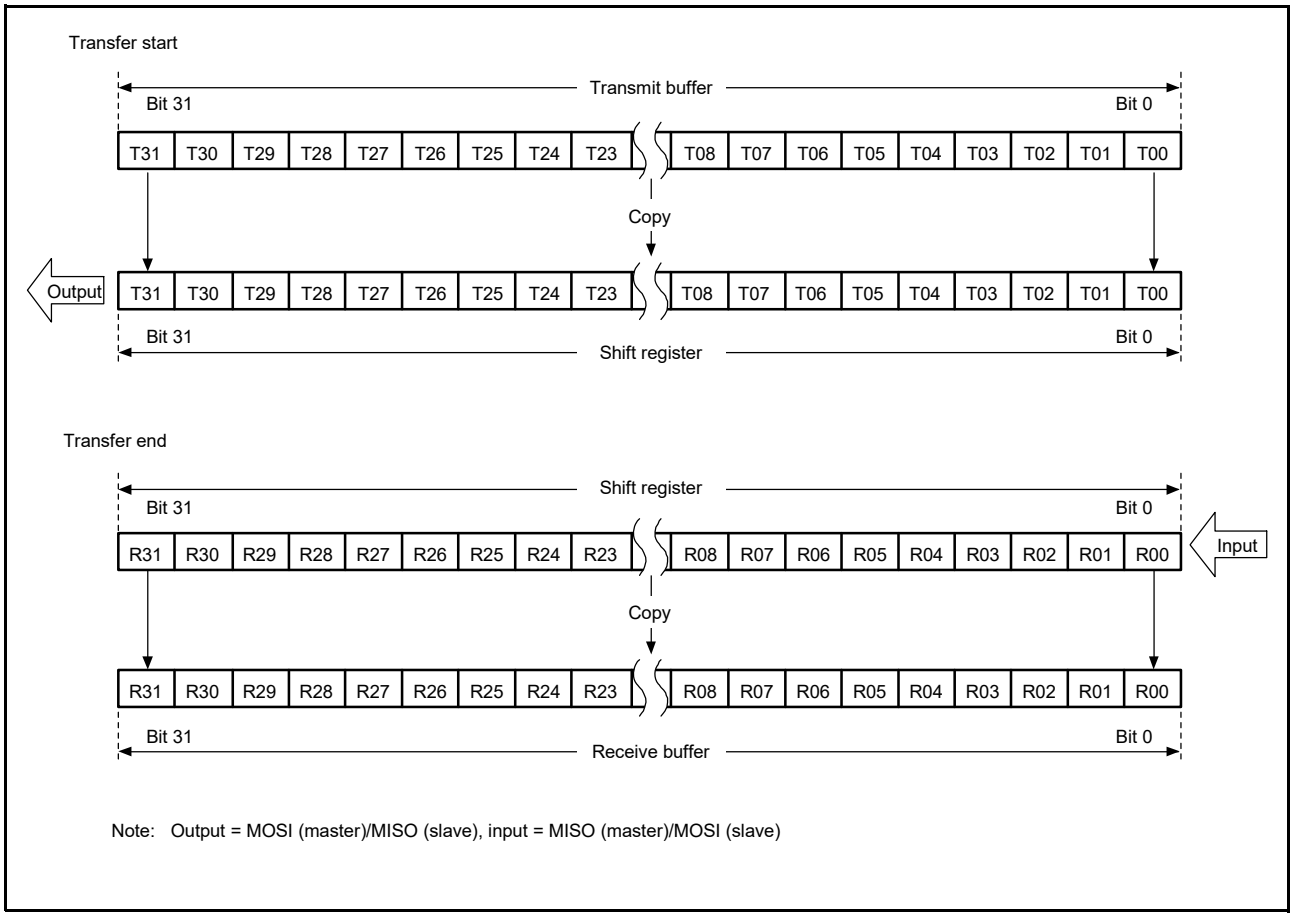


Figure 35.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 35.15 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, 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.

Figure 35.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 35.16 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.

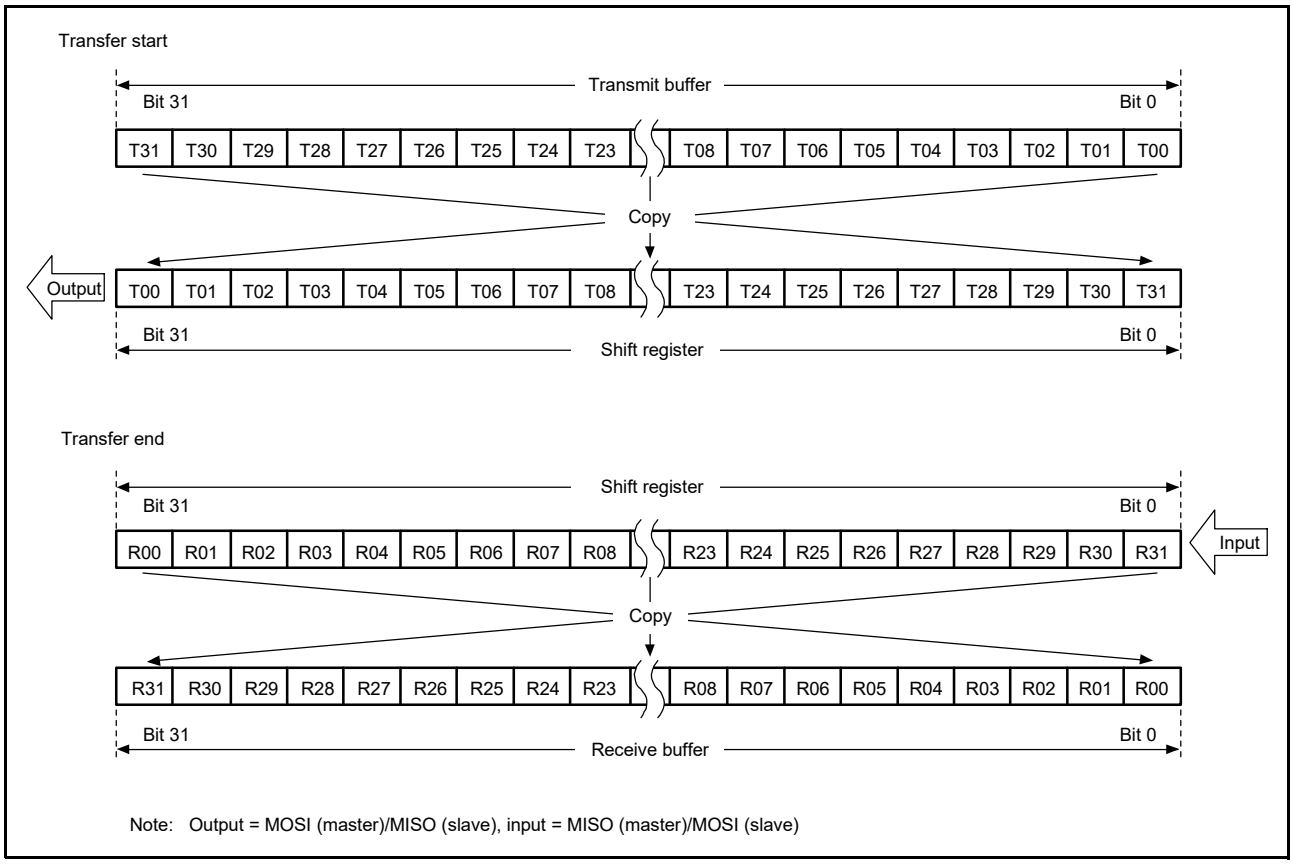


Figure 35.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 35.17 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, 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.

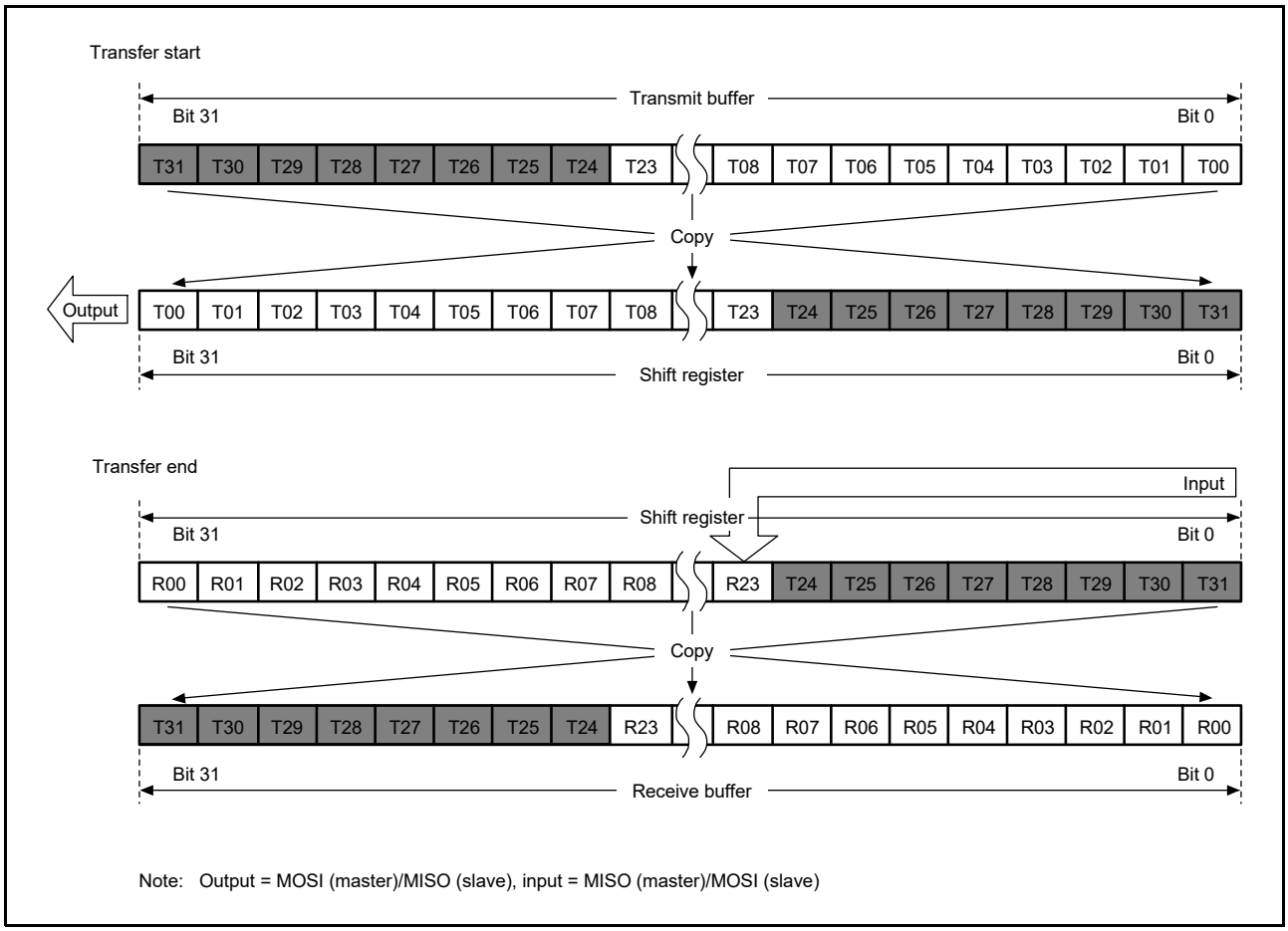


Figure 35.17 LSB First Transfer (24-Bit Data, Parity Disabled)

35.3.4.2 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 35.18 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.

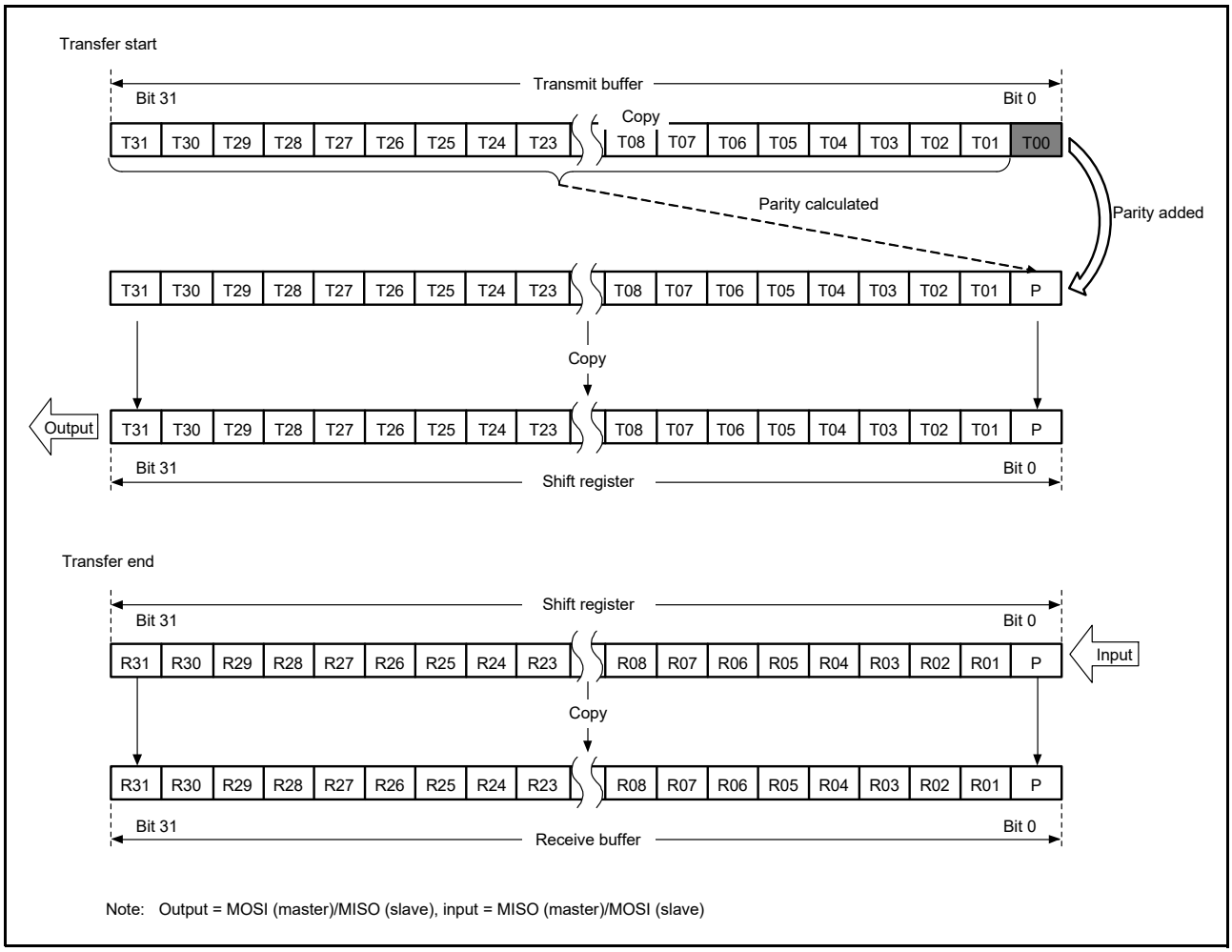


Figure 35.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 35.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the 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.

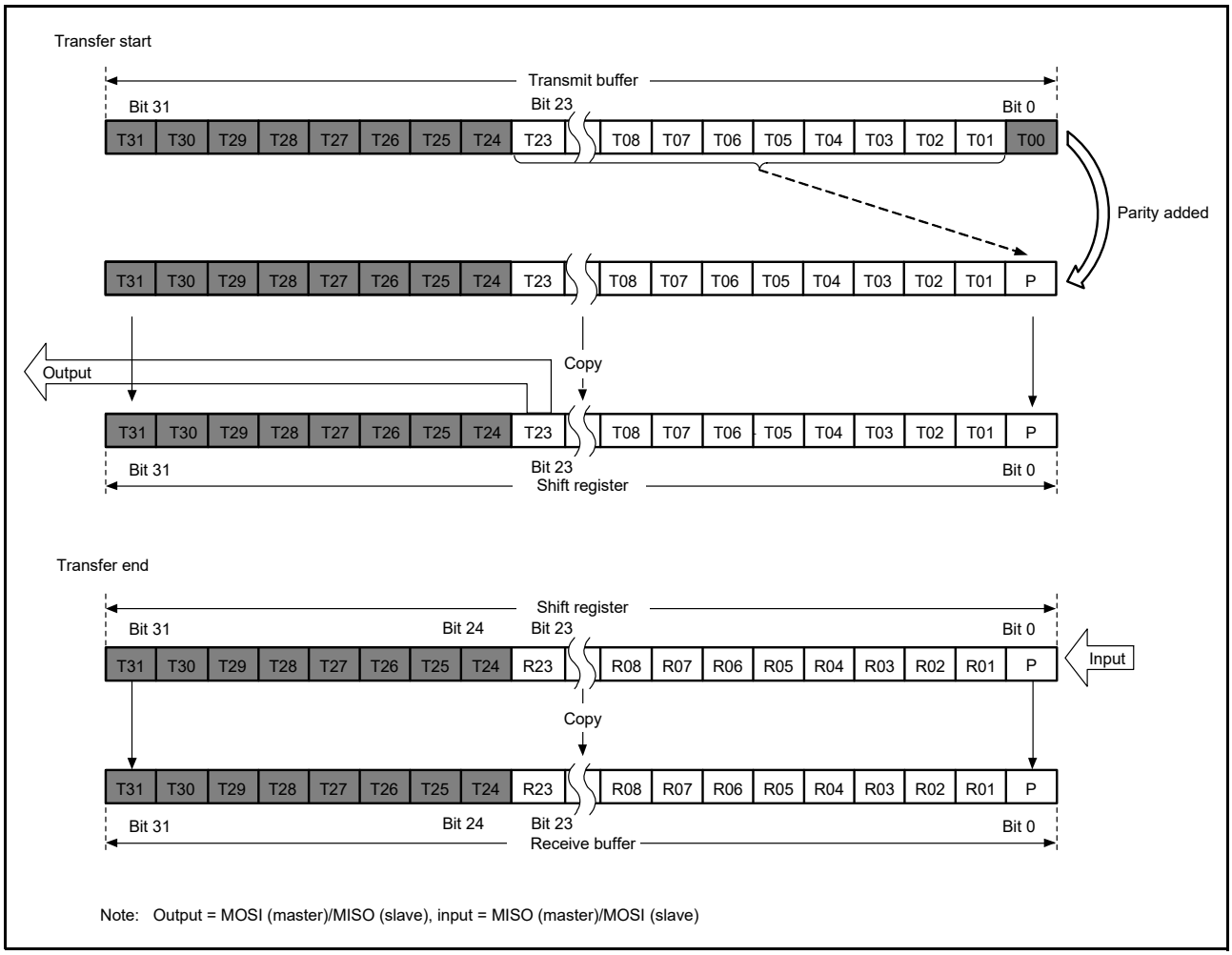


Figure 35.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 35.20 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.

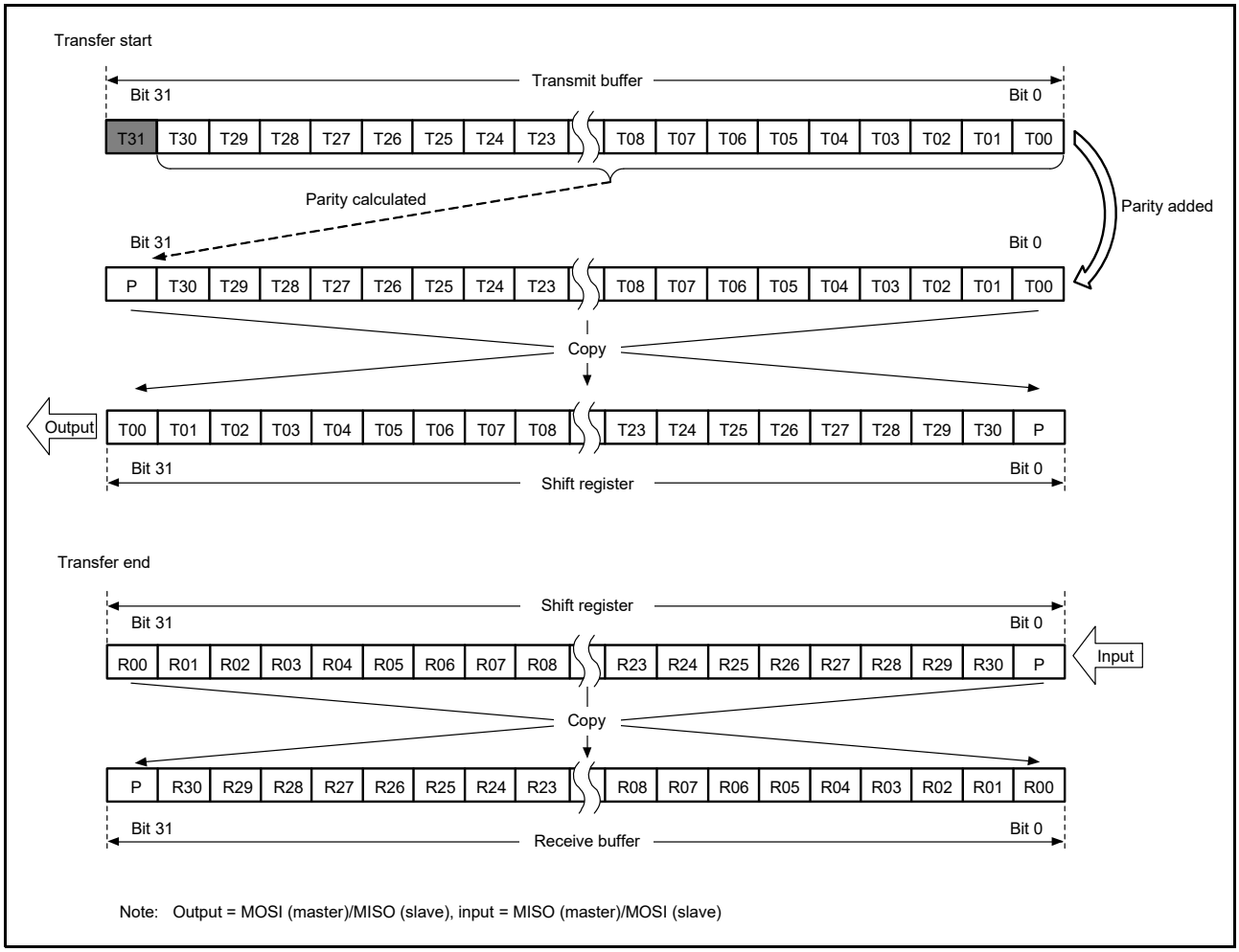


Figure 35.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 35.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the 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.

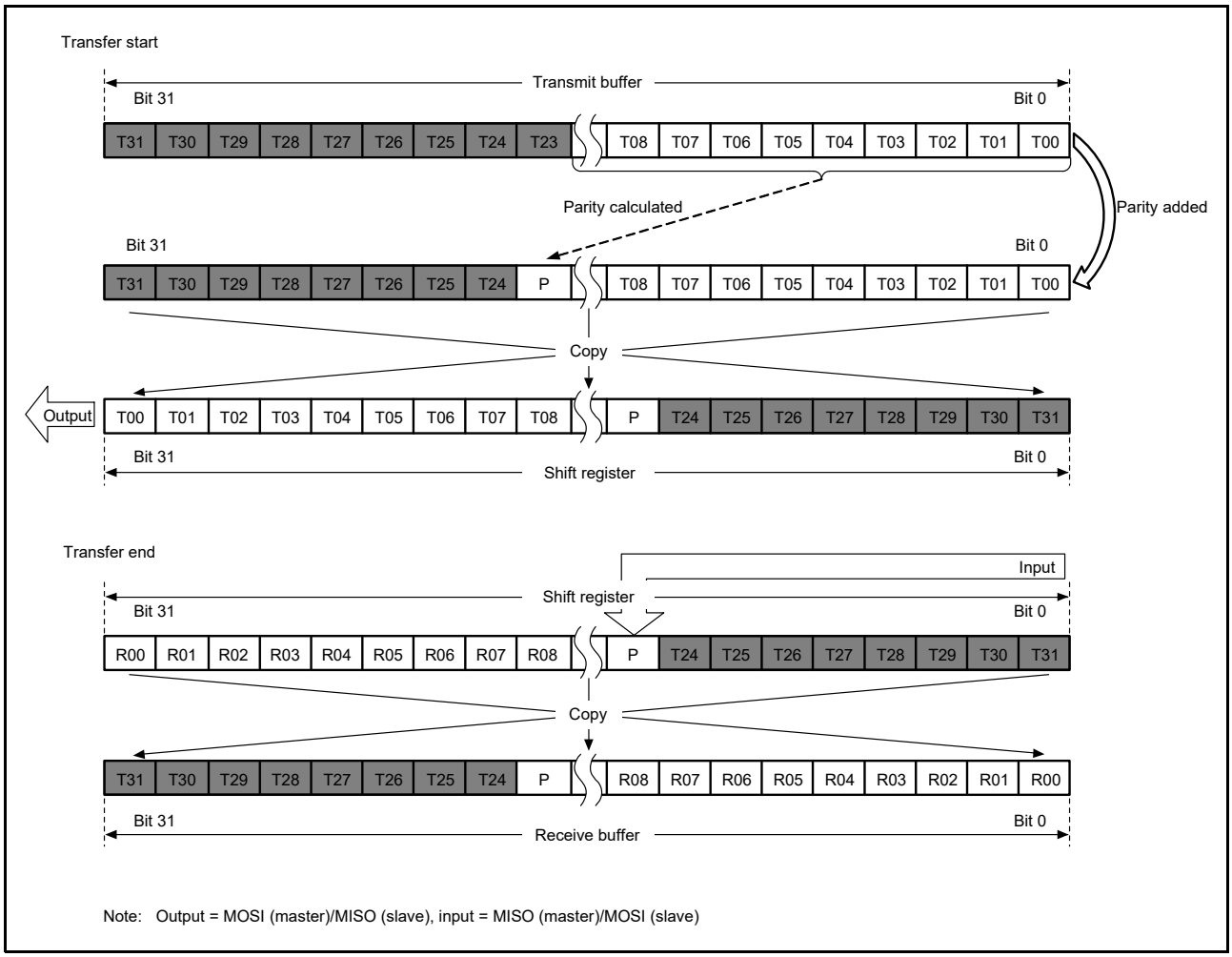


Figure 35.21 LSB First Transfer (24-Bit Data, Parity Enabled)

35.3.4.3 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 35.22 shows data transfer between the SPDR register and the shift register when data length is 32 bits.

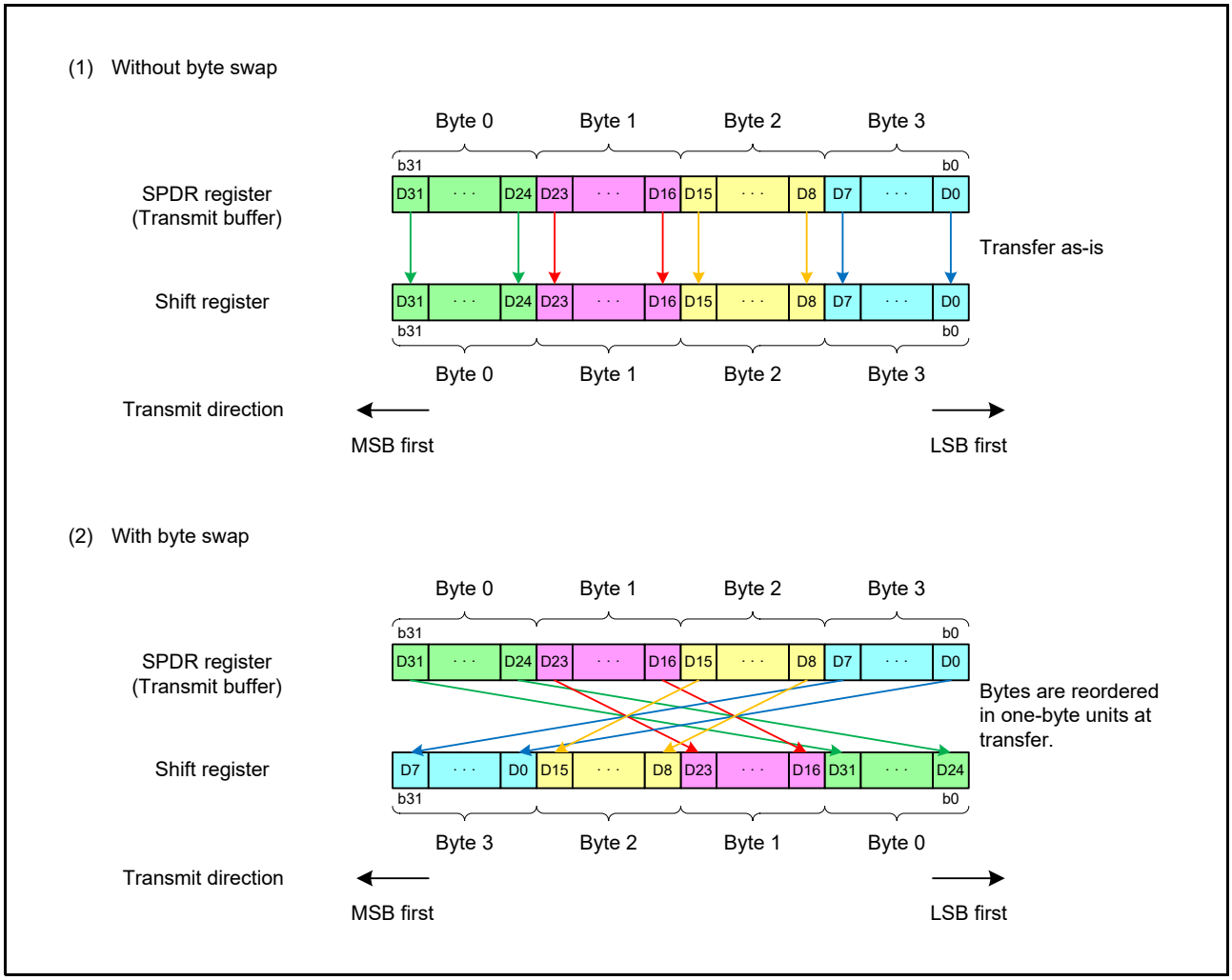


Figure 35.22 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

35.3.4.4 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 35.23 shows data transfer between the shift register and the SPDR register when data length is 32 bits.

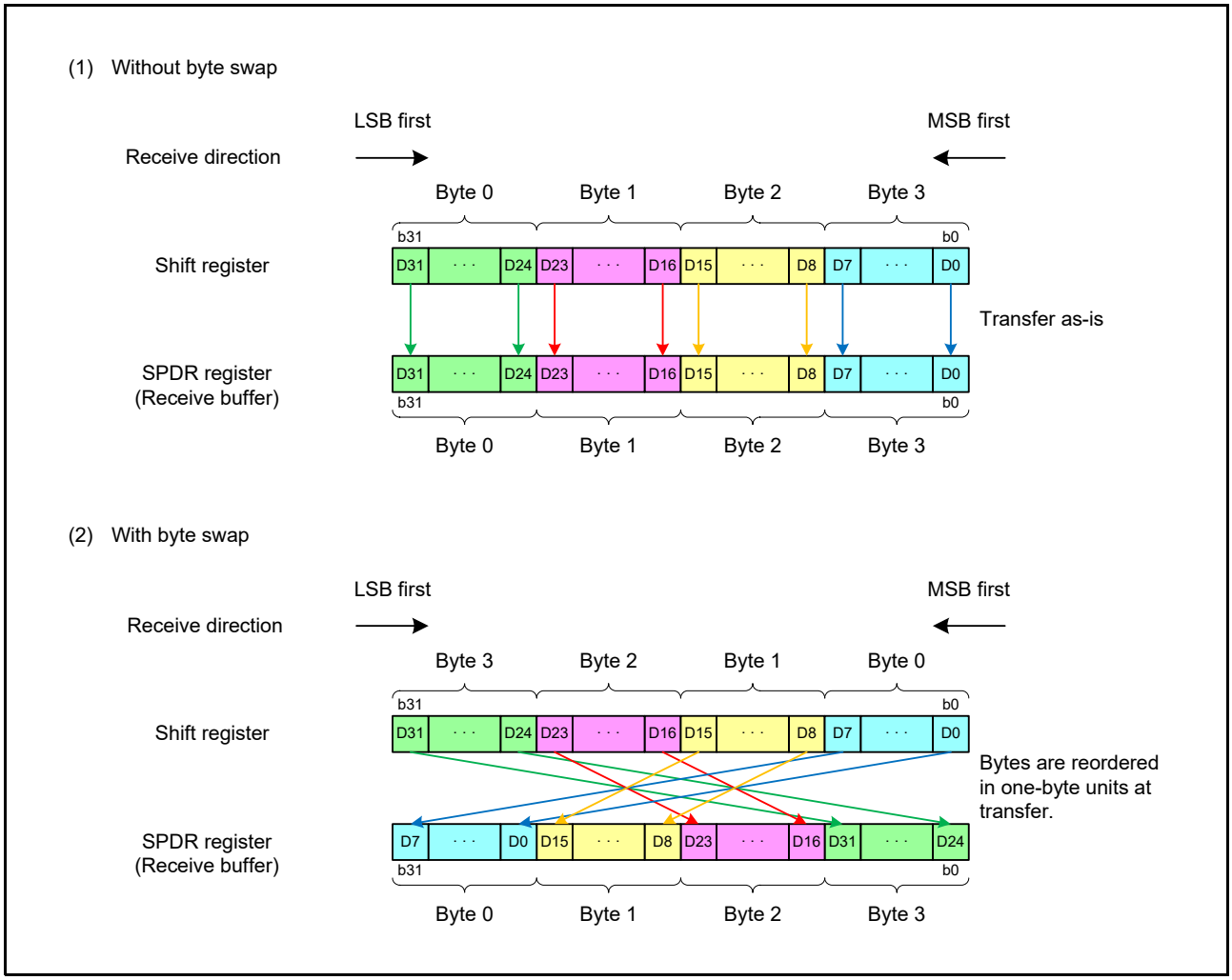


Figure 35.23 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

35.3.5 Transfer Format

35.3.5.1 CPHA = 0

Figure 35.24 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 35.24, 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 35.3.2, 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 35.3.11.1, Master Mode Operation.

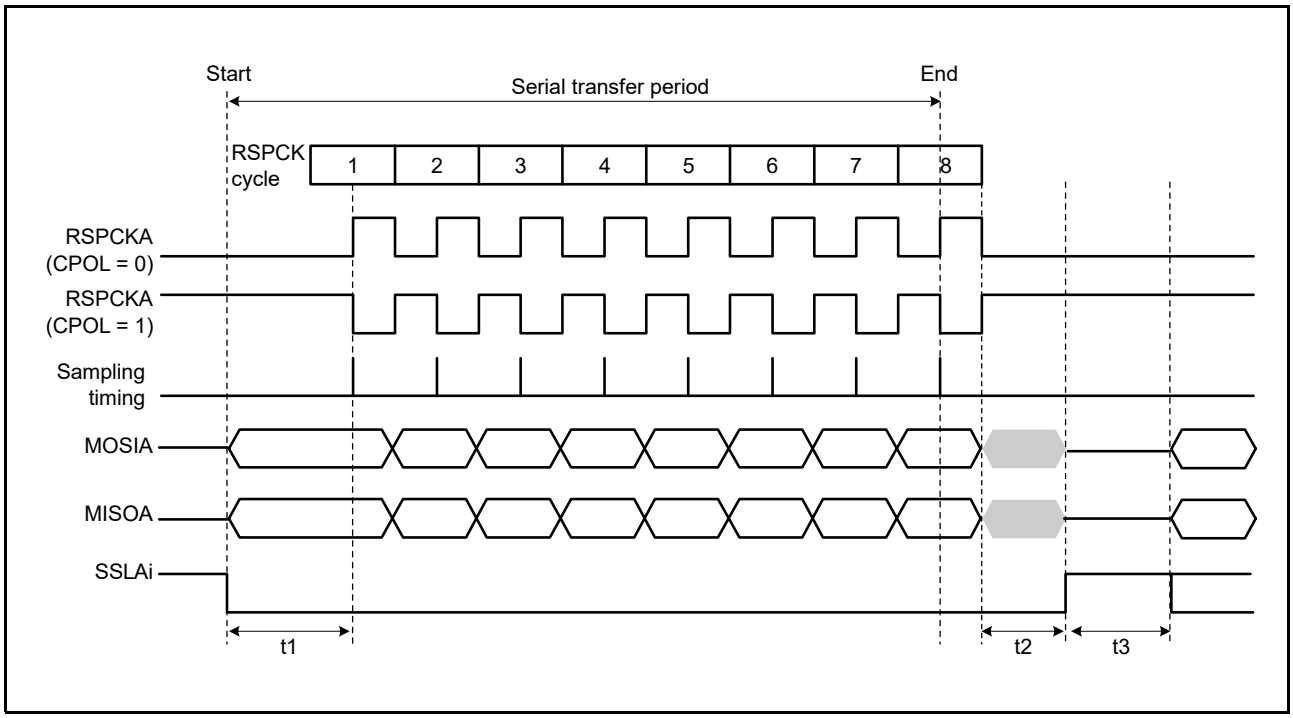


Figure 35.24 RSPI Transfer Format (CPHA = 0)

35.3.5.2 CPHA = 1

Figure 35.25 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 35.25, 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 35.3.2, 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 35.3.11.1, Master Mode Operation.

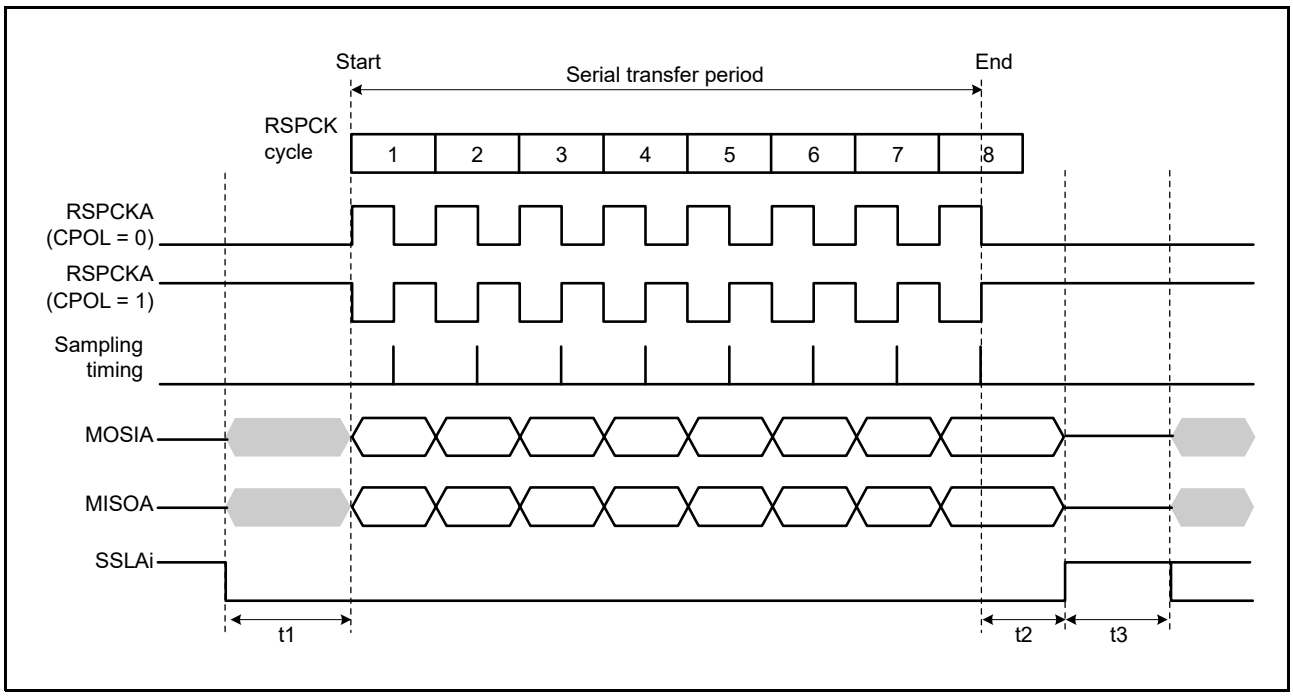


Figure 35.25 RSPI Transfer Format (CPHA = 1)

35.3.6 Communications Operating Mode

Full-duplex communications or transmit-only simplex communications can be selected by the SPCR.TXMD bit. 'SPDR access' shown in Figure 35.26 and Figure 35.27 indicate the condition of access to the SPDR register, where 'W' denotes a write cycle.

35.3.6.1 Full-Duplex Communications (SPCR.TXMD = 0)

Figure 35.26 shows an example of operation when the SPCR.TXMD bit is set to 0. In the example in Figure 35.26, 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).

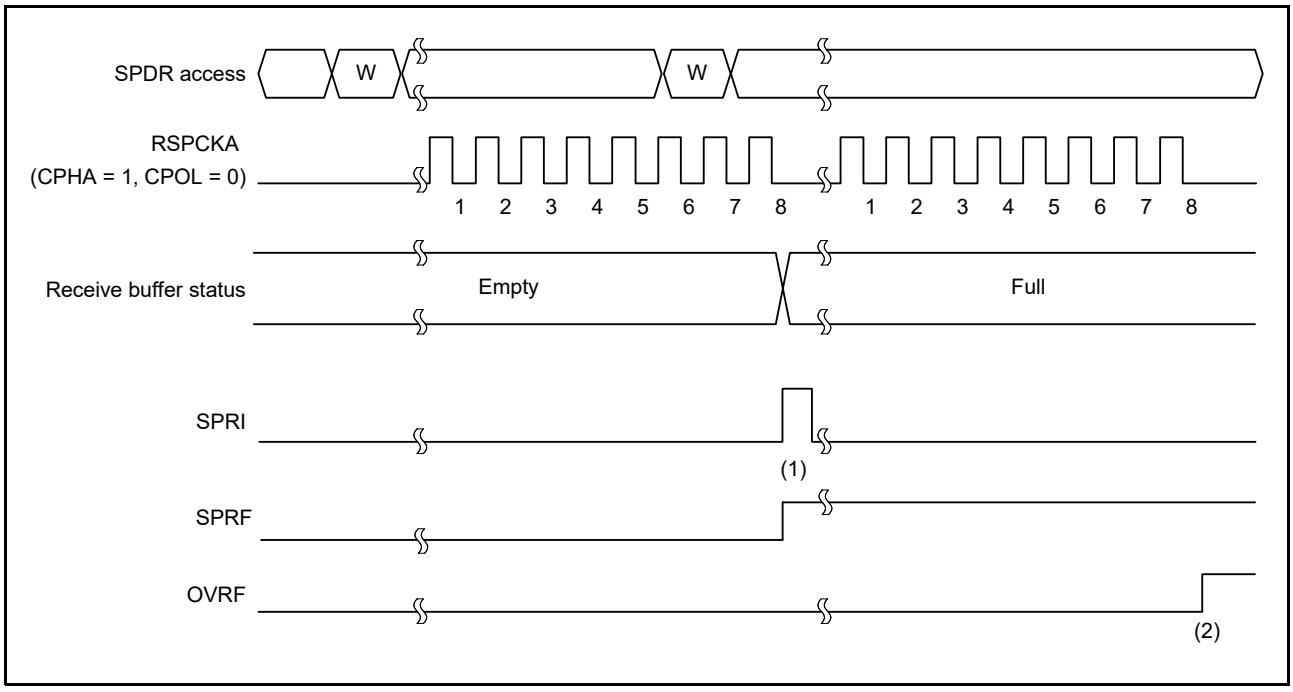


Figure 35.26 Operation Example of SPCR.TXMD = 0

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of 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) 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.

35.3.6.2 Transmit-only Simplex Communications (SPCR.TXMD = 1)

Figure 35.27 shows an example of operation when the SPCR.TXMD bit is set to 1. In the example in Figure 35.27, 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).

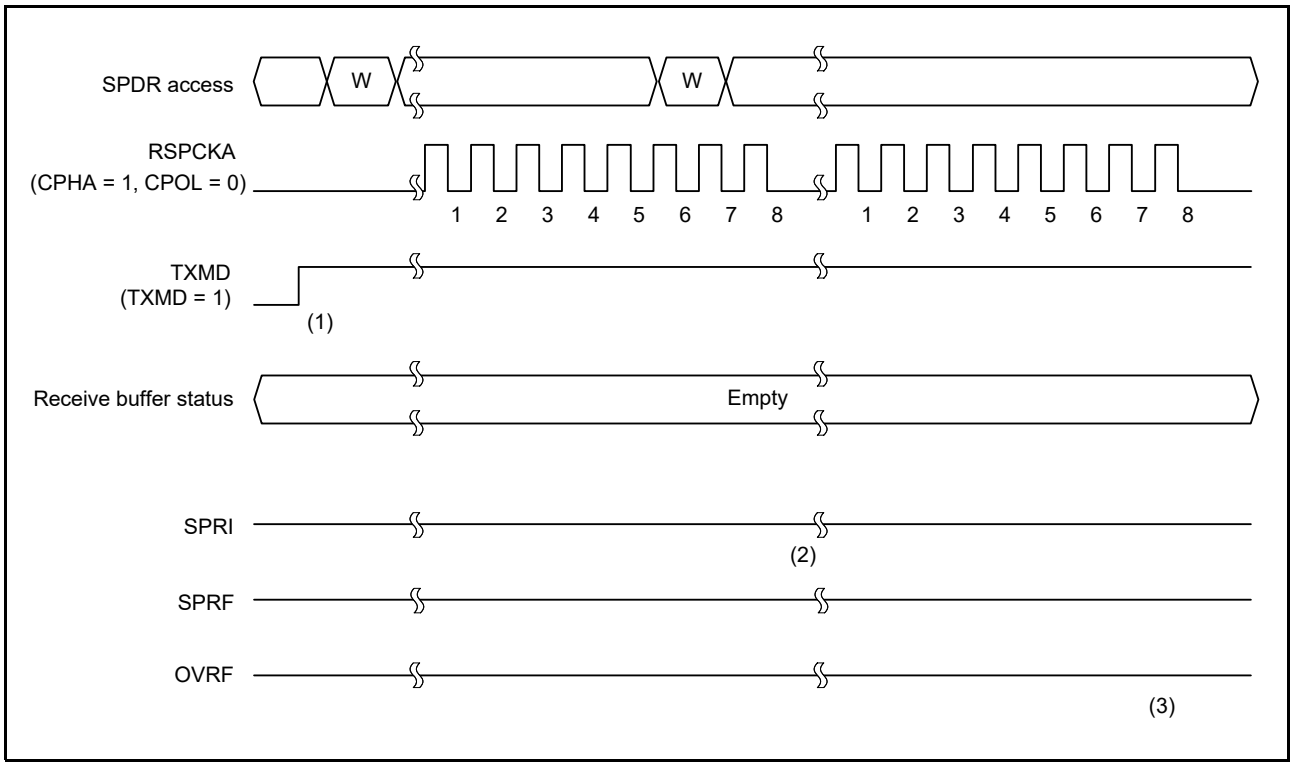


Figure 35.27 Operation Example of SPCR.TXMD = 1

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit-only simplex communications (SPCR.TXMD = 1).

(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), 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), the RSPI transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

35.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 35.28 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). 'SPDR access' shown in Figure 35.28 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 35.28, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

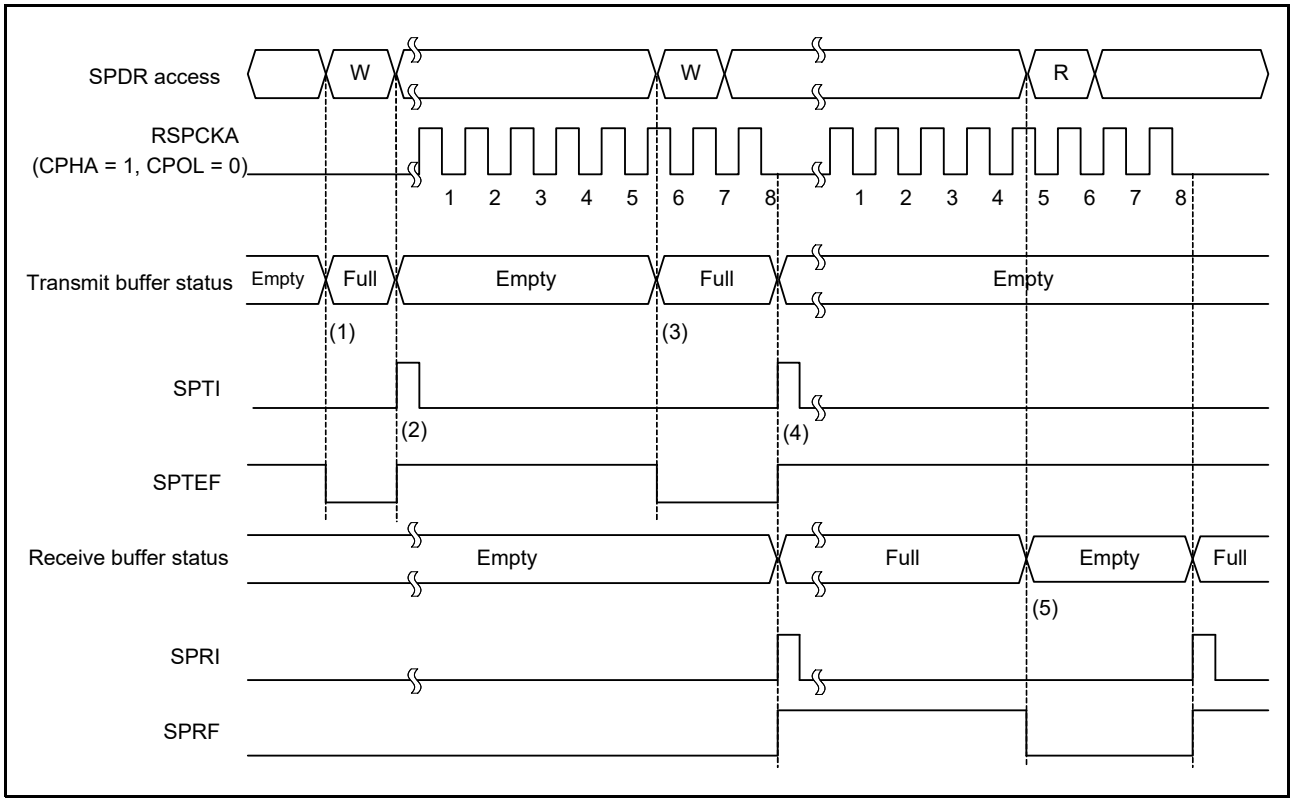


Figure 35.28 Operation Example of SPTI and SPRI Interrupts

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 35.3.11, SPI Operation, and section 35.3.12, 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 35.3.9, 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 (ICUb), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

35.3.8 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.

35.3.9 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 35.7 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 35.7 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 used 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 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 35.7, 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 35.3.9.4, Underrun Error. An overrun error shown in 4 is described in section 35.3.9.1, Overrun Error. A parity error shown in 5 is described in section 35.3.9.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 35.3.9.3, Mode Fault Error.

For the transmit and receive interrupts, refer to section 35.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

35.3.9.1 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 35.29 shows an example of operations of the SPRF and OVRF flags. 'SPSR access' and 'SPDR access' shown in Figure 35.29 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 35.29, 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).

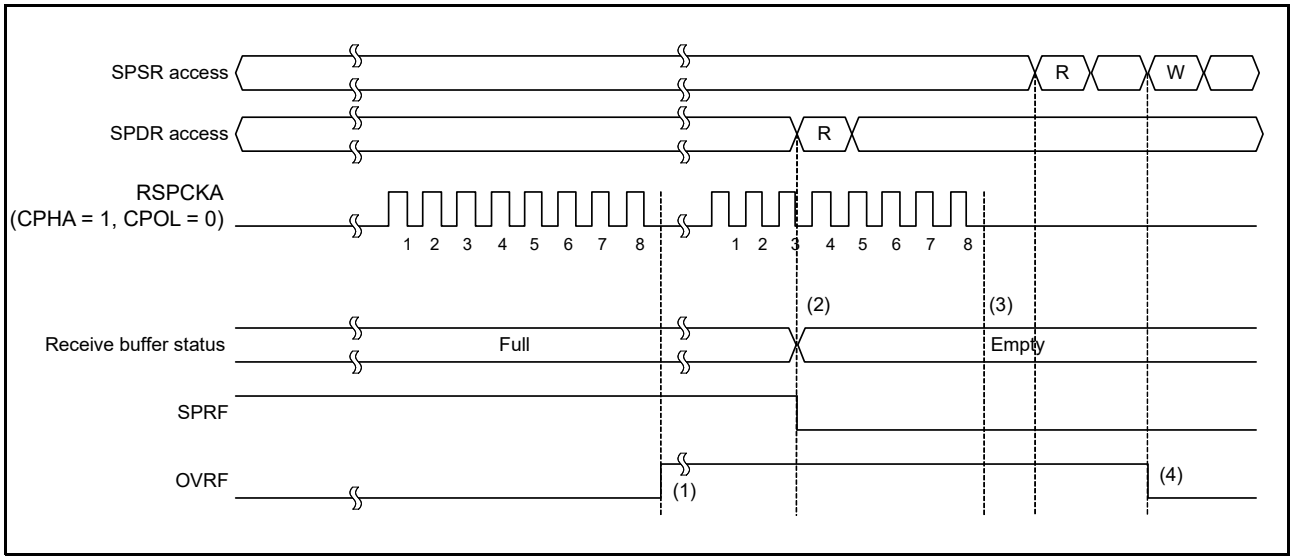


Figure 35.29 Operation Example of the SPRF and OVRF Flags

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 35.30 and Figure 35.31 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

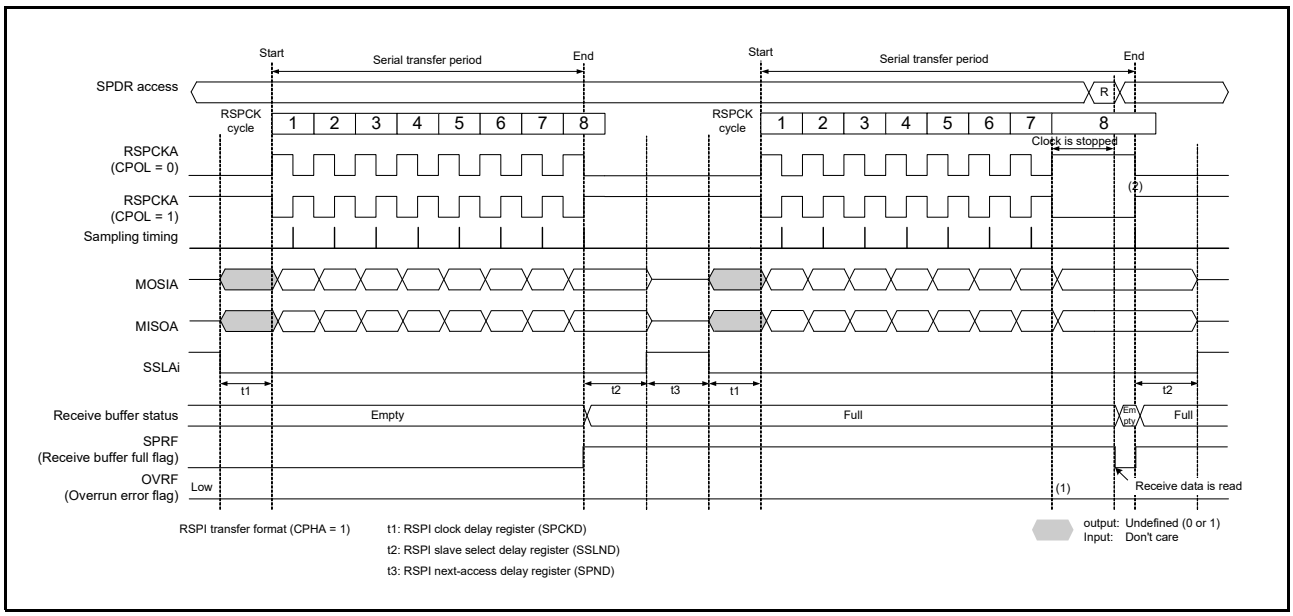


Figure 35.30 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

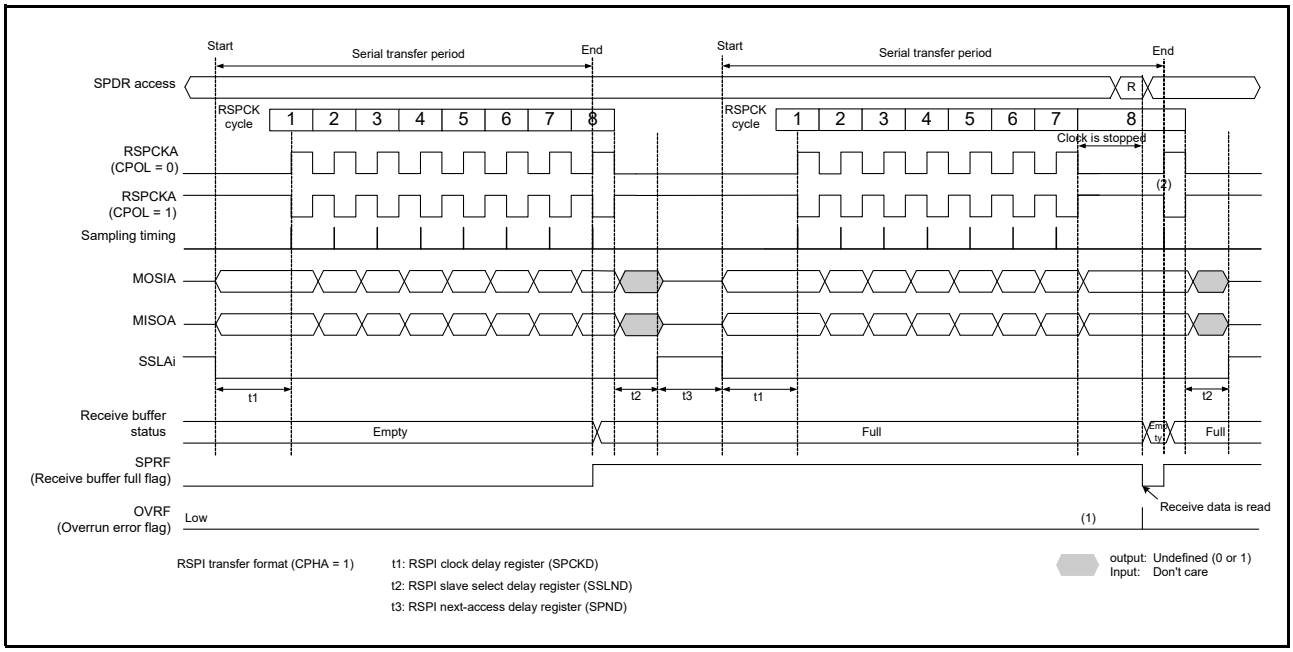


Figure 35.31 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

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).

35.3.9.2 Parity Error

If full-duplex 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 35.32 shows an example of operation of the OVRF and PERF flags. 'SPSR access' shown in Figure 35.32 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 35.32, full-duplex 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).

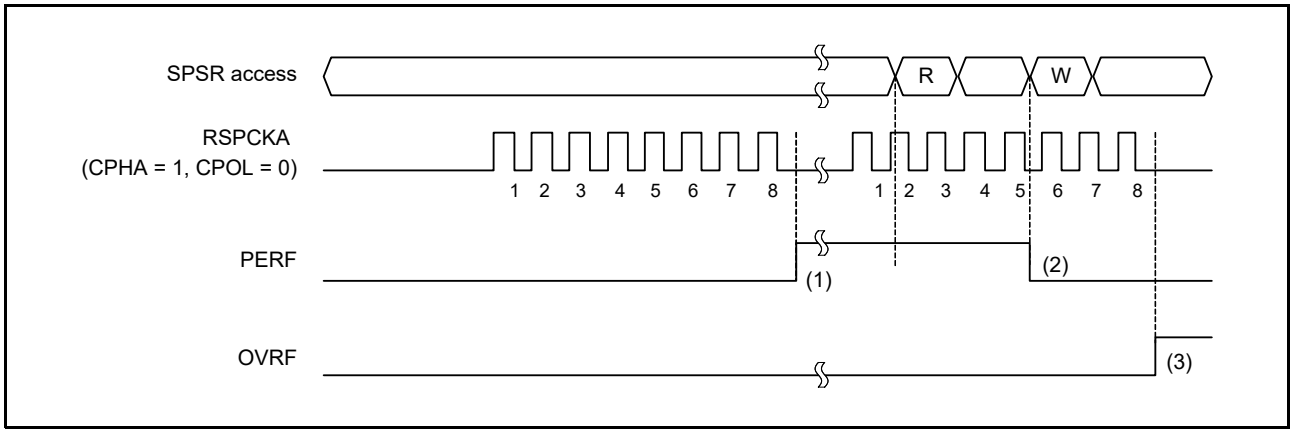


Figure 35.32 Operation Example of PERF Flag

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.

35.3.9.3 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 35.3.10, 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.

35.3.9.4 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), 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 35.3.10, 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.

35.3.10 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.

35.3.10.1 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, 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 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.

35.3.10.2 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 35.3.10.1, Initialization by Clearing the SPE Bit.

35.3.11 SPI Operation

35.3.11.1 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 35.3.9, 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 35.3.5, 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 35.3.5, 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.

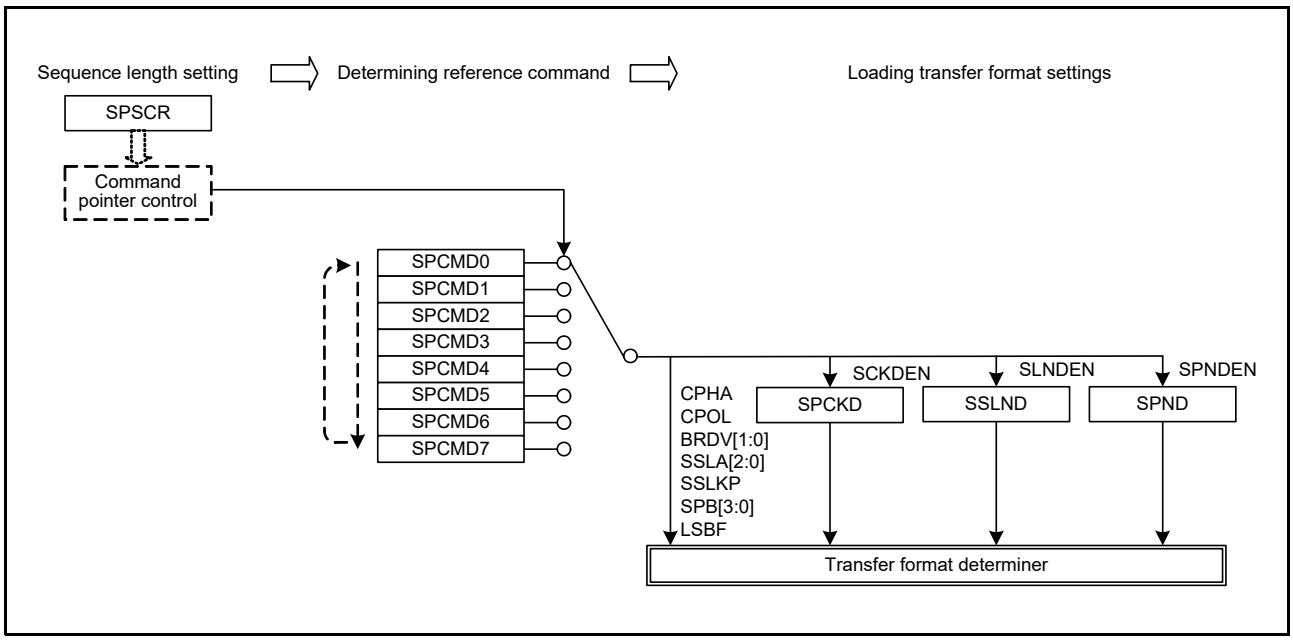


Figure 35.33 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

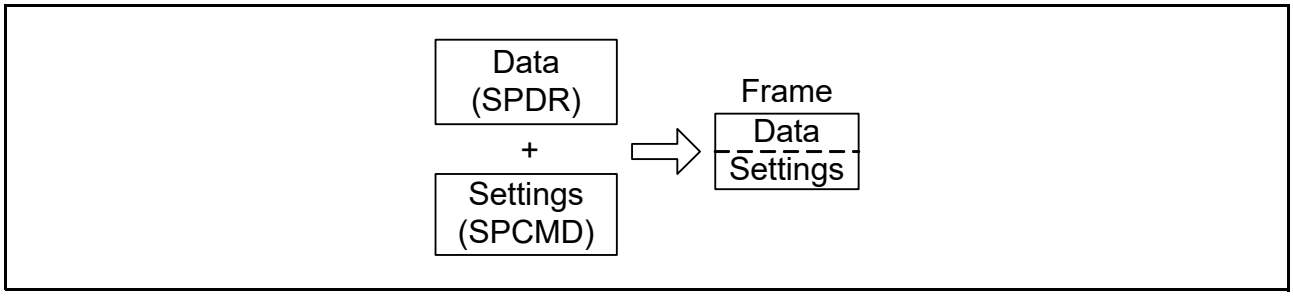


Figure 35.34 Concept of a Frame

Figure 35.35 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 35.4.

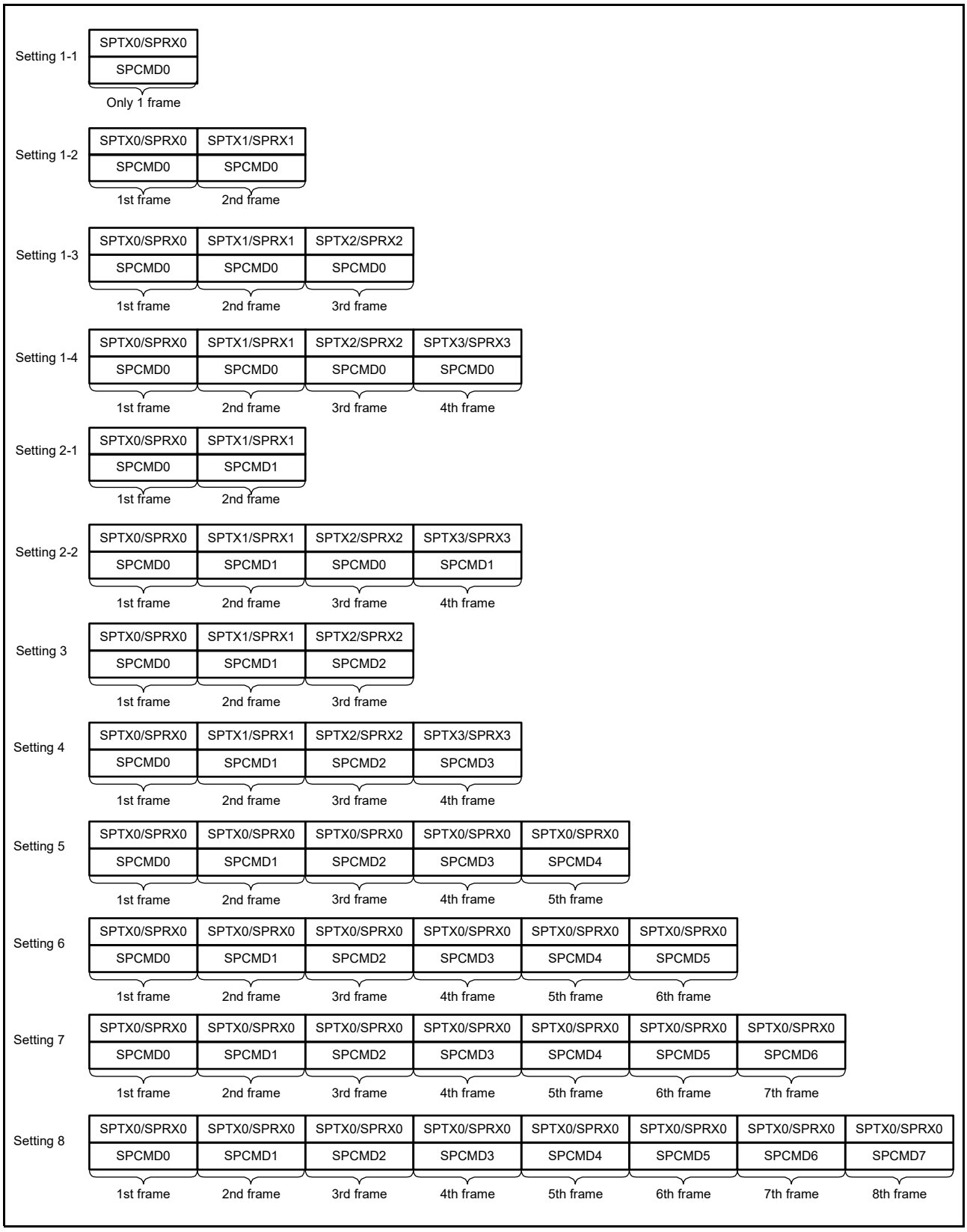


Figure 35.35 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(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 35.36 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 35.36. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

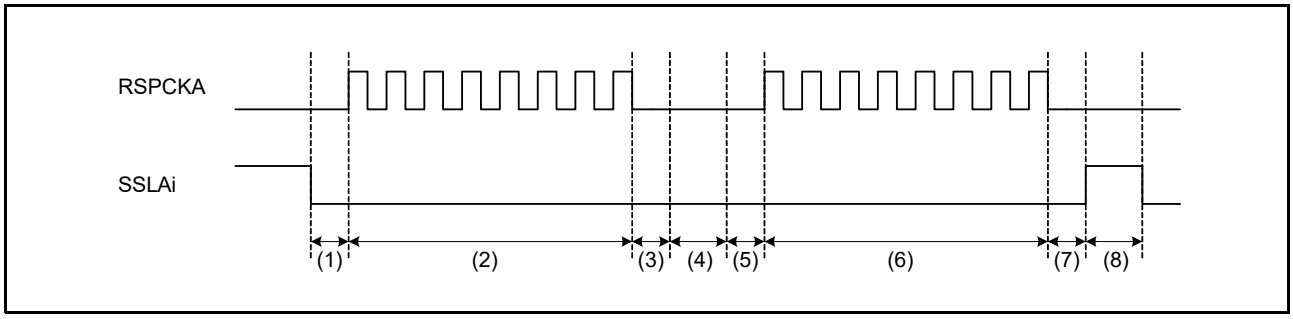


Figure 35.36 Example of Burst Transfer Operation Using SSLKP Bit (CPHA = 1, CPOL = 0)

(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 35.36) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.

(5) RSPCK Delay (t1)

The RSPCK delay value 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 35.8. For a definition of RSPCK delay, refer to section 35.3.5, Transfer Format.

Table 35.8 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 35.9. For a definition of SSL negation delay, refer to section 35.3.5, Transfer Format.

Table 35.9 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 35.10. For a definition of next-access delay, refer to section 35.3.5, Transfer Format.

Table 35.10 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 35.37 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.

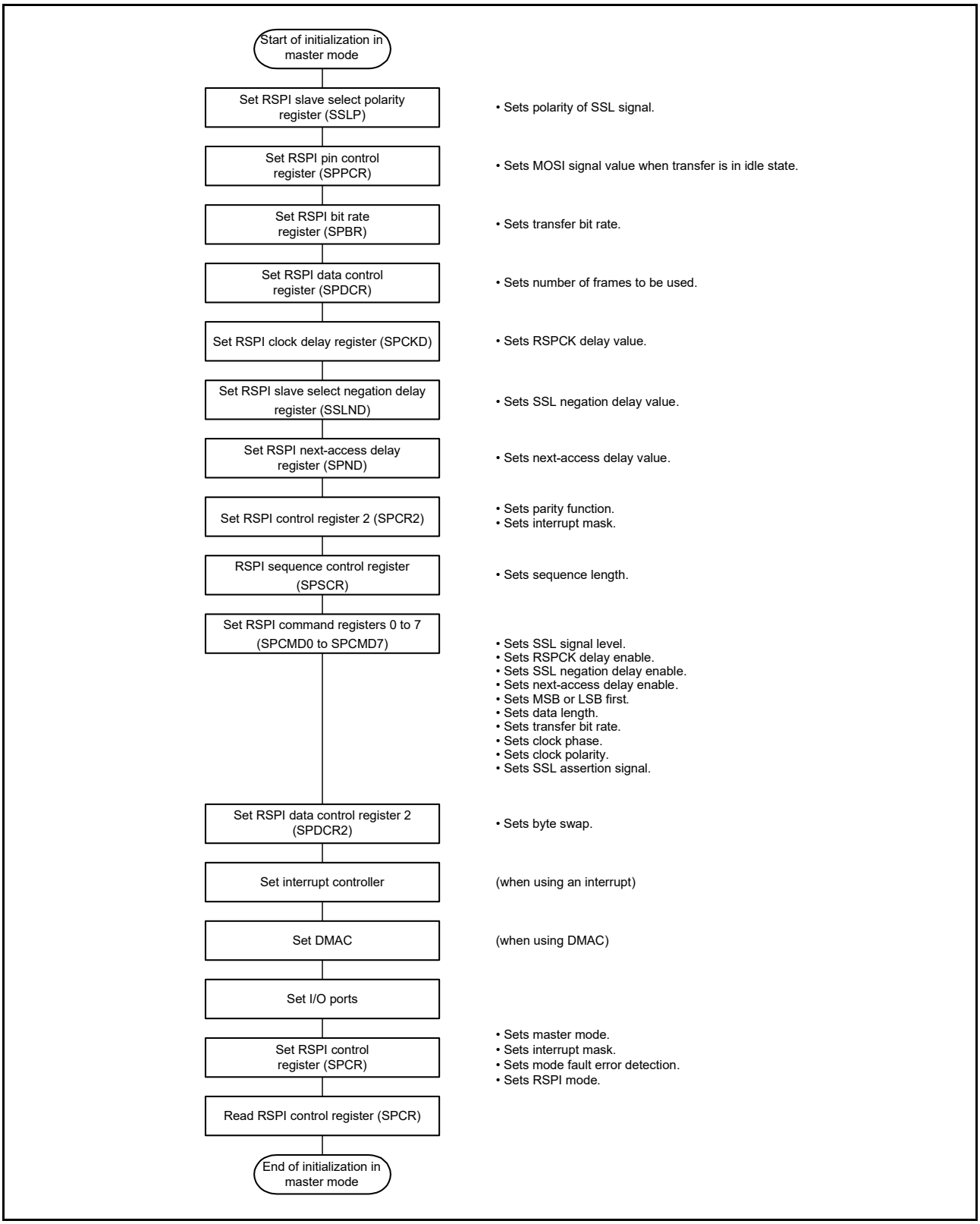


Figure 35.37 Example of Initialization Flowchart in Master Mode (SPI Operation)

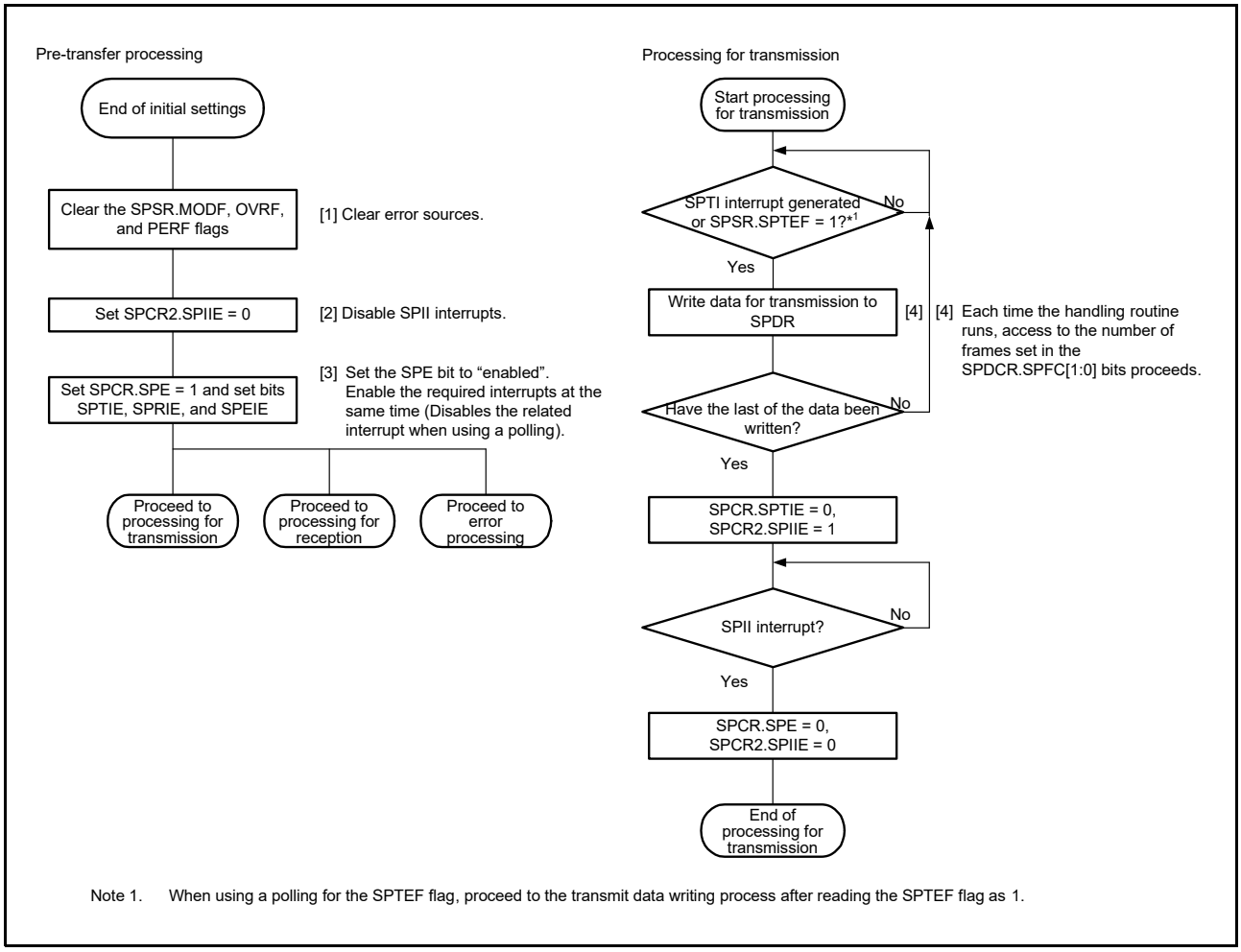
(9) Software Processing Flow

Figure 35.38 to Figure 35.40 show examples of the flow of software processing.

(a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission by enabling the SPII interrupt after the last writing of data for transmission.

The completion of data transmission can also be checked by polling to see if the SPSR.IDLNF flag has become 0, instead of using the SPII interrupt. However, one cycle of PCLK is required for the time from when data for transmission is written in the SPDR register to when the IDLNF flag becomes 1. After the last data is written in the SPDR register, discard the value of the SPSR register once not to judge the condition with the IDLNF flag which has not yet become 1, and read and use the value of the IDLNF flag to confirm the completion of data transmission.



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Figure 35.38 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not support receive-only simplex communications, so processing for transmission is required.

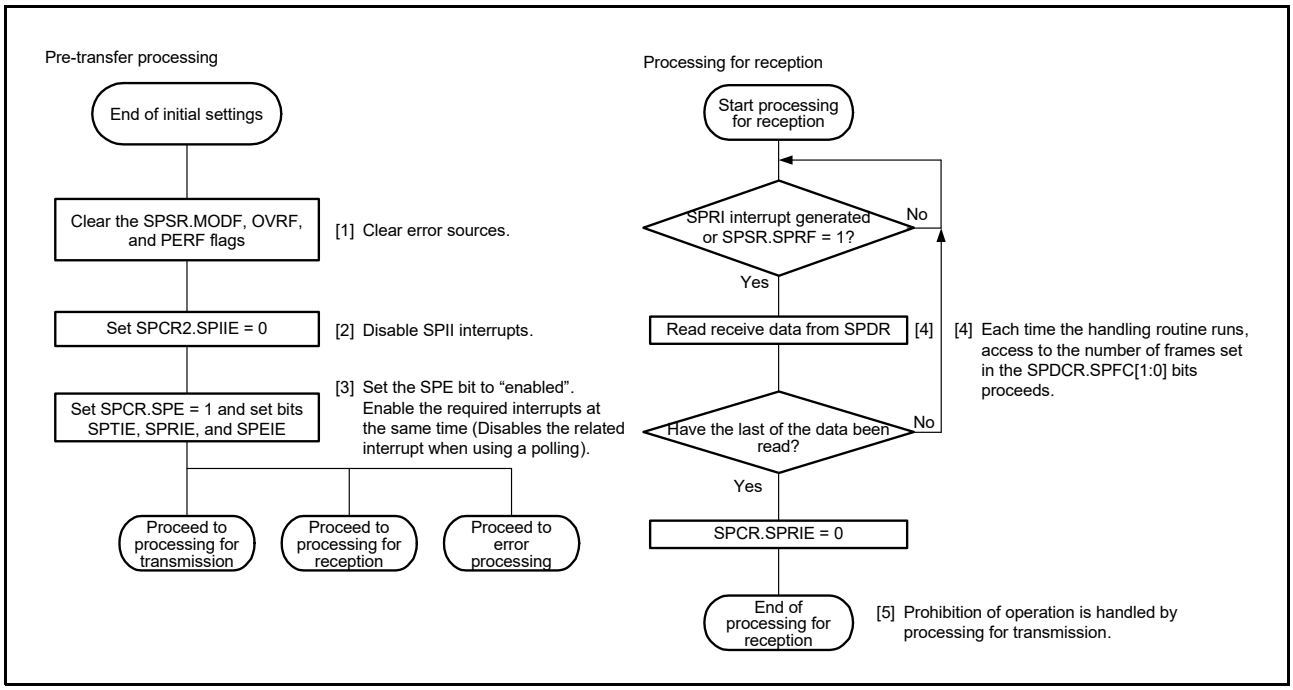


Figure 35.39 Flowchart in Master Mode (Reception)

(c) 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.

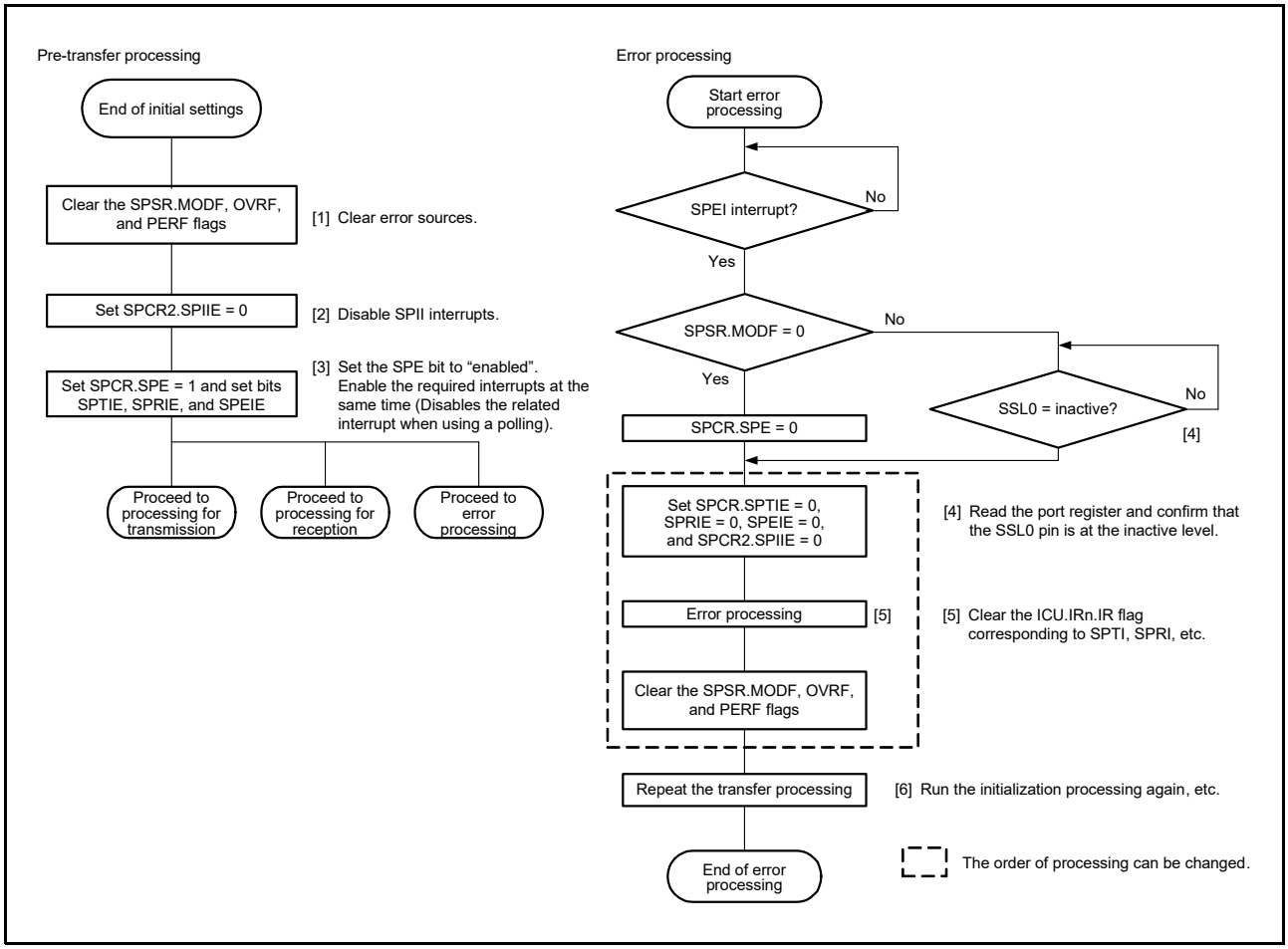


Figure 35.40 Flowchart for Master Mode (Error Processing)

35.3.11.2 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 35.3.5, 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 35.3.9, 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 35.3.5, 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 35.7 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 35.41 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.

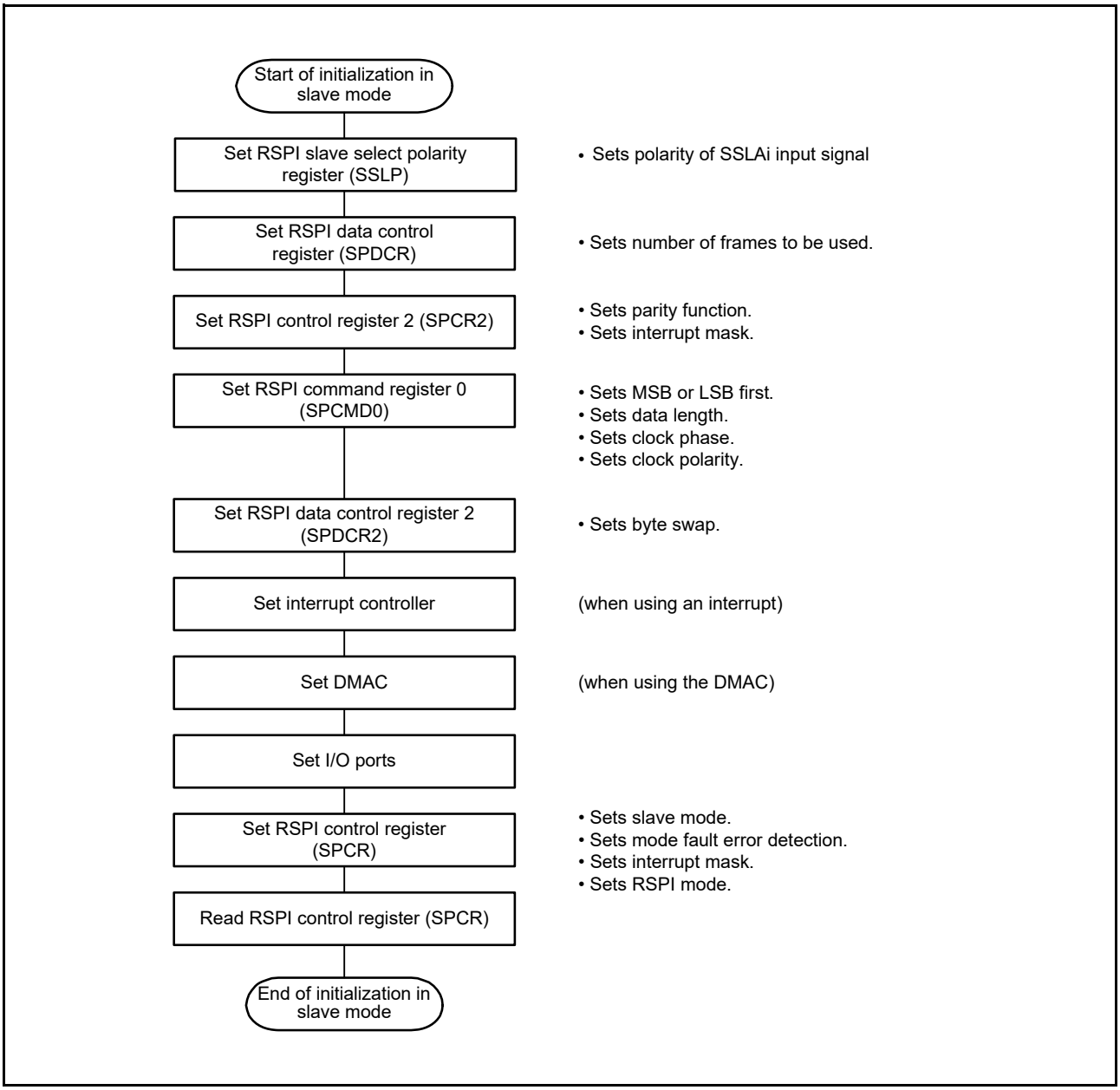


Figure 35.41 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 35.42 to Figure 35.44 show examples of the flow of software processing.

(a) Transmit Processing Flow

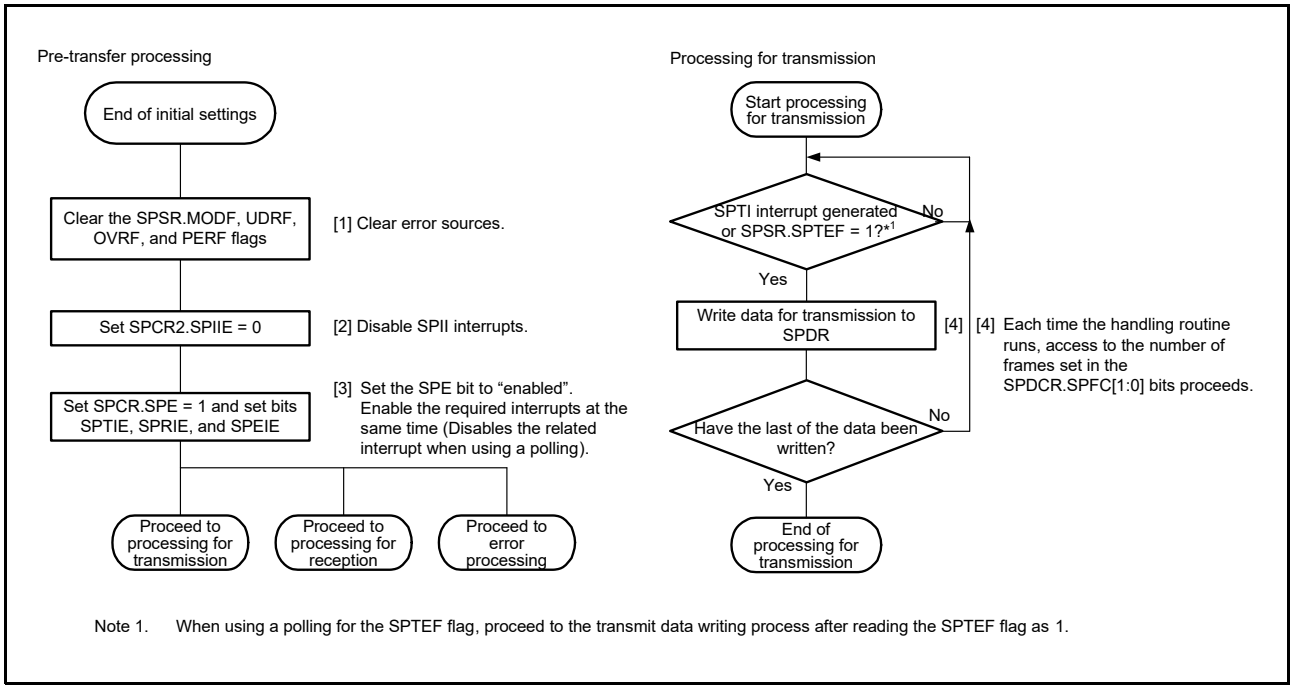


Figure 35.42 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not support receive-only simplex communications, so processing for transmission is required.

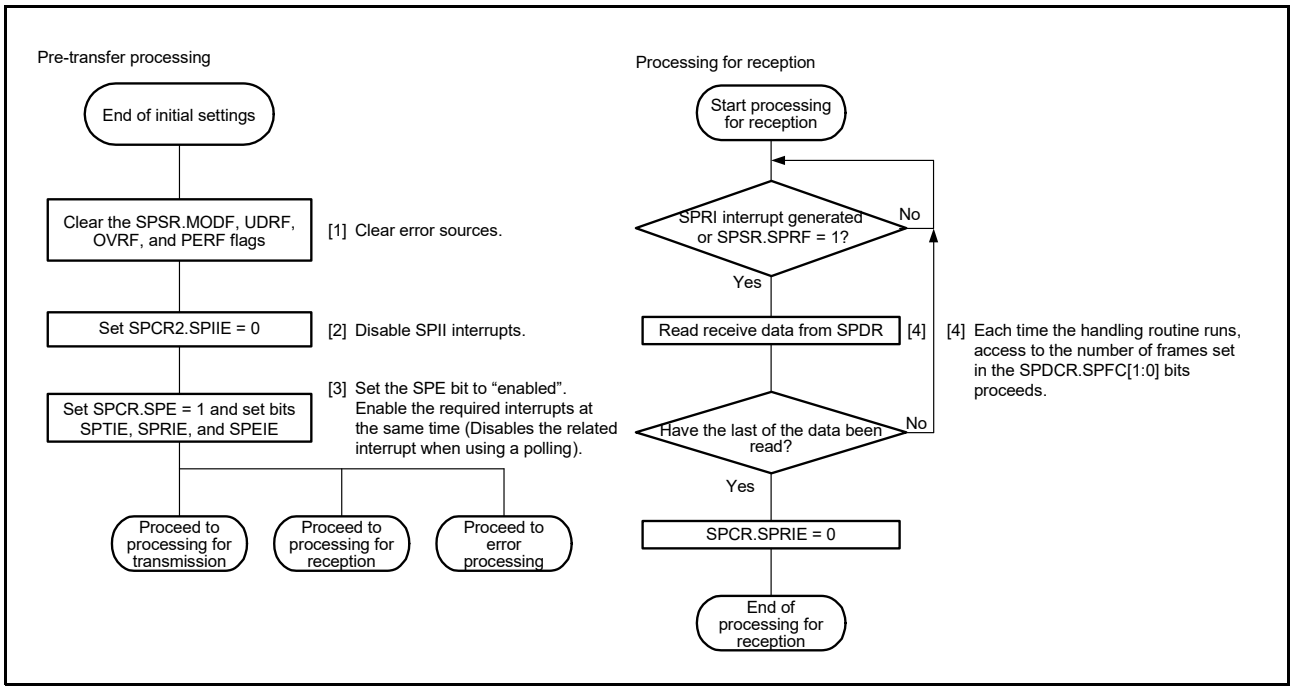


Figure 35.43 Flowchart in Slave Mode (Reception)

(c) 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.

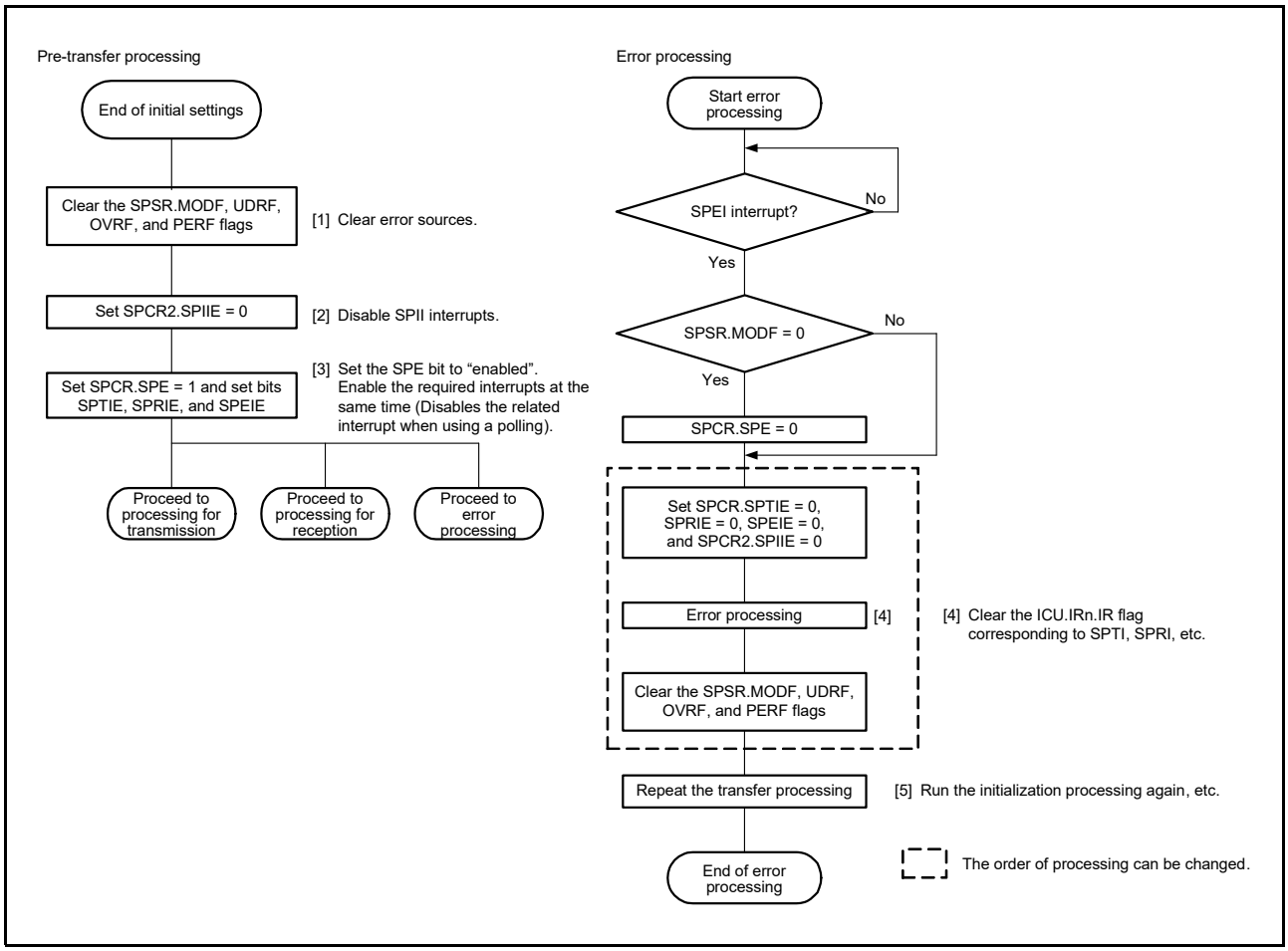


Figure 35.44 Flowchart for Slave Mode (Error Processing)

35.3.12 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).

35.3.12.1 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 35.3.5, 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 35.3.5, 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.

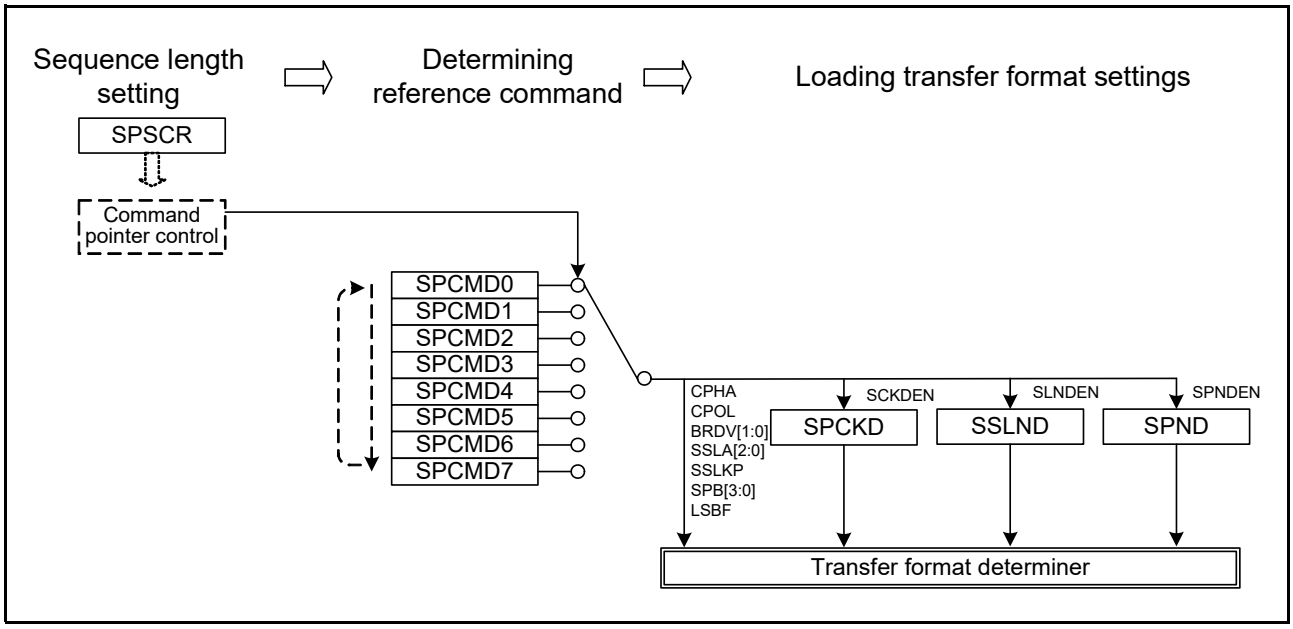


Figure 35.45 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

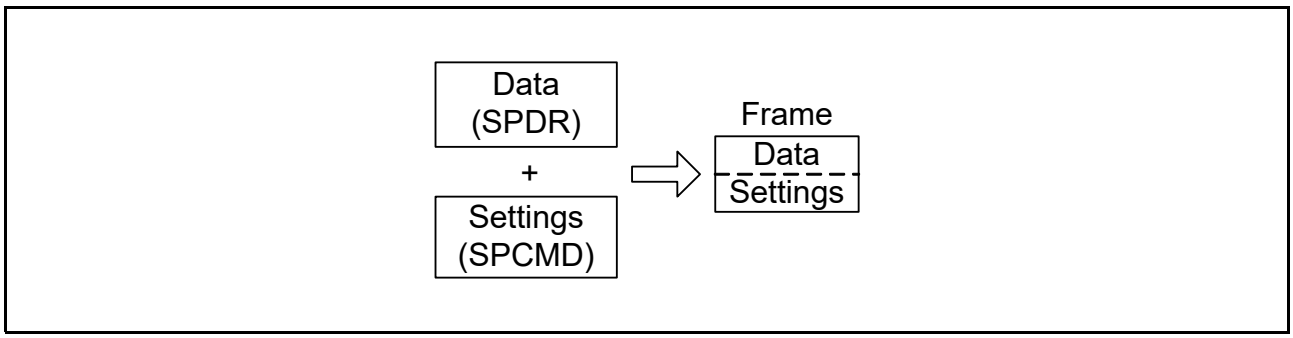


Figure 35.46 Concept of a Frame

Figure 35.47 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 35.4.

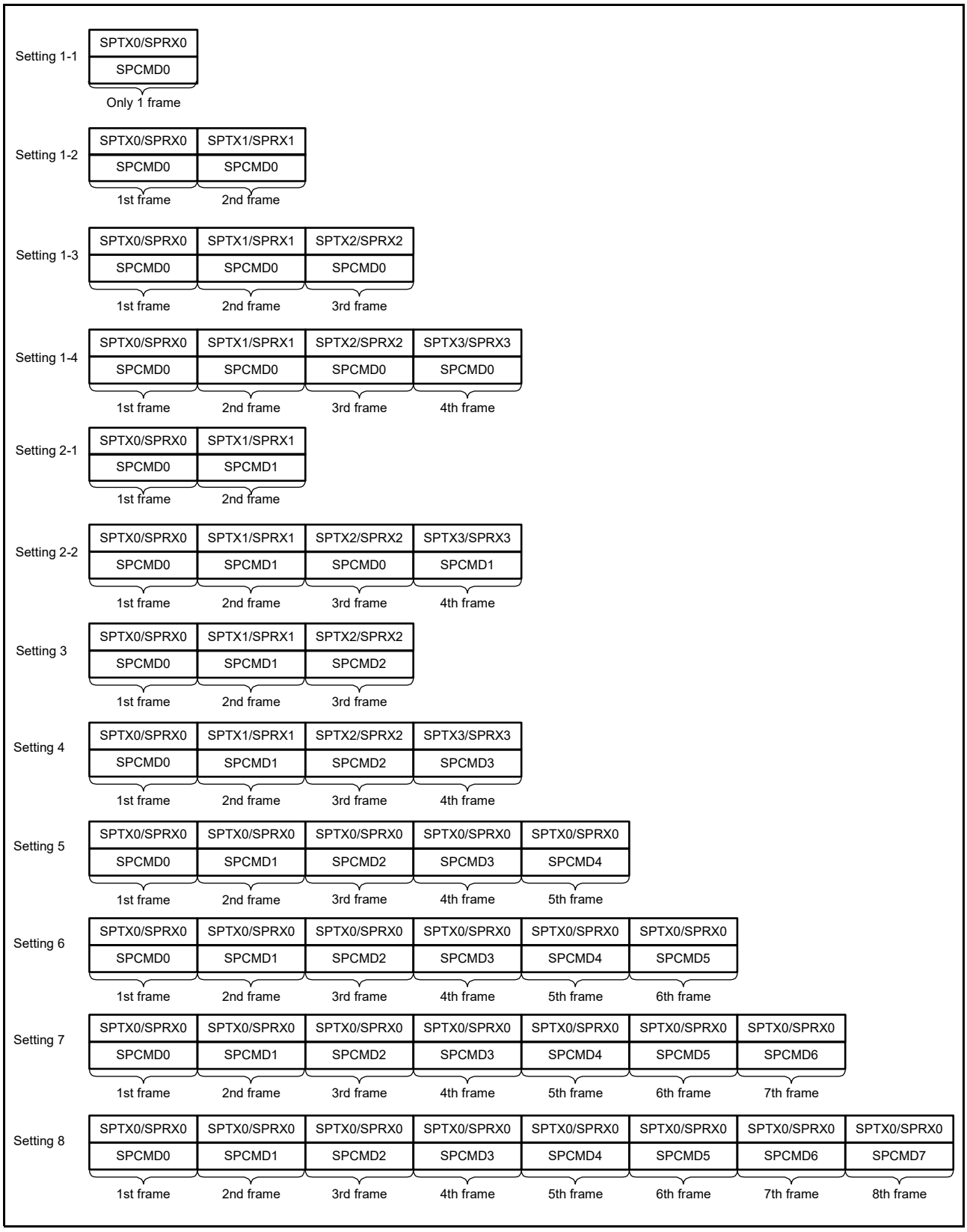


Figure 35.47 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 35.48 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.

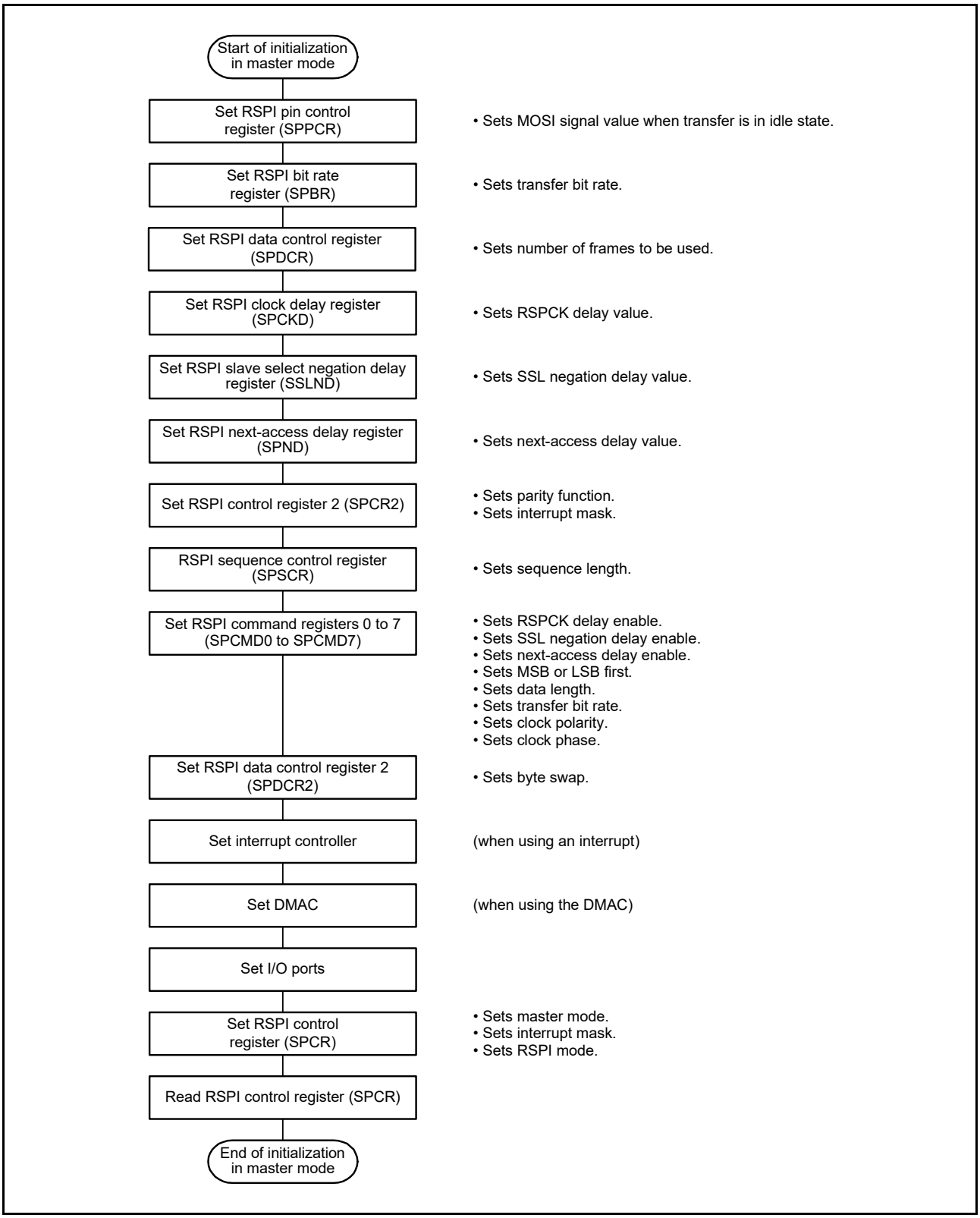


Figure 35.48 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(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 35.3.11.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

35.3.12.2 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 35.3.5, 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 35.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 35.49 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.

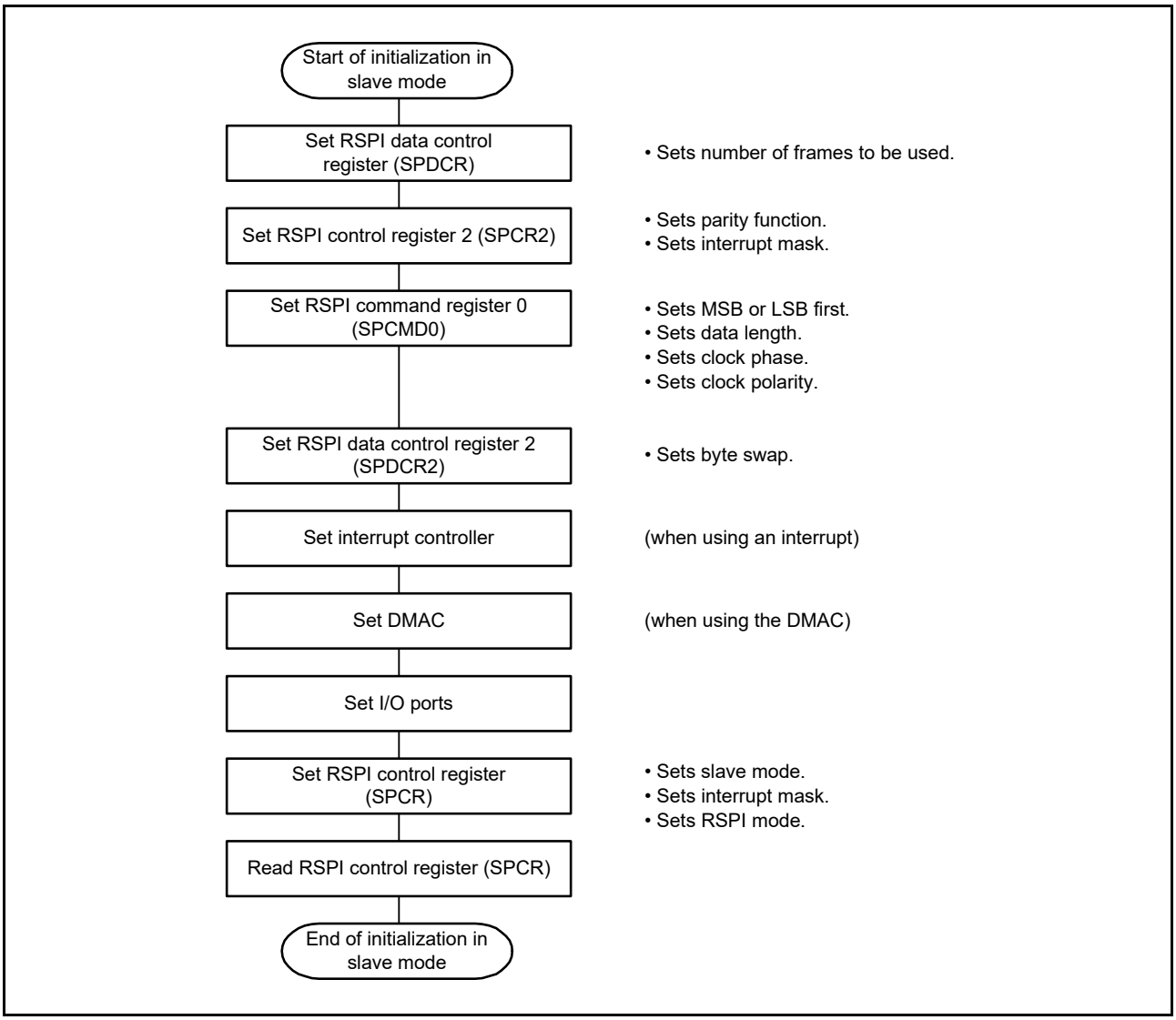


Figure 35.49 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(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 35.3.11.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

35.3.13 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 35.11 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 35.50 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 35.11 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

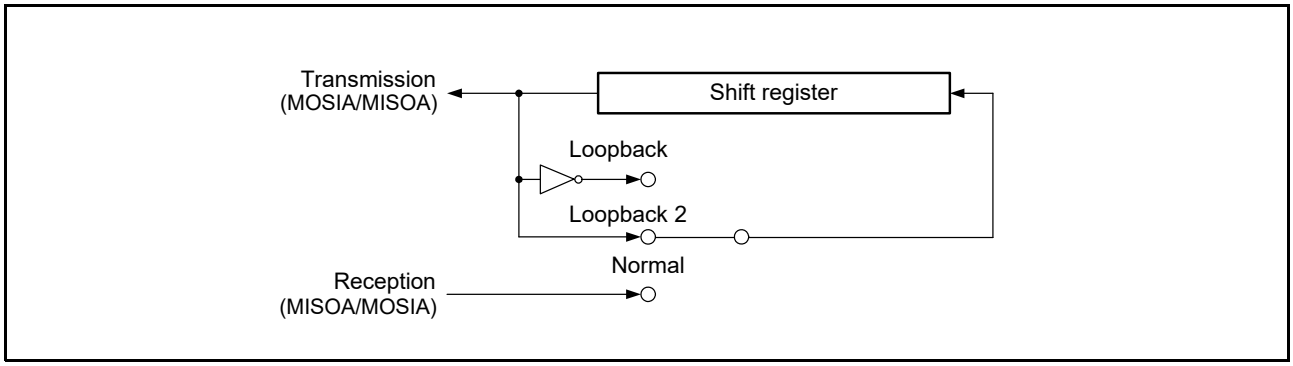


Figure 35.50 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

35.3.14 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 35.51.

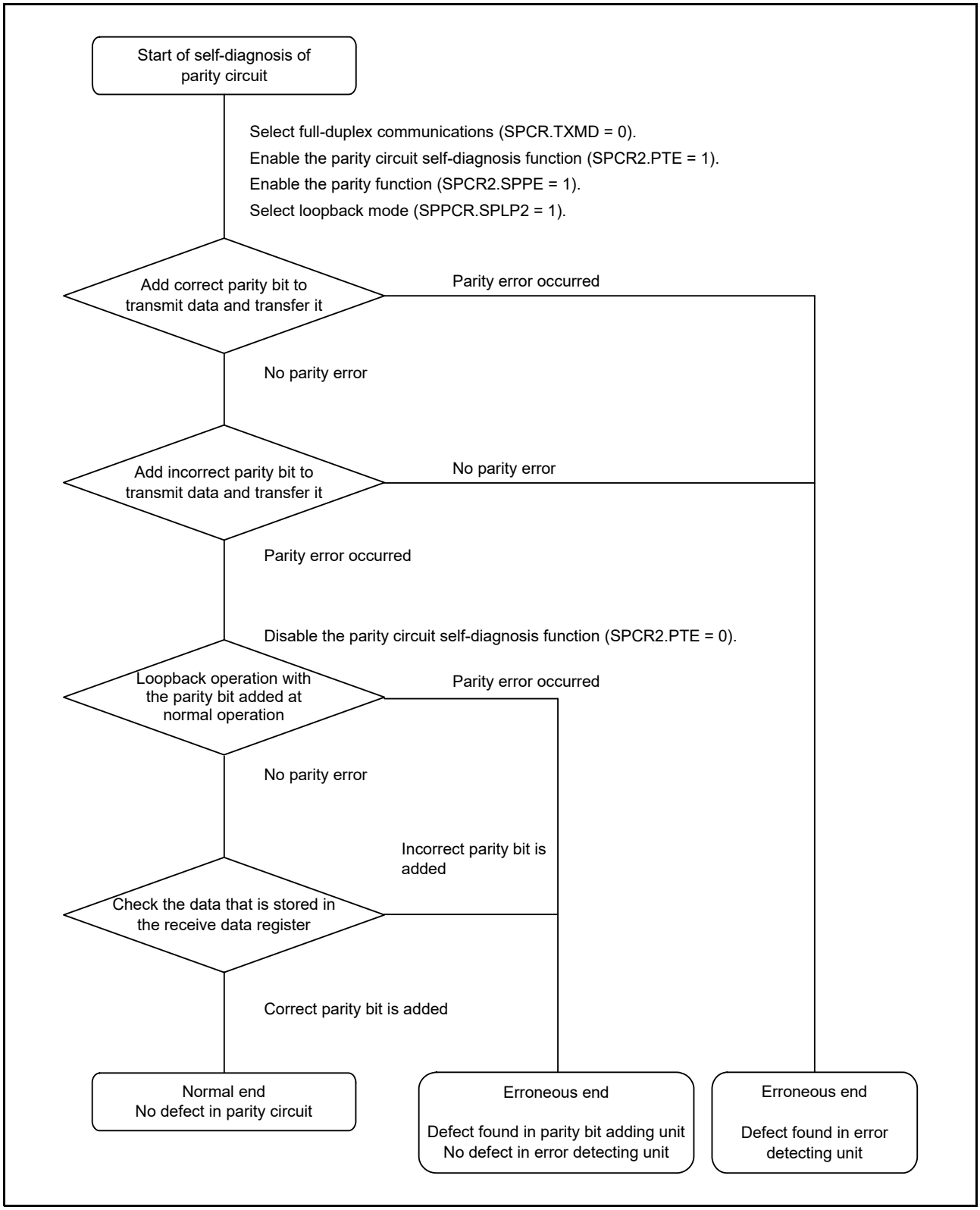


Figure 35.51 Flowchart for Self-Diagnosis of Parity Circuit

35.3.15 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, error (mode fault, underrun, overrun, and parity error), and idle. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 35.12. An interrupt is generated on satisfaction of an interrupt condition in Table 35.12. 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 (DMACA), 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 35.12 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

35.4 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.

35.4.1 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.

35.4.2 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.

35.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 35.13 lists the occurrence conditions of a mode fault event.

Table 35.13 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.

35.4.4 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).

35.4.5 Transmit 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 35.14.

Table 35.14 Generating Conditions of Transmit End Event (Slave mode)

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

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 transmission or the SPCR.SPE bit is cleared by the mode fault error.

35.5 Usage Notes

35.5.1 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.

35.5.2 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.

35.5.3 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.

35.5.4 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.

36. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

36.1 Overview

Table 36.1 lists the specifications of the CRC calculator, and Figure 36.1 shows a block diagram of the CRC calculator.

Table 36.1 CRC Specifications

Item	Description
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	Module stop state can be set.

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit units.

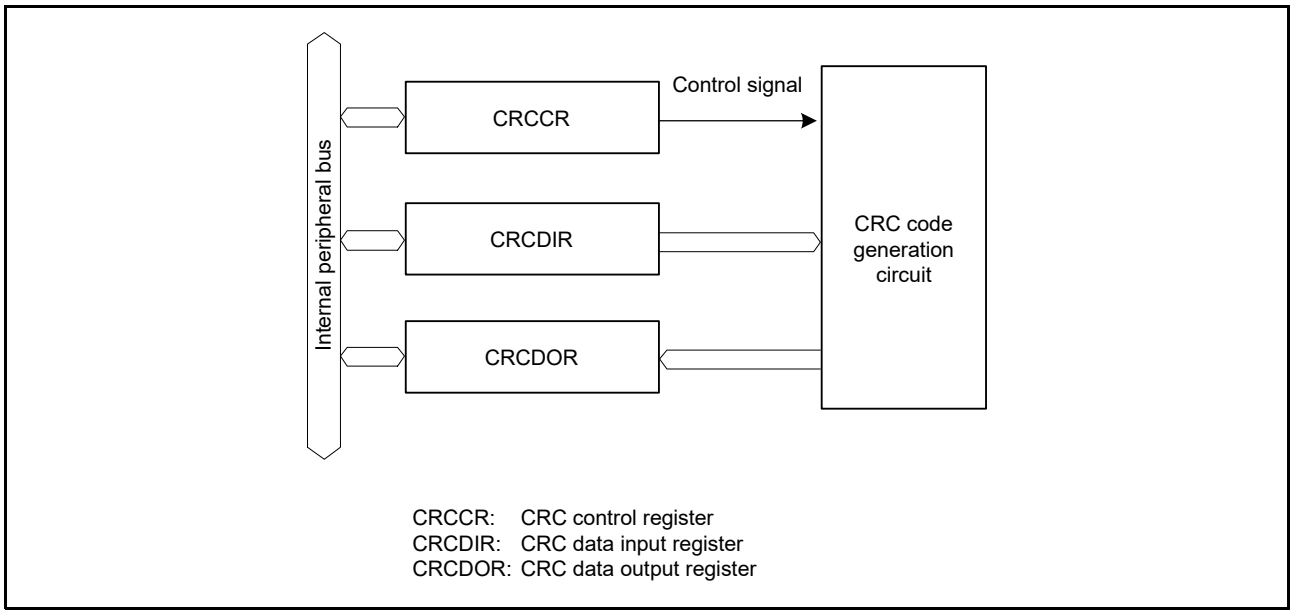


Figure 36.1 CRC Block Diagram

36.2 Register Descriptions

36.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed. 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$)	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	R/W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

This bit selects the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 36.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

36.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

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

CRCDIR is a readable and writable register. Write data for CRC calculation to this register.

36.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

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

CRCDOR is a readable and writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

36.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of the CRCDOR register.

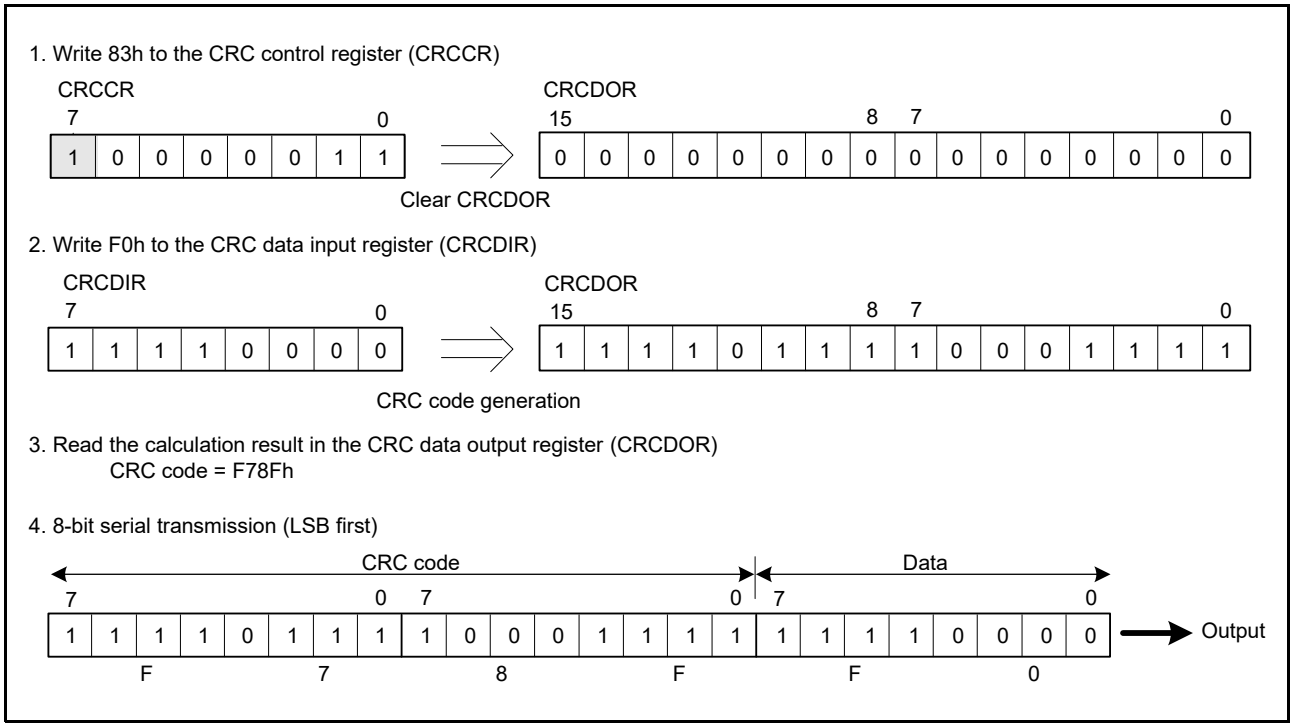


Figure 36.2 LSB First Data Transmission

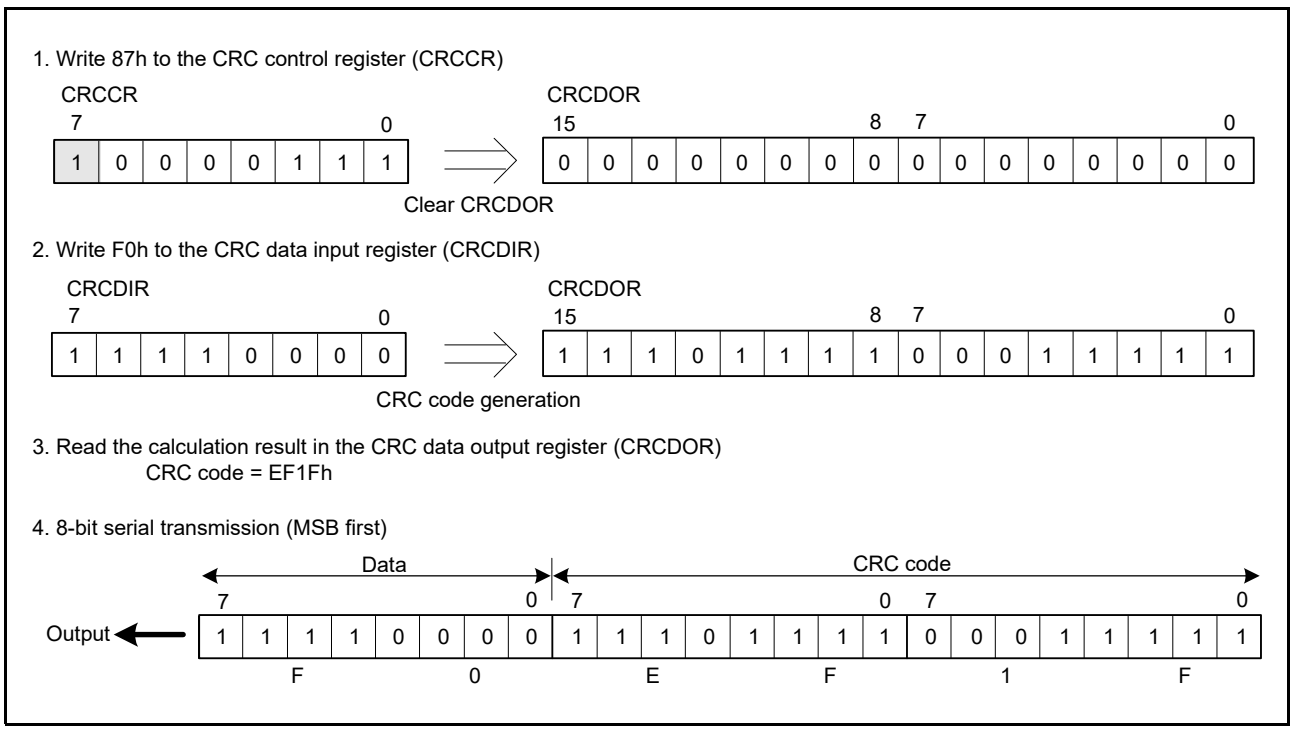


Figure 36.3 MSB First Data Transmission

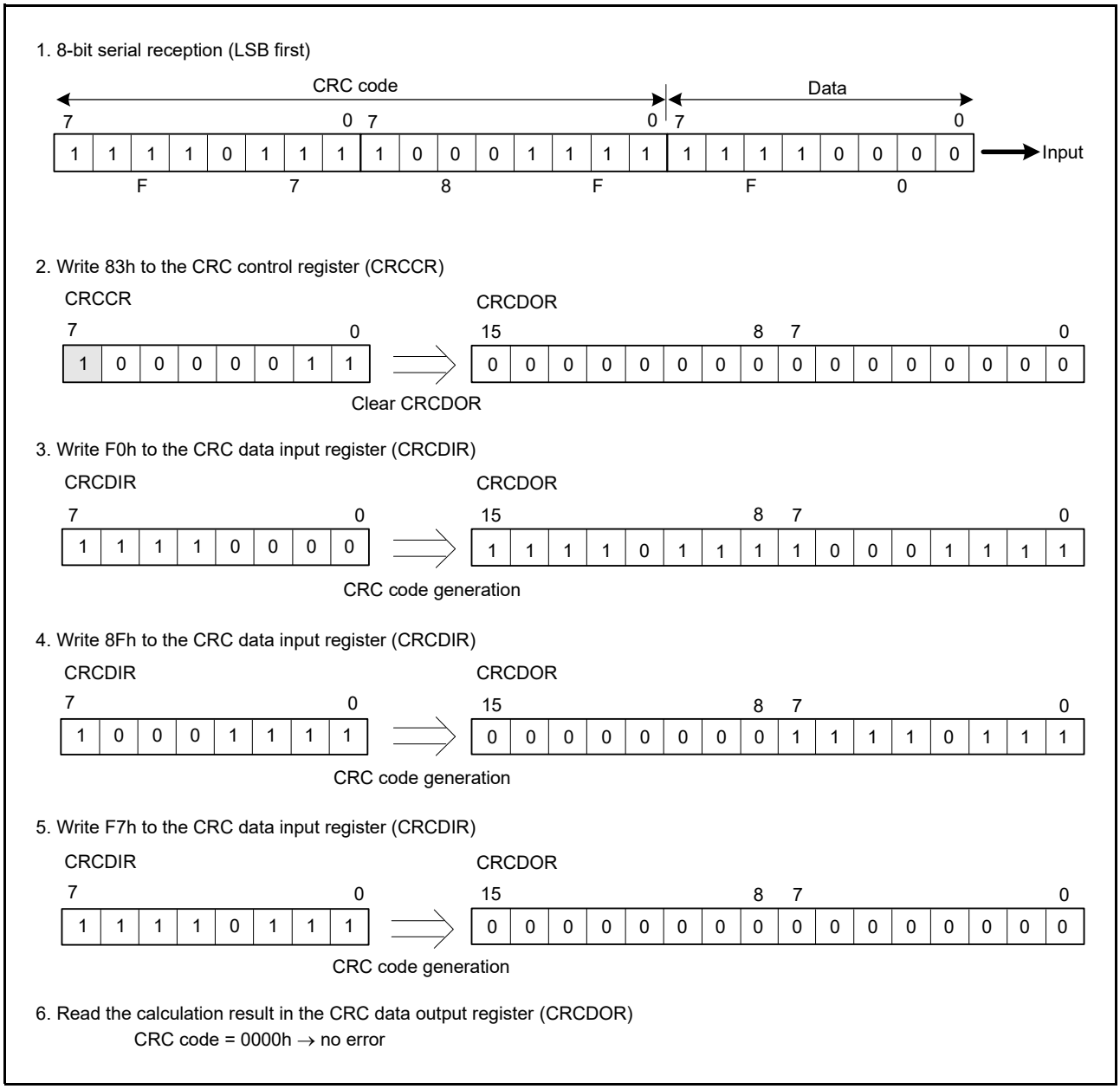


Figure 36.4 LSB First Data Reception

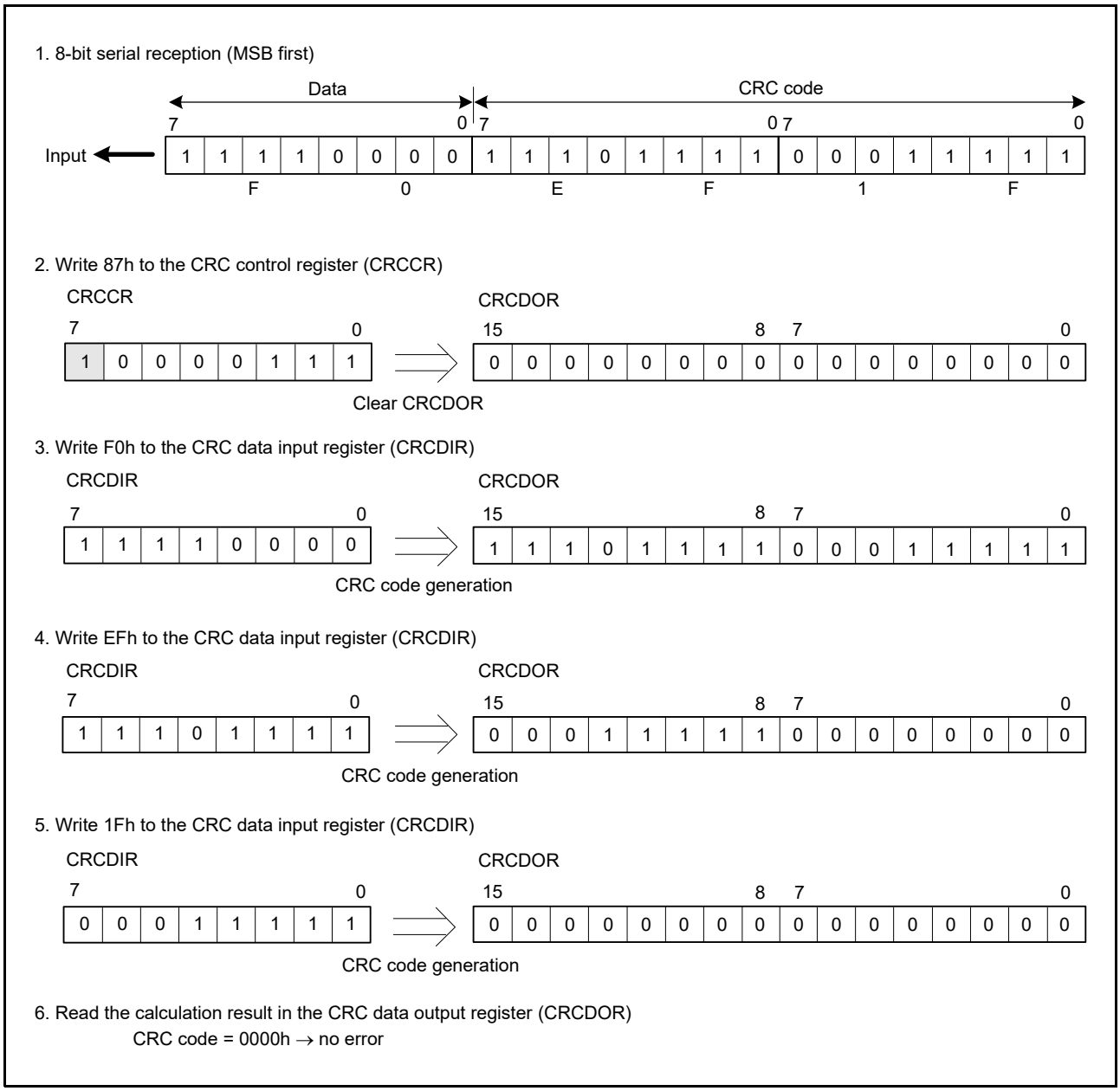


Figure 36.5 MSB First Data Reception

36.4 Usage Notes

36.4.1 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.

36.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

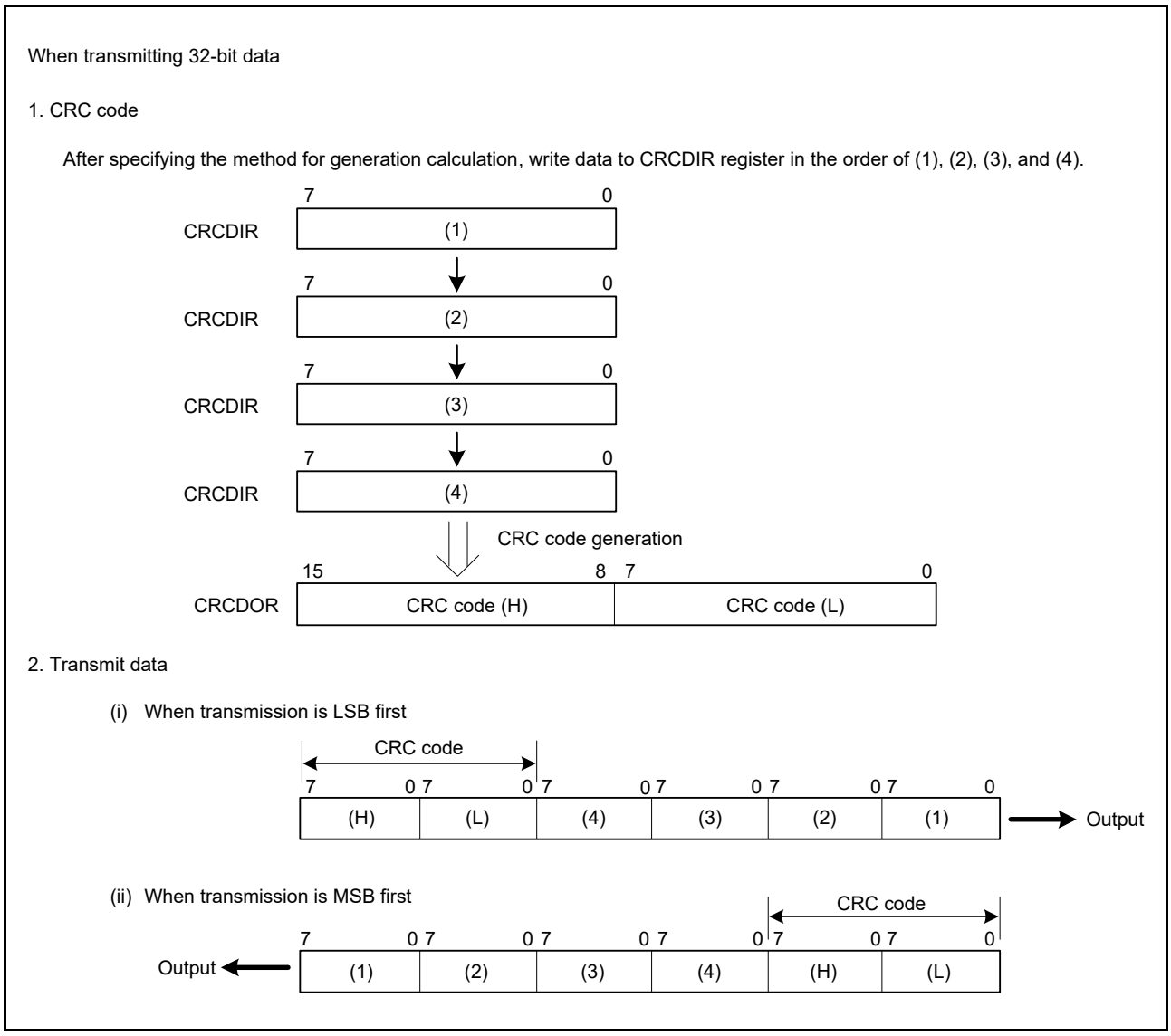


Figure 36.6 LSB First and MSB First Data Transmission

37. 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.

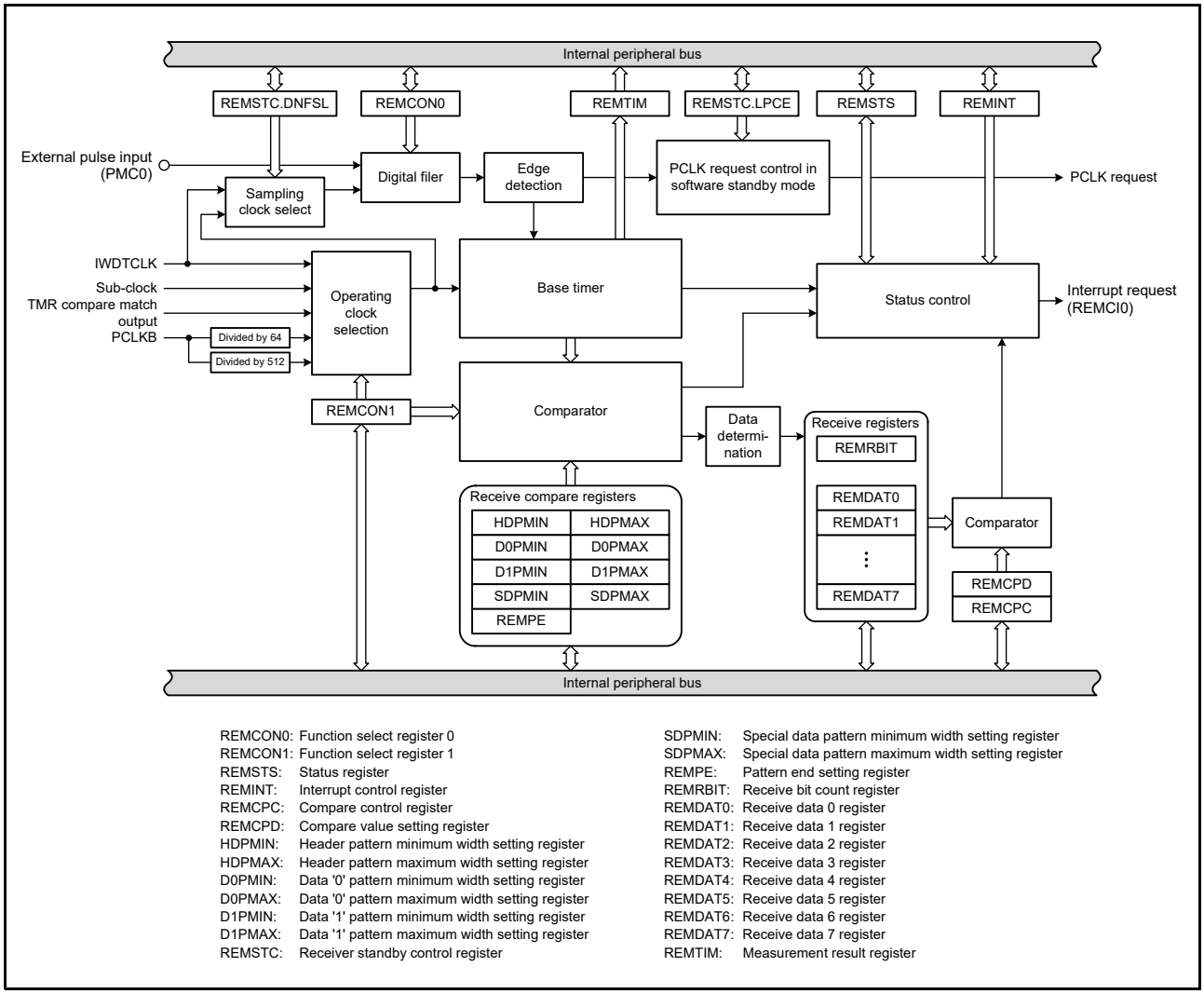
37.1 Overview

Table 37.1 lists the REMC specifications. Figure 37.1 shows a block diagram of the REMC.

Table 37.1 REMC Specifications

Item	Description
External pulse input	PMC0
Operating clock sources*1	• IWDTCLK*2 • 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)*3 • 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. IWDTCLK is a clock supplied from the IWDT-dedicated on-chip oscillator.
Note 3. The sampling clock of the digital filter is an operating clock selected by the REMCON1.CSRC[3:0] bits, or the IWDTCLK.



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
REMSTC: Receiver standby control 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 37.1 REMC Block Diagram

Table 37.2 lists the input pins used for the REMC.

Table 37.2 REMC Pin Configuration

Channel	Pin Name	I/O	Function
REMC0	PMC0	Input	External pulse signal input

37.2 Registers

37.2.1 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.

37.2.2 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 37.3.3, Pattern Setting. 0 1: Format B shown in section 37.3.3, Pattern Setting. 1 0: Format C shown in section 37.3.3, 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 0 0: IWDTCLK 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 37.3.12, Interrupts.

37.2.3 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 37.4.9, 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 37.3, 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 37.3, Measurement Results and Flags.

D0FLG Flag (Data '0' Pattern Match Flag)

[Setting condition]

- See Table 37.3, 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 37.3, Measurement Results and Flags.

D1FLG Flag (Data '1' Pattern Match Flag)

[Setting condition]

- See Table 37.3, 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 37.3, Measurement Results and Flags.

SDFLG Flag (Special Data Pattern Match Flag)

[Setting condition]

- See Table 37.3, 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 37.3, Measurement Results and Flags.

Table 37.3 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.

37.2.4 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).

37.2.5 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).

37.2.6 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).

37.2.7 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).

37.2.8 Header Pattern Maximum Width Setting Register (HDPMAX)

Address(es): REMC0.HDPMAX 000A 0B0Ah

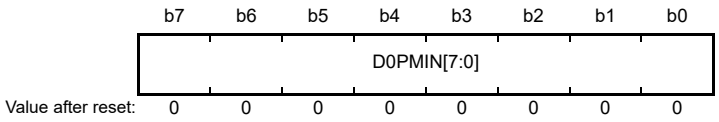
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HDPMAX[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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).

37.2.9 Data ‘0’ Pattern Minimum Width Setting Register (D0PMIN)

Address(es): REMC0.D0PMIN 000A 0B0Ch

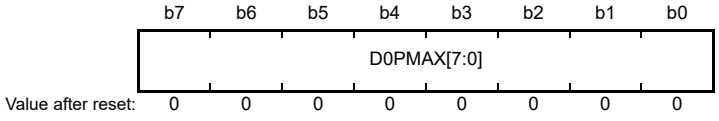


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D0PMIN[7:0]	Data ‘0’ Pattern Minimum Width Setting *1	Set the minimum width of data ‘0’ pattern. Setting range: 00h to FFh	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

37.2.10 Data ‘0’ Pattern Maximum Width Setting Register (D0PMAX)

Address(es): REMC0.D0PMAX 000A 0B0Dh

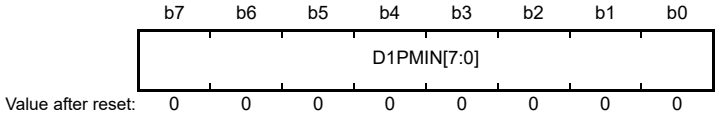


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D0PMAX[7:0]	Data ‘0’ Pattern Maximum Width Setting *1	Set the maximum width of data ‘0’ pattern. Setting range: 00h to FFh	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

37.2.11 Data ‘1’ Pattern Minimum Width Setting Register (D1PMIN)

Address(es): REMC0.D1PMIN 000A 0B0Eh



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).

37.2.12 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).

37.2.13 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).

37.2.14 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).

37.2.15 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).

37.2.16 Receiver Standby Control Register (REMSTC)

Address(es): REMC0.REMSTC 000A 0B16h

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

Bit	Symbol	Bit Name	Description	R/W
b0	LPCE	Low Power Control Enable*1	0: The PCLK supply in software standby mode is disabled. 1: The PCLK supply in software standby mode is enabled.	R/W
b1	DNFSL	Digital Filter Clock Selection*2	0: REMC operating clock is selected as a sampling clock 1: IWDTCLK is selected as a sampling clock	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 1 when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped)
Note 2. This bit can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

LPCE Bit (Low Power Control Enable)

This bit enables or disables the PCLK supply when the input level of the PMC0 pin is changed in the software standby mode.

When this bit is set to 1 (the PCLK supply in software standby mode is enabled) and the input level of the PMC0 pin is changed while the PCLK request signal is set to the snooze mode transition condition (SNZCR.REMCSNZSEL[1:0] bits = 10b), the MCU is placed in the snooze mode, and each oscillator that was once operating before transition to the software standby mode resumes its operation and starts supplying the PCLK. Therefore, data can be received even if PCLKB/64 or PCLKB/512 is selected as the REMC operating clock. In addition, the PCLK supply is restarted when the oscillation stabilization time of the oscillator is elapsed after the change of the input level of the PMC0 pin. During that time, the REMC operating clock is not supplied and the base timer is stopped. Select the high-speed on-chip oscillator (HOCO) as the system clock source.

When setting this bit to 1, set the REMCON0.FIL bit to 1 (enables the digital filter) and set the REMSTC.DNFSL bit to 1 (IWDTCLK is selected as a sampling clock).

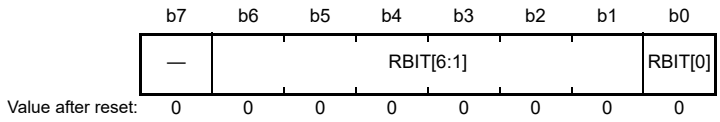
After returning from the software standby mode due to a compare match interrupt or header pattern match interrupt, set this bit to 0.

DNFSL Bit (Digital Filter Clock Selection)

This bit is used to select the sampling clock of the digital filter. Set this bit to 1 when setting the LPCE bit to 1 (the PCLK supply in software standby mode is enabled).

37.2.17 Receive Bit Count Register (REMRBIT)

Address(es): REMC0.REMRBIT 000A 0B17h

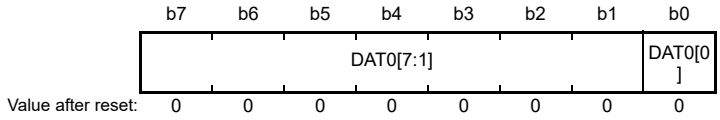


Bit	Symbol	Bit Name	Description	R/W
b0	RBIT[0]	Receive Bit Count Check 0	Receive bit count can be read. These bits indicate the bit position of the buffer to be stored by counting the detected data ‘0’ pattern or data ‘1’ pattern. • When the receive bit count exceeds 64 (40h), the value returns to 1. • 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 37.4.9, Reading Registers.
Note: The values of this register are initialized when the REMCON1.EN bit is 0.

37.2.18 Receive Data 0 Register (REMDAT0)

Address(es): REMC0.REMDAT0 000A 0B18h



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 37.4.9, Reading Registers.
Note: The values of this register are initialized when the REMCON1.EN bit is 0.

37.2.19 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 37.4.9, 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 37.3.8, Receive Data Buffer.

37.2.20 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 37.4.9, Reading Registers.

Note: The values of this register are initialized when the REMCON1.EN bit is 0.

37.3 Operation

37.3.1 Overview of REMC Operation

Figure 37.2 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.

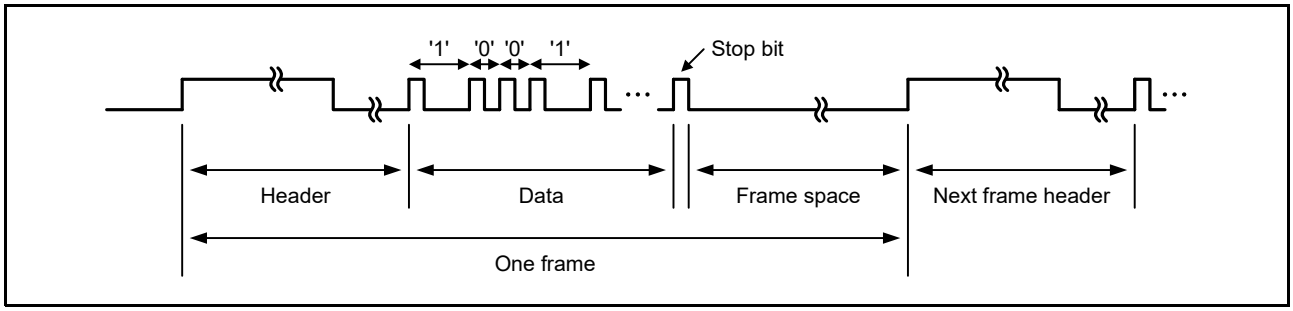


Figure 37.2 Example of Remote Control Signal

37.3.2 Initial Setting

Initialize the REMC according to the procedure shown in Figure 37.3 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, REMCON0.FILSEL, and REMSTC.DNFSL 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.

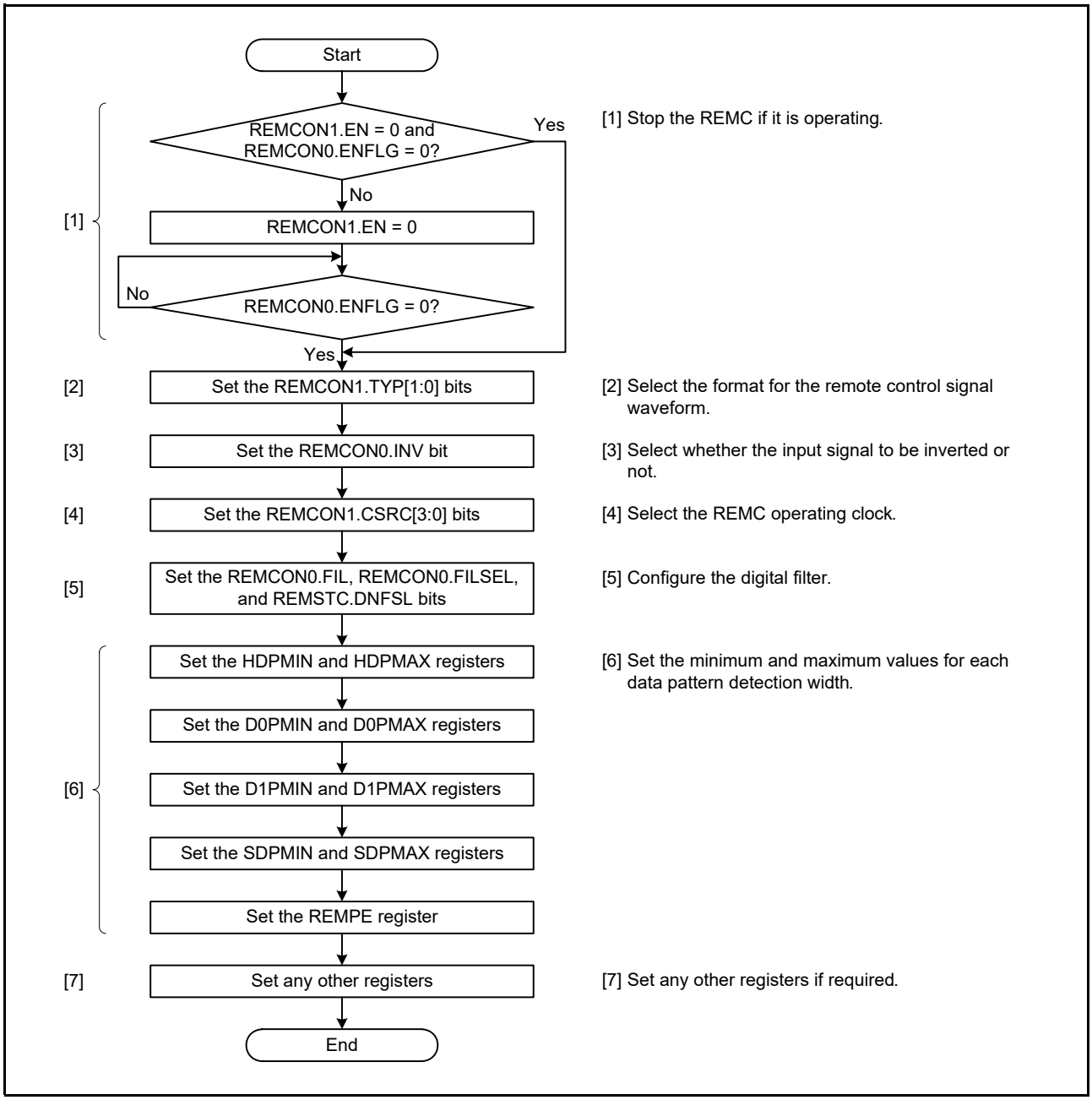


Figure 37.3 Example of Flowchart for Initial Settings of REMC

37.3.3 Pattern Setting

The format for capturing the remote control signal reception waveform can be set by setting the REMCON1.TYP[1:0] bits. Figure 37.4 and Figure 37.5 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)

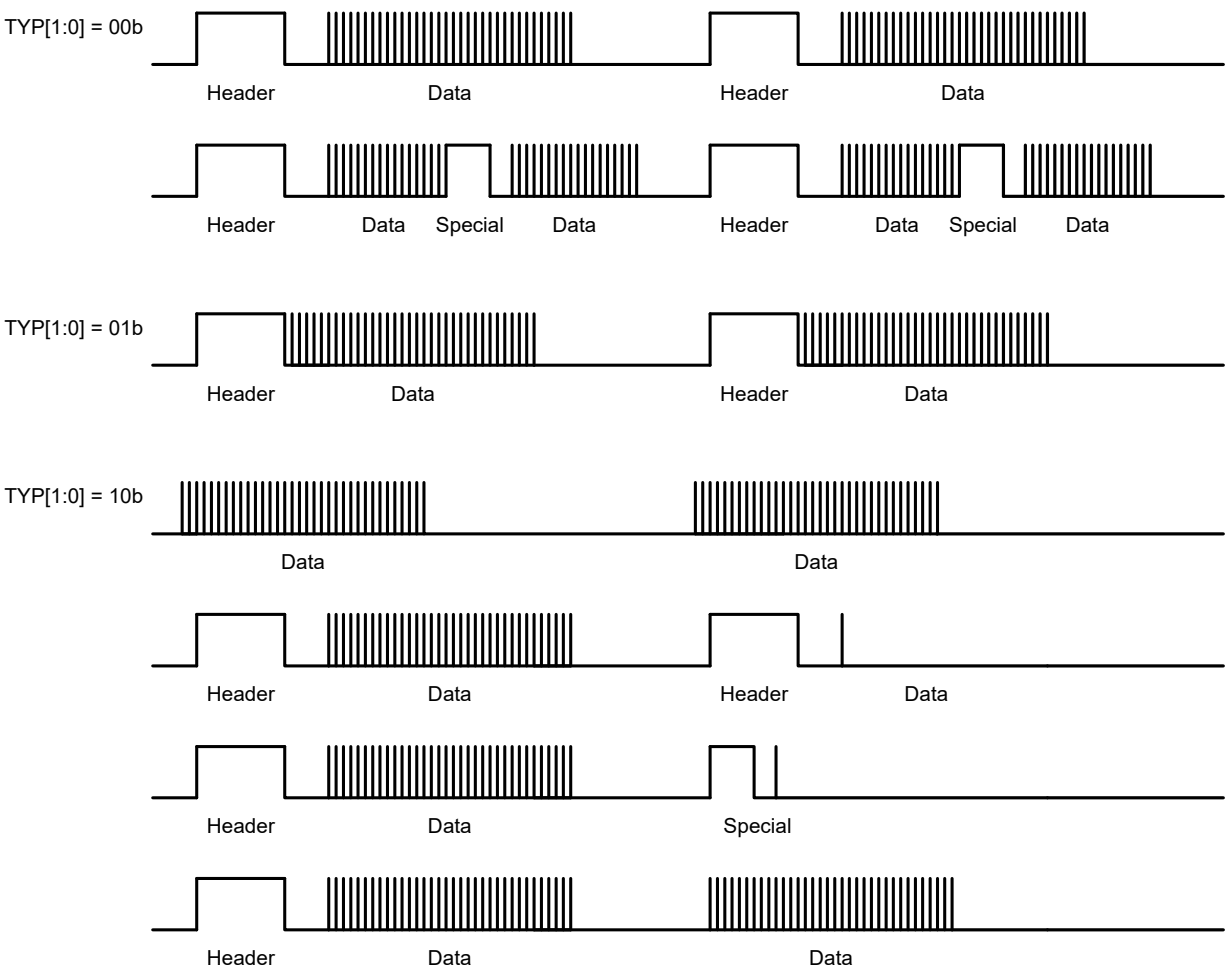
The measured result is determined from the setting value of the header pattern at the rising edge of the internal input signal.
When the header pattern is received, the measured result is determined from the setting values of the data ‘0’, data ‘1’ and special data patterns at the rising edge of the 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				

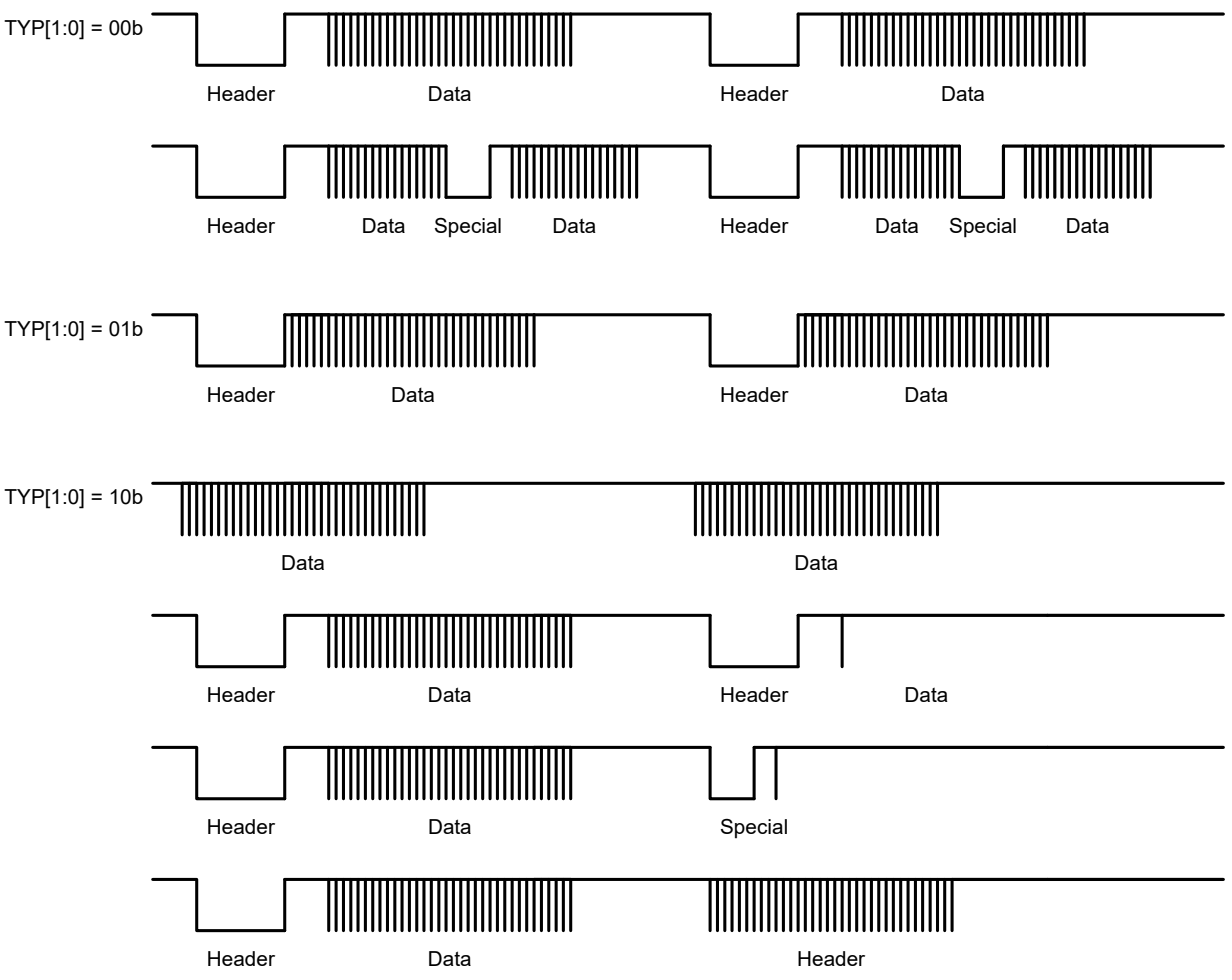
TYP[1:0]: Bits in the REMCON1 register

↑↓ · · · 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 37.4 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 37.5 Example of Remote Control Signal Reception Waveform Captured by Setting REMCON1.TYP[1:0] Bits (REMCON0.INV = 1)

37.3.4 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 IWDTCLK supplied from the IWDT-dedicated on-chip oscillator, the sub-clock supplied from the sub-clock oscillator, or TMR compare match output.
When using the IWDTCLK as the sampling clock of the digital filter, it is necessary to supply the IWDTCLK. When supplying the IWDTCLK sub-clock to the REMC, take note of the respective procedures for supplying these clocks. The following describes how to supply these clocks.

37.3.4.1 When Using IWDTCLK as REMC Operating Clock

This section describes the procedure for using the IWDTCLK supplied from the IWDT-dedicated on-chip oscillator as the REMC operating clock.
When the ILOCOCR.ILCSTP bit is set to 0, the IWDT-dedicated on-chip oscillator starts operating. After oscillation starts, the operating clock is supplied to the REMC when the oscillation stabilization wait time has elapsed. When continuing operation of the IWDT-dedicated on-chip oscillator in the software standby mode, set the IWDTCSTPR.SLCSTP bit to 0.
For details on the ILOCOCR register, refer to section 9.2.10, IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR). For details on the IWDTCSTPR register, refer to section 29.2.5, IWDT Count Stop Control Register (IWDTCSTPR).
Note that the operating clock is also supplied to the IWDT while the IWDT-dedicated on-chip oscillator is operating. When using the IWDTCLK as the REMC operating clock, do not use the IWDT function in order to prevent an unexpected reset or an interrupt from being generated.
Figure 37.6 shows an example of the flowchart for starting the IWDTCLK supply to the REMC.

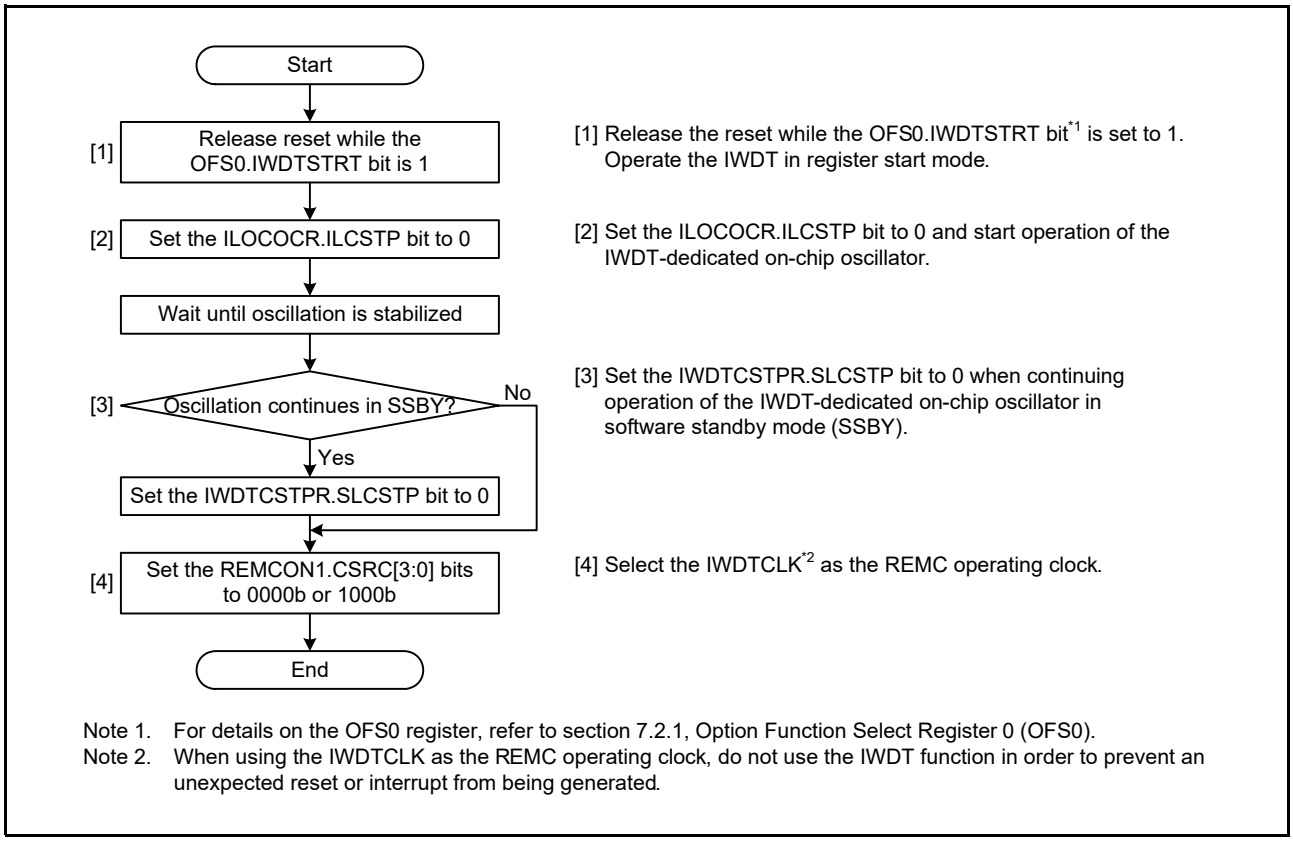


Figure 37.6 Example of Flowchart for Starting IWDTCLK Supply to REMC

37.3.4.2 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).

37.3.4.3 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 (TMRa).

37.3.5 PMC0 Input

The options below can be selected in PMC0 input.

- Input polarity
- Digital filter

Figure 37.7 shows the configuration of PMC0 internal input signal generation.

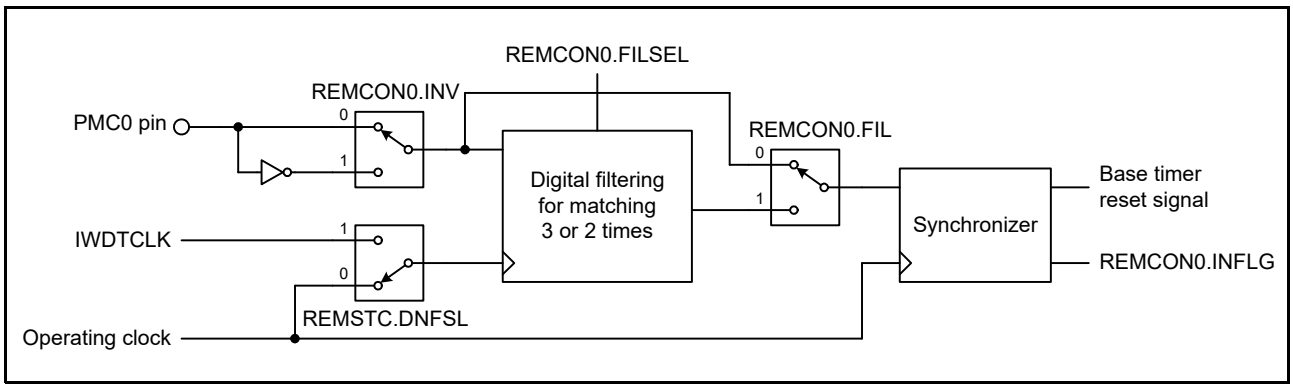
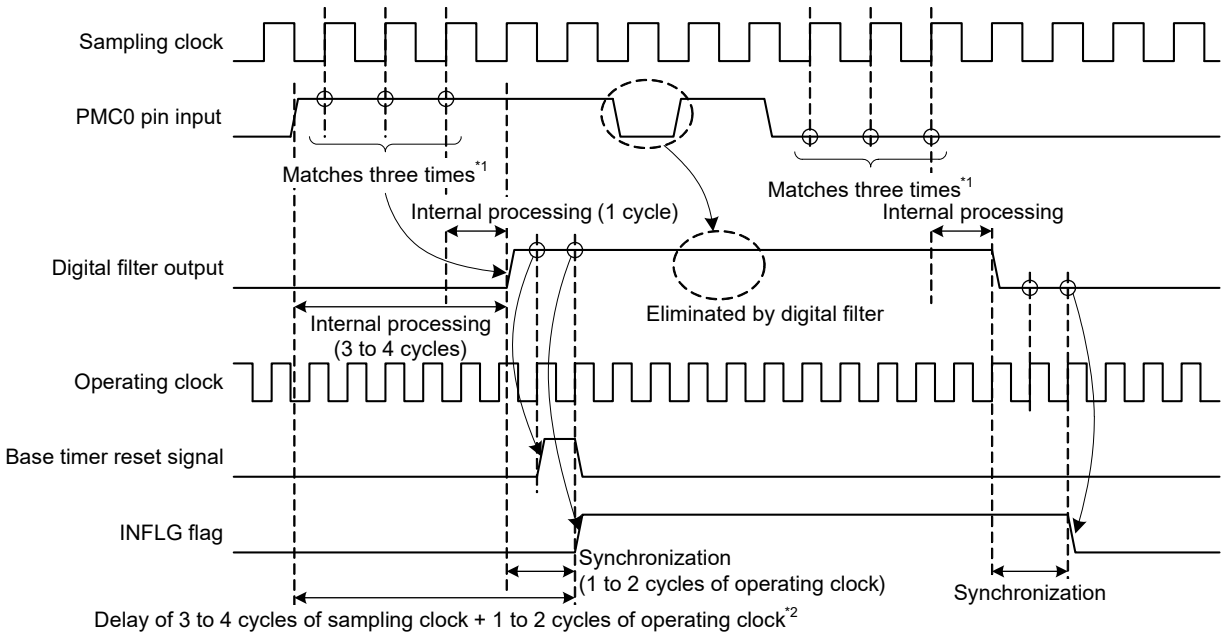


Figure 37.7 PMC0 Internal Input Signal Generation Configuration

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. The sampling clock of the digital filter is selectable from the REMC operating clock and IWDTCLK by setting the REMSTC.DNFSL bit.
When setting the REMSTC.LPCE bit to 1 (the PCLK supply in software standby mode is enabled), set the REMCON0.FIL bit to 1 (the digital filter is enabled), and the REMSTC.DNFSL bit to 1 (IWDTCLK is selected as a 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 37.8 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 sampling 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 37.8 Digital Filtering for PMC0 Input

37.3.6 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 37.9 shows the waveform of REMC operation.

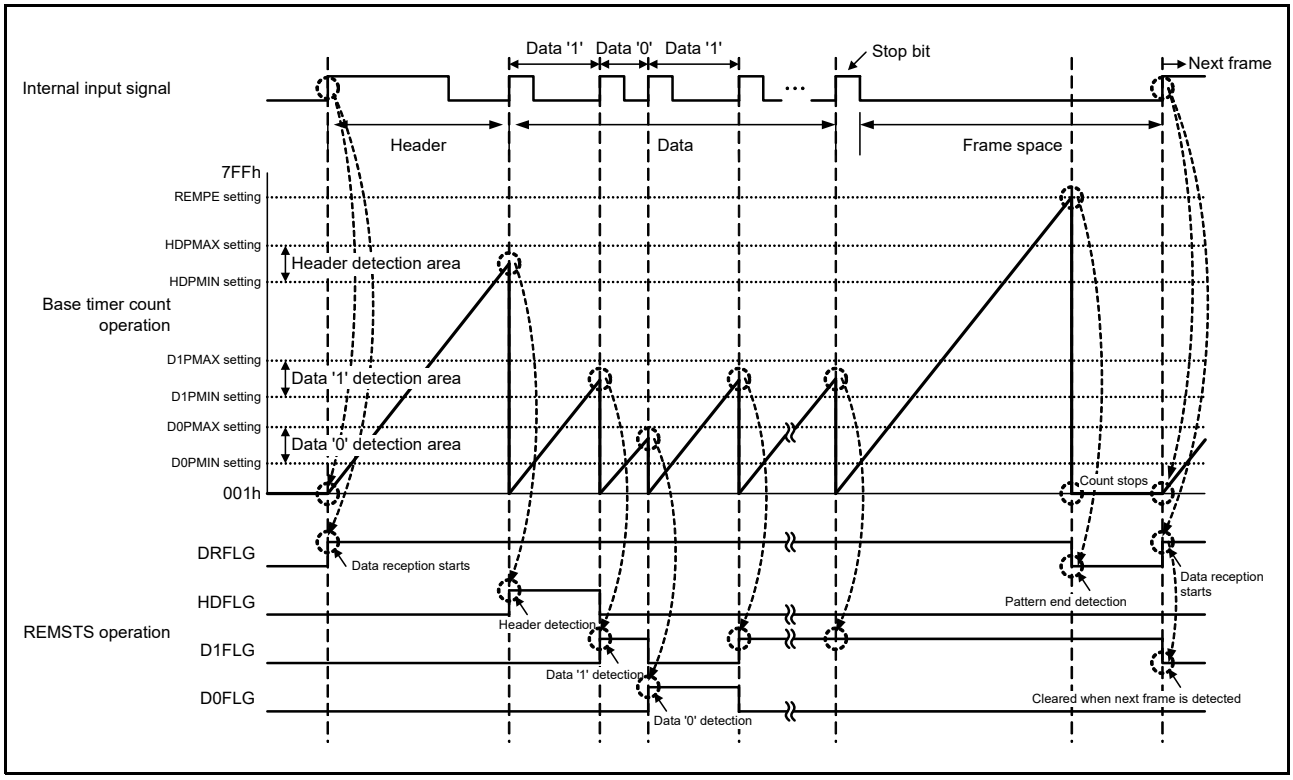


Figure 37.9 Waveform of REMC Operation

37.3.6.1 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.

37.3.6.2 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.

37.3.6.3 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.

37.3.6.4 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.

37.3.6.5 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 37.10.

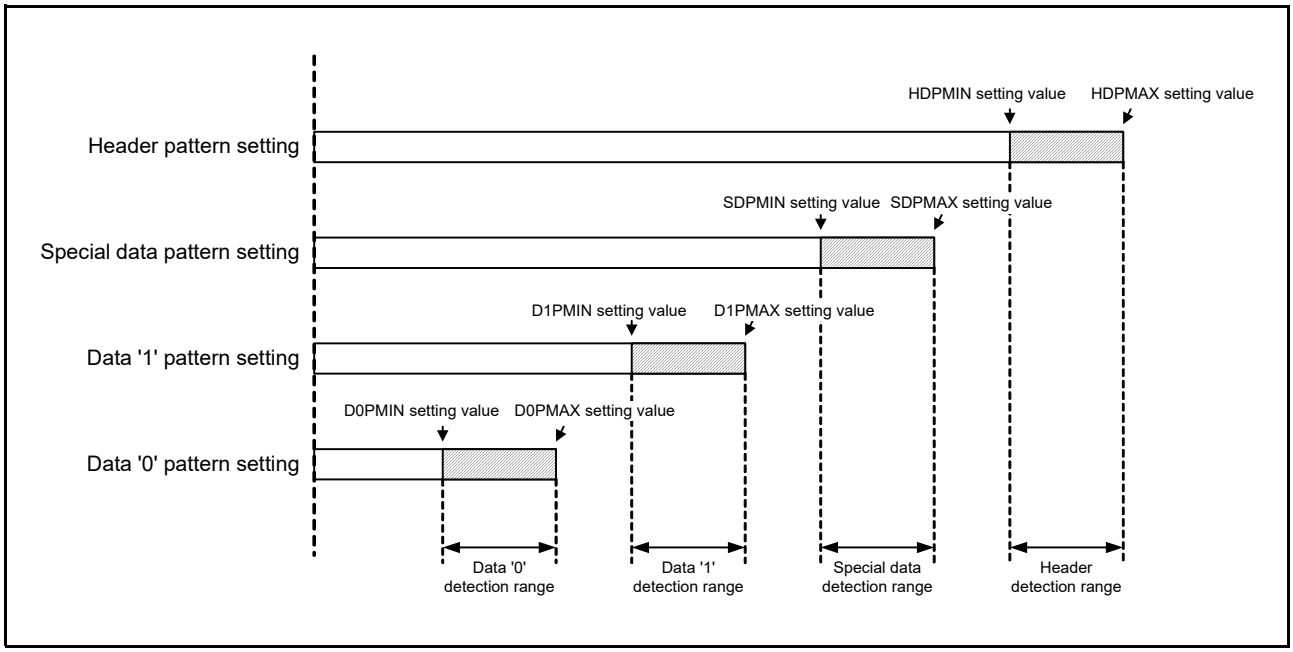


Figure 37.10 Examples of Setting Pattern Setting Registers

37.3.6.6 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 37.11 shows pattern detection and an example of flag operation.

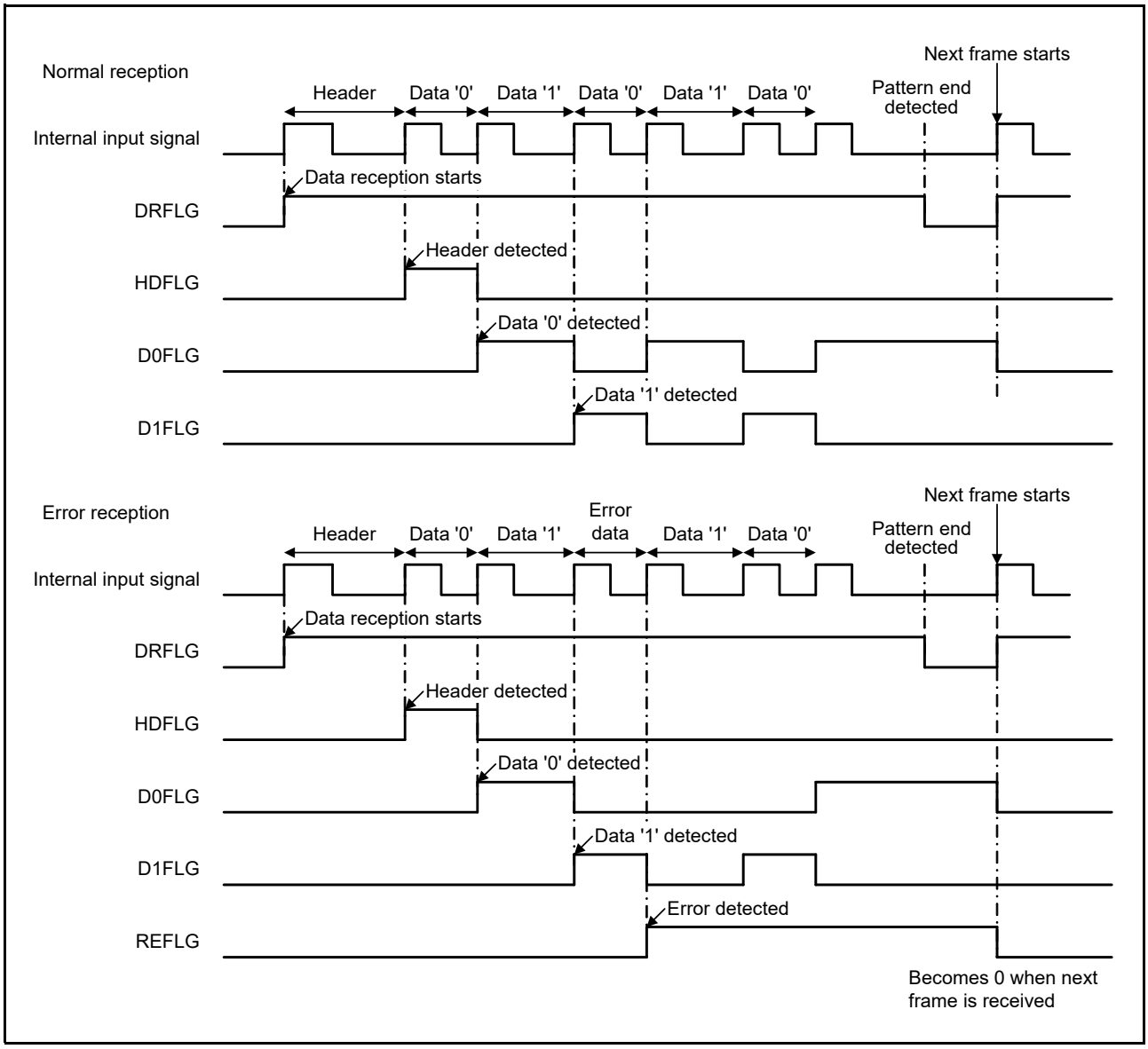


Figure 37.11 Example of Flag Operation

37.3.7 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 37.12 shows operation of the data reception complete flag for each pattern end setting.

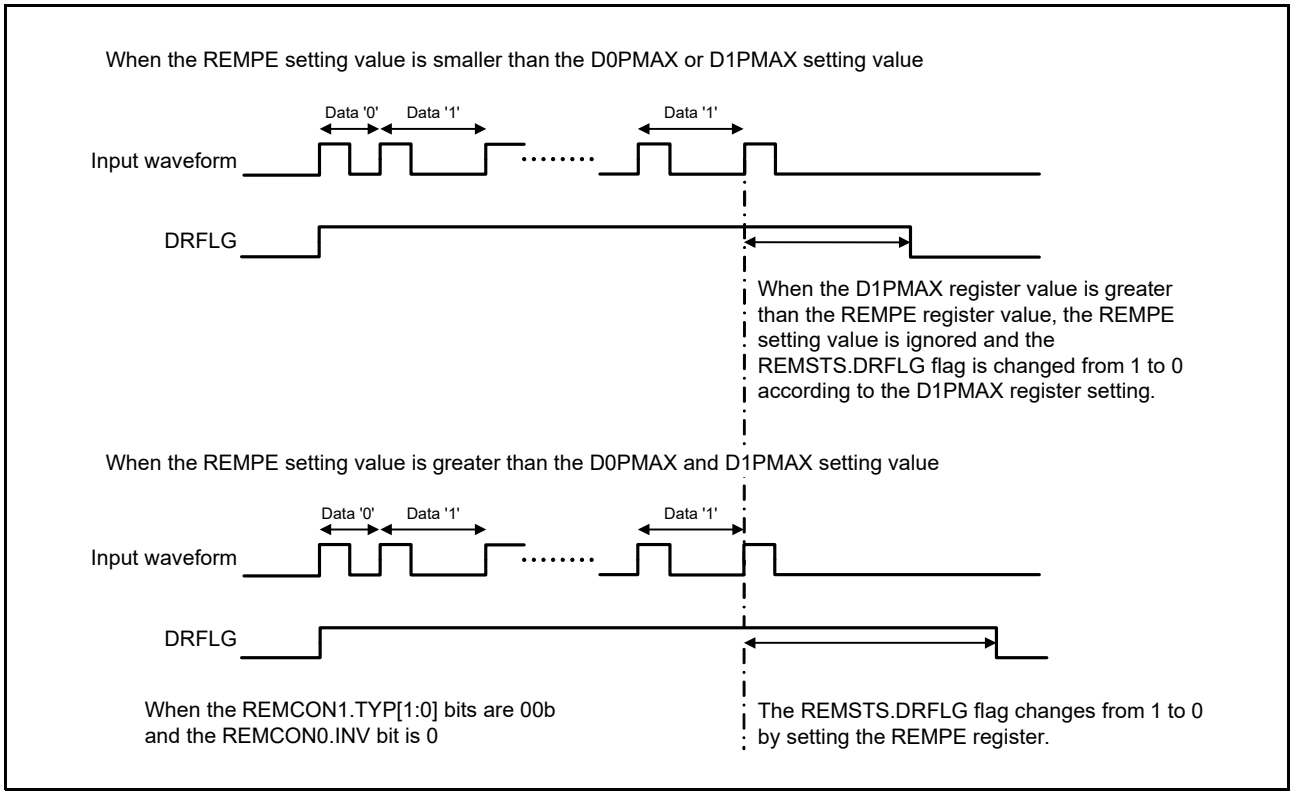


Figure 37.12 Operation of Data Reception Complete Flag for Each Pattern End Setting

37.3.8 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 37.13. 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 37.4 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.

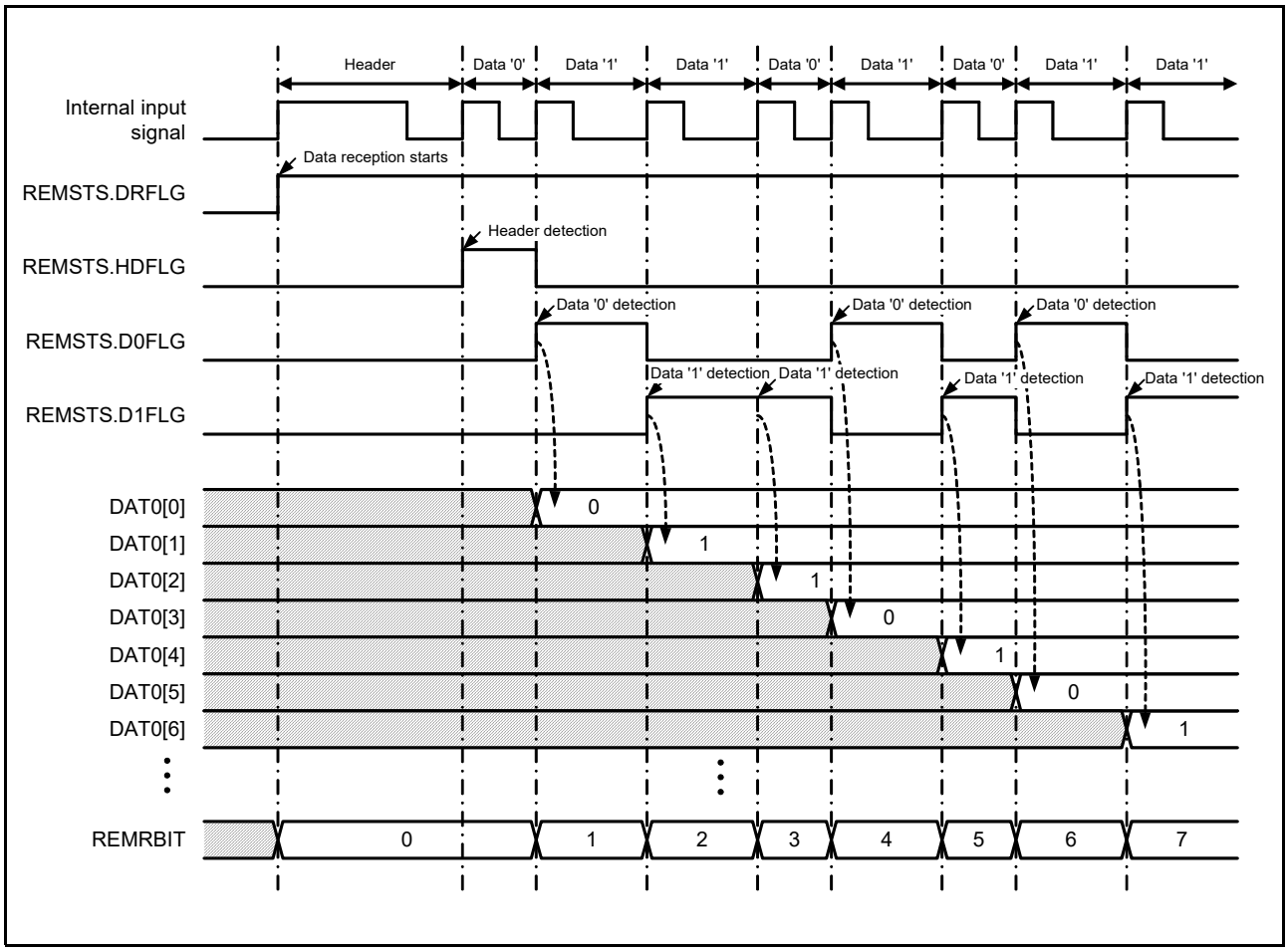


Figure 37.13 Operation of Receive Data Buffer

Table 37.4 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 37.14 shows the REMDATj/REMRBIT register operation when 00h is written to the REMDAT0 register.

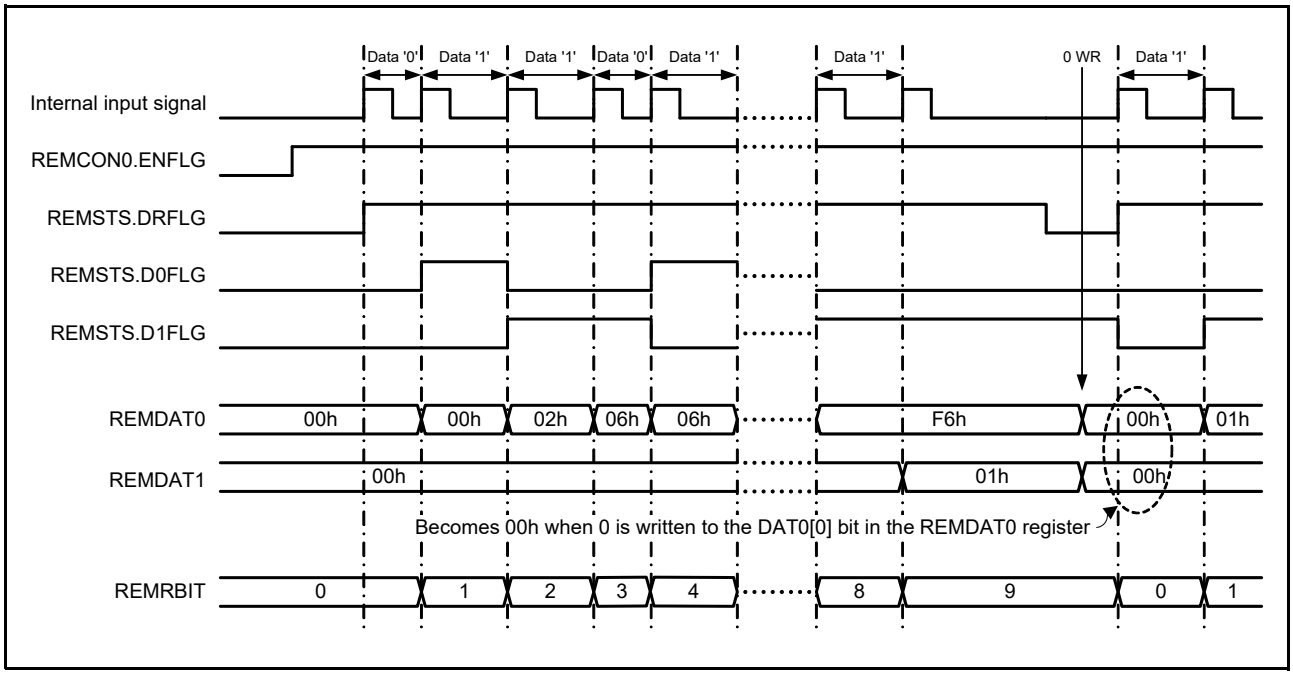


Figure 37.14 REMDATj/REMRBIT Register Operation (00h is Written to REMDAT0 Register)

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 37.15 shows operation of header pattern detection during data reception.

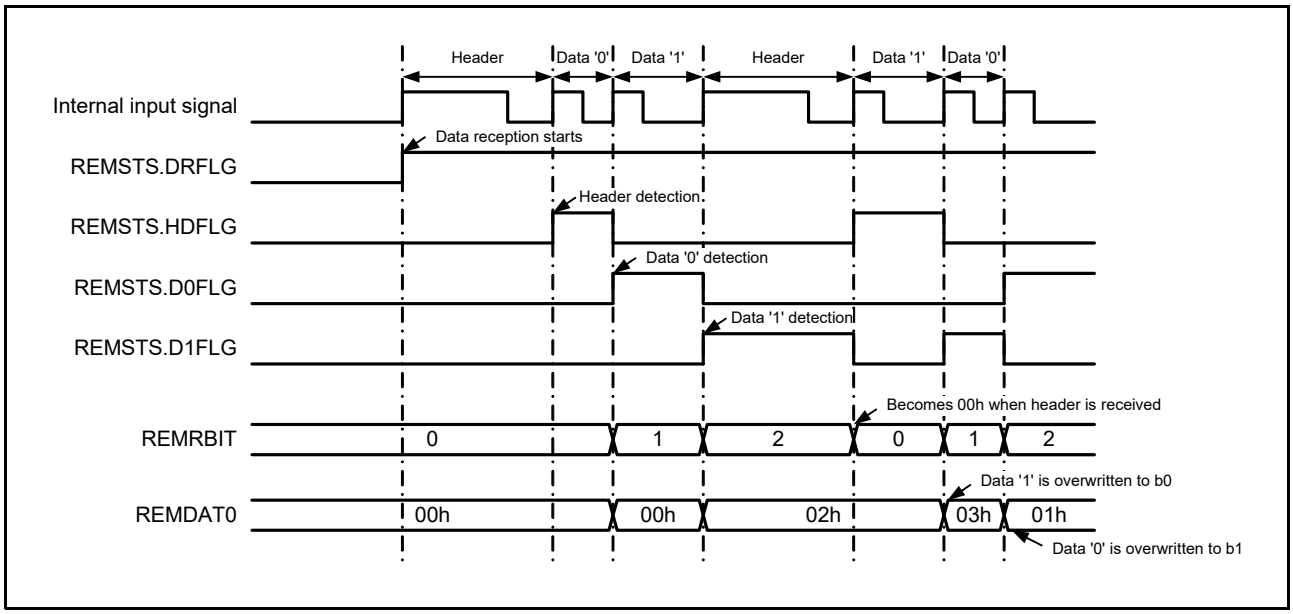


Figure 37.15 Operation of Header Pattern Detection during Data Reception

When the data exceeds 64 bits, the buffer is sequentially overwritten from the first bit. Figure 37.16 shows the REMRBIT register operation when the REMSTS.BFULFLG flag becomes 1.

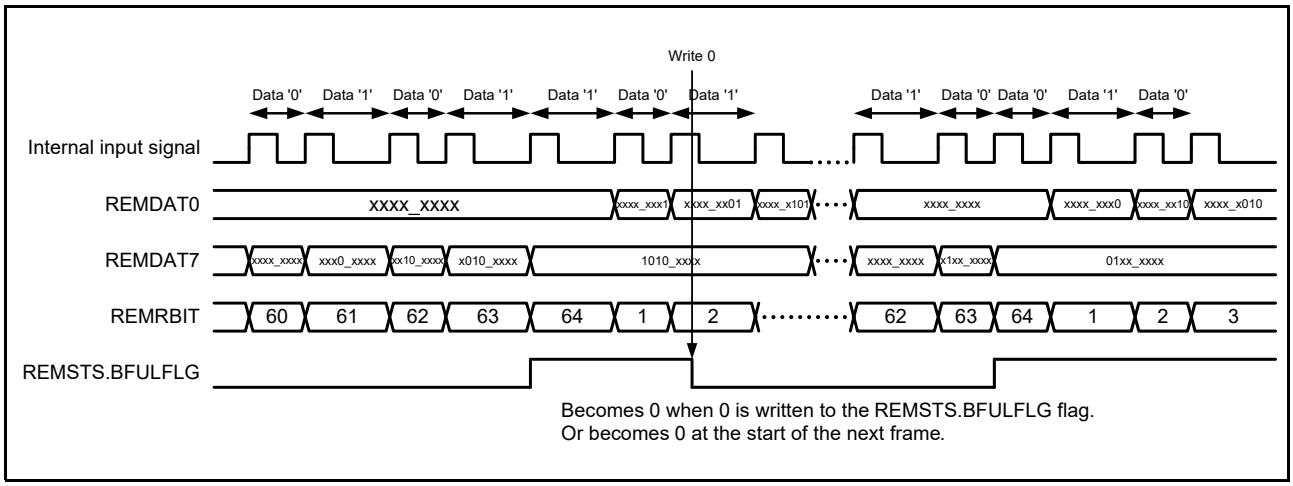


Figure 37.16 REMRBIT Register Operation (REMSTS.BFULFLG Flag = 1)

37.3.9 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 37.17 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).

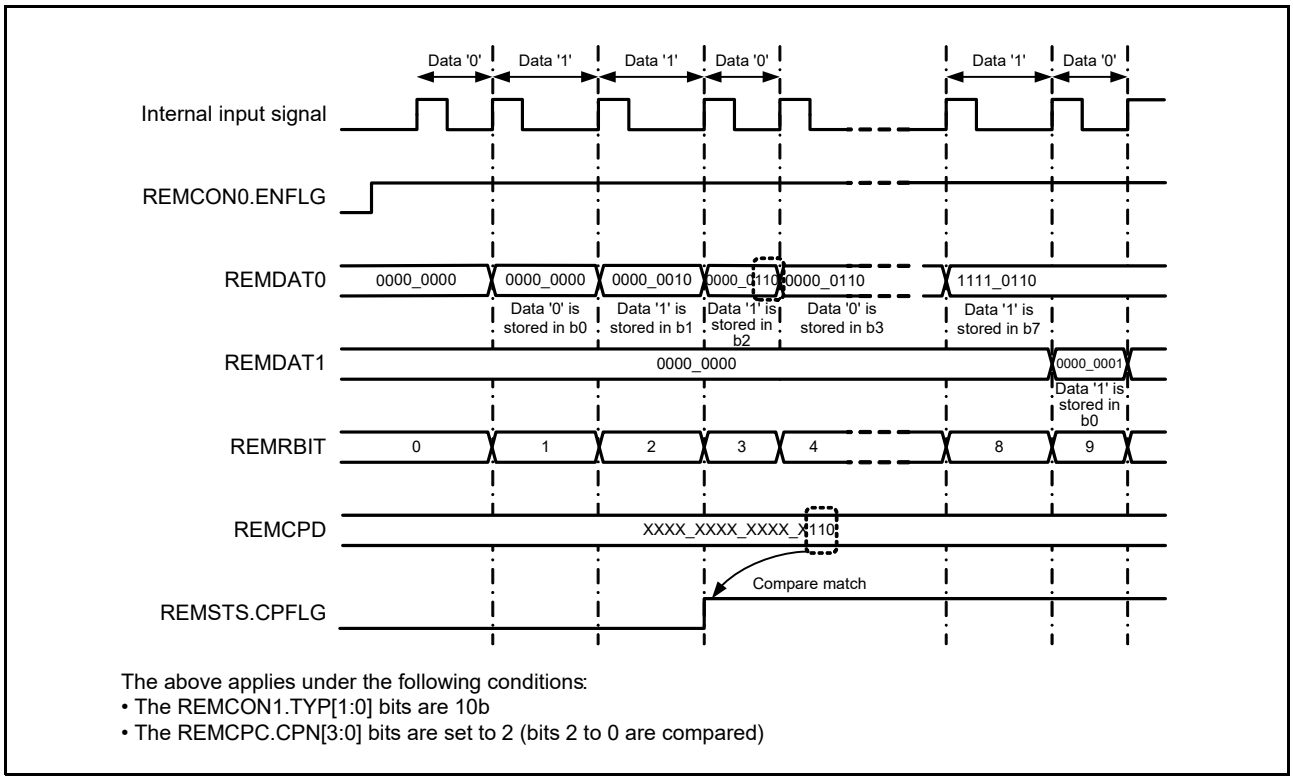


Figure 37.17 Receive Buffer and Compare Function

37.3.10 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 37.18 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 37.19 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.

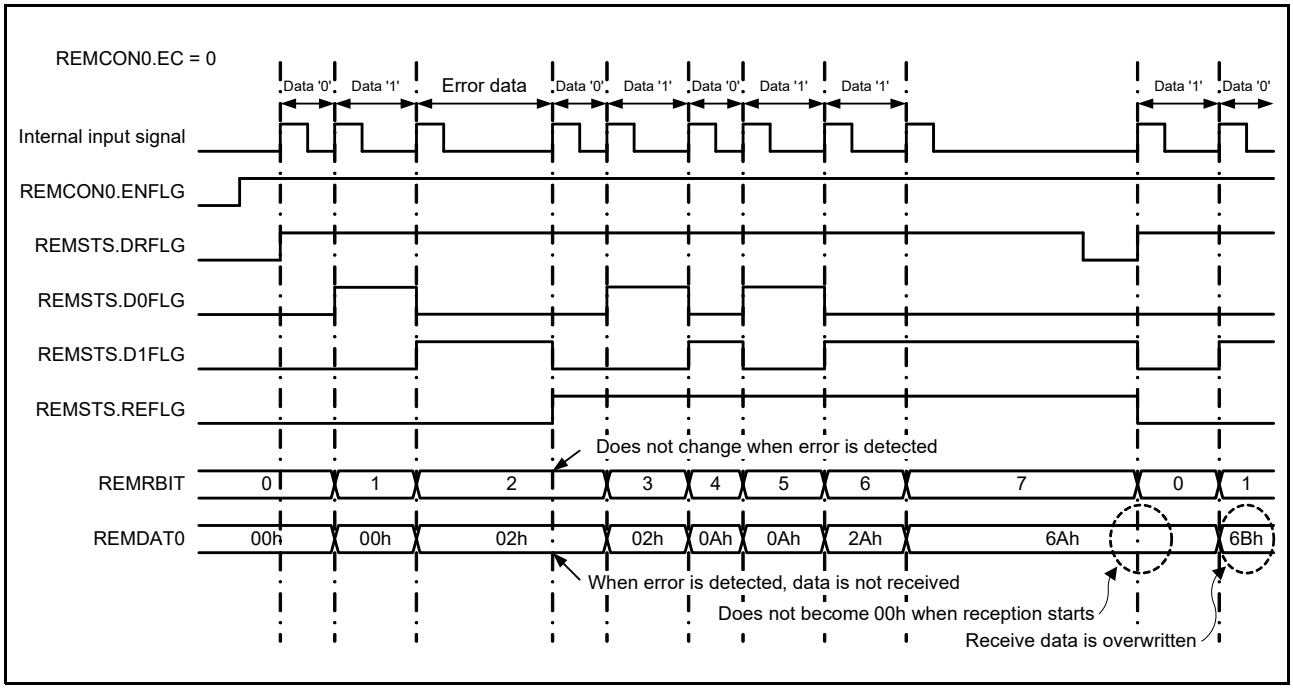


Figure 37.18 REMDAT0 and REMRBIT Registers Operation upon Error Detection (REMCON0.EC Bit = 0)

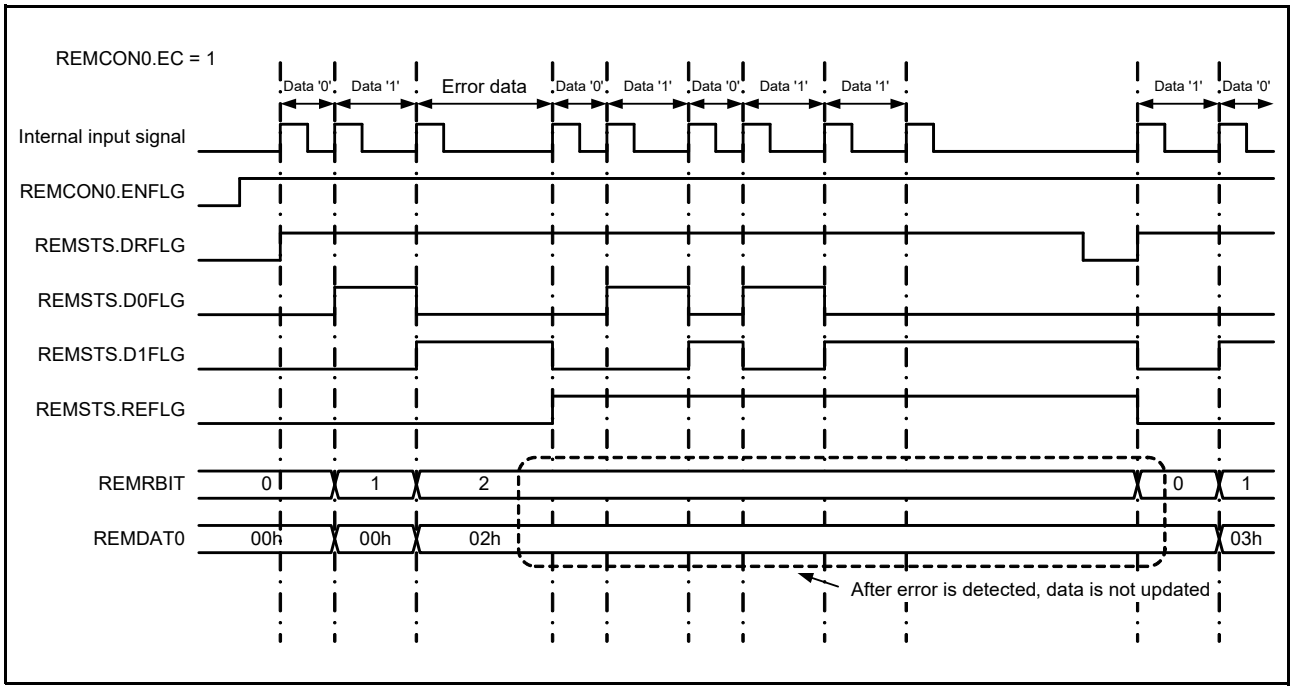


Figure 37.19 REMDAT0 and REMRBIT Registers Operation upon Error Detection (REMCON0.EC Bit = 1)

37.3.11 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 37.20 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)

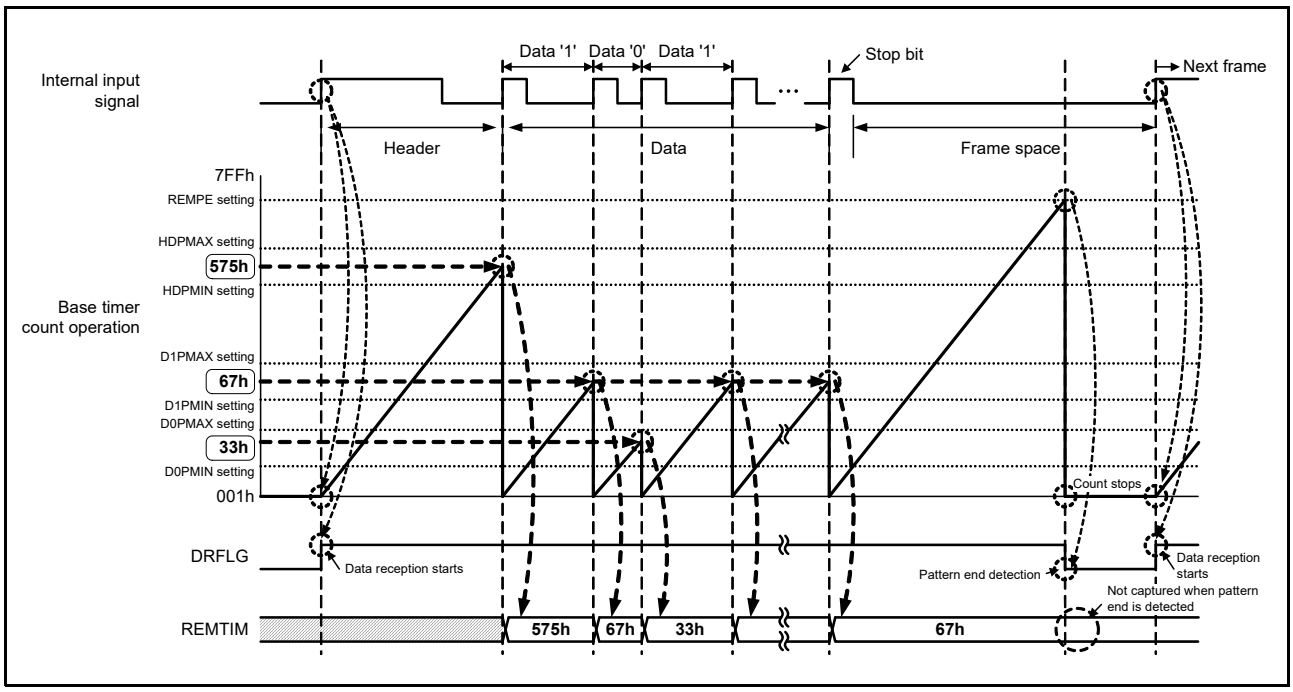


Figure 37.20 Operation Example of Measurement Function

37.3.12 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 37.5 lists the REMC interrupt sources, and Table 37.6 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 (ICUb) for details on interrupt control.

Table 37.5 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 37.6 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

37.3.13 Data Reception in Low Power Consumption State

In this MCU, data can be received in a low power consumption state (sleep mode, deep sleep mode, snooze mode, or software standby mode).

To receive data in a low power consumption state, REMC communications should be set before transitioning to the state.

37.3.13.1 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 37.21). Pattern detection and compare function enable returning from the low power consumption state only when specified data is received.

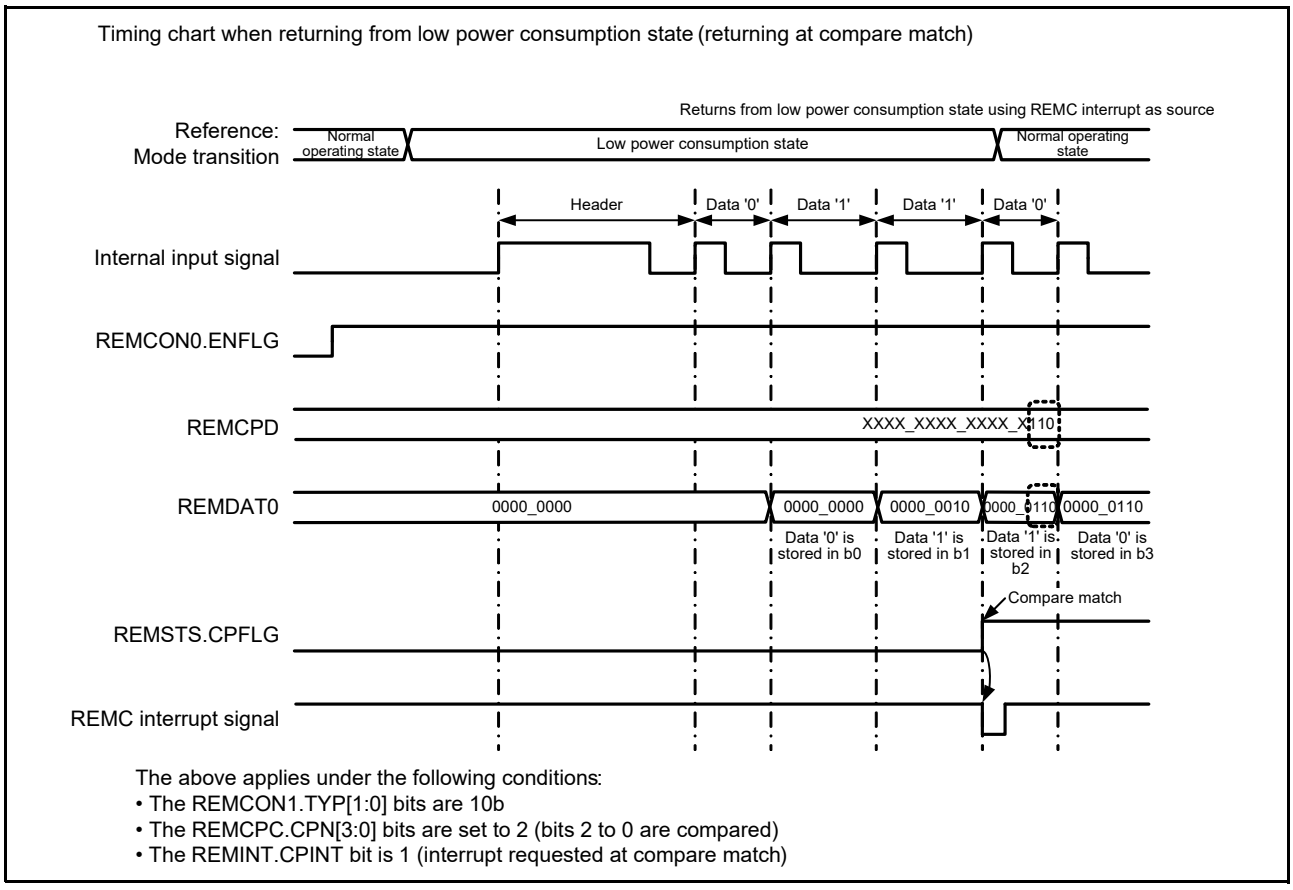


Figure 37.21 Using REMC Interrupt Request to Return from Low Power Consumption State (Normal Interrupt Mode)

37.3.13.2 Data Reception in Software Standby Mode

In software standby mode, data reception is possible with the combination of the settings shown in Table 37.7.

Table 37.7 Possible Combinations of the Settings for Data Reception in Software Standby Mode

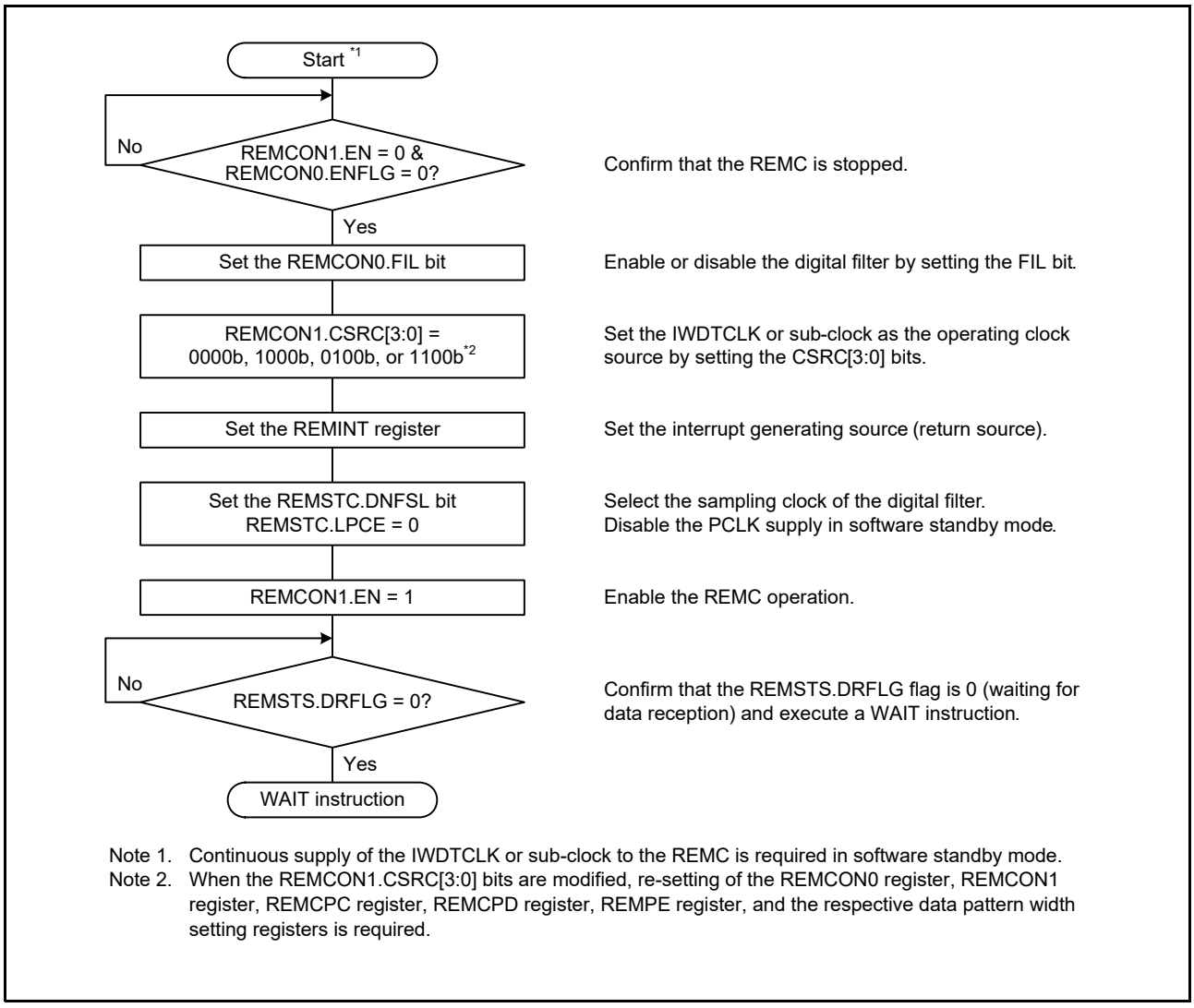
REMC Operating Clock	Setting of the REMSTC Register	Other Settings and Restrictions
IWDTCLK	LPCE = 0, DNFSL = 0 or 1	IWDTCSTPR.SLCSTP = 0
Sub-clock		SOSCCR.SOSTP = 0
PCLKB/64	LPCE = 1, DNFSL = 1	SCKCR3.CKSEL[2:0] = 001b (HOCO is selected as a system clock source) REMCON0.FIL = 1 IWDTCSTPR.SLCSTP = 0 Do not use the LPT function
PCLKB/512		

(1) When IWDTCLK or Sub-clock is Selected as the REMC Operating Clock

When IWDTCLK or sub-clock is selected as the REMC operating clock, set the REMSTC.LPCE bit to 0 (the PCLK supply in software standby mode is disabled). The selected clock need to be continuously supplied in the software standby mode. For the procedure to supply each clock, see section 37.3.4, Operating Clocks.

Select the REMC interrupt request that is generated when the data is received as the source for returning from the software standby mode. The MCU can return from the software standby mode only when a specific data has been received by using the pattern detection and compare function.

Figure 37.22 shows an example of a flowchart of the procedure for setting up data reception in the software standby mode.



Note 1. Continuous supply of the IWDTCLK or sub-clock to the REMC is required in software standby mode.
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 37.22 Flowchart for Setting Data Reception in Software Standby Mode (REMSTC.LPCE = 0)

(2) When PCLKB/64 or PCLKB/512 is Selected as the REMC Operating Clock

When PCLKB/64 or PCLKB/512 is selected as the REMC operating clock, set the SNZCR.REMCSNZSEL[1:0] bits to 10b and the REMSTC.LPCE bit to 1 (the PCLK supply in software standby mode is enabled). After transition to the software standby mode, when a change of the input level of the PMC0 pin is detected, the PCLK request signal is output from the REMC. When the MCU transitions to the snooze mode, the oscillator resume its operation and restarts the PCLK supply. Select the HOCO as the system clock source. It takes up to t_{SNZHO} from detection of the change in the input level of the PMC0 pin to the PCLK supply. During that time, the REMC operating clock is not supplied and the base timer is stopped. Set the value minus t_{SNZHO} from the value to be set for data reception in normal mode for the minimum value of the pattern setting of the first data. For the value of t_{SNZHO}, refer to section 47, Electrical Characteristics.

When a header pattern match interrupt or compare match interrupt is generated during data reception, the MCU returns to the normal operating mode. When an interrupt is not generated, the PCLK request signal is negated due to detection of the pattern end, the MCU returns to the software standby mode, and operation of the oscillator is stopped (the data reception complete interrupt should be disabled). Afterwards, when a change of the input level of the PMC0 pin is detected again, the PCLK request signal is output and data reception becomes possible. Furthermore, a header pattern match interrupt or compare match interrupt is generated, the PCLK request signal is not negated even if a pattern end is detected. The PCLK request signal is negated by setting the REMSTC.LPCE bit to 0 during interrupt handling of the

recovery source.

Figure 37.23 shows operation of continuing the software standby mode in response to a compare mismatch, and Figure 37.24 shows operation of returning from the software standby mode in response to a compare match. Figure 37.25 shows an example of the flowchart for setting data reception in the software standby mode.

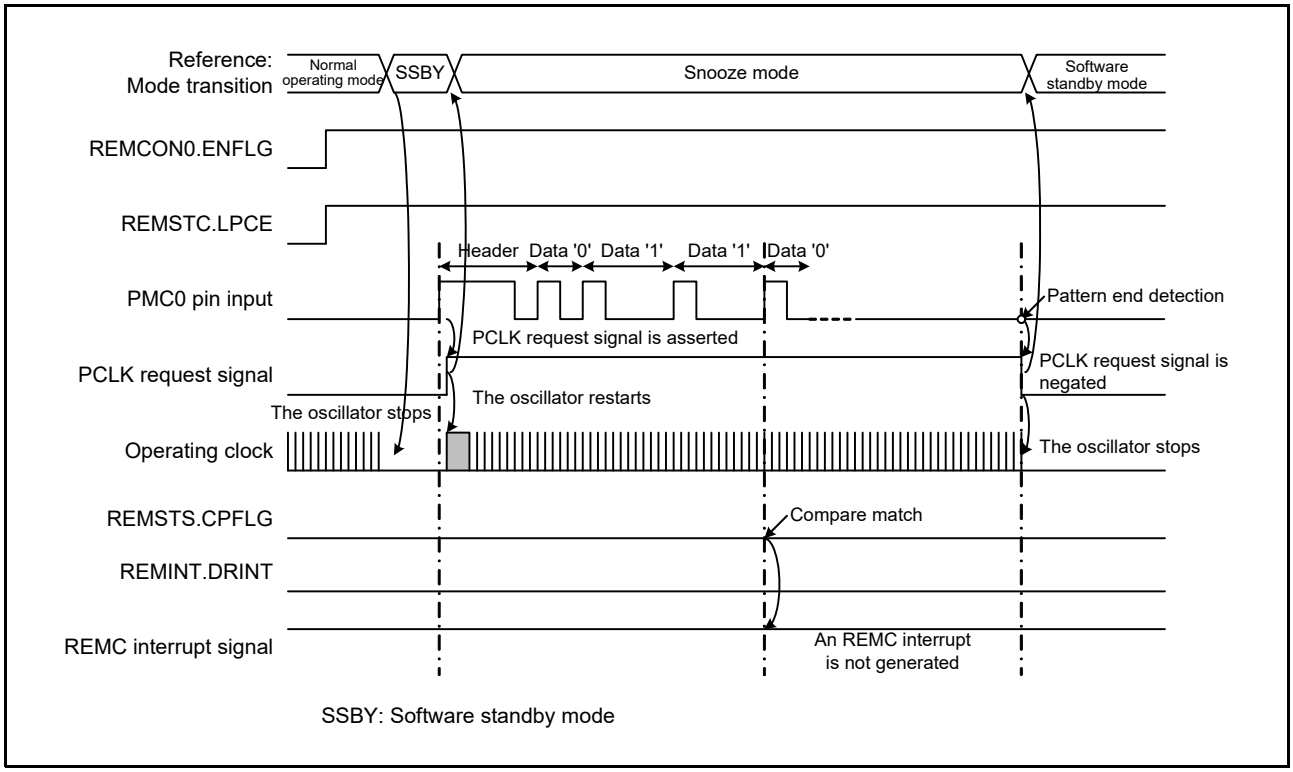


Figure 37.23 Continuation of Software Standby Mode in Response to a Compare Mismatch

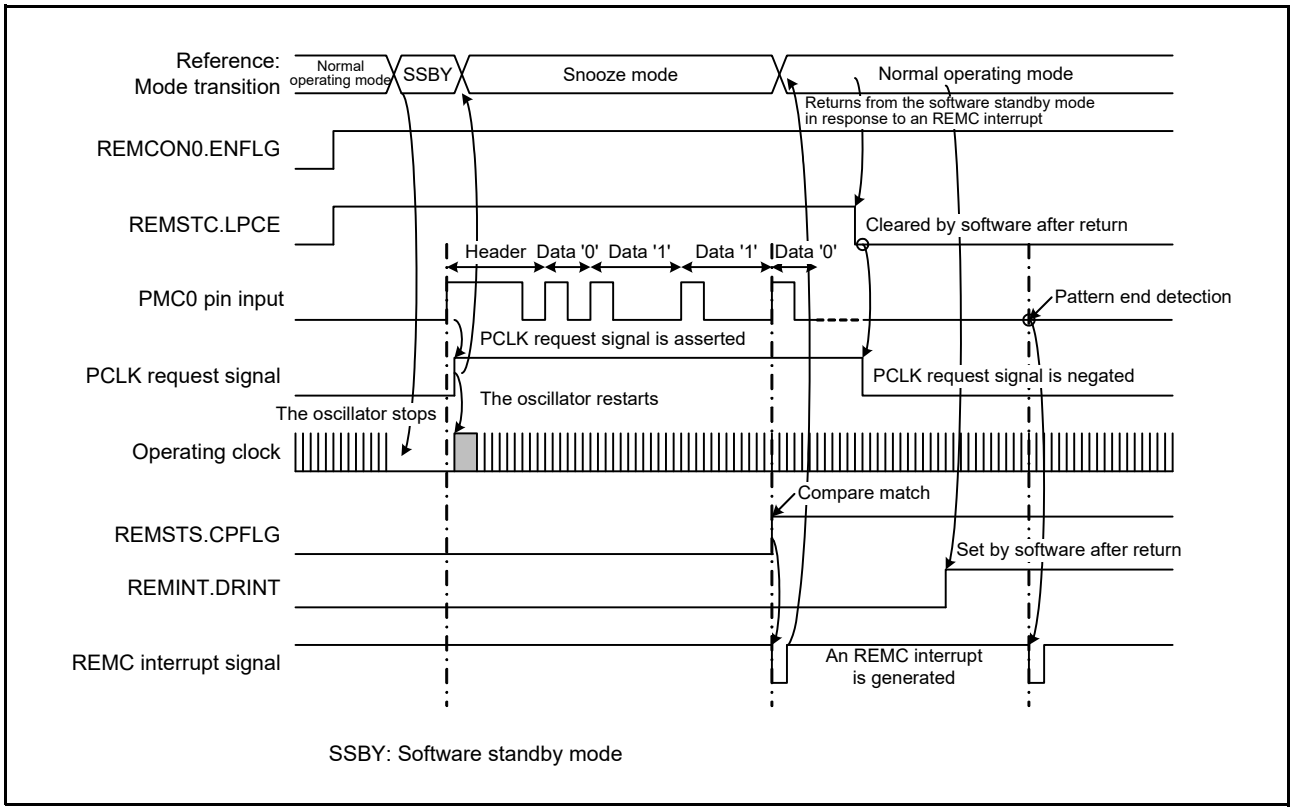
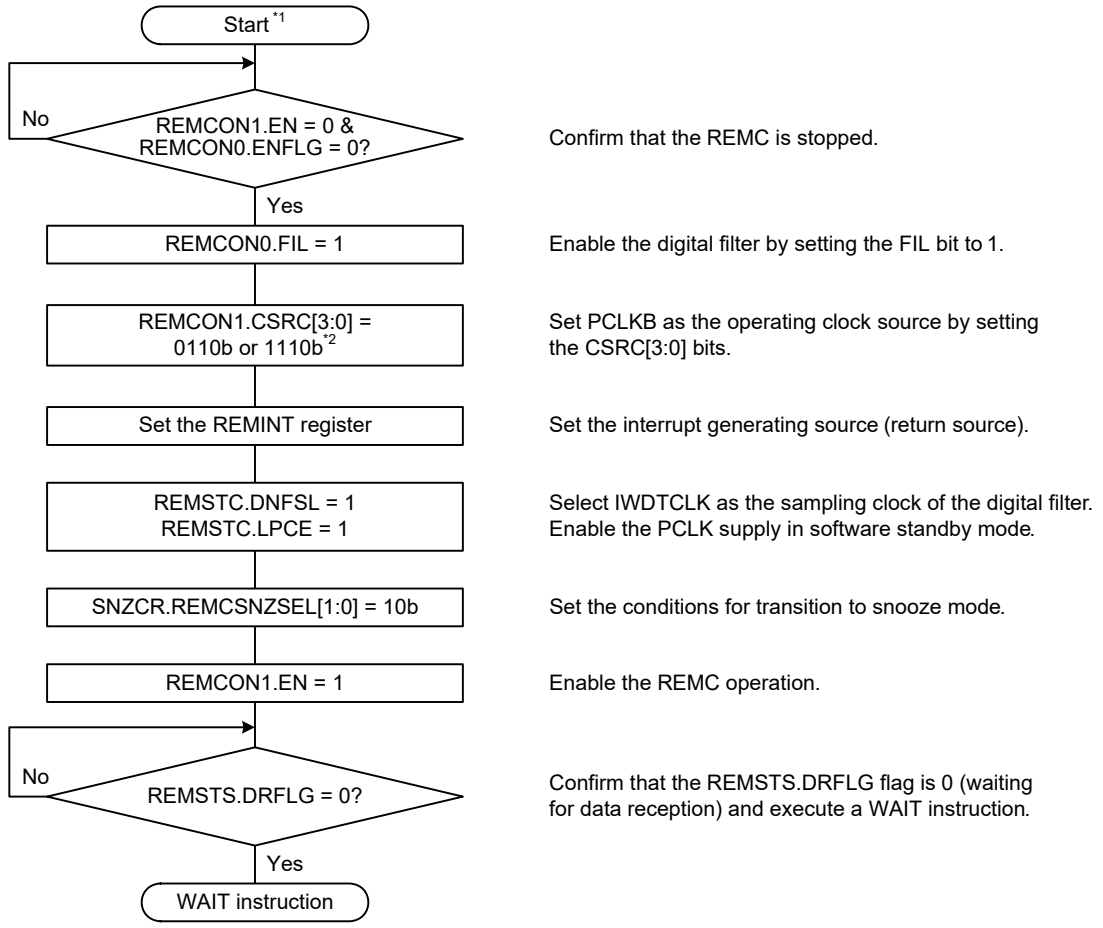


Figure 37.24 Return from Software Standby Mode in Response to a Compare Match



Note 1. The HOCO should be selected as the system clock source and IWDTCLK should be supplied for the sampling clock of the digital filter.

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 37.25 Flowchart for Setting Data Reception in Software Standby Mode (REMSTC.LPCE = 1)

37.4 Usage Notes

37.4.1 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.

37.4.2 Settings for Peripheral Module Clock and REMC Operating Clock

Set the peripheral module clock (PCLKB) frequency to higher than the REMC operating clock frequency.

37.4.3 Restriction on Using Independent Watchdog Timer (IWDT)

Do not use the IWDT function when using IWDTCLK as the REMC operating clock or the sampling clock of the digital filter.

37.4.4 Restriction on Using Low-Power Timer (LPT)

Do not use the LPT function when setting the REMSTC.LPCE bit to 1 to receive data in the software standby mode.

37.4.5 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 37.2.1 to section 37.2.20) except for the REMCON0.ENFLG flag.

37.4.6 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
- REMSTC 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.

37.4.7 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.

37.4.8 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.

37.4.9 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.

38. Renesas Secure IP (RSIP-E11A)

This MCU incorporates a Renesas Secure IP (RSIP-E11A) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the RSIP library, the RSIP can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).

Key information to be used in encrypting and decrypting data is only stored within the RSIP, and any external access can be shut out to obtain a system with strong security.

38.1 Overview

Table 38.1 summarizes the specifications of the RSIP. Figure 38.1 shows a block diagram of the RSIP.

Table 38.1 Specifications of RSIP

Item	Description
Access control	Access management circuit • In case of irregular access to the RSIP due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the RSIP.
Symmetric-key cryptography	AES: Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128 or 256 bits • Block sizes: 128 bits • Block cipher mode of operation ECB, CBC, CTR: Compliant with NIST SP 800-38A CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D • Message Authentication Code Algorithm CMAC: Compliant with NIST SP 800-38B GMAC: Compliant with NIST SP 800-38D • Number of cycles for execution*1 ECB, CBC, CTR, CCM, GCM, CMAC, GMAC: 44 cycles of PCLKB for 128-bit keys 60 cycles of PCLKB for 256-bit keys
Public-key cryptography	ECC • Key sizes: Up to 256 bits • Block sizes: Up to 256 bits • Signing, Signature verification, Key generation
Message digest function	HASH • Secure hash algorithm SHA-224, SHA-256: Compliant with FIPS PUB 180-4 • Key sizes: Up to 512-bit • Block sizes: 512 bits (SHA-224, SHA-256)
Generation of random numbers	128-bit true random number generator
Hardware unique key (HUK)	• 256-bit read-only key unique to each individual MCU • HUK is wrapped with RSIP unique ID and stored in an area isolated from CPU. • HUK can only be read from the access management circuit through the dedicated bus. HUK is not accessible from CPU.
RSIP unique ID	• 64-bit read-only ID code to identify individual MCUs. • RSIP unique ID is accessible from the access management circuit through the dedicated bus. • RSIP unique ID is used to unwrap the HUK.
Key management	• User key can be securely stored in flash memory or RAM by wrapping with HUK. Raw key is not stored. • Keys wrapped with HUK are valid only within the corresponding RSIP. Even if the wrapped key is leaked, it cannot be unwrapped by RSIP on other MCUs.
Supervisor mode	• The supervisor mode signal is connected to the access management circuit and is used to limit control of the RSIP module to supervisor mode only.
Interrupt sources	10
Low power consumption	Setting of the module stop state is possible.

Note 1. This does not include the overhead for calling functions of the RSIP library.

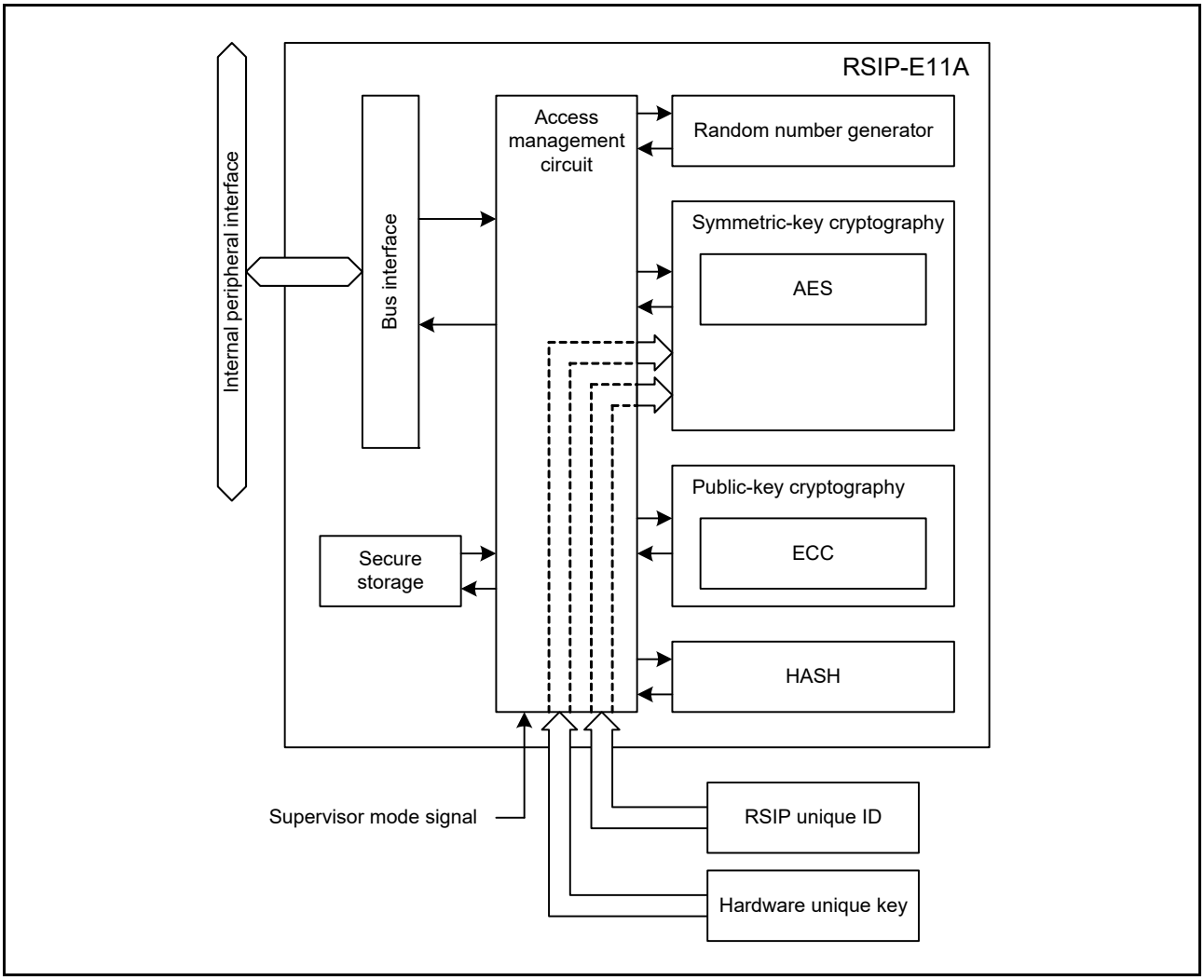


Figure 38.1 RSIP Block Diagram

38.2 Operation

38.2.1 Encryption Engine

Figure 38.2 shows processes of the encryption engine integrated in the RSIP.
The encryption engine, using the wrapped user key, performs plaintext to ciphertext encryption and ciphertext to plaintext decryption by hardware.
In no part of the encryption or decryption process, is user key or intermediate data ever exposed outside of the RSIP.

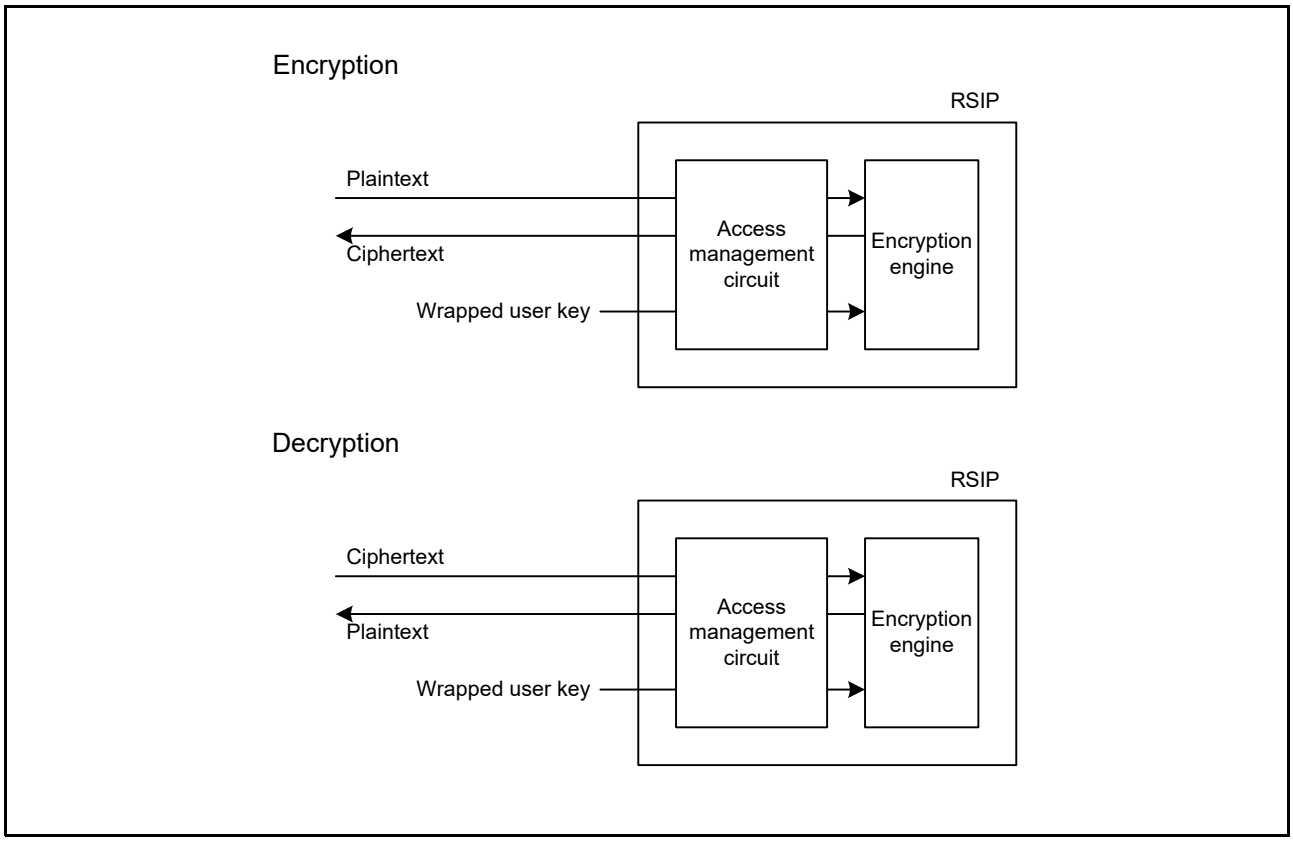


Figure 38.2 Encryption and Decryption processes by Encryption Engine

38.2.2 Encryption and Decryption

The procedures for encrypting and decrypting data are given below.

(1) Input the wrapped user key into the RSIP, and recover the user key.
(2) Input the data to encrypt or decrypt into the RSIP. This converts plaintext into ciphertext, and ciphertext into plaintext.
(3) Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output.
Figure 38.3 shows the timing diagram.

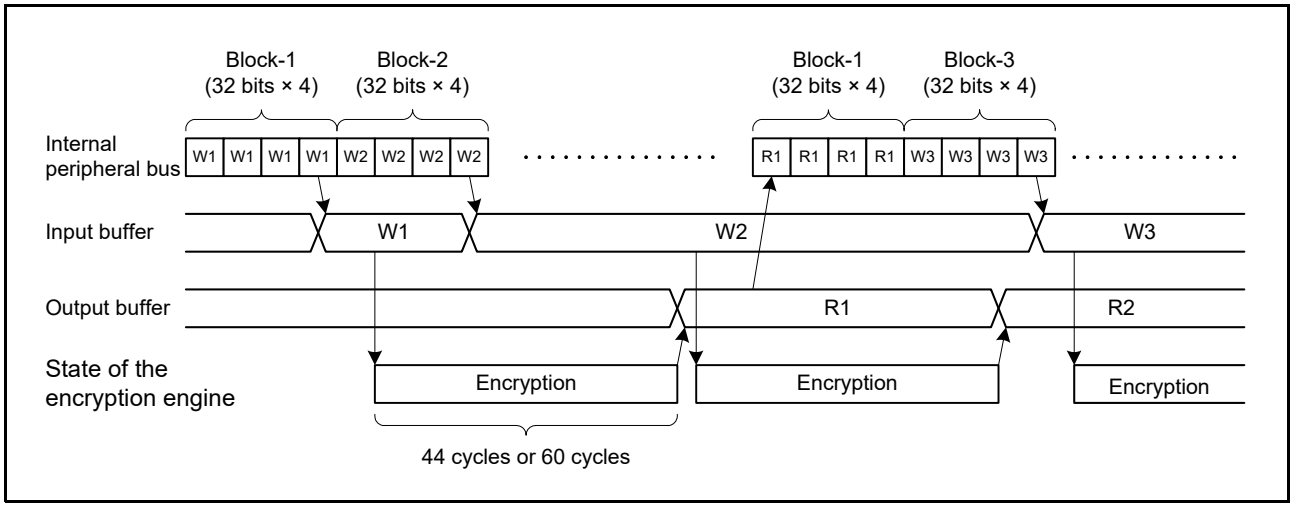


Figure 38.3 Encryption and Decryption Timing Diagram

38.3 Interrupt

Table 38.2 lists the interrupt sources.

Table 38.2 RSIP Interrupt Sources

Name	Interrupt Source
PROC_BUSY	Procedure completed interrupt
ROMOK	ROM falsification detected interrupt
LONG_PLG	Operation completed interrupt
TEST_BUSY	Test busy
WRRDY0	Write ready 0
WRRDY2	Write ready 2
RDRDY0	Read ready 0
INTEGRATE_WRRDY	Integrate write ready
INTEGRATE_RDRDY	Integrate read ready
ECCERR	ECC error detection interrupt

38.4 Usage Notes

38.4.1 Standby Mode

When standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after standby mode is exited. Standby mode should therefore be entered only when the encryption engine is stopped.

38.4.2 Setting the Module Stop Function

The module stop control register D (MSTPCRD) enables or disables operation of the RSIP. After a reset, the RSIP is stopped. After exiting the module stop state, the RSIP can be accessed. Refer to section 11, Low Power Consumption for details.

38.4.3 RSIP Library

Use of the RSIP requires the RSIP library provided by Renesas Electronics.

39. Capacitive Touch Sensing Unit (CTSU2SLa)

The capacitive touch sensing unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with a dielectric so that a finger does not come into contact with the electrode.

As shown in Figure 39.1, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.

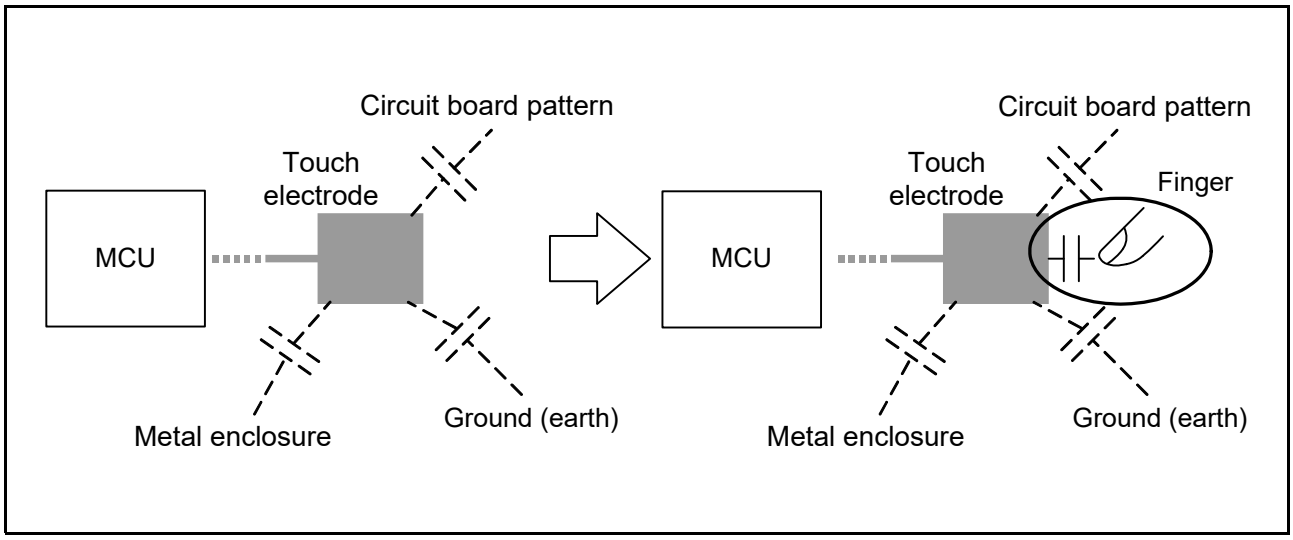


Figure 39.1 Increased Electrostatic Capacitance Due to Presence of Finger

Electrostatic capacitance is detected by the following methods: Self-capacitance and mutual capacitance.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

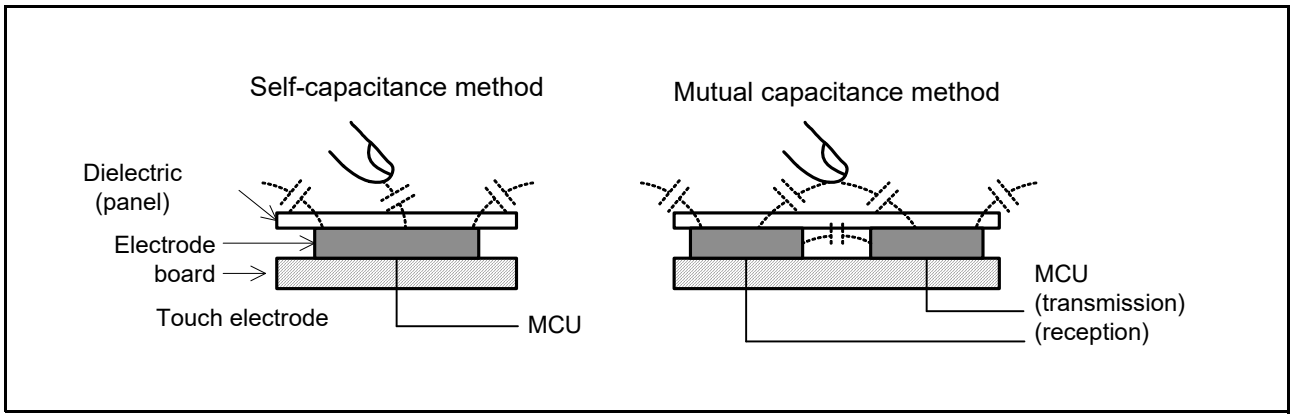


Figure 39.2 Self-Capacitance Method and Mutual Capacitance Method

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged/discharged current, for a specified period.

For details on the measurement principles of the CTSU, refer to section 39.3.1, Principles of Measurement Operation.

39.1 Overview

Table 39.1 lists the specifications of the CTSU, and Figure 39.3 shows a block diagram of the CTSU.

Table 39.1 CTSU Specifications

Item		Description
Operating clock		Selectable from PCLKB (at least 1 MHz), PCLKB/2, PCLKB/4, or PCLKB/8
I/O Pins	Electrostatic capacitance measurement pins	TS0 to TS35 pins (36 channels)
	External capacitor connecting pin for measurement power supply	TSCAP pin (0.01 µF)
Measurement methods	Self-capacitance method	The electrostatic capacitance is measured from the current that charges and discharges the capacitance of the electrode.
	Mutual capacitance method	The electrostatic capacitance is measured from the current that charges and discharges the capacitance between the transmitting electrode and the receiving electrode.
	Current measurement mode	Direct measurement of current flowing through a pin.
Scan mode	Single scan mode	Electrostatic capacitance on a channel is measured.
	Multi-scan mode	Electrostatic capacitance on multiple channels is measured successively.
Noise prevention		• Spread spectrum • Random phase shift • Noise hopping by multiple sensor drive pulses with different frequency
Calibration per pin		• Offset current • Frequency of sensor drive pulse • Measurement time
Measurement start conditions		• Software trigger • External trigger (event input from the event link controller (ELC))
Automatic operations		• Automatic correction • Automatic judgment
Low power operation		Measurement under snooze mode is available. • Measurement starts by an external trigger input via ELC. • Snooze mode can be ended by non-touch judgment using the automatic judgment function. • Snooze mode can be released by the measurement end interrupt.
Interrupt sources		• Register setting request (CTSUWR) • Measurement result read request (CTSURD) • Measurement end (CTSUFN)
Event link		Measurement start trigger input
Low power consumption function		Module stop state can be set.

As shown in Figure 39.3, the CTSU consists of the state control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and control registers.

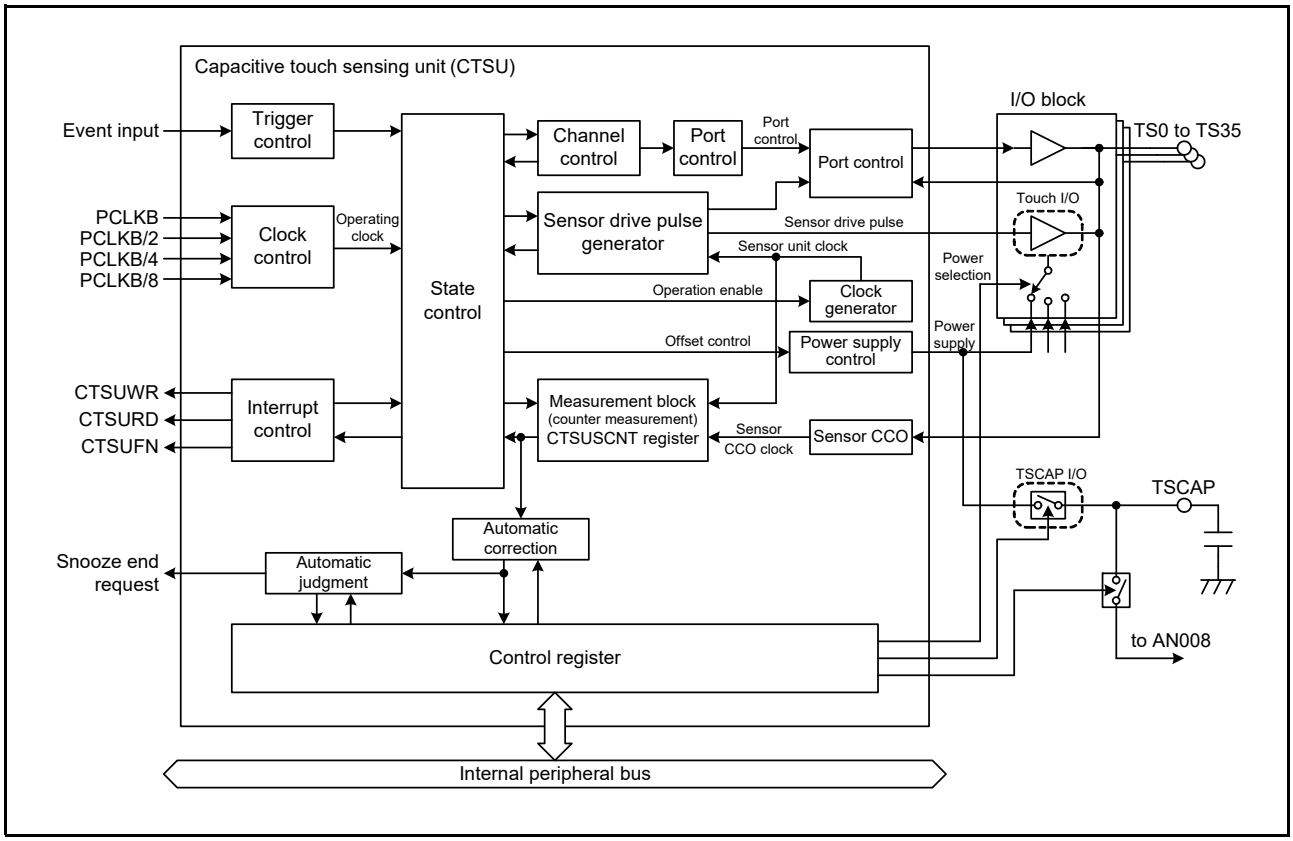


Figure 39.3 CTSU Block Diagram

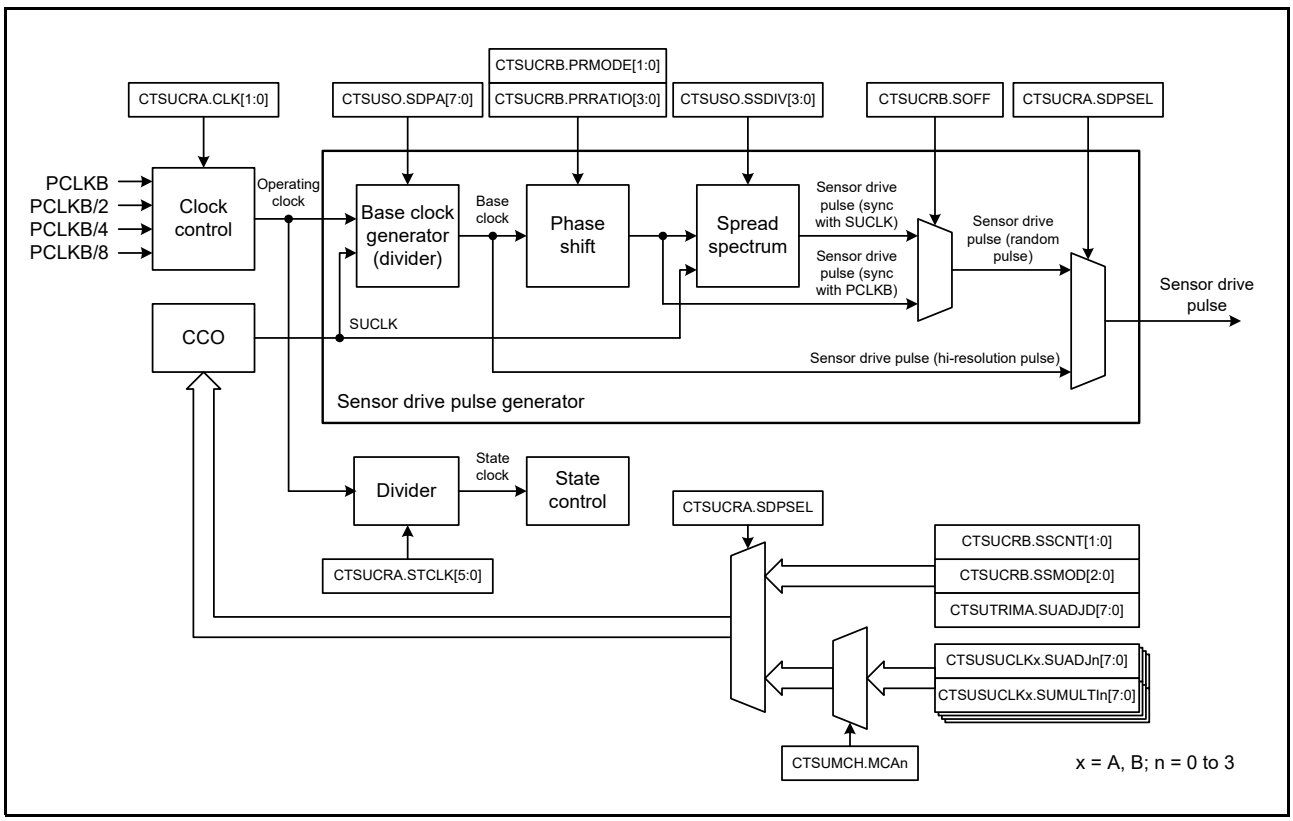


Figure 39.4 Generation of Sensor Drive Pulse

Table 39.2 CTSU Pin Configuration

Pin Name	I/O	Function
TS0 to TS35	I/O	Electrostatic capacitive measurement pins (touch pins)
TSCAP	—	Secondary power supply for measurement (capacitor) connection pin

39.2 Register Descriptions

39.2.1 CTSU Control Register A (CTSUCRA)

Address(es): CTSU.CTSUCRA 000A 0900h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DCBACK	DCMODE	STCLK[5:0]						PCSEL	SDPSEL	POSEL[1:0]		LOAD[1:0]		ATUNE2	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD1	MD0	CLK[1:0]		ATUNE1	ATUNE0	CSW	PON	TXVSEL[1:0]		PUMPON	INIT	—	SNZ	CAP	STRT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STRT	Measurement Operation Start	0: Measurement operation stops 1: Measurement operation starts	R/W
b1	CAP	Measurement Operation Start Trigger Select	0: Software trigger 1: External trigger	R/W
b2	SNZ	Snooze Function Enable	0: Measurement during snooze mode is disabled. 1: Measurement during snooze mode is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	INIT	Control Block Initialization	Writing 1 to this bit initializes the CTSU control block and some registers*1. This bit is read as 0.	R/W
b5	PUMPON	Voltage Booster Enable*2	0: Voltage booster is disabled. 1: Voltage booster is enabled.	R/W
b7, b6	TXVSEL[1:0]	Transmission Power Supply Select	b7 b6 0 0: Power supply for I/O port is selected. x 1: Low-noise VCC is selected.*3 1 0: Power supply for internal logic is selected.	R/W
b8	PON	Measurement Power Supply Enable	0: Powered off 1: Powered on	R/W
b9	CSW	Capacitor for Measurement Power Supply Connection	0: TSCAP is disconnected. 1: TSCAP is connected.	R/W
b10	ATUNE0	Power Supply Voltage Setting*4	0: Measurement power supply voltage = 1.5 V 1: Measurement power supply voltage = 1.2 V	R/W
b11	ATUNE1	Current Range Switch	This bit selects the measurement current range. Refer to Table 39.5 for details.	R/W
b13, b12	CLK[1:0]	Operating Clock Select	b13 b12 0 0: PCLKB 0 1: PCLKB/2 (PCLKB divided by 2) 1 0: PCLKB/4 (PCLKB divided by 4) 1 1: PCLKB/8 (PCLKB divided by 8)	R/W
b14	MD0	Measurement Mode Select 0	0: Single scan mode 1: Multi-scan mode	R/W
b15	MD1	Measurement Mode Select 1	0: Self-capacitance method 1: Mutual capacitance method	R/W
b16	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b17	ATUNE2	Current Range Switch 2	This bit selects the measurement current range. Refer to Table 39.5 for details.	R/W
b19, b18	LOAD[1:0]	Measurement Load Control	b19 b18 0 0: Constant current load mode (for normal measurement) 0 1: No load mode 1 0: Constant current load mode (for calibration) 1 1: Resistive load mode (for calibration)	R/W

Bit	Symbol	Bit Name	Description	R/W
b21, b20	POSEL[1:0]	Non-measurement Channel Output Select	b21 b20 0 0: Outputs low 0 1: Open (Hi-Z) 1 0: Outputs low (uses power supply selected by TXVSEL[1:0] bits) 1 1: Outputs pulse in phase with transmit channel	R/W
b22	SDPSEL	Sensor Drive Pulse Select	0: Random pulse mode 1: High resolution pulse mode (SUCLK mode)	R/W
b23	PCSEL	Voltage Booster Clock Select	0: Sensor drive pulse 1: State clock	R/W
b29 to b24	STCLK[5:0]	State Clock Select	When the setting value is n, the frequency of the state clock (STCLK) is 1/2(n+1) of that of operating clock. n = 0 to 63, STCLK = Operating clock / (2 to 128) Set the frequency of STCLK to 500 kHz.	R/W
b30	DCMODE	Current measurement Mode Select	0: Normal mode 1: Current measurement mode	R/W
b31	DCBACK	Current Measurement Feedback Select	0: Feedback from TSCAP pin 1: Feedback from TSm pin	R/W

Note: Set bits other than the STRT and INIT bits while the STRT bit is 0.
Note 1. The CTSUSCNT, CTSUMCH, and CTSUSR registers are initialized.
Note 2. When the voltage of VCC is less than 4.5 V, set this bit to 1.
Note 3. Low-noise VCC is a power supply supplied from the VCC pin using wiring dedicated to the CTSU transmission power supply. Since there is no common impedance with other circuits, it is less susceptible to switching noise caused by the operation of surrounding pins.
Note 4. When the voltage of VCC is less than 2.4 V, set this bit to 1.

STRT Bit (Measurement Operation Start)

This bit controls whether CTSU operation starts or stops.

Writing 1 to the STRT bit while the CAP bit is 0 (software trigger) starts measurement. The STRT bit becomes 0 when the measurement is finished.

When the STRT bit is set to 1 while the CAP bit is 1 (external trigger), the CTSU waits for an external trigger. When an external trigger is input, measurement is started. When measurement is finished, the CTSU waits for the next external trigger and operation is continued.

Table 39.3 lists the CTSU status.

Table 39.3 CTSU Status

CAP Bit	STRT Bit	CTSUSR.STC[2:0] Flag	CTSU Status
0	0	—	Stopped
0	1	—	During measurement
1	0	—	Stopped
1	1	000b	Waiting for an external trigger
1	1	Other than 000b	During measurement

If the STRT bit is set to 1 when the STRT bit is 1, the writing is ignored and operation is continued.

To forcibly stop operation (forced stop) when the STRT bit is 1, set the STRT bit to 0 and the INIT bit to 1 simultaneously.

SNZ Bit (Snooze Function Enable)

This bit enables or disables measurement during a snooze mode when the external trigger is selected. This bit can also be used to suspend the CTSU, which decreases power consumption during the wait state.

Table 39.4 CTSU Internal Status

PON Bit	SNZ Bit	CAP Bit	STRT Bit	External Trigger	CTSU Internal State
0	0	0	0	—	Stopped
1	0	—	—	—	Operating
1	1	0	0	—	Suspended (low power consumption mode)*1
1	1	1	0	—	Suspended
1	1	1	1	None	Suspended (waiting a trigger)
1	1	1	1	Inputted	Measuring

Note: Settings other than the above are prohibited.
Note 1. This mode suspends CTSU to reduce power consumption. Measurement is not possible.

A trigger can cause the MCU to transition form software standby mode to snooze mode and allow CTSU to make measurement.
Measurement in snooze mode can be performed by the following procedure.

(1) Set the CTSU external trigger to LPT compare match 1 on the event link controller (ELC).
(2) Set the transition condition to snooze mode to LPT compare match 1, and set the release condition from snooze mode to the measurement end interrupt (CTSUFN interrupt).
(3) After setting the CAP and SNZ bits to 1, set the STRT bit to 1 to wait for an external trigger.
(4) Transition the CPU to software standby mode.
(5) When an external trigger is detected, the CPU transitions to snooze mode and CTSU starts measurement.
(6) When the measurement in completed, a CTSUFN interrupt is generated and the CPU enters normal operation mode.
(7) Set the SNZ bit to 0 to release CTSU from the suspended state. To continue the suspended state, set the SNZ bit to 1 again.

When the sensor drive pulse is set to high resolution pulse mode (CTSUCRA.SDPSEL bit = 1) and the snooze function is used, the sensor drive pulse cannot be selected for the clock of voltage booster (CTSUCRA.PCSEL bit = 0).

INIT Bit (Control Block Initialization)

Writing this bit to 1 can initialize the internal control registers. To forcibly stop the current measurement, set the STRT bit to 0 and the INIT bit to 1 simultaneously.
Do not write 1 to the INIT bit at the same time as setting the STRT bit to 1 (CTSU operation starts).

PUMPON Bit (Voltage Booster Enable)

This bit controls the voltage booster to generate the power supply for the analog switch. When this bit is 0, the output voltage is the same as VCC, and when it is 1, it is about 4.5 V. If the VCC voltage is less than 4.5 V, set this bit to 1.

TXVSEL[1:0] Bits (Transmission Power Supply Select)

These bits are used to select the I/O power supply of the TSm pin which is set for transmission in the mutual capacitance method. The selected power supply is also used as the power supply for an active shield electrode in the self-capacitance method.
When using the TSm pin for active shield in the self-capacitance method, set these bits to 10b.

PON Bit (Measurement Power Supply Enable)

This bit controls the supply of measurement power (1.5 V).
Before setting this bit to 1, Set the PUMPON bit and set the CSW bit to 1.

CSW Bit (Capacitor for Measurement Power Supply Connection)

This bit controls charging of the capacitor connected to the TSCAP pin.
Before setting this bit to 1, output low from the general-purpose I/O port (the PC4 pin) multiplexed to the TSCAP pin to discharge the electric charge of the capacitor.
After setting this bit to 1, wait until the capacitor connected to the TSCAP pin is charged before starting measurement.

ATUNE0 Bit (Power Supply Voltage Setting)

This bit is used to control the voltage of measurement power supply.
Set this bit according to the VCC voltage. Set this bit to 1 if the VCC voltage is less than 2.4 V. If the VCC voltage is 2.4 V or higher, either 0 or 1 can be set.

ATUNE1 Bit (Current Range Switch), ATUNE2 Bit (Current Range Switch 2)

These bits are used to select the maximum supply current (measurement range) of the measurement power supply.

Table 39.5 The ATUNE2 and ATUNE1 Bits and Measurement Range

ATUNE2 Bit	ATUNE1 Bit	Measurement Range
0	0	80 µA
0	1	40 µA
1	0	20 µA
1	1	160 µA (current measurement mode)

MD1 Bit (Measurement Mode Select 1)

This bit is used to select the measurement method (self-capacitance or mutual capacitance).
When setting this bit to 0 (self-capacitance method), set the channel to be measured for reception (CHTRCm bit = 0 (m = 0 to 35)). It can be set for transmission (CHTRCm bit = 1) only when used as an active shield. If there is a channel set for transmission, a pulse with the same phase as the channel being measured is output from the transmit pin.
When setting this bit to 1 (mutual capacitance method), arrange the transmit pin and receive pin evenly. Measurement is not possible if all channels are set for reception.
For details, refer to section 39.3.4, Measurement Modes.

LOAD[1:0] Bits (Measurement Load Control)

These bits are used to select the load for measurement.
When these bits are 00b, a constant current load of 2.5 µA is selected, when they are 10b, a constant current load of 20 µA is selected, and when they are 11b, a resistive load of 15 kΩ is selected.
When switching from the constant current mode to the resistive load mode, first switch to the no-load mode and then the resistive load mode.

POSEL[1:0] Bits (Non-measurement Channel Output Select)

These bits are used to select the output of the pins that are not being measured (non-measured pins).
For example, with the electrode arrangement shown in Figure 39.5, when measuring by the self-capacitance method using the TS0 to TS2 pins and using the TS3 pin and non-measurement pins as an active shield, set as follows.

- Set the TS3 pin for transmission: set the CTSUCHACA.CHAC3 and CTSUCHTRCA.CHTRC3 bits to 1.
- Set the TS0 to TS2 pins for reception: set the CTSUCHACA.CHAC0 to CHAC2 bits to 1 and the CTSUCHTRCA.CHTRC0 to CHTRC2 bits to 0.
- Set the output of the non-measured pins to in-phase pulse: Set the POSEL[1:0] bits to 11b.

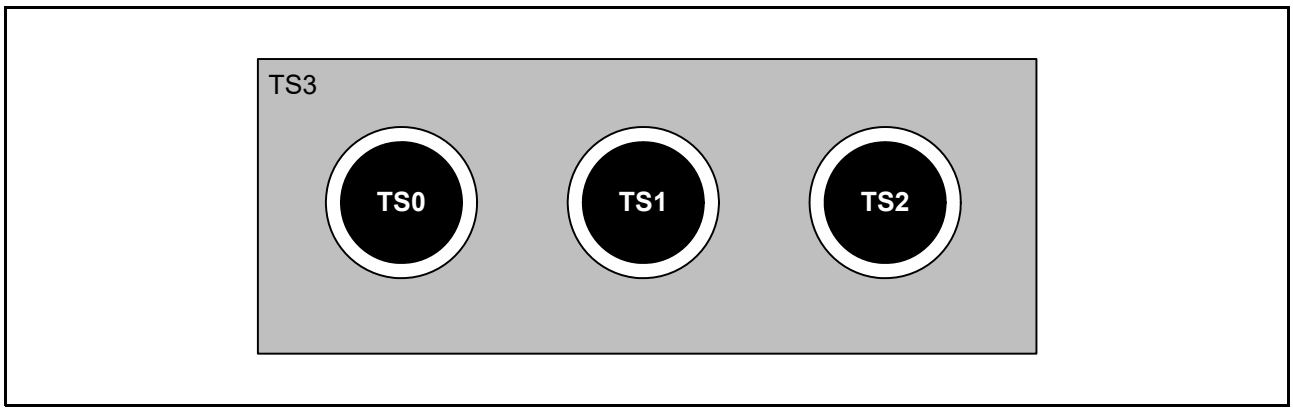


Figure 39.5 Example of Electrode Arrangement

SDPSEL Bit (Sensor Drive Pulse Select)

This bit is used to select the sensor drive pulse and sensor unit clock (SUCLK).
When this bit is set to 0 (random pulse mode), a sensor drive pulse synchronized with PCLKB is generated by randomly extending the low width of the clock (base clock), which is generated by dividing the operating clock by the value specified in the CTSUSO.SDPA[7:0] bits, using a pseudo-random number. By resampling this pulse with SUCLK, a clock with random high and low widths can be used as a sensor drive pulse. SUCLK is generated according to the settings of the CTSUTRIMA.SUADJD[7:0], CTSUCRB.SSCNT[1:0], and CTSUCRB.SSMOD[2:0] bits. In random pulse mode, SUCLK is used only for resampling (spread spectrum) the sensor drive pulse.
When this bit is set to 1 (high resolution pulse mode), the clock (base clock) obtained by dividing SUCLK by the value specified in the CTSUSO.SDPA[7:0] bits becomes the sensor drive pulse. SUCLK is generated according to the settings of the CTSUSUCLKA and CTSUSUCLKB registers. Set the CTSUSUCLKA and CTSUSUCLKB registers before setting the SDPSEL bit to 1, and do not rewrite them after setting it to 1. When the CTSUCALIB.CCOCLK bit is set to 1, the SUCLK frequency is adjusted for each measurement.

STCLK[5:0] Bits (State Clock Select)

These bits are used to specify the frequency division value of the state clock (STCLK) relative to the operating clock.
STCLK is related to the measurement time and the frequency adjustment cycle of SUCLK.
If the set value of these bits is n, the frequency division value is calculated by the following formula.

Frequency division value = (n + 1) × 2

Set the value so that the STCLK is 500 kHz.

DCMODE Bit (Current measurement Mode Select)

This bit is used to measure current without using the sensor drive pulse.
When this bit is set to 1, a fixed level specified by the CTSUCALIB.IOC bit is output from the receive pin, and the sensor drive pulse stops.

DCBACK Bit (Current Measurement Feedback Select)

This bit is used to select the feedback input for a measurement power supply (LDO). This is valid only when the DCMODE bit is 1.

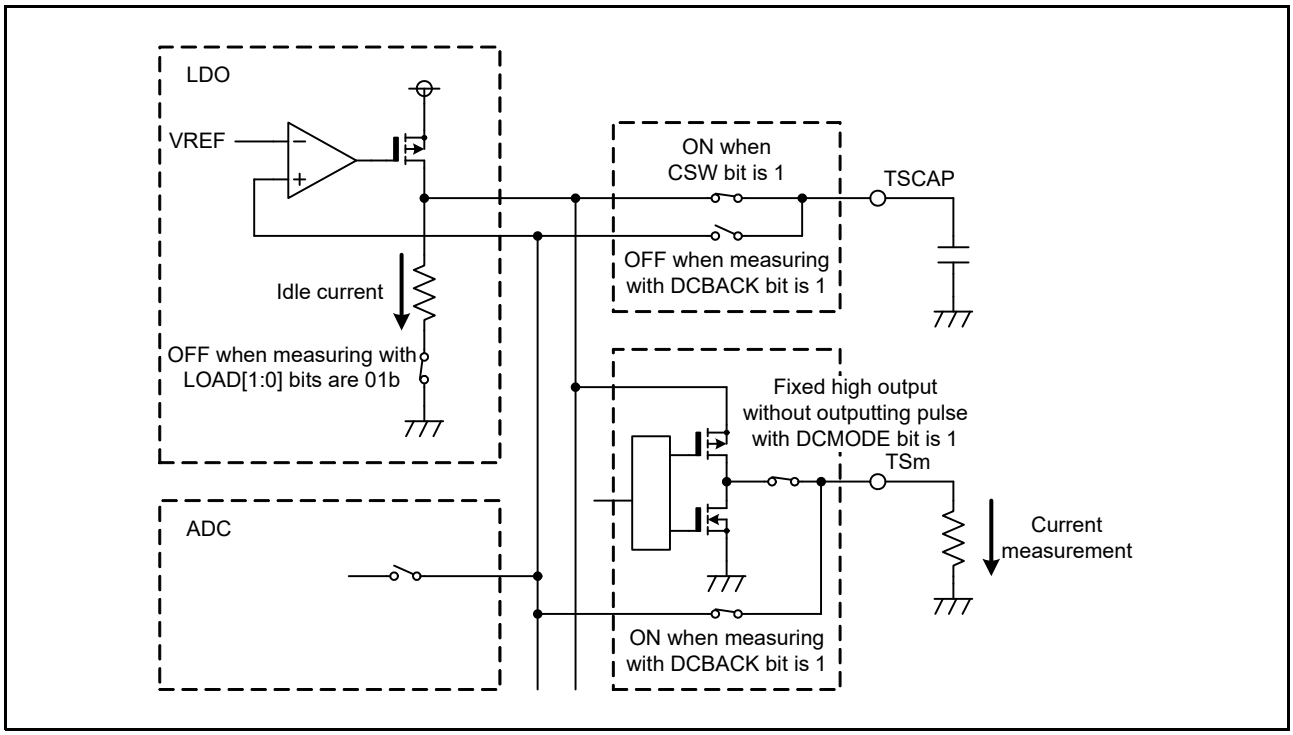


Figure 39.6 Feedback Loop for LDO

39.2.2 CTSU Control Register B (CTSUCRB)

Address(es): CTSU.CTSUCRB 000A 0904h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	SSCNT[1:0]		—	SSMOD[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
	SST[7:0]								PROFF	SOFF	PRMODE[1:0]		PRRATIO[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	PRRATIO[3:0]	Pseudo-Random Number Update Cycle Setting*1	Sets the shift period of the linear feedback shift register (LFSR) for pseudo-random number generation.	R/W
b5, b4	PRMODE[1:0]	Pseudo-Random Number Generation Cycle Setting*1	b5 b4 0 0: 255 cycles 0 1: 63 cycles 1 0: 31 cycles 1 1: 3 cycles	R/W
b6	SOFF	Spread-Spectrum Function Off*1	0: Spread-spectrum function is enabled, 1: Spread-spectrum function is disabled.	R/W
b7	PROFF	Pseudo-Random Number Off	0: Perform pseudo-random number control 1: Do not perform pseudo-random number control	R/W
b15 to b8	SST[7:0]	Sensor Stabilization Wait Time Setting	• Random pulse mode (CTSUCRA.SDPSEL = 0) When the setting value is n, the stabilization wait time is 2(n + 1) cycles of the sensor drive pulse synchronized with PCLKB. • High resolution pulse mode (CTSUCRA.SDPSEL = 1) When the setting value is n, the stabilization wait time is n + 1 cycles of STCLK.	R/W
b23 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	SSMOD[2:0]	SUCLK Spread-Spectrum Mode Select	b26 b24 0 0 0: 256 cycles 0 0 1: 384 cycles 0 1 0: 512 cycles 0 1 1: 1024 cycles 1 1 1: No spread-spectrum Settings other than above are prohibited.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b29, b28	SSCNT[1:0]	SUCLK Spread-Spectrum Control	b29 b28 0 0: CTSUTRIMA.SUADJD[7:0] + 00h 0 1: CTSUTRIMA.SUADJD[7:0] + 20h 1 0: CTSUTRIMA.SUADJD[7:0] + 40h 1 1: CTSUTRIMA.SUADJD[7:0] + 60h	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The CTSUCRB register should be set when the CTSUCRA.STRT bit is 0.
Note 1. The PRRATIO[3:0], PRMODE[1:0], SOFF bits are valid when the CTSUCRA.SDPSEL bit is 0 (random pulse mode).

PRRATIO[3:0] Bits (Pseudo-Random Number Update Cycle Setting)

These bits specifies the shift period of the linear feedback shift register (LFSR) used to generate pseudo-random numbers. Specify by the number of clocks of the base clock.
In addition, the value of this bit affects the number of measurement pulses and measurement time. These can be calculated by the following formula.
The number of basic pulses is twice the period selected by the PRMODE[1:0] bits.
Number of measurement pulses = number of basic pulses × (PRRATIO[3:0] bits + 1)
Measurement time = (number of basic pulses × (PRRATIO[3:0] bits + 1) + (number of basic pulses – 2) × 0.25) × period of base clock

PRMODE[1:0] Bits (Pseudo-Random Number Generation Cycle Setting)

These bits specifies the period at which the linear feedback shift register (LFSR) generated polynomial used to generate pseudo-random numbers is updated.
The number of basic pulses is twice the period selected by this bit.

SOFF Bit (Spread-Spectrum Function Off)

This bit is used to turn off the SUCLK resampling function for frequency diffusion. Set this bit to 1 when using a sensor drive pulse synchronized with PCLKB.

PROFF Bit (Pseudo-Random Number Off)

This bit is used to turn off pseudo-random number control. When this bit is set to 1, the LFSR is not used for pseudo-random number generation, and 1 or 0 is output for each cycle.

SST[7:0] Bits (Sensor Stabilization Wait Time Setting)

These bits are used to set the period from the supply of the sensor drive pulse to the stabilization of the voltage at the TSCAP pin.
The voltage at the TSCAP pin is stabilized by supplying a sensor drive pulse. The relationship between the value of these bits and the number of cycles is as follows.

- When the CTSUCRA.SDPSEL bit is 0
  Specify the stabilization wait time based on the number of cycles of the sensor drive pulse synchronized with PCLKB.
  Number of cycles = 2 × (SST[7:0] bit value + 1)
  Set the stabilization wait time within the following range.
  Number of cycles set by SST[7: 0] bits ≥ (PRRATIO[3:0] bits + 1)
- When the CTSUCRA.SDPSEL bit is 1
  Specify the stabilization wait time by the number of STCLK cycles.
  Number of cycles = 1 × (SST[7:0] bit value + 1)

SSMOD[2:0] Bits (SUCLK Spread-Spectrum Mode Select)

These bits are used to set the spread-spectrum period of SUCLK oscillator. The oscillation frequency of SUCLK is upspread based on the frequency specified by the CTSUTRIMA.SUADJD[7:0] and SSCNT[1:0] bits.

SSCNT[1:0] Bits (SUCLK Spread-Spectrum Control)

These bits adjust the oscillation frequency of the SUCLK oscillator specified by the CTSUTRIMA.SUADJD[7:0] bits.

39.2.3 CTSU Measurement Channel Register (CTSUMCH)

Address(es): CTSU.CTSUMCH 000A 0908h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	MCA3	MCA2	MCA1	MCA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MCH1[5:0]						—	—	MCH0[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	MCH0[5:0]	Measurement Channel 0	• Single scan mode Specify the number of the receive channel to be measured. • Multi-scan modes The number of the receive channel being measured is indicated.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	MCH1[5:0]	Measurement Channel 1	• Single scan mode Specify the number of the transmit channel to be measured. • Multi-scan modes The number of the transmit channel being measured is indicated.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	MCA0	Multi Clock 0 Enable	0: Disabled 1: Enabled*1	R/W
b17	MCA1	Multi Clock 1 Enable		
b18	MCA2	Multi Clock 2 Enable		
b19	MCA3	Multi Clock 3 Enable		
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The CTSUMCH register should be set when the CTSUCRA.STRT bit is 0.
Note 1. When using the majority mode (CTSUOPT.MAJIRIMD bit = 1), set only three of the four bits to 1.

MCH0[5:0] Bits (Measurement Channel 0)

In single scan mode, specify the receive channel to measure. Do not specify a channel that is set to 0 (not measurement target) in the CTSUCHACA and CTSUCHACB registers. If specified, the measurement will be completed immediately after the measurement starts.
In multi-scan mode, these bits indicates the receive channel being measured. Even if the value is rewritten, it will be cleared at the start of measurement.
When the measurement is completed, it becomes 111111b.

MCH1[5:0] Bits (Measurement Channel 1)

In single scan mode, specify the transmit channel to measure. Do not specify a channel that is set to 0 (not measurement target) in the CTSUCHACA and CTSUCHACB registers. If specified, the measurement will be completed immediately after the measurement starts.
In multi-scan mode, these bits indicates the transmit channel being measured. Even if the value is rewritten, it will be cleared at the start of measurement.
When the measurement is completed, it becomes 111111b.

MCAn Bit (Multi Clock n Enable) (n = 0 to 3)

These bits allow the use of each clock when measuring using multiple sensor unit clocks (SUCLK) with different frequencies (multi-clock measurement).

When the MCAn bit is set to 1, SUCLKn is used for measurement.

When multiple MCAn bits are enabled and measurement is started, the specified channel is measured while switching the clock in ascending order from SUCLK0. When the measurement is completed using all the allowed SUCLKs, the measurement of the next channel is started.

The frequencies of SUCLK0 to SUCLK3 are specified by the CTSUSUCLKA and CTSUSUCLKB registers.

39.2.4 CTSU Channel Enable Control Register A (CTSUCHACA)

Address(es): CTSU.CTSUCHACA 000A 090Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CHAC31	CHAC30	CHAC29	CHAC28	CHAC27	CHAC26	CHAC25	CHAC24	CHAC23	CHAC22	CHAC21	CHAC20	CHAC19	CHAC18	CHAC17	CHAC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CHAC15	CHAC14	CHAC13	CHAC12	CHAC11	CHAC10	CHAC9	CHAC8	CHAC7	CHAC6	CHAC5	CHAC4	CHAC3	CHAC2	CHAC1	CHAC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CHAC0	Channel 0 Enable Control	0: Not measurement target 1: Measurement target	R/W
b1	CHAC1	Channel 1 Enable Control		R/W
b2	CHAC2	Channel 2 Enable Control		R/W
b3	CHAC3	Channel 3 Enable Control		R/W
b4	CHAC4	Channel 4 Enable Control		R/W
b5	CHAC5	Channel 5 Enable Control		R/W
b6	CHAC6	Channel 6 Enable Control		R/W
b7	CHAC7	Channel 7 Enable Control		R/W
b8	CHAC8	Channel 8 Enable Control		R/W
b9	CHAC9	Channel 9 Enable Control		R/W
b10	CHAC10	Channel 10 Enable Control		R/W
b11	CHAC11	Channel 11 Enable Control		R/W
b12	CHAC12	Channel 12 Enable Control		R/W
b13	CHAC13	Channel 13 Enable Control		R/W
b14	CHAC14	Channel 14 Enable Control		R/W
b15	CHAC15	Channel 15 Enable Control		R/W
b16	CHAC16	Channel 16 Enable Control		R/W
b17	CHAC17	Channel 17 Enable Control		R/W
b18	CHAC18	Channel 18 Enable Control		R/W
b19	CHAC19	Channel 19 Enable Control		R/W
b20	CHAC20	Channel 20 Enable Control		R/W
b21	CHAC21	Channel 21 Enable Control		R/W
b22	CHAC22	Channel 22 Enable Control		R/W
b23	CHAC23	Channel 23 Enable Control		R/W
b24	CHAC24	Channel 24 Enable Control		R/W
b25	CHAC25	Channel 25 Enable Control		R/W
b26	CHAC26	Channel 26 Enable Control		R/W
b27	CHAC27	Channel 27 Enable Control		R/W
b28	CHAC28	Channel 28 Enable Control		R/W
b29	CHAC29	Channel 29 Enable Control		R/W
b30	CHAC30	Channel 30 Enable Control		R/W
b31	CHAC31	Channel 31 Enable Control		R/W

Note: The CTSUCHACA register should be set when the CTSUCRA.STRT bit is 0.

39.2.5 CTSU Channel Enable Control Register B (CTSUCHACB)

Address(es): CTSU.CTSUCHACB 000A 0910h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CHAC35	CHAC34	CHAC33	CHAC32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CHAC32	Channel 32 Enable Control	0: Not measurement target 1: Measurement target	R/W
b1	CHAC33	Channel 33 Enable Control		R/W
b2	CHAC34	Channel 34 Enable Control		R/W
b3	CHAC35	Channel 35 Enable Control		R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The CTSUCHACB register should be set when the CTSUCRA.STRT bit is 0.

CHACm Bit (Channel m Enable Control) (m = 32 to 63)

This bit sets the pin (for both reception and transmission) whose electrostatic capacitance is to be measured.
Set the bit corresponding to the non-existent pin to 0.

39.2.6 CTSU Channel Transmit/Receive Control Register A (CTSUCHTRCA)

Address(es): CTSU.CTSUCHTRCA 000A 0914h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CHTRC31	CHTRC30	CHTRC29	CHTRC28	CHTRC27	CHTRC26	CHTRC25	CHTRC24	CHTRC23	CHTRC22	CHTRC21	CHTRC20	CHTRC19	CHTRC18	CHTRC17	CHTRC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CHTRC15	CHTRC14	CHTRC13	CHTRC12	CHTRC11	CHTRC10	CHTRC9	CHTRC8	CHTRC7	CHTRC6	CHTRC5	CHTRC4	CHTRC3	CHTRC2	CHTRC1	CHTRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CHTRC0	Channel 0 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b1	CHTRC1	Channel 1 Transmit/Receive Control		R/W
b2	CHTRC2	Channel 2 Transmit/Receive Control		R/W
b3	CHTRC3	Channel 3 Transmit/Receive Control		R/W
b4	CHTRC4	Channel 4 Transmit/Receive Control		R/W
b5	CHTRC5	Channel 5 Transmit/Receive Control		R/W
b6	CHTRC6	Channel 6 Transmit/Receive Control		R/W
b7	CHTRC7	Channel 7 Transmit/Receive Control		R/W
b8	CHTRC8	Channel 8 Transmit/Receive Control		R/W
b9	CHTRC9	Channel 9 Transmit/Receive Control		R/W
b10	CHTRC10	Channel 10 Transmit/Receive Control		R/W
b11	CHTRC11	Channel 11 Transmit/Receive Control		R/W
b12	CHTRC12	Channel 12 Transmit/Receive Control		R/W
b13	CHTRC13	Channel 13 Transmit/Receive Control		R/W
b14	CHTRC14	Channel 14 Transmit/Receive Control		R/W
b15	CHTRC15	Channel 15 Transmit/Receive Control		R/W
b16	CHTRC16	Channel 16 Transmit/Receive Control		R/W
b17	CHTRC17	Channel 17 Transmit/Receive Control		R/W
b18	CHTRC18	Channel 18 Transmit/Receive Control		R/W
b19	CHTRC19	Channel 19 Transmit/Receive Control		R/W
b20	CHTRC20	Channel 20 Transmit/Receive Control		R/W
b21	CHTRC21	Channel 21 Transmit/Receive Control		R/W
b22	CHTRC22	Channel 22 Transmit/Receive Control		R/W
b23	CHTRC23	Channel 23 Transmit/Receive Control		R/W
b24	CHTRC24	Channel 24 Transmit/Receive Control		R/W
b25	CHTRC25	Channel 25 Transmit/Receive Control		R/W
b26	CHTRC26	Channel 26 Transmit/Receive Control		R/W
b27	CHTRC27	Channel 27 Transmit/Receive Control		R/W
b28	CHTRC28	Channel 28 Transmit/Receive Control		R/W
b29	CHTRC29	Channel 29 Transmit/Receive Control		R/W
b30	CHTRC30	Channel 30 Transmit/Receive Control		R/W
b31	CHTRC31	Channel 31 Transmit/Receive Control		R/W

Note: The CTSUCHTRCA register should be set when the CTSUCRA.STRT bit is 0.

39.2.7 CTSU Channel Transmit/Receive Control Register B (CTSUCHTRCB)

Address(es): CTSU.CTSUCHTRCB 000A 0918h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CHTRC35	CHTRC34	CHTRC33	CHTRC32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CHTRC32	Channel 32 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b1	CHTRC33	Channel 33 Transmit/Receive Control		R/W
b2	CHTRC34	Channel 34 Transmit/Receive Control		R/W
b3	CHTRC35	Channel 35 Transmit/Receive Control		R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The CTSUCHTRCB register should be set when the CTSUCRA.STRT bit is 0.

CHTRCm Bit (Channel m Transmit/Receive Control) (m = 0 to 35)

This bit is used to assign reception or transmission to channel m (TSm pin) in the mutual capacitance method.
It can be used as an active shield signal output by setting the pin to transmission in the self-capacitance method (CTSUCRA.MD1 bit = 0). However, when using a transmit channel as an active shield output, do not set more than 2 bits to 1.

39.2.8 CTSU Status Register (CTSUSR)

Address(es): CTSU.CTSUSR 000A 091Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PS	UCOVF	SCOVF	DTSR	—	STC[2:0]			ICOMP0	ICOMP1	ICOMPRST	—	—	—	MFC[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	MFC[1:0]	Multi-Clock Counter	b1 b0 0 0: SUCLK0 0 1: SUCLK1 1 0: SUCLK2 1 1: SUCLK3	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ICOMPRST	ICOMP0 and ICOMP1 Flags Reset	Writing this bit to 1 clears the ICOMP0 and ICOMP1 flags. This bit is read as 0.	R/W
b6	ICOMP1	Over-Current Detection Flag	0: Normal 1: Over-current is detected.	R
b7	ICOMP0	Over-Voltage Detection Flag	0: Normal 1: Over-voltage is detected.	R
b10 to b8	STC[2:0]	Measurement State Counter	b10 b8 0 0 0: State 0 0 0 1: State 1 0 1 0: State 2 0 1 1: State 3 1 0 0: State 4 1 0 1: State 5	R
b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	DTSR	Data Transfer Status Flag	0: Measurement result has been read. 1: Measurement result has not been read.	R
b13	SCOVF	Sensor Counter Overflow Flag*1	0: Does not overflow 1: Overflows	R/(W)*2
b14	UCOVF	Sensor Unit Clock Counter Overflow Flag*1	0: Does not overflow 1: Overflows	R/(W)*2
b15	PS	Mutual Capacitance Measurement Status Flag	0: First measurement 1: Second measurement	R
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When using the CTSUCRA.INIT bit to clear the SCOVF and UCOVF flags, make sure that the CTSUCRA.STRT bit is 0.
Note 2. 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.

MFC[1:0] Bits (Multi-Clock Counter)

These bits indicate the SUCLK number used when multi-clock measurement is being executed (CTSUCRA.SDPSEL bit = 1, at least 2 bits of the CTSUMCH.MCAn bit are 1 (n = 0 to 3)).
When writing a value to the CTSUSR register, write 00b to these bits.

ICOMP1 Flag (Over-Current Detection Flag)

This flag indicates an abnormal output current of the measurement power supply. This flag is cleared when the CTSUCRA.PON bit is set to 0 or the ICOMPRST bit is set to 1.

When this flag becomes 1, the value of the CTSUSCNT.SC[15:0] bits is FFFFh.

ICOMP0 Flag (Over-Voltage Detection Flag)

This flag indicates an abnormal output voltage of the measurement power supply. This flag is cleared when the CTSUCRA.PON bit is set to 0 or the ICOMPRST bit is set to 1.
When this flag becomes 1, the value of the CTSUSCNT.SC[15:0] bits is 0000h.

STC[2:0] Bits (Measurement State Counter)

These bits indicate the current measurement state. For details on each state, refer to section 39.3.3, Measurement State.

DTSR Flag (Data Transfer Status Flag)

This flag indicates whether the measurement result stored in the sensor counter has been read.
[Setting condition]

- When measurement is completed and the measurement result is stored in the CTSUSCNT register.

[Clearing conditions]

- When the CTSUSCNT register is read.
- When the CTSUCRA.INIT bit is set to 1.

SCOVF Flag (Sensor Counter Overflow Flag)

This flag indicates whether the sensor counter has overflowed. The value of the CTSUSCNT.SC[15:0] bits is FFFFh when this flag becomes 1.
Even if an overflow occurs during the measurement, the measurement process continues until the set period.
Determine which channel the overflow occurred during measurement by reading the measurement results of each channel after the measurement of all channels is completed (after the CTSUFN interrupt occurs).
[Setting condition]

- When the CTSUSCNT.SC[15:0] bits overflows during measurement.

[Clearing conditions]

- When writing 0 after confirming that the SCOVF flag is 1.
- When the CTSUCRA.INIT bit is set to 1.

UCOVF Flag (Sensor Unit Clock Counter Overflow Flag)

This flag indicates whether the sensor unit clock counter has overflowed. The value of the CTSUSCNT.UC[15:0] bits is FFFFh when this flag becomes 1.
Even if an overflow occurs during the measurement, the measurement process continues until the set period.
Determine which channel the overflow occurred during measurement by reading the measurement results of each channel after the measurement of all channels is completed (after the CTSUFN interrupt occurs).
[Setting condition]

- When the CTSUSCNT.UC[15:0] bits overflows during measurement.

[Clearing conditions]

- When writing 0 after confirming that the UCOVF flag is 1.
- When the CTSUCRA.INIT bit is set to 1.

PS Flag (Mutual Capacitance Measurement Status Flag)

This flag indicates whether the measurement is the first or second of two measurements for each channel in mutual capacitance method (CTSUCRA.MD1 bit = 1).
This flag indicates 0 while measurement is stopped or in self-capacitance method (MD1 bit = 0).

39.2.9 CTSU Sensor Offset Register (CTSUSO)

Address(es): CTSU.CTSUSO 000A 0920h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SDPA[7:0]								SSDIV[3:0]				—	—	SNUM[7:6]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SNUM[5:0]						SO[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SO[9:0]	Sensor Offset Adjustment	These bits adjust the input offset current of sensor CCO.	R/W
b17 to b10	SNUM[7:0]	Measurement Period Setting	• Random pulse mode (CTSUCRA.SDPSEL = 0) Set the measurement period by the number of repetitions of the basic measurement unit. The range of values that can be set is 00h to 3Fh. When the set value is n, the basic unit of measurement is repeated n + 1 times. • High resolution pulse mode (CTSUCRA.SDPSEL = 1) Set the measurement period based on the STCLK cycle. When the set value is n, measurement is performed for 8(n + 1) cycles of STCLK.	R/W
b19 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b20	SSDIV[3:0]	Spread Spectrum Sampling Period Control	b23 b20 0 0 0 0: SUCLK 0 0 0 1: SUCLK divided by 2 : : 1 1 1 0: SUCLK divided by 15 1 1 1 1: SUCLK divided by 16	R/W
b31 to b24	SDPA[7:0]	Base Clock Setting*1	• Random pulse mode (CTSUCRA.SDPSEL = 0) When the set value is n, the base clock frequency is the operating clock divided by 2(n + 1). • High resolution pulse mode (CTSUCRA.SDPSEL = 1) When the set value is n, the base clock frequency is SUCLK divided by 2(n + 1).	R/W

Note 1. When the spread spectrum function is disabled (CTSUCRB.SOFF bit = 1) or in the mutual capacitance method (CTSUCRA.MD1 bit = 1), do not set the SDPA[7:0] bits to 00h.

Write this register after a CTSUWR interrupt is generated. Writing to this register causes a transition to State 3. Set this register to all 32-bit values at once.

SO[9:0] Bits (Sensor Offset Adjustment)

These bits adjust the input offset current of the sensor CCO.
These bits are used to offset the sensor CCO input current generated from electrostatic capacitance while the electrode is not being touched during touch measurement, thus preventing overflow of the CTSU sensor counter.

SNUM[7:0] Bits (Measurement Period Setting)

In random pulse mode (CTSUCRA.SDPSEL bit = 0), these bits specify how many times the number of measurement pulses (basic measurement unit) specified by the CTSUCRB.PRRATIO[3:0] and CTSUCRB.PRMODE[1:0] bits is repeated during measurement. The number of iterations is (SNUM[7:0] bits + 1).
In high resolution pulse mode (CTSUCRA.SDPSEL bit = 1), these bits specify the measurement period based on the STCLK cycle. The measurement is performed during (8 × (SNUM[7: 0] bit + 1) cycles of the STCLK.

SSDIV[3:0] Bits (Spread Spectrum Sampling Period Control)

In random pulse mode (CTSUCRA.SDPSEL bit = 0), this bit sets which clock is used to resample the PCLKB-synchronized sensor drive pulse with SUCLK. Set the frequency of the selected clock to be at least four times the frequency of the sensor drive pulse.

SDPA[7:0] Bits (Base Clock Setting)

In random pulse mode (SDPSEL bit = 0), these bits specify the frequency division value of the base clock with respect to the operating clock. When the set value is n, the frequency division value is expressed as 2(n + 1). In random pulse mode, the spread spectrum of the base clock is the sensor drive pulse.

In high resolution pulse mode (SDPSEL bit = 1), these bits specify the frequency division value of the base clock with respect to SUCLK. When the set value is n, the frequency division value is expressed as 2(n + 1). In high resolution pulse mode, the base clock is used as the sensor drive pulse.

39.2.10 CTSU Sensor Counter (CTSUSCNT)

Address(es): CTSU.CTSUSCNT 000A 0924h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	UC[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
	SC[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	SC[15:0]	Sensor Counter	These bits indicate the measurement result. When an overflow occurs, these bits become FFFFh.	R
b31 to b16	UC[15:0]	Sensor unit Counter	These bits indicate the count value of the clock (twice the frequency of SUCLK) output from the CCO of SUCLK. When an overflow occurs, these bits become FFFFh.	R

Read this register after the CTSURD interrupt occurs.
When the value of this register is read the number of times set by the CTSUCALIB.CNTRDSEL bit, the measurement state transitions to State 0 (CTSUSR.STC[2:0] flags are 000b) or State 2 (STC[2:0] flags are 010b).
The values of the counters are cleared just before the measurement state transitions to State 4 (STC[2:0] flags are 100b) on the next measurement. These counters are also cleared by the CTSUCRA.INIT bit.

(a) When the CTSUCALIB.CNTRDSEL bit is 0

Reading this register once advances the measurement state.
For 16-bit access, reading either the SC[15:0] or UC[15:0] bits advances the measurement state and both counters are cleared before transitioning to State 4. If the measurement result is needed, read the SC[15:0] bits.

(b) When the CTSUCALIB.CNTRDSEL bit is 1

Reading this register twice advances the measurement state.
Use this mode when the results of both SC[15:0] and UC[15:0] bits are needed for 16-bit access.

SC[15:0] Bits (Sensor Counter)

These bits are a counter that counts the clock output from the sensor CCO during the measurement period.

UC[15:0] Bits (Sensor unit Counter)

These bits are a counter that counts the clock with a frequency twice that of the sensor unit clock.

39.2.11 CTSU Calibration Register (CTSUCALIB)

Address(es): CTSU.CTSUCALIB 000A 0928h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXREV	CCOCALIB	CCOCLK	DACCLK	SUCARRY	SUMSEL	DACCARRY	DACMSEL	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IOCSEL	DCOFF	—	IOC	CNTRDSEL	TSOC	SUCLKEN	CLKSEL[1:0]		DRV	TSOD	—	—
Value after 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	TSOD	All TS Pin Output Control	0: Capacitance measurement mode 1: Selected by the IOCSEL bit	R/W
b3	DRV	Calibration Setting 1	0: Capacitance measurement mode 1: Calibration mode 1	R/W
b5, b4	CLKSEL[1:0]	Monitor Clock Select	b5 b4 0 0: Low is output. 0 1: Sensor CCO clock divided by 4 1 0: Setting prohibited 1 1: SUCLK divided by 4	R/W
b6	SUCLKEN	SUCLK Enable	0: SUCLK is stopped. 1: SUCLK is oscillated.	R/W
b7	TSOC	Calibration Setting 2	0: Capacitance measurement mode 1: Calibration mode 2	R/W
b8	CNTRDSEL	Sensor Counter Register Read Count Select	0: Transition to the next state by reading once 1: Transition to the next state by reading twice	R/W
b9	IOC	TS Pin Output Control	• When TSOD = 1 and IOCSEL = 1 0: Low is output from all TSm pins. 1: High is output from all TSm pins. • When CTSUCRA.DCMODE = 1 0: High is output from all TSm pin. 1: Low is output from all TSm pin.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	DCOFF	Down Convert Off	0: Normal operating mode 1: Down Convert is Off	R/W
b12	IOCSEL	TS Pin Fixed Output Select	0: Capacitance measurement mode using multiple electrode connection function 1: TS pin output test mode (for calibration)	R/W
b23 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DACMSEL	DAC Current Matrix Fixing	0: Rotate current source 1: Fix the current source	R/W
b25	DACCARRY	DAC Upper Current Source Carry Input	0: Normal measurement 1: Add 64 to the data input	R/W
b26	SUMSEL	CCO Current Matrix Fixing	0: Rotate current source 1: Fix the current source	R/W
b27	SUCARRY	CCO Carry Input	0: Normal measurement 1: Add 32 to the data input	R/W
b28	DACCLK	DAC Modulation Circuit Clock Select	0: Operating clock 1: SUCLK	R/W
b29	CCOCLK	CCO Modulation Circuit Clock Select	0: Operating clock 1: SUCLK	R/W

Bit	Symbol	Bit Name	Description	R/W
b30	CCOCALIB	CCO Calibration Mode Select	• Normal mode 0: LDO current → Oscillator for current measurement 1: SSCNT current → SUCLK oscillator • Oscillator calibration mode 0: LDO current → SUCLK oscillator 1: SSCNT current → Oscillator for current measurement	R/W
b31	TXREV	Transmit Pin Inverting Output	• Self-capacitance method (CTSUCRA.MD1 = 0) 0: Outputs a signal with the same phase as the receive pin 1: Outputs a signal with the opposite phase to the receive pin • Mutual capacitance method (CTSUCRA.MD1 = 1) 0: Outputs a signal with the same phase as the receive pin at the first time, and outputs the signal with the opposite phase to the receive pin 1: Outputs a signal with the opposite phase to the receive pin at the first time, and outputs the signal with the same phase as the receive pin	R/W

TSOD Bit (All TS Pin Output Control)

This bit is used when testing the output of the TSm pins or measuring the total capacitance of multiple electrodes. The setting is valid only in self-capacitance method (the CTSUCRA.MD1 bit is 0) and single scan mode (the CTSUCRA.MD0 bit is 0). When measuring the capacitance, set this bit to 0.
When the TSOD and IOCSEL bits are set to 1, the signal specified by the IOC bit is output from all TSm pins.
When the TSOD bit is set to 1 and the IOCSEL bit is set to 0, the sensor drive pulses are output from all TSm pins set for measurement and reception. In this case, the total capacitance of all electrodes can be measured.

DRV Bit (Calibration Setting 1)

This bit is used to calibrate the CTSU. When measuring the capacitance, set this bit to 0.
When the DRV bit is set to “1”, the state is pseudo to State 3 or State 4. It differs from State 3 and State 4 during measurement in the following points.

- Sensor drive pulse is not output.
- Enable and reset for the measurement counter is not output.

CLKSEL[1:0] Bits (Monitor Clock Select)

These bits select the clock for monitoring the waveform from the two clocks generated in CTSU. The selected clock divided by 4 can be monitored with the CLKOUT pin.

TSOC Bit (Calibration Setting 2)

This bit is used to calibrate the CTSU. When measuring the capacitance, set this bit to 0.

CNTRDSEL Bit (Sensor Counter Register Read Count Select)

This bit is used to select the number of times to read the CTSUSCNT register. Set this bit to 1 in the following case.

- When reading both the CTSUSCNT.SC[15:0] and CTSUSCNT.UC[15:0] bits with 16-bit access.

IOC Bit (TS Pin Output Control)

This bit is used to select the level to be output from all TSm pins when the TSOD and IOCSEL bits are set to 1 or the CTSUCRA.DCMODE bit is set to 1.
If either the TSOD bit or the IOCSEL bit is 0 and the CTSUCRA.DCMODE bit is 0, this bit is ignored.

DCOFF Bit (Down Convert Off)

This bit is used to turn off the LDO output. When measuring the capacitance, set this bit to 0.

DACMSEL Bit (DAC Current Matrix Fixing)

This bit is mainly used for current source characterization and testing. When measuring the capacitance, set this bit to 0.

SUMSEL Bit (CCO Current Matrix Fixing)

This bit is mainly used for current source characteristic correction and test evaluation. When measuring the capacitance, set this bit to 0.

CCOCALIB Bit (CCO Calibration Mode Select)

This bit is used to compare the external current and the output of the current DAC to correct the oscillator characteristics.

TXREV Bit (Transmit Pin Inverting Output)

This bit is used to invert the output from the transmit pin.
When this bit is set to 1, the pulse output from the transmit pin can be inverted. Normally, set it to 0.

39.2.12 CTSU Sensor Unit Clock Control Register A (CTSUSUCLKA)

Address(es): CTSU.CTSUSUCLKA 000A 092Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SUMULTI1[7:0]								SUADJ1[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
	SUMULTI0[7:0]								SUADJ0[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	SUADJ0[7:0]	SUCLK0 Frequency Setting	Set the initial value of the frequency of SUCLK0	R/W
b15 to b8	SUMULTI0[7:0]	SUCLK0 Multiplier Setting	Set the multiplication ratio between the SUCLK0 frequency and the STCLK frequency. If the set value is n, the frequency of SUCLK0 will be n + 1 times the frequency of STCLK.	R/W
b23 to b16	SUADJ1[7:0]	SUCLK1 Frequency Setting	Set the initial value of the frequency of SUCLK1	R/W
b31 to b24	SUMULTI1[7:0]	SUCLK1 Multiplier Setting	Set the multiplication ratio between the SUCLK1 frequency and the STCLK frequency. If the set value is n, the frequency of SUCLK1 will be n + 1 times the frequency of STCLK.	R/W

This register is used to set the feedback loop in the sensor unit clock oscillator.

SUADJn[7:0] Bits (SUCLKn Frequency Setting) (n = 0, 1)

These bits are used to set the initial value of the SUCLKn frequency.
The value of these bits is updated to correct the frequency deviation with each measurement.

SUMULTIn[7:0] Bits (SUCLKn Multiplier Setting) (n = 0, 1)

These bits are used to set the frequency divider ratio of the divider in the PLL.
The clock generated by the PLL is divided by (the value set in these bits + 1) and compared with the phase of STCLK.
The value of the SUADJn bit is updated based on the result of the comparison.
Set the value so that the frequency of SUCLKn is in the range of 16 MHz to 32 MHz.

39.2.13 CTSU Sensor Unit Clock Control Register B (CTSUSUCLKB)

Address(es): CTSU.CTSUSUCLKB 000A 0930h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SUMULTI3[7:0]								SUADJ3[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
	SUMULTI2[7:0]								SUADJ2[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	SUADJ2[7:0]	SUCLK2 Frequency Setting	Set the initial value of the frequency of SUCLK2	R/W
b15 to b8	SUMULTI2[7:0]	SUCLK2 Multiplier Setting	Set the multiplication ratio between the SUCLK2 frequency and the STCLK frequency. If the set value is n, the frequency of SUCLK2 will be n + 1 times the frequency of STCLK.	R/W
b23 to b16	SUADJ3[7:0]	SUCLK3 Frequency Setting	Set the initial value of the frequency of SUCLK3	R/W
b31 to b24	SUMULTI3[7:0]	SUCLK3 Multiplier Setting	Set the multiplication ratio between the SUCLK3 frequency and the STCLK frequency. If the set value is n, the frequency of SUCLK3 will be n + 1 times the frequency of STCLK.	R/W

This register is used to set the feedback loop in the sensor unit clock oscillator.

SUADJn[7:0] Bits (SUCLKn Frequency Setting) (n = 2, 3)

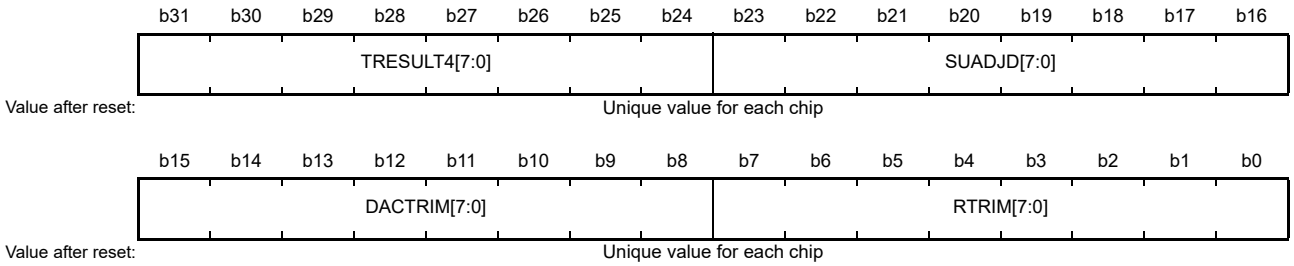
These bits are used to set the initial value of the SUCLKn frequency.
The value of these bits is updated to correct the frequency deviation with each measurement.

SUMULTIn[7:0] Bits (SUCLKn Multiplier Setting) (n = 2, 3)

These bits are used to set the frequency divider ratio of the divider in the PLL.
The clock generated by the PLL is divided by (the value set in these bits + 1) and compared with the phase of STCLK.
The value of the SUADJn bit is updated based on the result of the comparison.
Set the value so that the frequency of SUCLKn is in the range of 16 MHz to 32 MHz.

39.2.14 CTSU Trimming Register A (CTSUTRIMA)

Address(es): CTSU.CTSUTRIMA 007F C3A4h

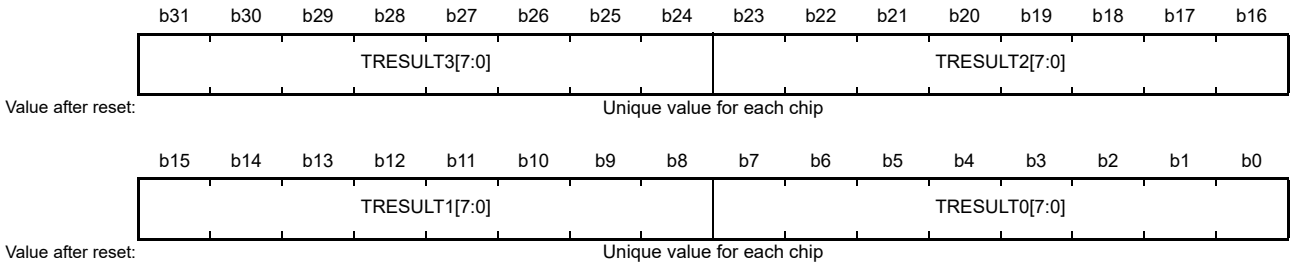


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RTRIM[7:0]	Reference Resistor Adjust Input	These bits are used to adjust the resistance value of the internal reference resistor. The value is written so that the specified current value is reached at 25°C.	R/W
b15 to b8	DACTRIM[7:0]	Offset Current Adjustment	These bits are used to adjust the coefficient of the current DAC for the current measurement oscillator.	R/W
b23 to b16	SUADJD[7:0]	SUCLK Frequency Adjustment	These bits are used to adjust the SUCLK frequency in random pulse mode (CTSUCRA.SDPSEL = 0). The value is written so that the frequency of SUCLK becomes about 32 MHz	R/W
b31 to b24	TRESULT4[7:0]	Test Result 4 Storage	These bits stores the coefficient of variation when the load resistance for the LDO is 120 kΩ.	R/W

This register stores adjustment data measured at the factory for each individual chip. Do not rewrite this register.

39.2.15 CTSU Trimming Register B (CTSUTRIMB)

Address(es): CTSU.CTSUTRIMB 007F C3A8h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRESULT0[7:0]	Test Result 0 Storage	These bits stores the coefficient of variation when the load resistance for the LDO is 7.5 kΩ.	R/W
b15 to b8	TRESULT1[7:0]	Test Result 1 Storage	These bits stores the coefficient of variation when the load resistance for the LDO is 15 kΩ.	R/W
b23 to b16	TRESULT2[7:0]	Test Result 2 Storage	These bits stores the coefficient of variation when the load resistance for the LDO is 30 kΩ.	R/W
b31 to b24	TRESULT3[7:0]	Test Result 3 Storage	These bits stores the coefficient of variation when the load resistance for the LDO is 60 kΩ.	R/W

This register stores adjustment data measured at the factory for each individual chip. Do not rewrite this register.

39.2.16 CTSU Option Setting Register (CTSUOPT)

Address(es): CTSU.CTSUOPT 000A 0940h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	SCACTB[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
	—	—	—	—	—	—	AJINTC	AJFEN	—	—	MTUCFEN	DTCLESS	—	MAJIRIMD	MCACFEN	CCOCFEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCOCFEN	CCO Characteristics Correction Function Enable	0: CCO characteristics correction function is disabled. 1: CCO characteristics correction function is enabled.	R/W
b1	MCACFEN	Multi-Clock Correction Function Enable	0: Multi-clock correction function is disabled. 1: Multi-clock correction function is enabled.*1	R/W
b2	MAJIRIMD	Majority Mode Enable	0: Majority mode is disabled. 1: Majority mode is enabled.*2	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DTCLESS	Data Transfer Request Disable*3	0: Data transfer request is enabled. 1: Data transfer request is disabled.	R/W
b5	MTUCFEN	Mutual Capacitance Calculation Enable *4	0: Do not subtract the first measurement data from the second measurement data 1: Subtract the first measurement data from the second measurement data	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	AJFEN	Automatic Judgment Function Enable	0: Automatic judgment function is disabled. 1: Automatic judgment function is enabled.*1	R/W
b9	AJINTC	Automatic Judgment Interrupt Control	0: Snooze end request is output when “non-touch” is detected. 1: CTSUFN interrupt is output regardless of judgment result.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	SCACTB[3:0]	Sensor Counter Automatic Correction Table Number Setting	Set the number of the sensor counter automatic correction table to be accessed. The range of values is 0 to 11. When a value is written to the CTSUSCNTACT register, the value of these bits is automatically incremented.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register while the CTSUCRA.STRT bit is 0.
Note 1. When setting this bit to 1, also set the CCOCFEN bit to 1.
Note 2. When setting this bit to 1, also set the CCOCFEN and MCACFEN bits to 1.
Note 3. When the CTSUCRA.MD0 bit is 1 (multi-scan mode), set this bit to 0.
Note 4. When the CTSUCRA.MD1 bit is 1 (mutual capacitance method) and AJFEN bit is 1, set this bit to 1.

This register is used to enable the automatic correction function and automatic judgment function.

CCOCFEN Bit (CCO Characteristics Correction Function Enable)

When this bit is set to 1, the measured value is corrected using the sensor counter automatic correction table n (n = 0 to 11).If the corrected result exceeds FFFFh, the value of the sensor counter becomes FFFFh and the CTSUSR.SCOVF flag becomes 1.

MAJIRIMD Bit (Majority Mode Enable)

When this bit is set to 1, the sum of the two closest values of the three values measured by three sensor unit clocks (SUCLK) with different frequencies is stored as the sensor counter value (VMM: value majority mode). When the AJFEN bit is set to 1, this sum is used to determine “touch” or “non-touch” and the result is stored in the CTSUAJRR.TJR0 flag. If the result of the addition exceeds FFFFh, the value of the sensor counter becomes FFFFh, but the CTSUSR.SCOVF flag does not become 1.

When this bit is set to 0 and the AJFEN bit is set to 1, each measurement result is used to determine “touch” or “non-touch” without a majority vote based on the measurement values, and the results are stored in the CTSUAJRR.TJR0 to TJR3 flags. When the number of SUCLK is three and the judgment condition is “two or more” (CTSUAJCR.JC[1:0] bits = 01b), a majority vote is performed based on the judgment results, and the result is stored in the FJR flag (JMM: judgment majority mode).

DTCLESS Bit (Data Transfer Request Disable)

When this bit is set to 1, the CTSUWR and CTSURD interrupts are not generated. In addition, the state counter transitions from State 2 to State 3 without writing to the CTSUSO register. Set a value in the CTSUSO register before starting measurement. Also, the CTSUSR.DTSR flag does not become 1.

MTUCFEN Bit (Mutual Capacitance Calculation Enable)

When this bit is set to 1 while the CTSUCRA.MD1 bit is 1 (mutual capacitance method), the second measurement result becomes the difference from the first measurement value. When the first measurement is completed, the CTSUSR.DTSR flag does not become 1 and no CTSURD interrupt is generated. When the second measurement is completed, the DTSR flag becomes 1 and a CTSURD interrupt is generated.

If the result of subtracting the first measurement value from the second measurement value is less than 0000h, the CTSUSCNT.SC[15:0] bits become 0000h. If the first or second measurement value overflows, the CTSUSR.SCOVF flag becomes 1, but the difference is stored in the SC[15:0] bits.

AJINTC Bit (Automatic Judgment Interrupt Control)

This bit specifies the condition for generating a CTSUFN interrupt when the CTSUCRA.SNZ bit is set to 1 (measurement during snooze mode is enabled) and the CTSUOPT.AJFEN bit is 1 (automatic judgment function is enabled). In other conditions, a CTSUFN interrupt is output when measurement is completed, regardless of the value of this bit.

Under the above conditions, if this bit is set to 0, a CTSUFN interrupt is output when any channel is determined to be touched, but a snooze end request is output when all channels are determined to be non-touched. If this bit is set to 1, a CTSUFN interrupt is output regardless of whether the judgment result is touch or non-touch.

39.2.17 CTSU Sensor Counter Automatic Correction Table Access Register (CTSUSCNTACT)

Address(es): CTSU.CTSUSCNTACT 000A 0944h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SCNTACCOUNT[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
	SCNTACCOEFF[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	SCNTACCOEFF[15:0]	Sensor Counter Correction Coefficient Setting	Set the correction coefficient for the measured value specified in the SCNTACCOUNT[15:0] bits. The coefficient is a fixed-point number with 4 bits for the integer part and 12 bits for the decimal part.	R/W
b31 to b16	SCNTACCOUNT[15:0]	Sensor Counter Measurement Value Setting	Set the measurement value to be compared.	R/W

Note: Set this register while the CTSUCRA.STRT bit is 0.

This register is used to access the sensor counter automatic correction table n (n = 0 to 11). The table to be accessed is selected by the CTSUOPT.SCACTB[3:0] bits.
The sensor counter automatic correction tables are used when the CTSUOPT.CCOCFEN bit is 1 (CCO characteristics correction function is enabled).
In the sensor counter automatic correction table n, set the n-th measurement value to be compared with measured value and the correction coefficient for the measurement value. The n-th measurement value should be larger than the (n-1)-th measurement value. Also, set the 0th measurement value to 0000h and 11th measurement value to FFFFh.
Writing to the SCNTACCOUNT[15:0] bits automatically increments the value of the CTSUOPT.SCACTB[3:0] bits to allow access to the next table.

39.2.18 CTSU Multi-Clock Automatic Correction Table n (CTSUMCACTn) (n = 1 to 3)

Address(es): CTSU.CTSUMCACT1 000A 094Ch, CTSU.CTSUMCACT2 000A 0950h, CTSU.CTSUMCACT3 000A 0954h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	OFFSETCOEFFn[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
	—	—	—	—	—	—	SOn[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SOn[9:0]	Sensor Offset Adjustment	These bits adjust the input offset current of sensor CCO for multi-clock n.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	OFFSETCOEFFn[15:0]	Measured Value Offset Setting	Set the offset value used in the correction operation for multi-clock n with a signed integer.	R/W

Note: Set this register while the CTSUCRA.STRT bit is 0.

These registers are tables for correcting the measured values used in the multi-clock correction function.
These registers are used when the CTSUOPT.MCACFEN bit is 1 (multi-clock correction function is enabled).
When majority mode is enabled (CTSUOPT.MAJIRIMD bit = 1) in multi-scan mode (CTSUCRA.MD0 bit = 1), update these registers each time a CTSUWR interrupt occurs.
When majority mode is enabled (CTSUOPT.MAJIRIMD bit = 1) in single scan mode (CTSUCRA.MD0 bit = 0), it is not necessary to update these registers during measurement if they were set before the measurement.
If majority mode is not used (CTSUOPT.MAJIRIMD bit = 0), the CTSUMCACT2 and CTSUMCACT3 registers and the CTSUMCACT1.SO1[9:0] bits are not used. Set the correction values for each clock to the CTSUMCACT1.OFFSETCOEFF1[15:0] and CTSUSO.SO[9:0] bits each time a CTSUWR interrupt occurs. For SUCLK0, set the OFFSETCOEFF1[15:0] bits to 0000h.

SOn[9:0] Bits (Sensor Offset Adjustment)

These bits are the same function as the CTSUSO.SO[9:0] bits. SO1[9:0] bits to SO3[9:0] bits correspond to multi-clock 1 to 3 respectively. The CTSUSO.SO[9:0] bits correspond to multi-clock 0.

OFFSETCOEFFn[15:0] Bits (Measured Value Offset Setting)

These bits are used to set the offset value corresponding to multi-clock 1 to 3. There is no bits corresponding to multi-clock 0.

39.2.19 CTSU Automatic Judgment Control Register (CTSUAJCR)

Address(es): CTSU.CTSUAJCR 000A 0958h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AJBMAT[3:0]				AJMMAT[3:0]				—	—	JC[1:0]		—	—	—	BLINI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	THOT[7:0]								TLOT[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	TLOT[7:0]	Non-Touch Judgment Criterion Setting	These bits specify the number of consecutive “non-touch” detections before a “non-touch” is reported. If the set value is m, the CTSUAJRR.TJRn flag becomes 0 when “non-touch” is detected m + 1 times in a row.	R/W
b15 to b8	THOT[7:0]	Touch Judgment Criterion Setting	These bits specify the number of consecutive “touch” detections before a “touch” is reported. If the set value is m, the CTSUAJRR.TJRn flag becomes 1 when “touch” is detected m + 1 times in a row.	R/W
b16	BLINI	Baseline Initialization	0: Perform baseline operations 1: Initialize the results of baseline operations*1	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21 to b20	JC[1:0]	Judgment Condition Setting*2	b21 b20 0 0: Judges as “touch” when there is one or more touch judgments 0 1: Judges as “touch” when there are two or more touch judgments 1 0: Judges as “touch” when there are three or more touch judgments 1 1: Judges as “touch” only when there are four touch judgments	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	AJMMAT[3:0]	Measured Value Moving Average Number Setting	These bits specify the number of moving averages of the measured value. If the set value is n, the smoothing factor is $1/2^n$. The setting range is 0000b to 1011b.	R/W
b31 to b28	AJBMAT[3:0]	Baseline Average Number Setting	These bits specify the number of baseline averages. If the set value is n, the number of average is 2^{n+1}. If 0000b is set, the baseline is not updated.	R/W

Note: Set this register while the CTSUCRA.STRT bit is 0.
Note 1. To initialize the calculation results, it is necessary to perform measurement with this bit set to 1.
Note 2. These bits can only be set when performing multi-clock measurement. Set these bits to 00b when not performing multi-clock measurement or when using the majority mode (CTSUOPT.MAJIRIMD bit = 1).

This register is used to set baseline calculations, moving average calculations, and touch/non-touch judgment criteria.

39.2.20 CTSU Threshold Register (CTSUAJTHR)

Address(es): CTSU.CTSUAJTHR 000A 095Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AJTHH[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
	AJTHL[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	AJTHL[15:0]	Lower Threshold Setting	Set the non-touch judgment threshold as a relative value from the baseline. The value is a signed integer between –32768 and 32767.	R/W
b31 to b16	AJTHH[15:0]	Upper Threshold Setting	Set the touch judgment threshold as a relative value from the baseline. The value is a signed integer between –32768 and 32767.	R/W

Note: Set this register when the measurement state is State 0 or State 2.

This register is used to set the thresholds that are the basis for touch/non-touch judgment.
Set the absolute value of the AJTHH[15:0] bits to be larger than the absolute value of the AJTHL[15:0] bits.
If the value of this register is to be set when a CTSUWR interrupt occurs in multi-scan mode (CTSUCRA.MD0 bit = 1), save the value of this register in RAM when a CTSURD interrupt occurs.

39.2.21 CTSU Moving Average Result Register (CTSUAJMMAR)

Address(es): CTSU.CTSUAJMMAR 000A 0960h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AJMMAR[26:11]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AJMMAR[10:0]											—	AJMMATI[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	AJMMATI[3:0]	Moving Average Count	These bits indicate the number of moving averages for the current measurement. Set these bits to 0000b at the first measurement.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b5	AJMMAR[26:0]	Moving Average Result	These bits store the moving average calculation result for the measured value. The value is a fixed-point number with 16 bits for the integer part and 11 bits for the decimal part.	R/W

Note: Set this register when the measurement state is State 0 or State 2.

This register is used to store the internal state of the moving average operation.
In multi-scan mode (the CTSUCRA.MD0 bit = 1), save the value of this register in RAM each time a CTSURD interrupt occurs, and write it back when a CTSUWR interrupt occurs when measuring the same channel again.

39.2.22 CTSU Baseline Average Intermediate Result Register (CTSUAJBLACT)

Address(es): CTSU.CTSUAJBLACT 000A 0964h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	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: Set this register when the measurement state is State 0 or State 2.

This register is used to store intermediate value during the baseline averaging operation. The value is a fixed-point number with 16 bits for the integer part and 16 bits for the decimal part.
In multi-scan mode (the CTSUCRA.MD0 bit = 1), save the value of this register in RAM each time a CTSURD interrupt occurs, and write it back when a CTSUWR interrupt occurs when measuring the same channel again.

39.2.23 CTSU Baseline Average Result Register (CTSUAJBLAR)

Address(es): CTSU.CTSUAJBLAR 000A 0968h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AJBLAR[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
	AJBLAC[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	AJBLAC[15:0]	Baseline Average Count	These bits indicates the current count value for the baseline average operation.	R/W
b31 to b16	AJBLAR[15:0]	Baseline Average Result	These bits store the baseline average calculation result.	R/W

Note: Set this register when the measurement state is State 0 or State 2.

This register is used to store the result of the baseline averaging operation.
In multi-scan mode (the CTSUCRA.MD0 bit = 1), save the value of this register in RAM each time a CTSURD interrupt occurs, and write it back when a CTSUWR interrupt occurs when measuring the same channel again.

39.2.24 CTSU Automatic Judgment Result Register (CTSUAJRR)

Address(es): CTSU.CTSUAJRR 000A 096Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SJCCR[7:0]								—	—	—	FJR	TJR3	TJR2	TJR1	TJR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TJR0	Touch Judgment Result Flag 0	The judgment result when using random pulse or SUCLK0 is stored. 0: Non-touch 1: Touch	R/W
b1	TJR1	Touch Judgment Result Flag 1	The judgment result when using SUCLK1 is stored. 0: Non-touch 1: Touch	R/W
b2	TJR2	Touch Judgment Result Flag 2	The judgment result when using SUCLK2 is stored. 0: Non-touch 1: Touch	R/W
b3	TJR3	Touch Judgment Result Flag 3	The judgment result when using SUCLK3 is stored. 0: Non-touch 1: Touch	R/W
b4	FJR	Final Judgment Result Flag	The final judgment result on multi-clock measurement is stored. If multi-clock measurement is not specified, the same value as the TJR0 flag is stored. 0: Non-touch 1: Touch	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	SJCCR[7:0]	Remaining Number of Consecutive Detections	These bits indicate how many more consecutive touch or non-touch detections will invert the value of the TJRn flag.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register when the measurement state is State 0 or State 2.

This register is used to store the touch/non-touch judgment result.
In multi-scan mode (the CTSUCRA.MD0 bit = 1), save the value of this register in RAM each time a CTSURD interrupt occurs, and write it back when a CTSUWR interrupt occurs when measuring the same channel again.

39.2.25 CTSU A/D Converter Connection Control Register (CTSUADCC)

Address(es): CTSU.CTSUADCC 000A 0700h

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	CTADCS	TSCAP Voltage A/D Conversion	0: Does not measure the voltage of the TSCAP pin 1: Connect the TSCAP pin with AN008	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the signal connection between the CTSU and the A/D converter.

CTADCS Bit (TSCAP Voltage A/D Conversion)

This bit is used to measure the voltage of the TSCAP pin with the A/D converter.
When the CTSUCRA.DCMODE bit is set to 1 (current measurement mode) and the CTSUCRA.DCBACK bit is set to 1, the voltage at the TSm pin (m = 0 to 35) can also be measured.

(a) Register settings when measuring the voltage of the TSCAP pin

CTSUCRA.PUMPON bit = 0 or 1 (determined by the VCC voltage)
CTSUCRA.PON bit = 1
CTSUCRA.CSW bit = 1
CTSUCRA.DCBACK bit = 0

(b) Register settings when measuring the voltage of the TSm pin

CTSUCRA.PUMPON bit = 0 or 1 (determined by the VCC voltage)
CTSUCRA.PON bit = 1
CTSUCRA.CSW bit = 0
CTSUCRA.MD0 bit = 0
CTSUCRA.MD1 bit = 0
CTSUCRA.LOAD[1:0] bits = 01b
CTSUCRA.DCMODE bit = 1
CTSUCRA.DCBACK bit = 1
CTSUCHACA.CHACm bit or CTSUCHACB.CHACm bit = 1
CTSUCHTRCA.CHTRCm bit or CTSUCHTRCB.CHTRCm bit = 0
CTSUMCH.MCH0[5:0] bits = m
CTSUCALIB.DRV bit = 1
CTSUSO register = 0000 03C0h

39.3 Operation

39.3.1 Principles of Measurement Operation

Figure 39.7 shows the measurement circuit.

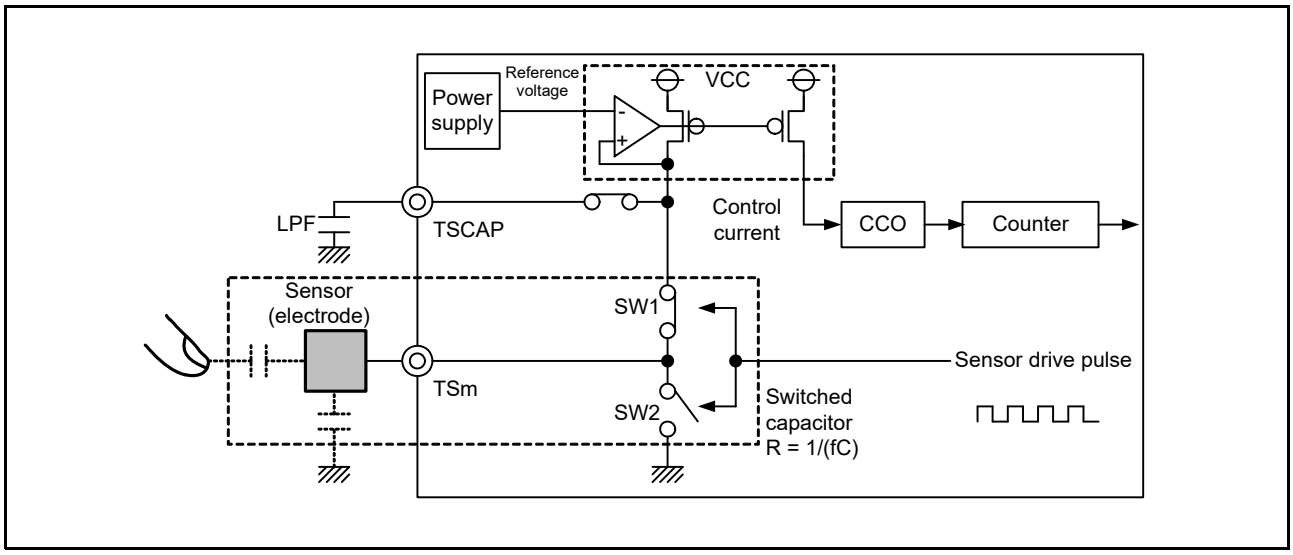


Figure 39.7 Measurement Circuit (m = 0 to 35)

Figure 39.8 to Figure 39.10 explain the electrostatic capacitance measurement operation principles of the CTSU current frequency conversion. The operation is as follows:

(1) The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 39.8).

(2) The charged capacitance is discharged by turning SW1 off and SW2 on (Figure 39.9).

Current flows to the switched capacitor by switching between charging and discharging. At this point, if a finger is in close proximity, the capacitance and the flowing current change. A clock is generated by supplying the control current, which is proportional to the amount of current flowing through the switched capacitor, from the circuit that generates the TSCAP power supply (LDO) to the current controlled oscillator (CCO). The counter measures the clock frequency that changes depending on whether a finger is in close proximity. The software uses the value read from the counter to determine contact with a finger (Figure 39.10).

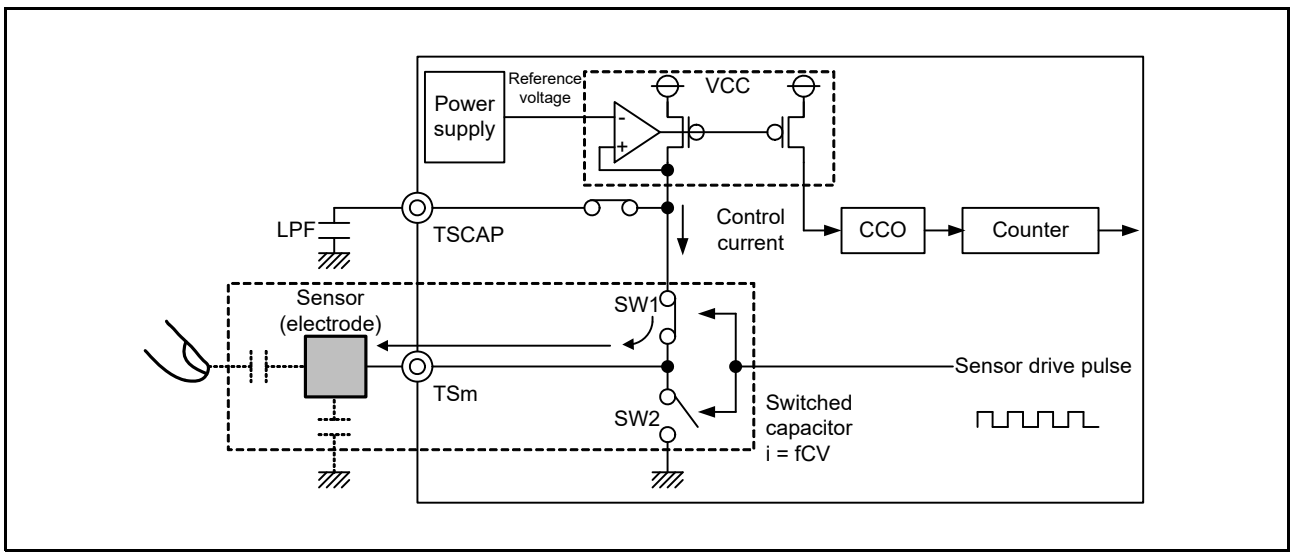


Figure 39.8 Charging Operation (m = 0 to 35)

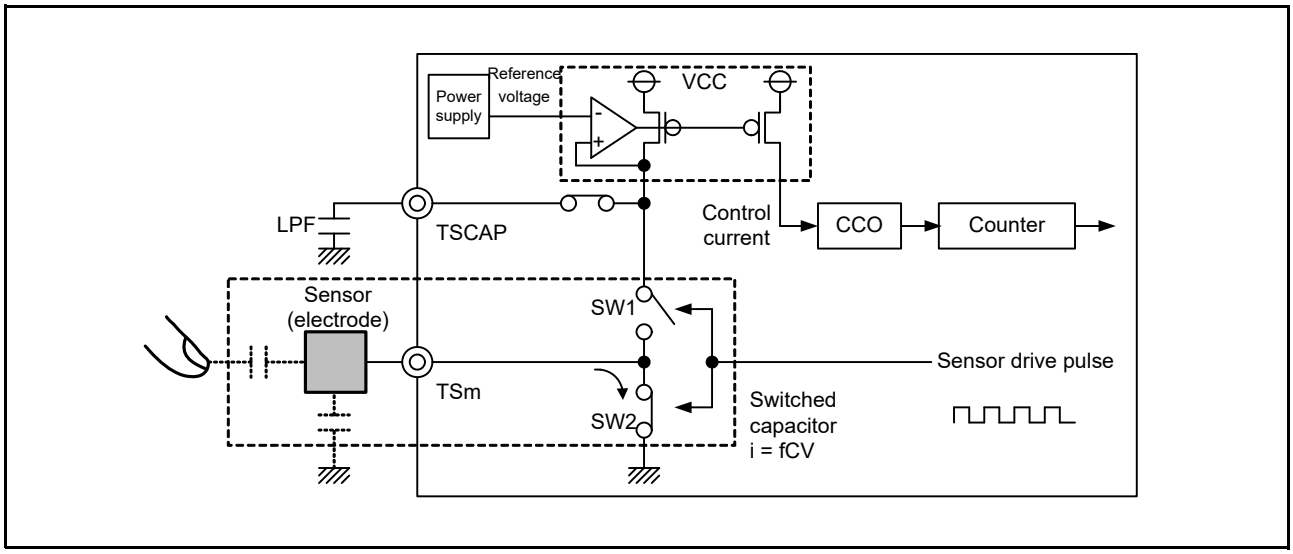


Figure 39.9 Discharging Operation (m = 0 to 35)

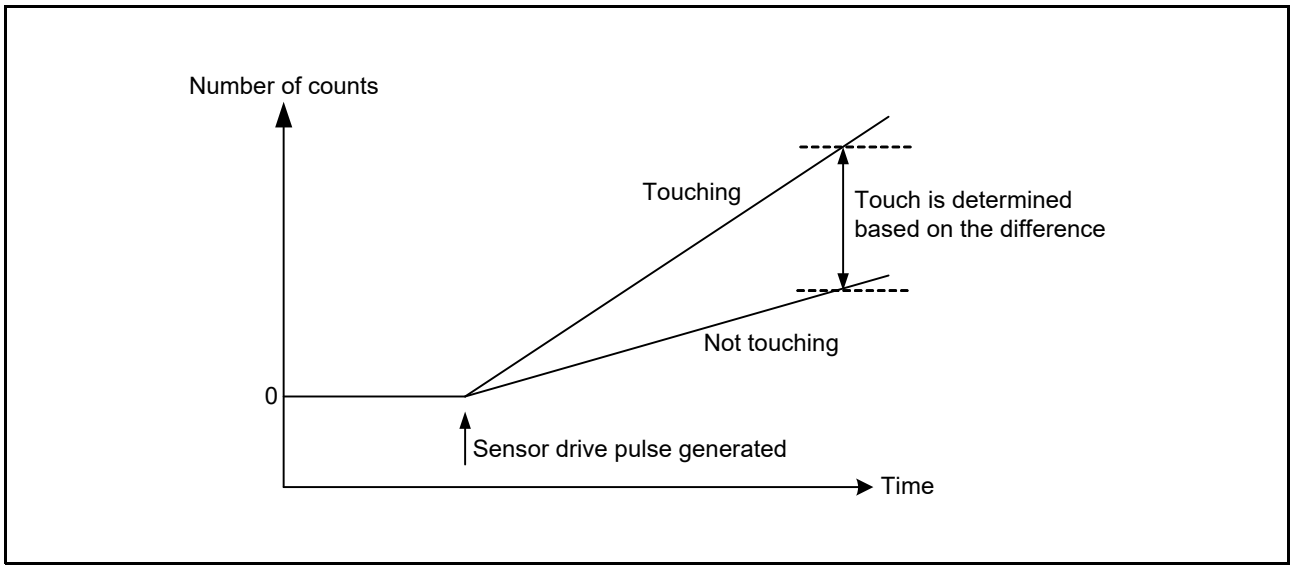


Figure 39.10 Change in Measured Value When Finger is Touching and Not Touching

39.3.2 Initial Setting Flowchart

Figure 39.11 shows the flowchart for CTSU initial setting.

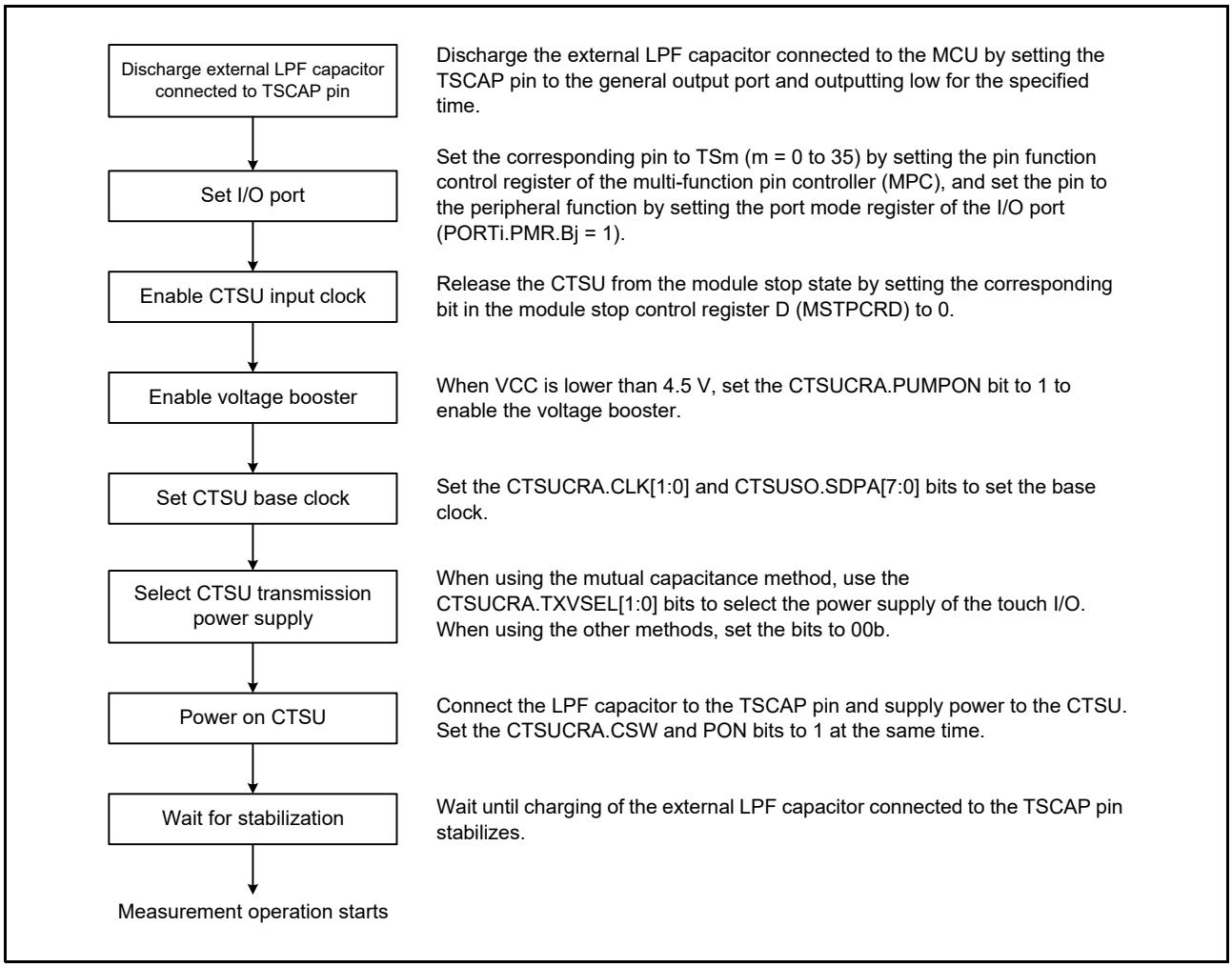


Figure 39.11 CTSU Initial Setting Flowchart

Figure 39.12 shows the flowchart for stopping CTSU operation and setting to the standby state.

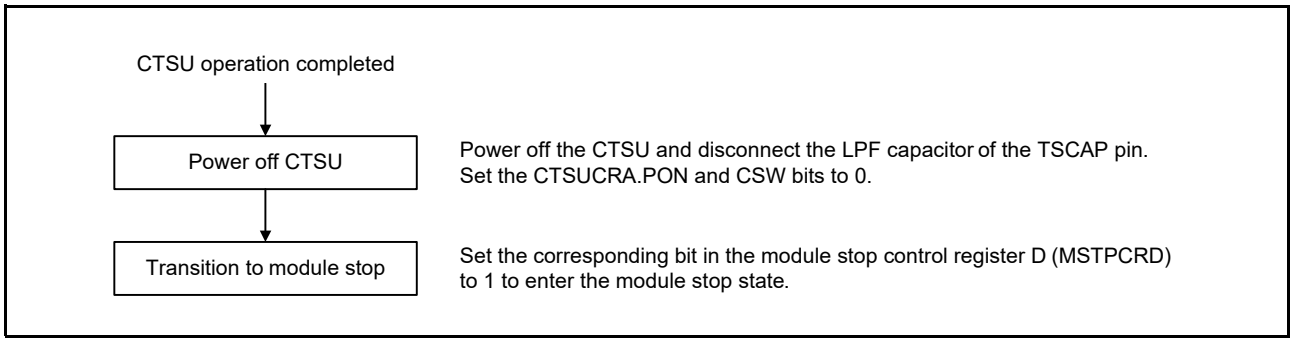


Figure 39.12 CTSU Stopping Flowchart

When restarting operation after it has been stopped, follow the initial setting flowchart shown in Figure 39.11.

39.3.3 Measurement State

The measurement state counter of the CTSU status register (CTSUSR) indicates the current measurement state. The measurement state is common to all four modes. Figure 39.13 shows state operation transitions.

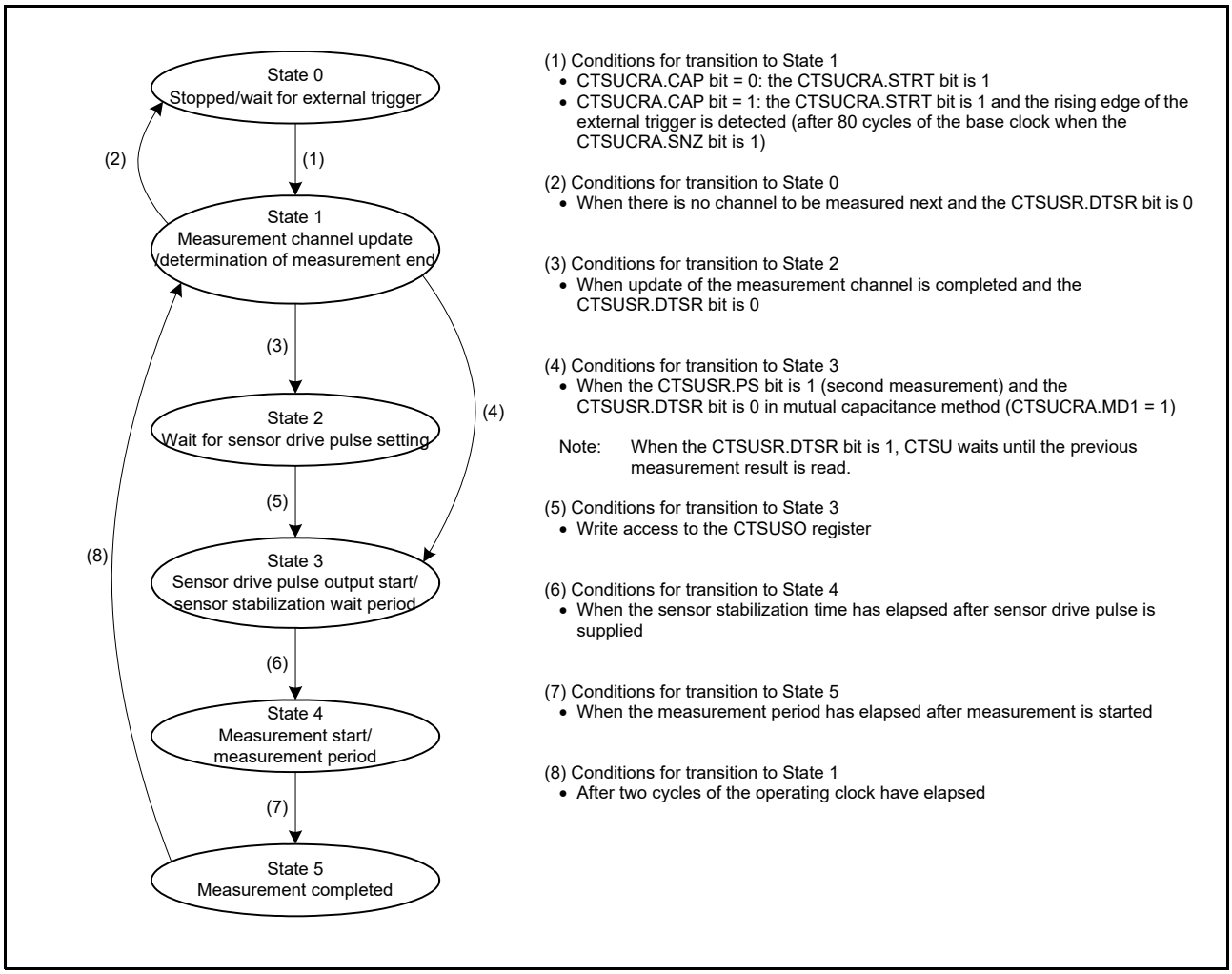


Figure 39.13 State Operation Transitions

The state counter transitions to State 0 when all of the specified measurement channels are measured.
The CTSUCRA.STRT bit is set to 0 by hardware when a software trigger is used. When an external trigger is used, the value of 1 is retained, and the CTSU waits for the next trigger.
When operation is forced to stop during measurement or the trigger wait state, by a simultaneous 0 write to the CTSUCRA.STRT bit and a 1 write to the CTSUCRA.INIT bit, the state transitions to State 0 and measurement stops.
If the channel to be measured is not set in the CTSUMCH, CTSUCHACx, and CTSUCHTRCx registers (x = A, B), a CTSUFN interrupt occurs immediately after a transition to State 1, and then the state transitions to State 0.

39.3.4 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods. Figure 39.14 illustrates these methods.

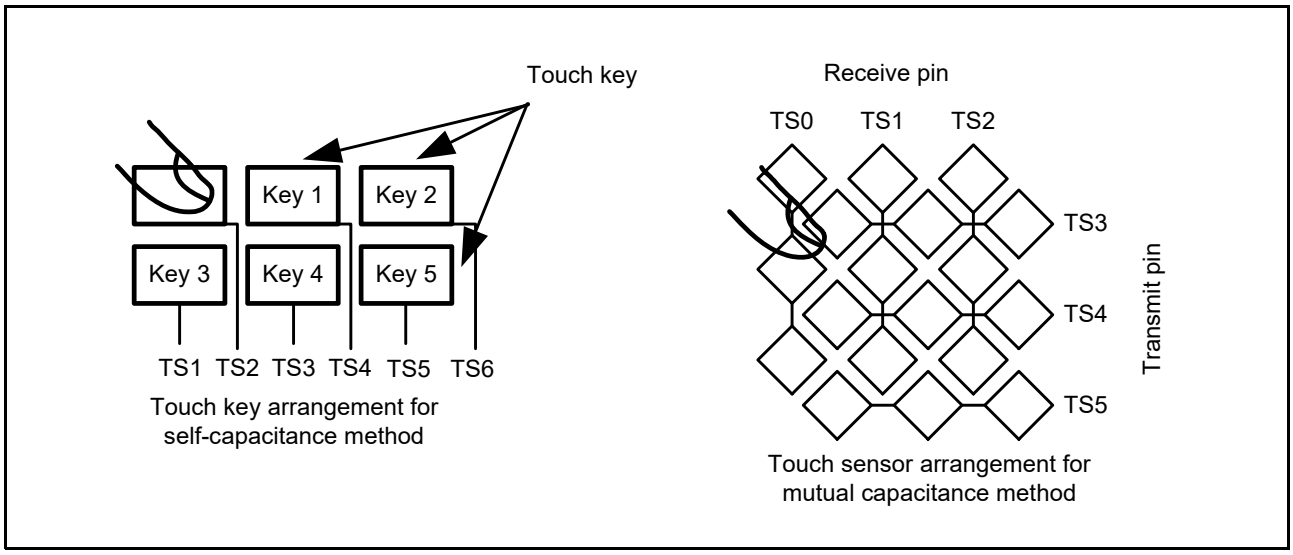


Figure 39.14 Overview of Self-Capacitance Method and Mutual Capacitance Method

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity.
In the mutual capacitance method, the capacitance between two opposing electrodes (transmit and receive pins) is measured.

39.3.4.1 Self-Capacitance Method Operation

In self-capacitance method, one measurement pin is assigned to each sensor, and the capacitance on each channel is measured. Scan mode and sensor drive pulse can be selected.

Figure 39.15 shows the software flow and an operation example.

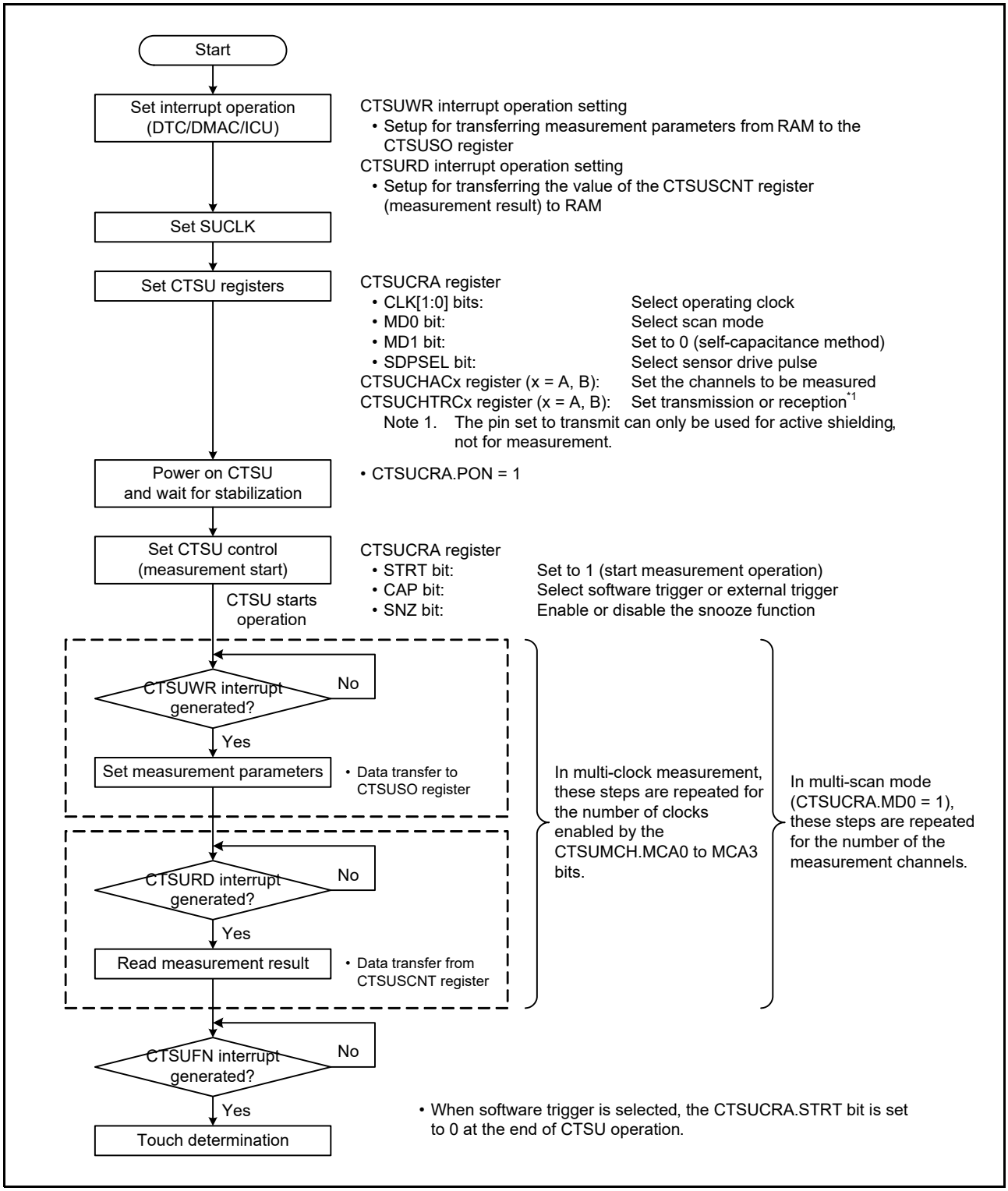


Figure 39.15 Software Flow and Operation Example of Self-Capacitance Method

39.3.4.2 Mutual Capacitance Method Operation

In mutual capacitance method, measurement is performed during the high period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, at the rising and falling edges. The difference between the data of these two measurements is used to determine whether or not the electrode is touched, thus achieving higher touch sensitivity. Scan mode and sensor drive pulse can be selected.

Capacitances are measured for all combinations of transmit and receive pins. Figure 39.16 shows the software flow and an operation example.

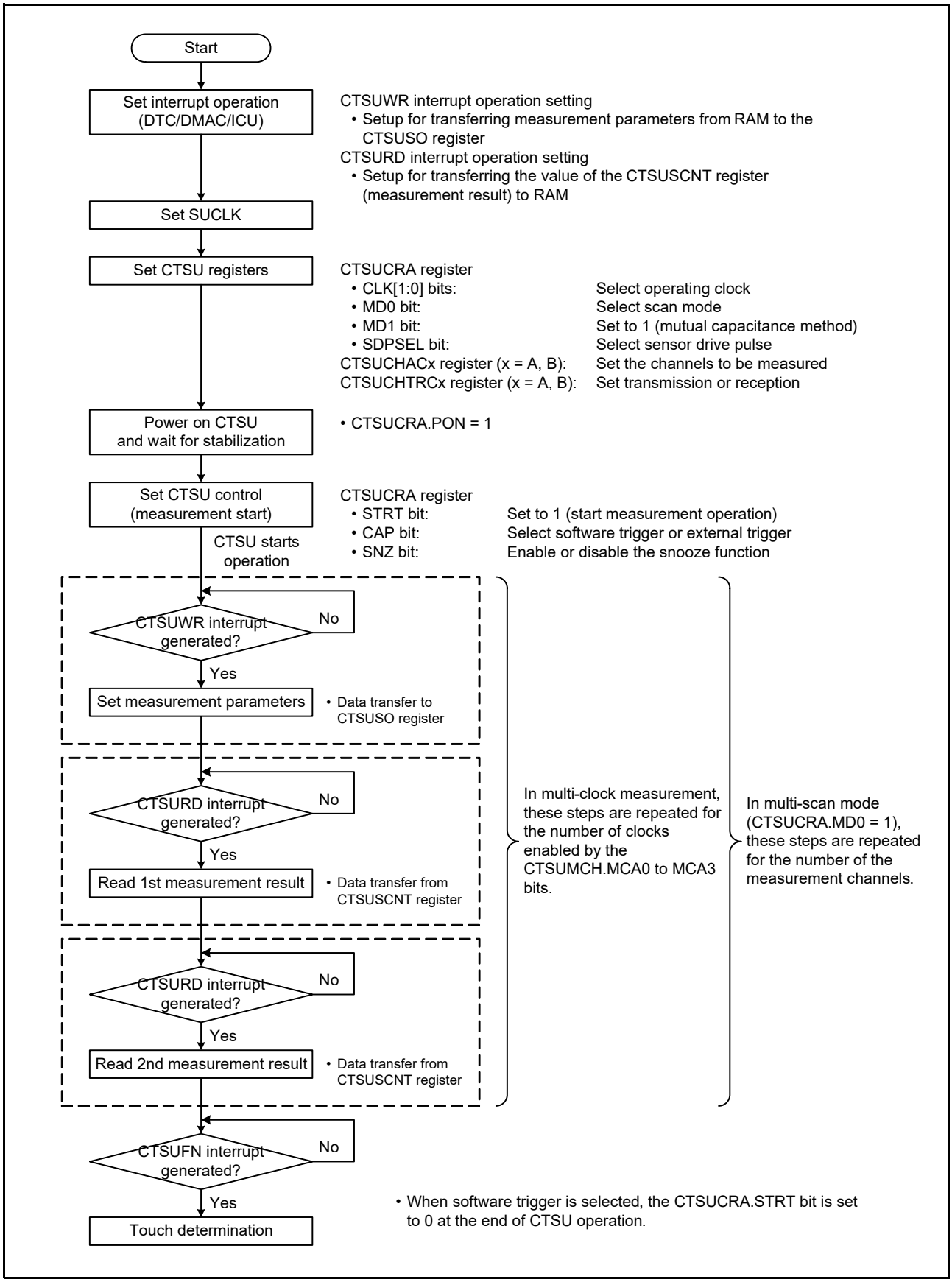


Figure 39.16 Software Flow and Operation Example of Mutual Capacitance Method

39.3.5 Scan Mode

CTSU has two scan modes: single scan mode and multi-scan mode.

(1) Single Scan Mode

Measurement is performed only once using the combination of the receive channel specified by the CTSUMCH.MCH0[5:0] bits and the transmit channel specified by the MCH1[5:0] bits.

(2) Multi-Scan Mode

Among the channels set as measurement targets in the CTSUCHACx register (x = A, B), measurements are performed once for each combination of receive and transmit channels set by the CTSUCHTRCx register.
Both receive and transmit channels are used in ascending order of channel number. The transmit channel is switched after the measurements for each combination of a transmit channel and all receive channels are completed.

39.3.6 Multi-Clock Measurement

In multi-clock measurement, multiple clocks with different frequencies are sequentially switched. Multi-clock measurement is valid only in high resolution pulse mode (SUCLK mode).
Specify the clock (SUCLKn) used for measurement using the CTSUMCH.MCAn bits (n = 0 to 3). The frequency of each clock can be set with the CTSUSUCLKA and CTSUSUCLKB registers.
Measurements are performed in ascending order starting from SUCLK0. When measurements with all clocks are completed, measurements for the next channel are started.

39.3.7 Automatic Judgment Function

The CTSU has a function to automatically judge whether a finger has touched or not.
The CTSUFN interrupt in the automatic judgment function can be used to return from the snooze mode to the normal operating mode, and the snooze end request can be used to return from the snooze mode to the software standby mode.
Figure 39.17 shows the configuration diagram of the automatic judgment function.

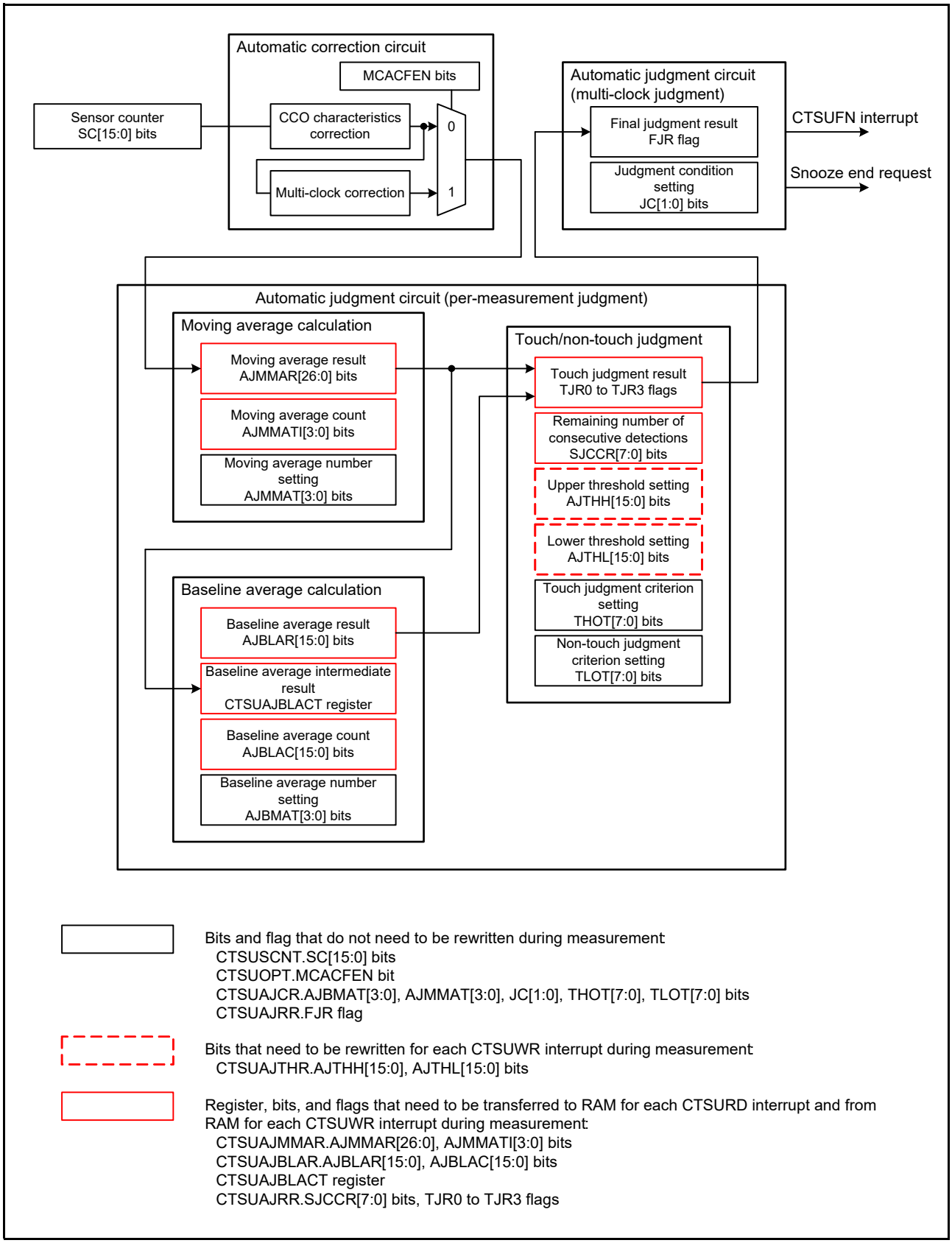
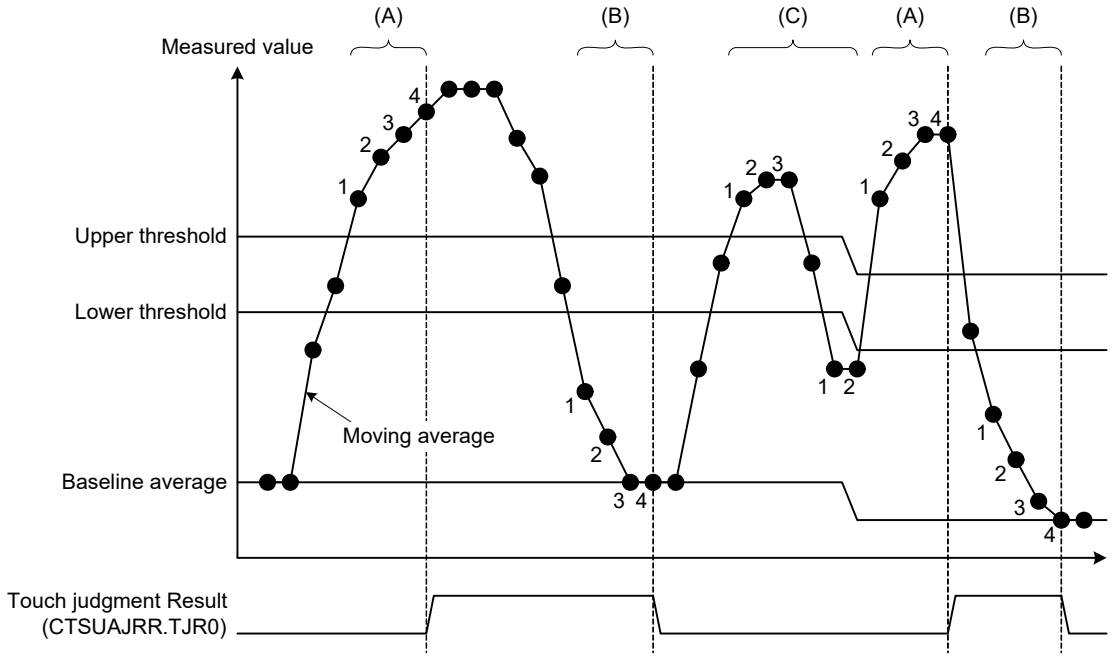


Figure 39.17 Configuration Diagram of Automatic Judgment Function

39.3.7.1 Operation of the Automatic Judgment Function

The measured value subject to automatic judgment is the value of the CTSUSCNT.SC[15:0] bits. The moving average and the baseline average are calculated from the measured values, and “touch” or “non-touch” is judged from the difference between them. The number of moving averages and the number of baseline averages can be set by the AJMMAT[3:0] and AJBMAT[3:0] bits in the CTSUAJCR register, respectively.
Figure 39.18 shows an operation example of the automatic judgment function.



(A)	(B)	(C)
When a finger touches the touch sensor, the moving average result begins to increase. When the measured value exceeds the upper threshold four times in a row, it is judged as “touch” and the TJR0 flag is set to 1. After that, the baseline average calculation stops.	When the finger leaves the touch sensor, the moving average result begins to decrease. When the measured value falls below the lower threshold four times in a row, it is judged as “non-touch” and the TJR0 flag is set to 0. After that, the baseline average calculation resumes.	Even if the moving average fluctuates due to noise to the touch sensor, etc., the judgment is not change unless the upper threshold is exceeded four times in a row. The same applies to the lower threshold.

Figure 39.18 Operation Example of Automatic Judgment Function (when both the CTSUAJCR.THOT[7:0] and TLOT[7:0] bits are 03h)

(1) Touch Judgment

When the difference between the moving average and the baseline average exceeds the upper threshold (CTSUAJTHR.AJTHH[15:0] bits) for a consecutive number of times specified by the CTSUAJCR.THOT[7:0] bits (or falls below if the upper threshold is negative), the CTSUAJRR.TJRn flag is set to 1 (n = 0 to 3).
When multi-clock measurement is not specified, the value of the TJR0 flag is directly reflected in the CTSUAJRR.FJR flag. When multi-clock measurement is specified, the FJR flag is set to 1 when the values of the TJR0 to TJR3 flags meet the judgment condition specified by the CTSUAJCR.JC[1:0] bits.

(2) Non-Touch Judgment

When the difference between the moving average and the baseline average falls below the lower threshold (CTSUAJTHR.AJTHL[15:0] bits) for a consecutive number of times specified by the CTSUAJCR.TLOT[7:0] bits (or exceeds if the lower threshold is negative), the TJRn flag is set to 0 (n = 0 to 3).
When multi-clock measurement is not specified, the value of the TJR0 flag is directly reflected in the CTSUAJRR.FJR flag. When multi-clock measurement is specified, the FJR flag is set to 1 when the values of the TJR0 to TJR3 flags no longer satisfy the judgment condition specified by the JC[1:0] bits.

(3) Baseline Average Calculation

When the TJRn flag is set to 1, the baseline average calculation is stopped from the next measurement (n = 0 to 3). When the baseline average calculation stops, the intermediate result (CTSUAJBLACT register) and the average count (CTSUAJBLAR.AJBLAC[15:0] bits) are initialized.
When the TJRn flags is set to 0, the baseline average calculation restarts with the next measurement.
When multi-clock measurement is specified, the baseline average calculation is stopped or restarted for each TJR0 to TJR3 flag.

39.3.7.2 Self-Capacitance Method Operation

Figure 39.19 shows the software flow of self-capacitance multi-scan mode using automatic judgment function.
The interrupt operation settings can be omitted only when using single scan mode and not performing multi-clock measurement. In this case, set the CTSUOPT.DTCLESS bit to 1 (data transfer request is disabled) and set the CTSUSO and CTSUAJTHR registers before starting measurement.

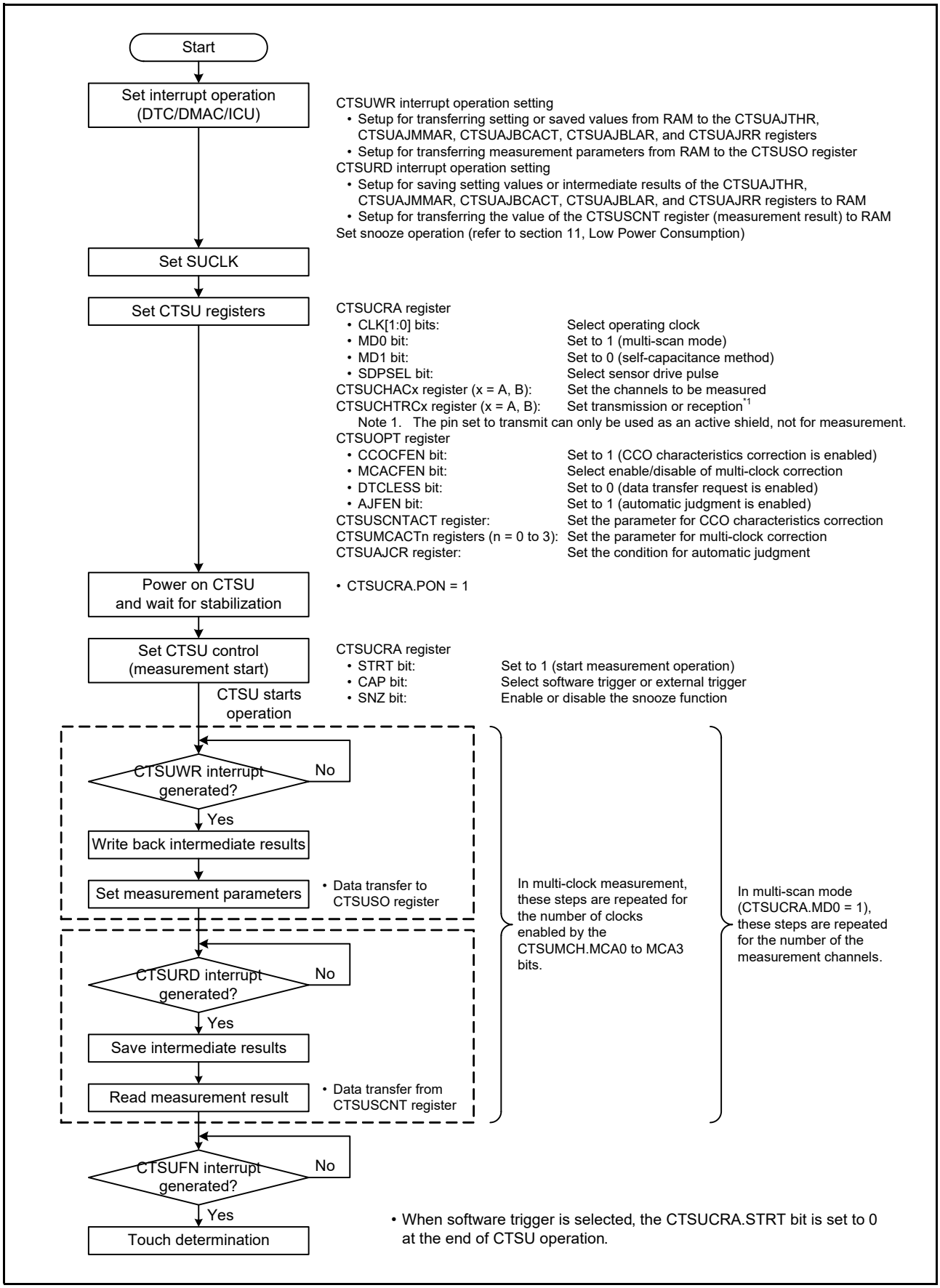


Figure 39.19 Software Flow of Self-Capacitance Multi-Scan Mode Using Automatic Judgment Function

39.3.7.3 Mutual Capacitance Method Operation

Figure 39.20 shows the software flow of mutual capacitance multi-scan mode using automatic judgment function.

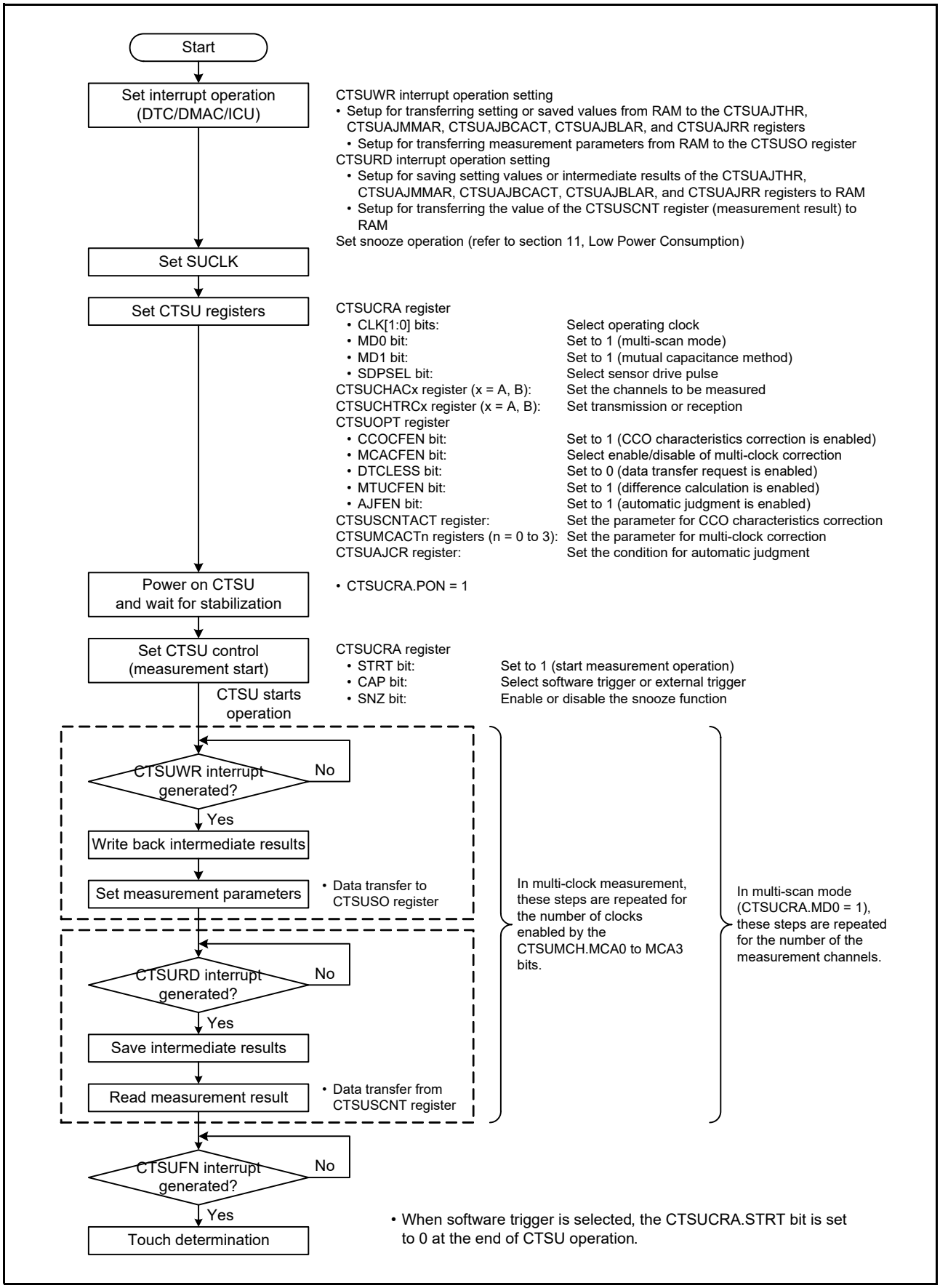


Figure 39.20 Software Flow of Mutual Capacitance Multi-Scan Mode Using Automatic Judgment Function

39.3.8 Multiple Electrode Connection Function

The multiple electrode connection (MEC) function is a function that makes multiple electrodes appear as if they are one electrode. This function allows multiple electrodes to be measured at once.

When the CTSUCRA.MD1 bit is set to 0 (self-capacitance method), the MD bit is set to 0 (single scan mode), the CTSUCALIB.TSOD bit is set to 1 (selected by the IOCSEL bit), and the IOCSEL bit is set to 0 (capacitance measurement mode using multiple electrode connection function), the sensor drive pulses are output from all TSm pins with the corresponding CTSUCHACx.CHACm bit set to 1 (measurement target) and the corresponding CTSUCHTRCx.CHTRCm bit set to 0 (reception) (x = A, B; m = 0 to 35). Therefore, the total capacitance of all electrodes connected to each TSm pin can be measured.

When “touch” is detected using this function, it is not possible to determine which electrode’s capacitance has changed. Set the TSOD bit to 0 (capacitance measurement mode) and measure the capacitance of each electrode individually.

39.4 Interrupts

The CTSU supports the following three interrupts:

- Register setting request interrupt (CTSUWR)
- Measurement result read request interrupt (CTSURD)
- Measurement end interrupt (CTSUFN)

39.4.1 Register Setting Request Interrupt (CTSUWR)

Store the settings for each measurement channel in the RAM, and set up the DTC or ICU transfer associated with the CTSUWR interrupt in advance. The CTSUWR interrupt is output when the transition from State 1 to State 2 is completed.
Write the settings for the selected channel from the RAM to the CTSUSO register (Figure 39.21). Write access to the CTSUSO register controls the transition to State 3.

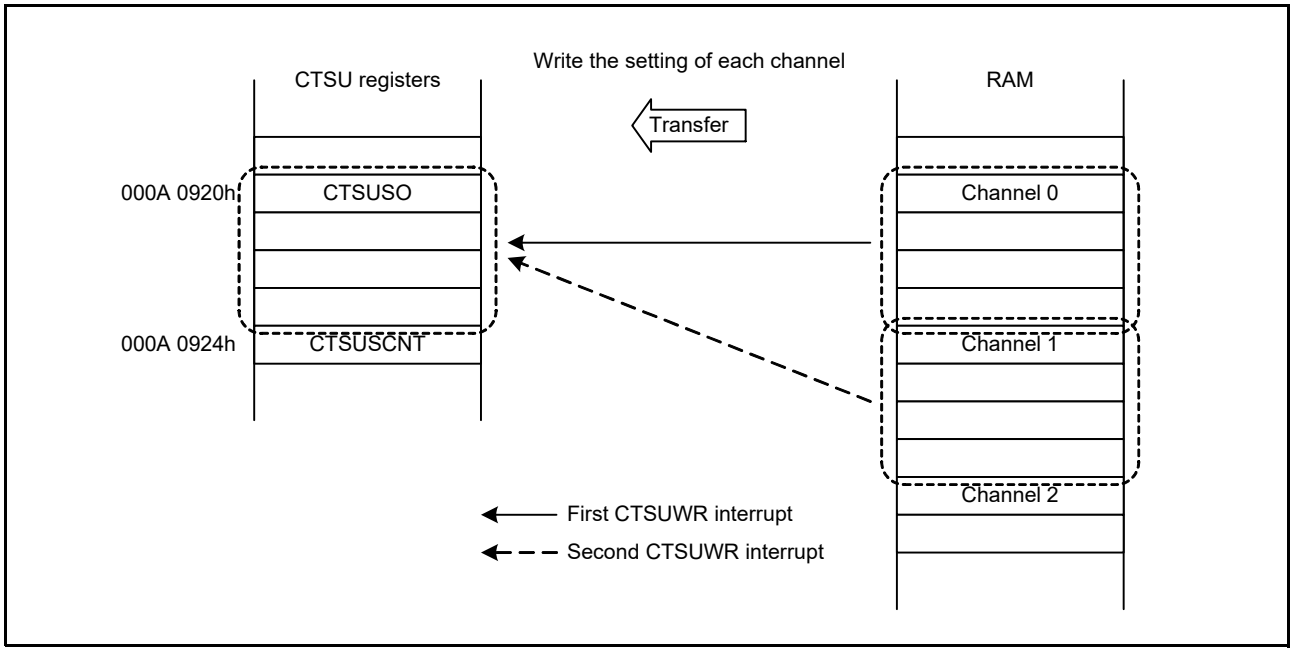


Figure 39.21 Example of DTC Transfer Operation Using the CTSUWR Interrupt

When performing DTC transfer by the CTSUWR interrupt, set as follows.

- Transfer destination address: CTSUSO register address
- Operation at the transfer destination address: Transfer 4 bytes of data once with a single interrupt (the address is fixed).
- Transfer source address: CTSUSO register data storage address for the lowest channel in the settings prepared on the RAM
- Operation at the transfer source address: Transfer 4 bytes of data once with a single interrupt (the address of the first byte is continued from the previous interrupt handling.
- Number of transfers per interrupt: Specify the number of measurements.

39.4.2 Measurement Data Read Request Interrupt (CTSURD)

Set up the DTC or ICU transfer associated with the CTSURD interrupt in advance. A CTSURD interrupt is output when the transition from State 5 to State 1 is completed after the measurement of a channel is completed. Read the measurement result from the CTSUSCNT register (Figure 39.22). Reading the CTSUSCNT register causes CTSU to transition to State o or State 2.

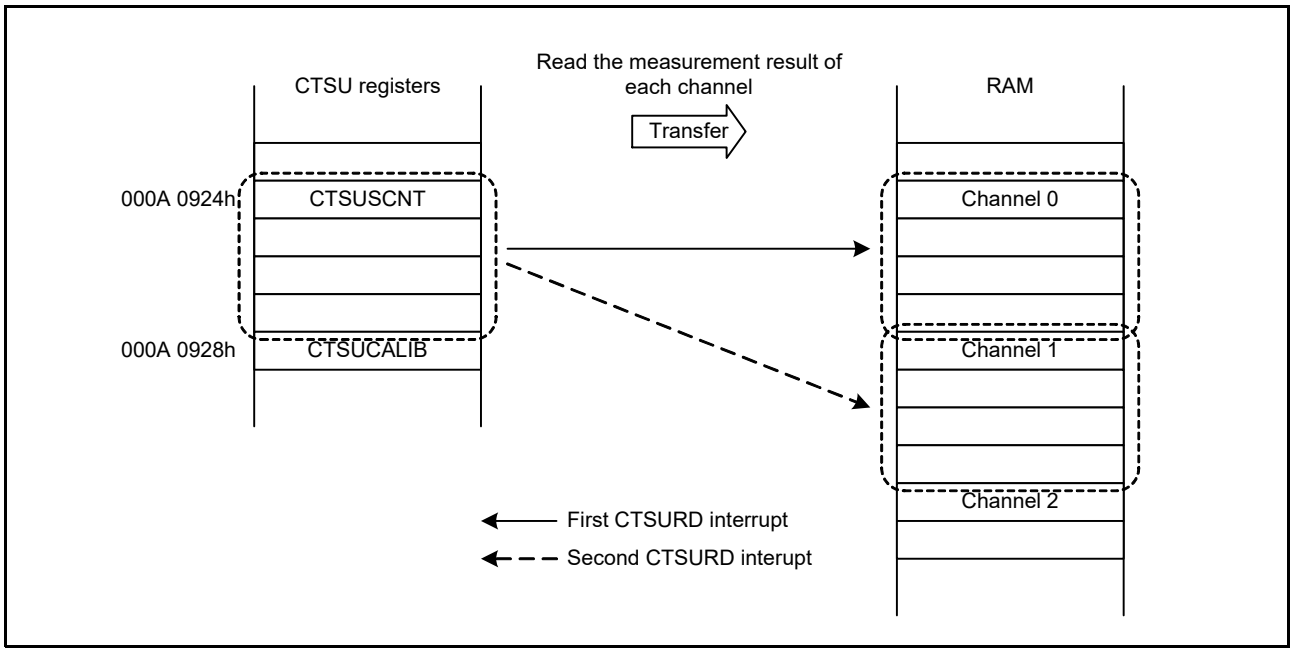


Figure 39.22 Example of DTC Transfer Operation Using the CTSURD Interrupt

When performing DTC transfer by CTSURD interrupt, set as follows.

- Transfer source address: CTSUSCNT register address
- Operation at the transfer source address: Transfer 4 bytes of data once with a single interrupt (the start address is fixed).
- Transfer destination address: Measurement result data storage address for the lowest channel prepared on the RAM
- Operation at the transfer destination address: Transfer 4 bytes of data once with a single interrupt (the start address is continued from the previous interrupt handling).
- Number of transfers per interrupt: Specify the number of measurements.

39.4.3 Measurement End Interrupt (CTSUFN)

When the transition from State 1 to State 0 is completed after the measurement of all enabled channels is completed, a CTSUFN interrupt is output.
However, if the CTSUCRA.SNZ bit is set to 1 (measurement during snooze mode is enabled), the CTSUOPT.AJFEN bit is 1 (automatic judgment function is enabled), and the CTSUOPT.AJINTC bit is 0 (snooze end request is output when “non-touch” is detected), the CTSUFN interrupt is output only when at least one of the following conditions is met.

- When there is a channel that has been judged as “touch”
- When an over-voltage is detected during measurement of the last channel

Note that the MCU is not released from the snooze mode and returns to the software standby mode when an error other than those listed above occurs during measurement and both of the following conditions are met.

- The CTSUFN interrupt is selected as the interrupt for release from the snooze mode.
- All measurement results are judged as "non-touch".

If this creates a problem, use another interrupt as the trigger for exit from the software standby mode.

39.5 Snooze End Request

When the CTSUCRA.SNZ bit is set to 1 (measurement during snooze mode is enabled), the CTSUOPT.AJFEN bit is 1 (automatic judgment function is enabled), and the CTSUOPT.AJINTC bit is 0 (snooze end request is output when “non-touch” is detected), the snooze end request is output when all channels are judged as “non-touch”.
However, if an over-voltage is detected during measurement of the last channel, the CTSUFN interrupt is output instead of the snooze end request.
For details on snooze mode, refer to section 11, Low Power Consumption.

39.6 Usage Notes

39.6.1 Setting the Module Stop Function

Module stop control register D (MSTPCRD) is used to stop and start CTSU operations. With the value after a reset, CTSU 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.

39.6.2 Measurement Result Data (CTSUSCNT Resister)

Do not read the register during measurement. If read, the value cannot be guaranteed.

39.6.3 Software Trigger

When 10b (PCLKB/4) or 11b (PCLKB/8) is set to the CTSUCRA.CLK[1:0] bits, to restart measurement by writing 1 to the CTSUCRA.STRT bit after measurement is complete, wait for at least following cycles to elapse after an interrupt occurs, and then write 1 to the CTSUCRA.STRT bit.

- When the CTSUCRA.CLK[1:0] bits are 10b: At least 3 cycles
- When the CTSUCRA.CLK[1:0] bits are 11b: At least 7 cycles

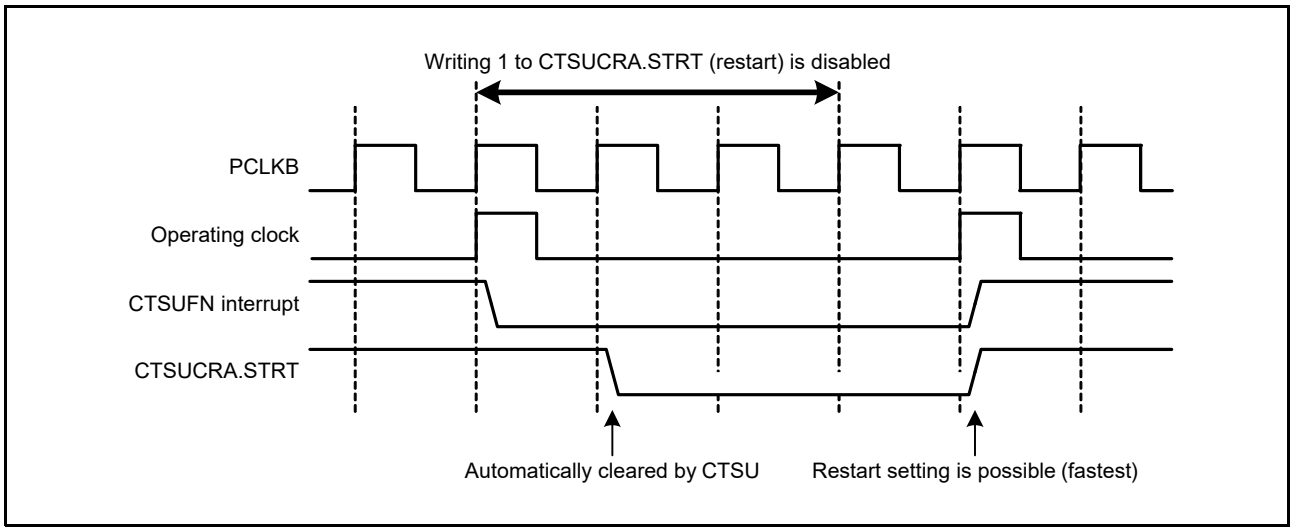


Figure 39.23 Notes on Restarting Measurement

39.6.4 External Trigger

If an external trigger is input during the measurement period, the trigger is ignored. The next external trigger is enabled after 1 cycle of the operating clock after a CTSUFN interrupt occurs.
To exit the external trigger mode, write 0 in the CTSUCRA.STRT bit and 1 in the CTSUCRA.INIT bit at the same time (forced stop).

39.6.5 Constraints on Forced Stops

To forcibly stop during measurement, write 0 in the CTSUCRA.STRT bit and 1 in the CTSUCRA.INIT bit at the same time. The operation stops and the internal control register is initialized.
Initialization with the INIT bit initializes the following registers in addition to initializing the internal measurement state.

- CTSUMCH register
- CTSUSR register
- CTSUSCNT register

If operation is forcibly stopped, an interrupt request may occur depending on the internal state. After the forced stop, also stop and disable the DTC or ICU.
If a DTC transfer is stopped for some reason, also perform forced stop and initialization for CTSU.

39.6.6 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.
The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (CTSUCRA.CSW bit = 1) to establish a connection.

39.6.7 Setting of the Sampling Cycle on Spread Spectrum

Set the CTSUCRA.CLK[1:0] and CTSUSO.SDPA[7:0] bits so that the sampling cycle of the sensor drive pulse when the spread spectrum function is enabled (CTSUCRB.SOFF = 0) is less than 1/4 of the sensor drive pulse period.

39.6.8 Notes on Measurement Operation (CTSUCRA.STRT Bit = 1)

Do not stop the peripheral module clock or change the port settings of the measurement pins (TSm and TSCAP pins) during the measurement operation (CTSUCRA.STRT bit = 1).
If such a setting is made, forcibly stop the operation (CTSUCRA.STRT bit = 0, CTSUCRA.INIT bit = 1), set the CTSUCRA.PON and CTSUCRA.CSW bits to 0 at the same time, set the CTSUCRA.SNZ bit to 1, and then start from the initial setting.

39.6.9 Transmit Pin in Self-Capacitance Method

In self-capacitance method, the transmit pin cannot be used for measurement. A pulse with the same phase as the measurement pulse is output from the transmission terminal. Use it as an active shield on the board.
Also, do not set multiple pins as transmission at the same time in self-capacitance method.

40. 12-Bit A/D Converter (S12ADE)

40.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. Up to 25 channel analog inputs, temperature sensor output, and internal reference voltage are selectable for conversion.

The 12-bit A/D converter converts a maximum of 25 selected channels of analog inputs, temperature sensor output, and internal reference voltage, which have been selected, into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 25 arbitrarily selected channels are converted only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 25 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 25 channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.

In group scan mode, the conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. When group-A priority control is selected along with operation as described above, if a request to start scanning for group A is received during A/D conversion for group B, the conversion operation for group B is discontinued and the conversion for group A starts, which is given priority.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.

It is prohibited to simultaneously select both temperature sensor output and internal reference voltage. Perform A/D conversion independently for the temperature sensor output or the internal reference voltage.

The external pin input (VREFH0) or the analog reference voltage (AVCC0) is selectable as the reference voltage on the high-potential side. The external pin input (VREFL0) or the analog reference voltage (AVSS0) is selectable as the reference voltage on the low-potential side.

This IP has a compare function (window A and window B). This function is used to specify the high-side reference value and low-side reference value for window A and window B, respectively. When the A/D-converted value of the selected channel meets the comparison conditions, the ELC event (S12ADWMELC/S12ADWUMELC) is output according to the event conditions (A or B, A and B, A exor B). Furthermore, the comparator operation to compare the A/D-converted value with the low-side reference value is also enabled.

The A/D data storage buffer is a ring buffer consisting of 16 buffers to sequentially store A/D converted data.

Table 40.1 lists the specifications of the 12-bit A/D converter and Table 40.2 lists the functions of the 12-bit A/D converter. Figure 40.1 shows a block diagram of the 12-bit A/D converter.

Table 40.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	One unit
Input channels	Up to 25 channels
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.7 µs per channel (ADCCR.CCS bit = 0), 0.5 µs per channel (ADCCR.CCS bit = 1) (when A/D conversion clock ADCLK = 64 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, 1:2, 2:1, 4:1, 8:1 ADCLK is set using the clock generation circuit.
Data registers	• 25 registers for analog input, 1 for A/D-converted data duplication in double trigger mode • One register for temperature sensor output • One register for internal reference voltage • One register for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*2 in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register.
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of up to 25 channels 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 of up to 25 channels arbitrarily selected. • Group scan mode: Analog inputs of up to 25 channels arbitrarily selected, are divided into group A and group B, and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. • Group scan mode (when group A is given priority): If a group A trigger is input during A/D conversion on group B, the A/D conversion on group B is stopped and A/D conversion is performed on group A. Restart (rescan) of A/D conversion on group B after completion of A/D conversion on group A can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the general PWM timer (GPTW), the event link controller (ELC). • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# pin.
Functions	• Variable sampling state count • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • Automatic clear function of A/D data registers • Compare function (window A and window B) • 16 ring buffers when the compare function is used

Table 40.1 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/D scan end interrupt request (S12ADI0) can be generated on completion of single scan. • In double trigger mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of double scan. • In group scan mode, an A/D scan end interrupt request (S12ADI0) can be generated on completion of group A scan, whereas an A/D scan end interrupt request (GBADI) for group B can be generated on completion of group B scan. • When double trigger mode is selected in group scan mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of double scan of group A, whereas A/D scan end interrupt request (GBADI) specially for group B can be generated on completion of group B scan. • The S12ADI0 and GBADI interrupts can activate the DMA controller (DMAC) and the data transfer controller (DTC).
Event link function	• An ELC event is generated on completion of scans other than group B scan in group scan mode. • An ELC event is generated on completion of group B scan in group scan mode. • An ELC event is generated on completion of all scans. • Scan can be started by a trigger output by the ELC. • An ELC event is generated according to the event conditions of the window compare function in single scan mode.
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 (addition zero to three times)
4-bit extension: 16-time conversion (addition 15 times)
Note 3. See section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

Table 40.2 Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation
Analog input channels			AN000 to AN008, AN016 to AN031, temperature sensor output, internal reference voltage
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger	ADTRG0#	Enabled
	Synchronous trigger	Compare match with GPTW0.GTADTRA	GTADTRA0N
		Compare match with GPTW0.GTADTRB	GTADTRB0N
		Compare match with GPTW1.GTADTRA	GTADTRA1N
		Compare match with GPTW1.GTADTRB	GTADTRB1N
		Compare match with GPTW2.GTADTRA	GTADTRA2N
		Compare match with GPTW2.GTADTRB	GTADTRB2N
		Compare match with GPTW0.GTADTRA, or compare match with GPTW0.GTADTRB	GTADTRA0N or GTADTRB0N
		Compare match with GPTW1.GTADTRA, or compare match with GPTW1.GTADTRB	GTADTRA1N or GTADTRB1N
		Compare match with GPTW2.GTADTRA, or compare match with GPTW2.GTADTRB	GTADTRA2N or GTADTRB2N
		ELC trigger	Enabled
Interrupt			S12ADI0, GBADI interrupt
Setting of the module stop function*1			MSTPCRA.MSTPA17 bit

Note 1. See section 11, Low Power Consumption for details.

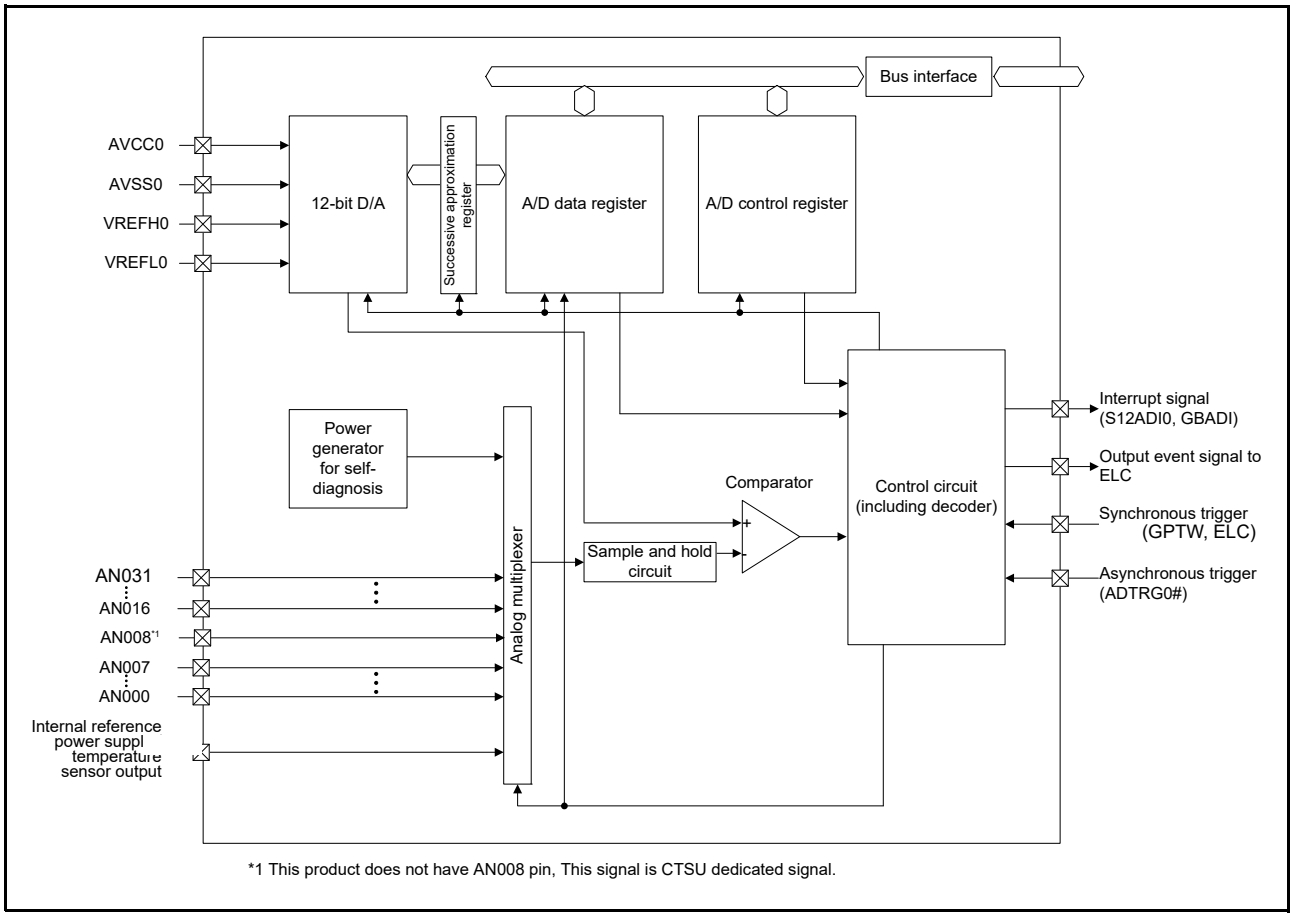


Figure 40.1 Block Diagram of 12-Bit A/D Converter

Table 40.3 lists the input pins of the 12-bit A/D converter.

Table 40.3 Pin Configuration of 12-Bit A/D Converter

Pin Name	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block ground pin
VREFH0	Input	Reference power supply pin
VREFL0	Input	Reference power supply ground pin
AN000 to AN007, AN016 to AN031	Input	Analog input pins 0 to 7, analog input pins 16 to 31
ADTRG0#	Input	External trigger input pin for starting A/D conversion

40.2 Register Descriptions

40.2.1 A/D Data Registers y (ADDRy) (y = 0 to 8, 16 to 31), A/D Data Duplication Register (ADDBLDR), 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.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.ADDR24 0008 9050h, S12AD.ADDR25 0008 9052h, S12AD.ADDR26 0008 9054h, S12AD.ADDR27 0008 9056h, S12AD.ADDR28 0008 9058h, S12AD.ADDR29 0008 905Ah, S12AD.ADDR30 0008 905Ch, S12AD.ADDR31 0008 905Eh, S12AD.ADDBLDR 0008 9018h, 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 8, 16 to 31) 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 ADTSDR register is a 16-bit read-only register that stores the A/D conversion results of the temperature sensor output.
The ADOCDR register is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (addition once, twice, three, or 15 times)
- Settings of the average mode enable bit (ADADC.AVEE) (addition or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left 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

- Flush-right 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.
- Flush-left 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 twice or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)

The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  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.

40.2.2 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) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 40.2.11, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Flush-right 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.
- Flush-left 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 40.4 Self-Diagnosis Status Description

Bits 15 and 14 for flush-right format setting Bits 1 and 0 for flush-left format setting	Self-diagnosis status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the reference voltage × 1/2 has been executed.
11b	Self-diagnosis using the reference voltage has been executed.

Note: For details of self-diagnosis, see section 40.2.11, A/D Control Extended Register (ADCER).

40.2.3 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	—	ADHSC	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables GBADI interrupt generation upon group B scan completion. 1: Enables GBADI interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select *1	0: A/D conversion is started by synchronous trigger. 1: A/D conversion is started by asynchronous trigger.	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by synchronous or asynchronous trigger.	R/W
b10	ADHSC	A/D Conversion Select	0: High-speed conversion 1: Low-current conversion	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan completion. 1: Enables S12ADI0 interrupt generation upon scan completion.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process. 1: Starts A/D conversion process.	R/W

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
After a high-level signal is input to the external pin (ADTRG0#), write 1 to both the TRGE and EXTRG bits in ADCSR and change the signals of ADTRG0# to low. Thus the falling edge of ADTRG0# is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLKB.

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 40.5 shows selection of the channel for double triggered operation.
When double trigger mode is selected, channel selection using the ADANSA0 and ADANSA1 registers is invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.
When double trigger mode is used, do not select A/D conversion for the self-diagnosis function, temperature sensor output, and internal reference voltage (temperature sensor output and internal reference voltage can be selected for A/D conversion for group B in group scan mode). The DBLANS[4:0] bits should be set while the ADST bit is 0. They should

not be set simultaneously when 1 is written to the ADST bit.
To enter A/D-converted value addition/average mode while double trigger mode is set, the channel selected by the DBLANS[4:0] bits should be selected in the ADANSA0 and ADANSA1 registers.

Table 40.5 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000b	AN000	10000b	AN016	11000b	AN024
00001b	AN001	10001b	AN017	11001b	AN025
00010b	AN002	10010b	AN018	11010b	AN026
00011b	AN003	10011b	AN019	11011b	AN027
00100b	AN004	10100b	AN020	11100b	AN028
00101b	AN005	10101b	AN021	11101b	AN029
00110b	AN006	10110b	AN022	11110b	AN030
00111b	AN007	10111b	AN023	11111b	AN031
01000b	AN008				

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (GBADI) in group scan mode.

DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.
When double trigger mode is selected, channel selection using the ADANSA0 and ADANSA1 registers is invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. Do not generate an asynchronous or software trigger.
The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is generated not upon completion of the first conversion but upon completion of the second conversion.
In continuous scan mode, double trigger mode should not be selected.
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.

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.

ADHSC Bit (A/D Conversion Select)

The ADHSC bit sets the operating mode of A/D conversion. When modifying this bit, set the 12-bit converter to the standby state. For the procedure for modifying the ADHSC bit, see section 40.8.10, ADHSC Bit Rewriting Procedure.

ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI0) in scans except for group B scan in group scan mode.
With double trigger mode deselected, the S12ADI0 interrupt is generated after the first scan is completed if the ADIE bit

is set to 1.
With double trigger mode selected, the S12ADI0 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.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 25 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.
In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 25 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.
In group scan mode, A/D conversion is performed for the analog inputs (group A) of 25 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 25 channels selected with the ADANSB0 and ADANSB1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.
When selecting group scan mode, different channels and triggers should be selected for group A and group B.
When selecting the temperature sensor output or internal reference voltage, select single scan mode, and deselect all the channels selected with the ADANSA0 and ADANSA1 registers before performing A/D conversion. When A/D conversion of the selected temperature sensor output or internal reference voltage is completed, A/D conversion is stopped.
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.

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B trigger is detected and A/D conversion of group B is started.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B is restarted.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of group B is started.

[Clearing conditions]

- 0 is written by software.

- The A/D conversion of all the selected channels, the temperature sensor output, or the internal reference voltage is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group A trigger is detected during group B A/D conversion and the scanning of group B is stopped.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of group B started by a resumption trigger is completed.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) the ADGSPCR.GBRP bit is set to 1 and the scanning of group B by a trigger is completed.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

40.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): S12AD.ADANSA0 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ANSA008	ANSA007	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	0: AN000 to AN008 are not subjected to conversion. 1: AN000 to AN008 are subjected to conversion.	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b8	ANSA008			R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSA0 register selects analog input channels for A/D conversion among AN000 to AN008. In group scan mode, this register selects group A channels.

ANSA0n Bit (n = 00 to 08) (A/D Conversion Channel Select)

The ANSA0n bit selects analog input channels for A/D conversion among AN000 to AN008. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA008 bit corresponds to AN008.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA0n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA0n bit should be set while the ADCSR.ADST bit is 0.

40.2.5 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
	ANSA115	ANSA114	ANSA113	ANSA112	ANSA111	ANSA110	ANSA109	ANSA108	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	0: AN016 to AN031 are not subjected to conversion. 1: AN016 to AN031 are subjected to 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
b8	ANSA108			R/W
b9	ANSA109			R/W
b10	ANSA110			R/W
b11	ANSA111			R/W
b12	ANSA112			R/W
b13	ANSA113			R/W
b14	ANSA114			R/W
b15	ANSA115			R/W

The ADANSA1 register selects analog input channels for A/D conversion among AN016 to AN031. In group scan mode, group A channels are to be selected.

ANSA1n Bit (n = 00 to 15) (A/D Conversion Channel Select)

The ANSA1n bit select analog input channels for A/D conversion among AN016 to AN031. The channels to be selected and the number of channels can be arbitrarily set. The ANSA100 bit corresponds to AN016 and the ANSA115 bit corresponds to AN031.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA1n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA1n bit should be set while the ADCSR.ADST bit is 0.

40.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): S12AD.ADANSB0 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ANSB008	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	0: AN000 to AN008 are not subjected to conversion. 1: AN000 to AN008 are subjected to conversion.	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b8	ANSB008			R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSB0 register selects analog input channels for A/D conversion among AN000 to AN008 in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (n = 00 to 08) (A/D Conversion Channel Select)

The ANSB0n bit selects analog input channels for A/D conversion among AN000 to AN008 in group B when group scan mode is selected. The ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB008 bit corresponds to AN008.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB0n bit should be set while the ADCSR.ADST bit is 0.

40.2.7 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
	ANSB115	ANSB114	ANSB113	ANSB112	ANSB111	ANSB110	ANSB109	ANSB108	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	0: AN016 to AN031 are not subjected to conversion. 1: AN016 to AN031 are subjected to 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
b8	ANSB108			R/W
b9	ANSB109			R/W
b10	ANSB110			R/W
b11	ANSB111			R/W
b12	ANSB112			R/W
b13	ANSB113			R/W
b14	ANSB114			R/W
b15	ANSB115			R/W

The ADANSB1 register selects analog input channels for A/D conversion among AN016 to AN031 in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

ANSB1n Bit (n = 00 to 15) (A/D Conversion Channel Select)

The ANSB1n bit selects analog input channels for A/D conversion among AN016 to AN031 in group B when group scan mode is selected. The ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB100 bit corresponds to AN016 and the ANSB115 bit corresponds to AN031.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB1n bit should be set while the ADCSR.ADST bit is 0.

40.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0)

Address(es): S12AD.ADADS0 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADS008	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN000 to AN008 is not selected. 1: A/D-converted value addition/average mode for AN000 to AN008 is selected.	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b8	ADS008			R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADADS0 register selects the channels 00 to 08 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (n = 00 to 08) (A/D-Converted Value Addition/Average Channel Select)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA0.ANSA0n bit or ADCSR.DBLANS[4:0] bits and ADANSB0.ANSB0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register. The ADS0n bit should be set while the ADCSR.ADST bit is 0.

40.2.9 A/D-Converted Value Addition/Average Function 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
	ADS115	ADS114	ADS113	ADS112	ADS111	ADS110	ADS109	ADS108	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	0: A/D-converted value addition/average mode for AN016 to AN031 is not selected. 1: A/D-converted value addition/average mode for AN016 to AN031 is selected.	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
b8	ADS108			R/W
b9	ADS109			R/W
b10	ADS110			R/W
b11	ADS111			R/W
b12	ADS112			R/W
b13	ADS113			R/W
b14	ADS114			R/W
b15	ADS115			R/W

The ADADS1 register selects the channels 16 to 31 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 (n = 00 to 15) (A/D-Converted Value Addition/Average Channel Select)

When the ADS1n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA1.ANSA1n bit or ADCSR.DBLANS[4:0] bits and ADANSB1.ANSB1n 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 ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register. The ADS1n bit should be set while the ADCSR.ADST bit is 0.

Figure 40.2 shows a scanning operation sequence in which both the ADS002 and ADS006 bits are set to 1. It is assumed that addition mode is selected (ADADC.AVEE = 0), the addition count is set to three times (ADADC.ADC[2:0] = 011b), and the channels AN000 to AN008 are selected (ADANSA0.ANSA0n = FFh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively four times (addition 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 4 times and the added (integrated) value is stored in A/D data register 6. After conversion of AN008, the conversion operation is once again performed in the same sequence from AN000.

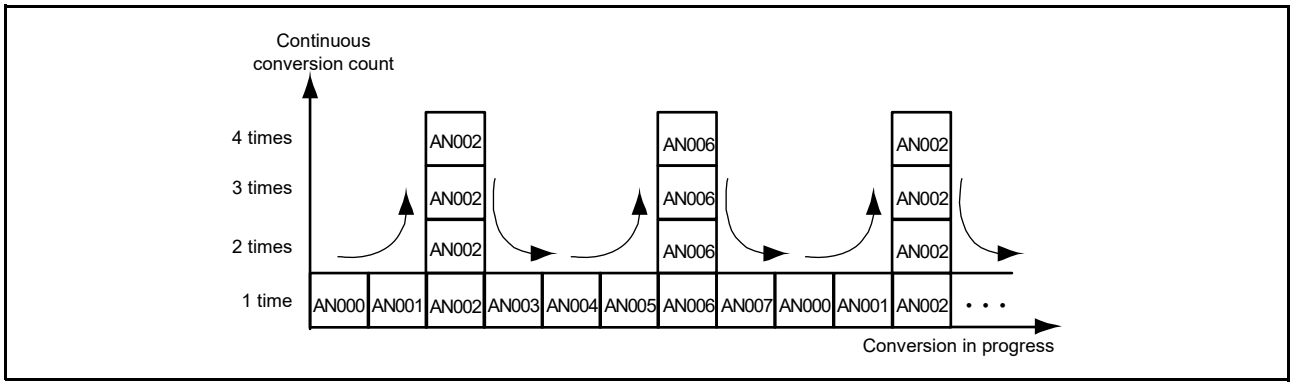


Figure 40.2 Scan Conversion Sequence with ADADC.ADC[2:0] = 011b, ADS002 = 1, and ADS006 = 1

40.2.10 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] = 010b) nor 16-time conversion (ADADC.ADC[2:0] = 101b).

The ADADC register sets the number of times addition is to proceed for the temperature sensor output, the internal reference voltage, and channels selected as being in A/D-converted value addition or average mode, and to select either addition or average mode for them.

ADC[2:0] Bits (Addition Count Select)

The ADC[2:0] bits set the number of times addition is to proceed as a common value for the temperature sensor output, the internal reference voltage, and channels for which A/D-converted value addition or average mode is selected, including those channels selected in double trigger mode (by the ADCSR.DBLANS[4:0] bits).
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b).
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for the temperature sensor output, the internal reference voltage, and the channels selected for which the addition or average mode of A/D conversion is selected, including those channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits).
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.
The AVEE bit should be set while the ADCSR.ADST bit is 0.

40.2.11 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 reference voltage x 1/2 for self-diagnosis.*1 1 1: Uses the reference voltage for self-diagnosis.*1	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Flush-right is selected for the A/D data register format. 1: Flush-left is selected for the A/D data register format.	R/W

Note 1. The reference voltage refers to the voltage on the pin selected in the ADHVREFCNT register.

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, ADTSDR, or ADOCDR register after any of these registers have been read by the CPU, DTC, or DMACA. 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)

These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference voltage × 1/2, and the reference voltage are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference voltage × 1/2, and the reference voltage is converted. 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 and B.
The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in the ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, 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, see section 40.2.1, A/D Data Registers y (ADDRy) (y = 0 to 8, 16 to 31), A/D Data Duplication Register (ADDBLDR), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR), section 40.2.2, A/D Self-Diagnosis Data Register (ADRD), section 40.2.25, A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), section 40.2.26, A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), section 40.2.33, A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), and section 40.2.34, A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB).

40.2.12 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]	A/D Conversion Start Trigger Select for Group B	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSTRGR register selects the A/D conversion start trigger.

TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.
When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh.
Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger may have no effect.
Table 40.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

TRSA[5:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, software trigger and asynchronous trigger cannot be used.

- 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.
Table 40.7 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 40.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits

Module	Source	Remarks	TRSB [5]	TRSB [4]	TRSB [3]	TRSB [2]	TRSB [1]	TRSB [0]
Trigger source deselection state			1	1	1	1	1	1
GPTW	GTADTRA0N	Compare match with GPTW0.GTADTRA	0	1	0	0	0	1
	GTADTRB0N	Compare match with GPTW0.GTADTRB	0	1	0	0	1	0
	GTADTRA1N	Compare match with GPTW1.GTADTRA	0	1	0	0	1	1
	GTADTRB1N	Compare match with GPTW1.GTADTRB	0	1	0	1	0	0
	GTADTRA2N	Compare match with GPTW2.GTADTRA	0	1	0	1	0	1
	GTADTRB2N	Compare match with GPTW2.GTADTRB	0	1	0	1	1	0
	GTADTRA0N or GTADTRB0N	Compare match with GPTW0.GTADTRA, or compare match with GPTW0.GTADTRB	0	1	1	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPTW1.GTADTRA, or compare match with GPTW1.GTADTRB	0	1	1	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPTW2.GTADTRA, or compare match with GPTW2.GTADTRB	0	1	1	0	1	1
ELC	ELCTRG00N		0	0	1	0	0	1
	ELCTRG01N		0	0	1	0	1	0
	ELCTRG00N or ELCTRG01N		0	0	1	0	1	1

Table 40.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits

Module	Source	Remarks	TRSA [5]	TRSA [4]	TRSA [3]	TRSA [2]	TRSA [1]	TRSA [0]
Trigger source deselection state			1	1	1	1	1	1
External pin	ADTRG0#	Input pin for the trigger	0	0	0	0	0	0
GPTW	GTADTRA0N	Compare match with GPTW0.GTADTRA	0	1	0	0	0	1
	GTADTRB0N	Compare match with GPTW0.GTADTRB	0	1	0	0	1	0
	GTADTRA1N	Compare match with GPTW1.GTADTRA	0	1	0	0	1	1
	GTADTRB1N	Compare match with GPTW1.GTADTRB	0	1	0	1	0	0
	GTADTRA2N	Compare match with GPTW2.GTADTRA	0	1	0	1	0	1
	GTADTRB2N	Compare match with GPTW2.GTADTRB	0	1	0	1	1	0
	GTADTRA0N or GTADTRB0N	Compare match with GPTW0.GTADTRA, or compare match with GPTW0.GTADTRB	0	1	1	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPTW1.GTADTRA, or compare match with GPTW1.GTADTRB	0	1	1	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPTW2.GTADTRA, or compare match with GPTW2.GTADTRB	0	1	1	0	1	1
ELC	ELCTRG00N		0	0	1	0	0	1
	ELCTRG01N		0	0	1	0	1	0
	ELCTRG00N or ELCTRG01N		0	0	1	0	1	1

40.2.13 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
	—	—	—	—	—	—	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Temperature sensor output A/D-converted value addition/average mode is not selected. 1: Temperature sensor output A/D-converted value addition/average mode is selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is not selected. 1: Internal reference voltage A/D-converted value addition/average mode is selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output is not performed. 1: A/D conversion of temperature sensor output is performed.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed. 1: A/D conversion of internal reference voltage is performed.	R/W
b15 to b10	—	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 and internal reference voltage.

TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D temperature sensor data register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is stored in the A/D temperature sensor data register (ADTSDR).
The TSSAD bit should be set while the ADCSR.ADST bit is 0.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in ADOCDR.
The OCSAD bit should be set while the ADCSR.ADST bit is 0.

TSSA Bit (Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output in single scan mode. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE and OCSA bits should all be set to 0 in single scan mode.
The TSSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the temperature sensor output, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The

sampling time should be 5 µs or longer.
Sampling starts after discharging is completed during A/D conversion of the temperature sensor output, an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage in single scan mode. When A/D conversion of the internal reference voltage is to be performed, set all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE bit and TSSA bit should be set to all 0 in single scan mode.
The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The sampling time should be 5 µs or longer.
Sampling starts after discharging is completed during A/D conversion of the internal reference voltage, so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

40.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 8, 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.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	0	1	1	0	1

The ADSSTRn register sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 64 MHz, one state is 15.625 ns. The initial value is 13 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. The lower-limit value for sampling time differs depending on the PCLKB to ADCLK frequency ratio.
Set a value that is 5 states or more when PCLKB to ADCLK frequency ratio = 1:1, 2:1, 4:1, or 8:1.
Set a value that is 6 states or more when PCLKB to ADCLK frequency ratio = 1:2.

Table 40.8 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 40.3.6, Analog Input Sampling Time and Scan Conversion Time.

Table 40.8 Relationship between A/D Sampling State Register and Relevant Channels

Register Name	Channels
ADSSTR0	AN000
ADSSTR1	AN001
ADSSTR2	AN002
ADSSTR3	AN003
ADSSTR4	AN004
ADSSTR5	AN005
ADSSTR6	AN006
ADSSTR7	AN007
ADSSTR8	AN008
ADSSTRL	AN016 to AN031
ADSSTRT	Temperature sensor output*1
ADSSTRO	Internal reference voltage*1

Note 1. When performing A/D conversion of the temperature sensor output or internal reference voltage, the sampling time should be 5 µs or longer. Since the maximum number of states that can be set by this register is 255, take note of the ADCLK frequency. For example, when ADCLK = 64 MHz, the sampling time does not reach 5 µs even if 255 states is set.

40.2.15 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
b4 to b0	ADNDIS[4:0]	A/D Disconnection Detection Assist Setting	b4 ADNDIS[4]: Discharge/precharge selected 0: Discharge 1: Precharge b3 to b0 ADNDIS[3:0]: Discharge/precharge period	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register sets the disconnection detection assist function.

ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)

These bits select either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge.

When the ADEXICR.OCSA or TSSA bit is set to 1 to perform A/D conversion of the temperature sensor output or internal reference voltage, ADNDIS[4:0] are automatically fixed to 0Fh, and discharging is executed prior to A/D conversion (auto-discharging). An auto-discharge period of 15 ADCLK cycles is inserted before sampling each time the temperature sensor output or internal reference voltage is A/D-converted.

40.2.16 A/D Event Link Control Register (ADELCCR)

Address(es): S12AD.ADELCCR 0008 907Dh

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ELCC[1:0]	Event Link Control	b1 b0 0 0: Event is generated on completion of the scan other than group B in group scan mode 0 1: Event is generated on completion of the scan of group B in group scan mode 1 x: Event is generated on completion of all scans	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

The ADELCCR register sets the generation conditions of the ELC scan end event (S12ADELC).

ELCC[1:0] Bits (Event Link Control)

These bits select the generation conditions of the scan end event (S12ADELC) for the ELC.

40.2.17 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	—	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Group-A Priority Control Setting *1	0: Operation is without group-A priority control 1: Operation is with group-A priority control	R/W
b1	GBRSCN	Group B Restart Setting *2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for group B is not restarted after having been discontinued due to group-A priority control. 1: Scanning for group B is restarted after having been discontinued due to group-A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	GBRP	Group B Single Scan Continuous Start *3	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan for group B is not continuously activated. 1: Single scan for group B is continuously activated.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
Note 2. When the GBRSCN bit is to be set to 1, the frequency ratio of peripheral module clock PCLKB to A/D conversion clock ADCLK should be set to 1:1.
Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for group B regardless of the setting of the GBRSCN bit.

The ADGSPCR register is used to make settings for priority control of A/D conversion for group A in group scan mode.

PGS Bit (Group-A Priority Control Setting)

This bit sets the priority of operation on group A. Set this bit to 1 when giving priority to operation on group A.
When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
When setting the PGS bit to 0, clearing should be performed by software according to section 40.8.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 40.3.4.3, Operation under Group-A Priority Control.

GBRSCN Bit (Group B Restart Setting)

This bit controls the restarting of scan operation on group B when operation on group A is given priority.
If a scan operation on group B has been stopped by a group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the A/D conversion on group A. Also, if a group B trigger is input during A/D conversion on group A, the scan operation on group B is restarted on completion of the A/D conversion on group A.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

GBRP Bit (Group B Single Scan Continuous Start)

This bit is set when a single scan operation is to be performed continuously on group B.
Setting the GBRP bit to 1 starts a single scan on group B. On completion of the scan, another single scan on group B is automatically started. If an A/D conversion on group B has been stopped due to an operation on group A that takes priority, single scan on group B is automatically restarted on completion of the A/D conversion on group A.
Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting of the

GBRSCN bit. The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

40.2.18 A/D Compare 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
	—	WCMPE	—	—	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 Composite Condition Setting	b1 b0 0 0: S12ADWMELC is output when window A comparison conditions are met OR window B comparison conditions are met. S12ADWUMELC is output in other cases. 0 1: S12ADWMELC is output when window A comparison conditions are met EXOR window B comparison conditions are met. S12ADWUMELC is output in other cases. 1 0: S12ADWMELC is output when window A comparison conditions are met AND window B comparison conditions are met. S12ADWUMELC is output in other cases. 1 1: Setting prohibited.	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	CMPBE	Compare Window B Operation Enable	0: Compare window B operation is disabled. S12ADWMELC and S12ADWUMELC outputs are disabled. 1: Compare window B operation is enabled.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	CMPAE	Compare Window A Operation Enable	0: Compare window A operation is disabled. S12ADWMELC and S12ADWUMELC outputs are disabled. 1: Compare window A operation is enabled.	R/W
b13, 12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	WCMPE	Window Function Setting	0: Window function is disabled. Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Window function is enabled. Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The ADCMPCR register sets the compare window A and window B functions.

CMPAB[1:0] Bits (Window A/B Composite Condition Setting)

These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits are used to select compare function match/mismatch event output conditions for the ELC or monitoring conditions of the ADWINMON.MONCOMB flag. The CMPAB[1:0] bits should be set while the ADCSR.ADST bit is 0.

CMPBE Bit (Compare Window B Operation Enable)

This bit enables or disables the compare window B operation. The CMPBE bit should be set while the ADCSR.ADST bit is 0.
Set this bit to 0 before setting the following registers.

- A/D channel select registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- The OCSA or TSSA bit in the A/D conversion extended input control register (ADEXICR.OCSA, TSSA)
- The CMPCHB[5:0] bits in the window B channel select register (ADCMPBNSR.CMPCHB[5:0])

CMPAE Bit (Compare Window A Operation Enable)

This bit enables or disables the compare window A operation. The CMPAE bit should be set while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers.

- A/D channel select registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- The OCSA or TSSA bit in the A/D conversion extended input control register (ADEXICR.OCSA, TSSA)
- Window A channel select registers 0/1 (ADCMPANSR0, ADCMPANSR1)
- Window A extended input select register (ADCMPANSER)

WCMPE Bit (Window Function Setting)

This bit enables or disables the window function. The WCMPE bit should be set while the ADCSR.ADST bit is 0.

40.2.19 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0)

Address(es): S12AD.ADCMPANSR0 0008 9094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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	Compare Window A Channel Select	0: The corresponding channel from among AN000 to AN008 is not a target for compare window A. 1: The corresponding channel from among AN000 to AN008 is a target for compare 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
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSR0 register is used to select analog input channels for comparison under compare window A conditions from among AN000 to AN008.

CMPCHA0n Bit (n = 00 to 08) (Compare Window A Channel Select)

Setting the CMPCHA0n bit which has the same number as the A/D channel selected by the ADANSA0.ANSA0n or ADANSB0.ANSB0n bit to 1 enables the compare function.

The CMPCHA0n bit should be set while ADCSR.ADST bit is 0.

40.2.20 A/D Compare 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
	CMPCHA115	CMPCHA114	CMPCHA113	CMPCHA112	CMPCHA111	CMPCHA110	CMPCHA109	CMPCHA108	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	Compare Window A Channel Select	0: The corresponding channel from among AN016 to AN031 is not a target for compare window A. 1: The corresponding channel from among AN016 to AN031 is a target for compare 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
b8	CMPCHA108			R/W
b9	CMPCHA109			R/W
b10	CMPCHA110			R/W
b11	CMPCHA111			R/W
b12	CMPCHA112			R/W
b13	CMPCHA113			R/W
b14	CMPCHA114			R/W
b15	CMPCHA115			R/W

The ADCMPANSR1 register is used to select analog input channels for comparison under compare window A conditions from among AN016 to AN031.

CMPCHA1n Bit (n = 00 to 15) (Compare Window A Channel Select)

Setting the CMPCHA1n bit which has the same number as the A/D channel selected by the ADANSA1.ANSA1n or ADANSB1.ANSB1n bit to 1 enables the compare function.

The CMPCHA1n bit should be set while ADCSR.ADST bit is 0.

40.2.21 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): S12AD.ADCMPANSER 0008 9092h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPTSA	Temperature Sensor Output Compare Select	0: Temperature sensor output is not a target for compare window A. 1: Temperature sensor output is a target for compare window A.	R/W
b1	CMPOCA	Internal Reference Voltage Compare Select	0: Internal reference voltage is not a target for compare window A. 1: Internal reference voltage is a target for compare 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 is used to select whether the temperature sensor output or internal reference voltage is compared under compare window A conditions.

CMPTSA Bit (Temperature Sensor Output Compare Select)

Setting the CMPTSA bit to 1 while the ADEXICR.TSSA bit is 1 enables the compare window A function. This bit should be set while the ADCSR.ADST bit is 0.

CMPOCA Bit (Internal Reference Voltage Compare Select)

Setting the CMPOCA bit to 1 while ADEXICR.OCSA bit is 1 enables the compare window A function. This bit should be set while the ADCSR.ADST bit is 0.

40.2.22 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

Address(es): S12AD.ADCMPLR0 0008 9098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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	Compare Window A Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	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
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLR0 register sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion. The ADCMPLR0 register should be set while ADCSR.ADST bit is 0.

CMPLCHA0n Bit (n = 00 to 08) (Compare Window A Comparison Condition Select)

This bit sets the condition for use in comparison with the selected channel from among AN000 to AN008 to which compare window A conditions are applied. A condition can be set for individual comparison of each analog input.
The CMPLCHA000 bit is used for AN000, the CMPLCHA008 bit is used for AN008.
When the result of comparison matches the set condition, the ADCMPSR0.CMPSTCHA0n flag is set to 1.
Figure 40.3 shows the comparison conditions.

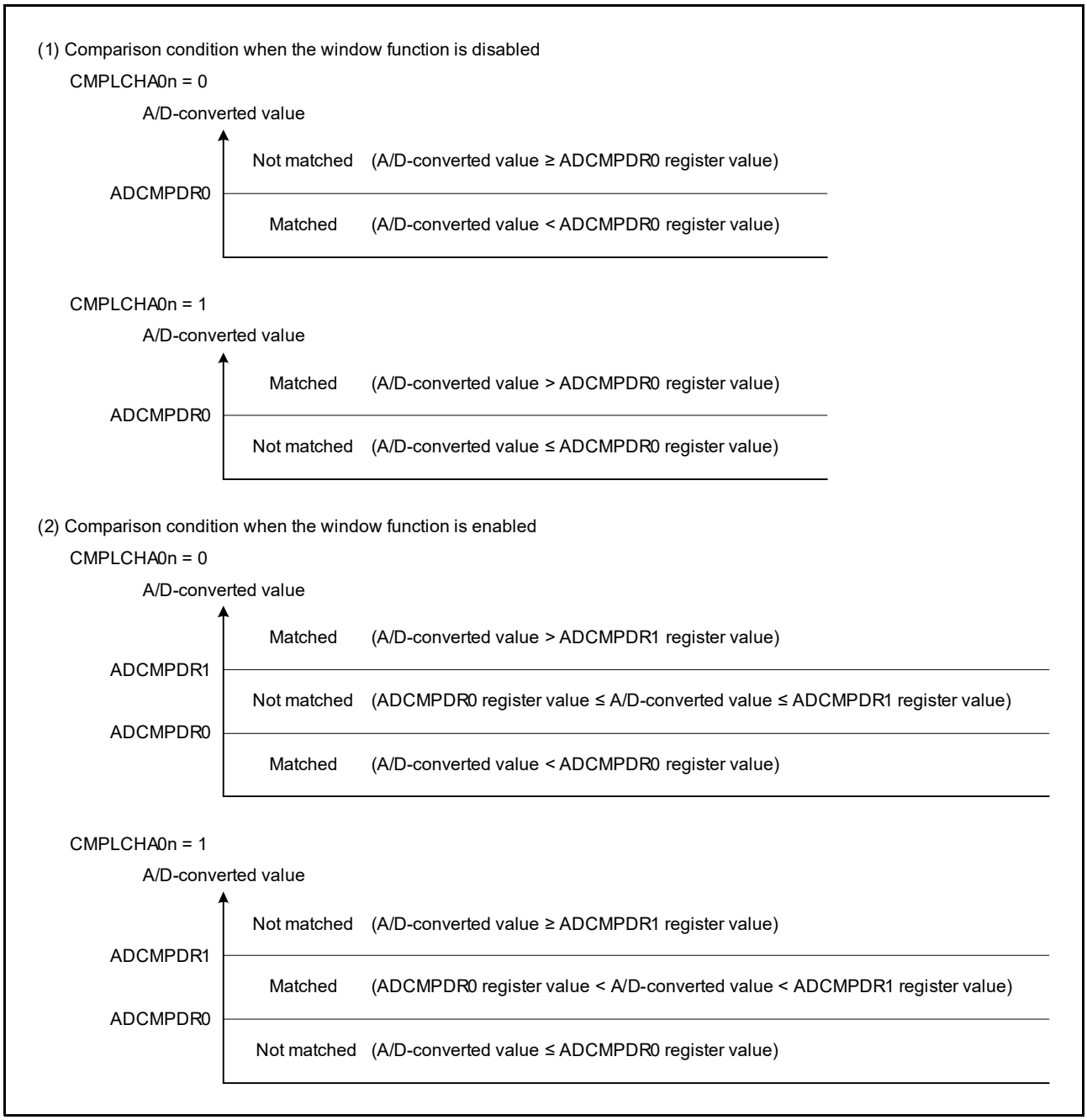


Figure 40.3 Explanation of Compare Function Window A Comparison Conditions

40.2.23 A/D Compare 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
	CMPLCHA115	CMPLCHA114	CMPLCHA113	CMPLCHA112	CMPLCHA111	CMPLCHA110	CMPLCHA109	CMPLCHA108	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	Compare Window A Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	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
b8	CMPLCHA108			R/W
b9	CMPLCHA109			R/W
b10	CMPLCHA110			R/W
b11	CMPLCHA111			R/W
b12	CMPLCHA112			R/W
b13	CMPLCHA113			R/W
b14	CMPLCHA114			R/W
b15	CMPLCHA115			R/W

The ADCMPLR1 register sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
The ADCMPLR1 register should be set while ADCSR.ADST bit is 0.

CMPLCHA1n Bit (n = 00 to 15) (Compare Window A Comparison Condition Select)

This bit sets the condition for use in comparison with the selected channel from among AN016 to AN031 to which compare window A conditions are applied. A condition can be set for individual comparison of each analog input.
The CMPLCHA100 bit is used for AN016 and the CMPLCHA115 bit is used for AN031.
When the result of comparison matches the set condition, the ADCMPSR1.CMPSTCHA1n flag is set to 1.
Figure 40.3 shows the comparison conditions.

40.2.24 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): S12AD.ADCMPLER 0008 9093h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLOCA	Internal Reference Voltage Comparison Condition Select	When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLER register sets the condition for use in comparing the values of ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
The ADCMPLER register should be set while ADCSR.ADST is 0.

CMPLTSA Bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

This bit sets the condition for use in comparison with temperature sensor output to which compare window A conditions are applied.
When the result of comparison matches the set condition, the ADCMPSER.CMPSTTSA flag is set to 1.
Figure 40.3 shows the comparison conditions.

CMPLOCA Bit (Internal Reference Voltage Comparison Condition Select)

This bit sets conditions for use in comparison with internal reference voltage to which compare window A conditions are applied.
When the result of comparison matches the set condition, the ADCMPSER.CMPSTOCA flag is set to 1.
Figure 40.3 shows the comparison conditions.

40.2.25 A/D Compare Function Window A Lower-Side 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/writable register that sets the reference data when the compare window A function is used. The ADCMPDR0 register sets the lower-side level of window A.
The ADCMPDR0 register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value).

The ADCMPDR0 register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

Set bits 15 to 2 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. 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 A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADCMPDR0 register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADCMPDR0 register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

40.2.26 A/D Compare Function Window A Upper-Side 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/writable register that sets the reference data when the compare window A function is used. The ADCMPDR1 register sets the upper-side level of window A.
The ADCMPDR1 register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value).
The ADCMPDR1 register is not used when the window function is disabled.

The ADCMPDR1 register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
- Set bits 15 to 2 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. 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 A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADCMPDR1 register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADCMPDR1 register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

40.2.27 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0)

Address(es): S12AD.ADCMPSR0 0008 90A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), these flags indicate the comparison result of channels (AN000 to AN008 to which window A comparison conditions are applied. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTCHA001			R/W
b2	CMPSTCHA002			R/W
b3	CMPSTCHA003			R/W
b4	CMPSTCHA004			R/W
b5	CMPSTCHA005			R/W
b6	CMPSTCHA006			R/W
b7	CMPSTCHA007			R/W
b8	CMPSTCHA008			R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSR0 register stores the comparison results of the compare window A function.

CMPSTCHA0n Flag (n = 00 to 08) (Compare Window A Flag)

This flag is comparison result status flag of channel (AN000 to AN008) to which window A comparison conditions are applied. When the comparison condition set by the ADCMPLR0.CMPLCHA0n bit is met at the end of A/D conversion, each of these flags is set to 1. The CMPSTCHA000 and CMPSTCHA008 flags correspond to AN000 and AN008, respectively.

Writing 1 to the CMPSTCHA0n flag is disabled.

[Setting condition]

- The condition set by the ADCMPLR0.CMPLCHA0n bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

40.2.28 A/D Compare 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 CHA115	CMPST CHA114	CMPST CHA113	CMPST CHA112	CMPST CHA111	CMPST CHA110	CMPST CHA109	CMPST CHA108	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	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), these flags indicate the comparison result of channels (AN016 to AN031) to which window A comparison conditions are applied. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTCHA101			R/W
b2	CMPSTCHA102			R/W
b3	CMPSTCHA103			R/W
b4	CMPSTCHA104			R/W
b5	CMPSTCHA105			R/W
b6	CMPSTCHA106			R/W
b7	CMPSTCHA107			R/W
b8	CMPSTCHA108			R/W
b9	CMPSTCHA109			R/W
b10	CMPSTCHA110			R/W
b11	CMPSTCHA111			R/W
b12	CMPSTCHA112			R/W
b13	CMPSTCHA113			R/W
b14	CMPSTCHA114			R/W
b15	CMPSTCHA115			R/W

The ADCMPSR1 register stores the comparison results of the compare window A function.

CMPSTCHA1n Flag (n = 00 to 15) (Compare Window A Flag)

This flag is comparison result status flag of channel (AN016 to AN031) to which window A comparison conditions are applied. When the comparison condition set by the ADCMPLR1.CMPLCHA1n bit is met at the end of A/D conversion, each of these flags is set to 1. The CMPSTCHA100 flag corresponds to AN016, the CMPSTCHA104 flag corresponds to AN020, and the CMPSTCHA115 flag corresponds to AN031.

Writing 1 to the CMPSTCHA1n flag is disabled.

[Setting condition]

- The condition set by the ADCMPLR1.CMPLCHA1n bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

40.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): S12AD.ADCMPSER 0008 90A4h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTOCA	Compare Window A Internal Reference Voltage Compare Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the internal reference voltage comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSER register stores the comparison result of the compare window A function.

CMPSTTSA Flag (Compare Window A Temperature Sensor Output Compare Flag)

This flag is a status flag that indicates the temperature sensor output comparison result. When the comparison condition set by the ADCMPLER.CMPLTSA bit is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTTSA flag is disabled.
[Setting condition]

- The condition set by the ADCMPLER.CMPLTSA bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

CMPSTOCA Flag (Compare Window A Internal Reference Voltage Compare Flag)

This flag is a status flag that indicates the internal reference voltage comparison result. When the comparison condition set by the ADCMPLER.CMPLOCA bit is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTOCA flag is disabled.
The value 1 cannot be written to the CMPSTOCA flag.
[Setting condition]

- The condition set in by the ADCMPLER.CMPLOCA bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

40.2.30 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT)

Address(es): S12AD.ADHVREFCNT 0008 908Ah

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HVSEL[1:0]	High-Potential Reference Voltage Select	b1 b0 0 0: AVCC0 is selected as the high-potential reference voltage. 0 1: VREFH0 is selected as the high-potential reference voltage.*1 Settings other than above are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	LVSEL	Low-Potential Reference Voltage Select	0: AVSS0 is selected as the low-potential reference voltage. 1: VREFL0 is selected as the low-potential reference voltage.*1	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ADSLP	Sleep	0: Normal operation 1: Standby state	R/W

Note 1. Set the PJ6 and PJ7 pins to analog inputs/outputs (PORTJ.PMR.B6 = 0, PORTJ.PMR.B7 = 0, PORTJ.PDR.B6 = 0, PORTJ.PDR.B7 = 0, PJ6PFS.ASEL = 1, and PJ7PFS.ASEL = 1).

The ADHVREFCNT register specifies the high-potential and low-potential reference voltages. Set this register before performing A/D conversion.

HVSEL[1:0] Bits (High-Potential Reference Voltage Select)

These bits are used to set the high-potential reference voltage. AVCC0 or VREFH0 is selectable as the high-potential reference voltage.

LVSEL Bit (Low-Potential Reference Voltage Select)

This bit is used to set the low-potential reference voltage. AVSS0 or VREFL0 is selectable as the low-potential reference voltage.

ADSLP Bit (Sleep)

This bit is used to transition the 12-bit A/D converter to the standby state. Set the ADSLP bit to 1 only when modifying the ADCSR.ADHSC bit. In other cases, setting the ADSLP bit to 1 is prohibited.

After the ADSLP bit is set to 1, wait at least 5 µs before clearing this bit to 0. Furthermore, after the ADSLP bit is cleared to 0, wait at least 1 µs and then start the A/D conversion.

For the ADHSC bit rewriting procedure, see section 40.8.10, ADHSC Bit Rewriting Procedure.

40.2.31 A/D Compare Function Window A/B Status Monitor 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 Monitor Flag	This flag indicates the combination result. This flag is valid when both window A operation and window B operation are enabled. 0: Window A/window B composite conditions are not met. 1: Window A/window B composite conditions are met.	R
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	MONCMPA	Comparison Result Monitor A Flag	0: Window A comparison conditions are not met. 1: Window A comparison conditions are met.	R
b5	MONCMPB	Comparison Result Monitor B Flag	0: Window B comparison conditions are not met. 1: Window B comparison conditions are met.	R
b7, b6	—	Reserved	These bits are read as 0.	R

The ADWINMON register can monitor the comparison result and the combination result.

MONCOMB Flag (Combination Result Monitor Flag)

This read-only flag indicates the result in combination of comparison condition result A and comparison result condition B with the combination condition set by the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set by the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set by the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA Flag (Comparison Result Monitor A Flag)

This read-only flag is read as 1 when the A/D-converted value of the window A target channel meets the condition set in the ADCMPLR0, ADCMPLR1, and ADCMPLER registers, and is read as 0 otherwise.

[Setting condition]

- The A/D-converted value meets the condition set by the ADCMPLR0.CMPLCHA0n bit when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set by the ADCMPLR0.CMPLCHA0n bit when ADCMPCR.CMPAE = 1.
- ADCMPCR.CMPAE = 0 (Automatically cleared when the ADCMPCR.CMPAE bit value changes from 1 to 0.)

MONCMPB Flag (Comparison Result Monitor B Flag)

This read-only flag is read as 1 when the A/D converted value of the window B target channel meets the condition set by the ADCMPBNSR.CMPLB bit, and is read as 0 in other cases.

[Setting condition]

- The A/D-converted value meets the condition set by ADCMPBNSR.CMPLB bit when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set by ADCMPBNSR.CMPLB bit when ADCMPCR.CMPBE = 1.
- ADCMPCR.CMPBE = 0 (Automatically cleared when the ADCMPCR.CMPBE bit value changes from 1 to 0.)

40.2.32 A/D Compare Function Window B Channel Select Register (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]	Compare Window B Channel Select	These bits select channels to be compared with the compare window B conditions. b5 b0 0 0 0 0 0 0: AN000 0 0 0 0 0 1: AN001 0 0 0 0 1 0: AN002 : : 0 0 0 1 1 0: AN006 0 0 0 1 1 1: AN007 0 0 1 0 0 0: AN008 0 1 0 0 0 0: AN016 0 1 0 0 0 1: AN017 : : 0 1 1 0 0 1: AN029 0 1 1 1 1 0: AN030 0 1 1 1 1 1: AN031 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	Compare Window B Comparison Condition Setting	When the window function is disabled (ADCMPCR.WCMPE bit = 0) 0: ADWINLLB register value > A/D-converted value 1: ADWINLLB register value < A/D-converted value When the window function is enabled (ADCMPCR.WCMPE bit = 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 ADCMPBNSR register is used to set the compare window B function.

CMPCHB[5:0] Bits (Compare Window B Channel Select)

These bits are used to select channels to be compared from among AN000 to AN008, AN016 to AN031, temperature sensor, and internal reference voltage against the conditions of comparison window B.
The compare window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected by the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers.
The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Compare Window B Comparison Condition Setting)

This bit is used to set comparison conditions of channels for window B. When the comparison result of each analog input

meets the set condition, the ADCMPBSR.CMPSTB flag is set to 1.
Figure 40.4 shows the comparison conditions.

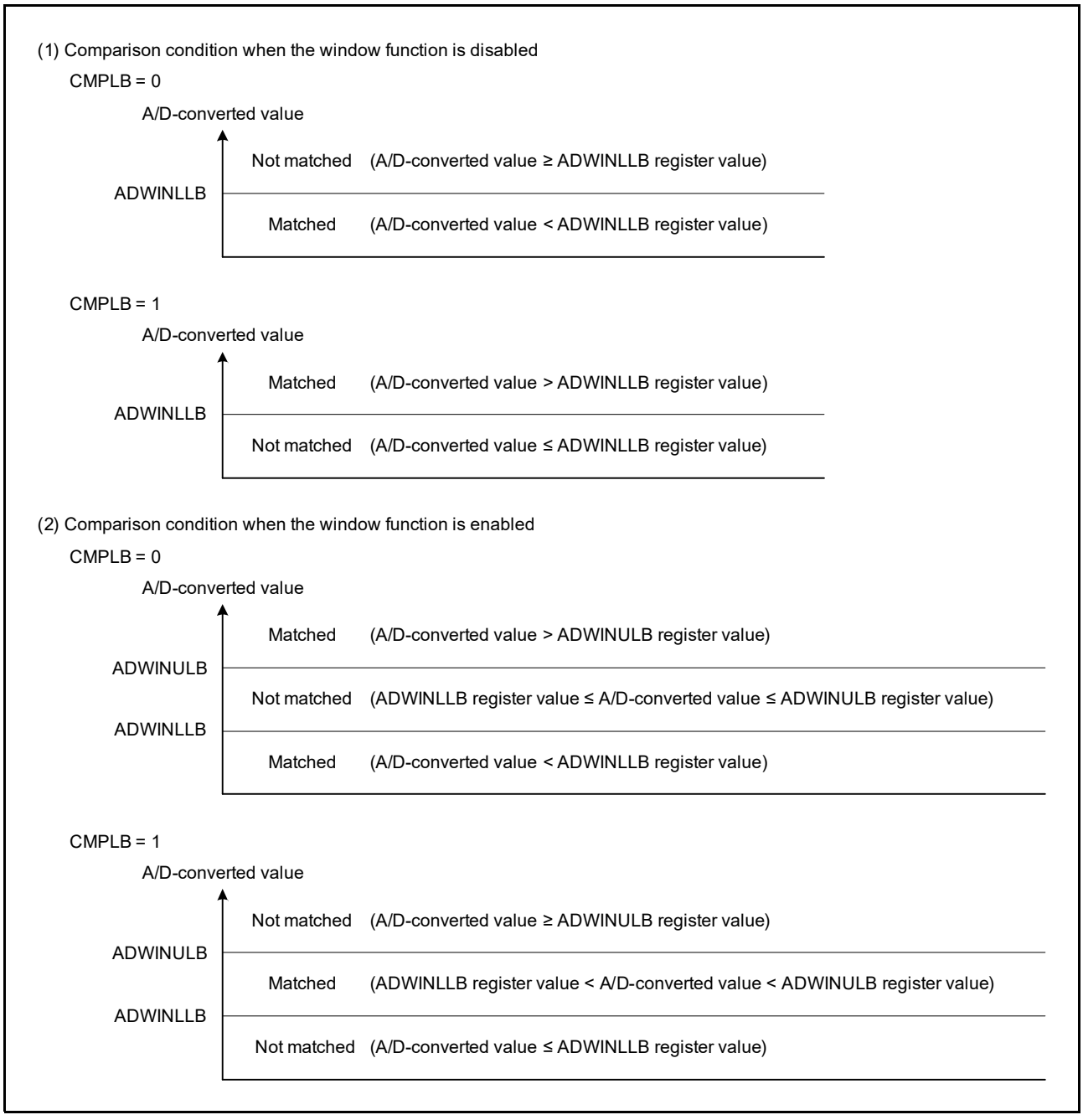


Figure 40.4 Explanation of Compare Function Window B Compare Conditions

40.2.33 A/D Compare Function Window B Lower-Side 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/writable register that sets the reference data when the compare window B function is used. The ADWINLLB register sets the lower-side level of window B.
The ADWINLLB register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADWINULB setting value ≥ ADWINLLB setting value).

The ADWINLLB register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y (ADDRy) is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

Set bits 15 to 2 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. 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 A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADWINLLB register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADWINLLB register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

40.2.34 A/D Compare Function Window B Upper-Side 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/writable register that sets the reference data when the compare window B function is used. The ADWINULB register sets the upper-side level of window B.
The ADWINULB register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADWINULB setting value ≥ ADWINLLB setting value)
The ADWINULB register is not used when the window function is disabled.

The ADWINULB register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y (ADDRy) is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 15 to 2 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. 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 A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADWINULB register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADWINULB register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

40.2.35 A/D Compare Function Window B Channel Status Register (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	Compare Window B Flag	0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPBSR register stores the comparison result of the compare window B function.

CMPSTB Flag (Compare Window B Flag)

This flag is a status flag that indicates the results of comparison for the channels (AN000 to AN008, AN016 to AN031, temperature sensor, and internal reference voltage) to which the window B comparison conditions are applied.
When the comparison condition set by ADCMPBNSR.CMPCHB[5:0] bits is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTB flag is disabled.
[Setting condition]

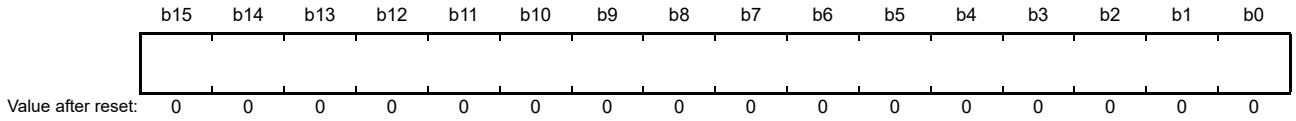
- The condition set by ADCMPBNSR.CMPLB bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

40.2.36 A/D Data Storage Buffer Register n (ADBUFn) (n = 0 to 15)

Address(es): S12AD.ADBUF0 0008 90B0h, S12AD.ADBUF1 0008 90B2h, S12AD.ADBUF2 0008 90B4h, S12AD.ADBUF3 0008 90B6h, S12AD.ADBUF4 0008 90B8h, S12AD.ADBUF5 0008 90BAh, S12AD.ADBUF6 0008 90BCh, S12AD.ADBUF7 0008 90BEh, S12AD.ADBUF8 0008 90C0h, S12AD.ADBUF9 0008 90C2h, S12AD.ADBUF10 0008 90C4h, S12AD.ADBUF11 0008 90C6h, S12AD.ADBUF12 0008 90C8h, S12AD.ADBUF13 0008 90CAh, S12AD.ADBUF14 0008 90CCh, S12AD.ADBUF15 0008 90CEh



	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

A/D data storage buffer registers n (ADBUFn) are 16-bit read-only registers that sequentially store all A/D converted values. The automatic clear function is not applied to these registers.

The ADBUFn register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left 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 Value Average Mode is Selected

- Flush-right 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.
- Flush-left 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 two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  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 conversion 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 ADBUFn 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 ADBUFn register as 4-bit extended data of the conversion accuracy bits.
Even if A/D-converted value addition mode is selected, the extended A/D-converted value is stored in the ADBUFn register according to the settings of the A/D data register format select bits.

40.2.37 A/D Data Storage Buffer Enable Register (ADBUFEN)

Address(es): S12AD.ADBUFEN 0008 90D0h

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

Bit	Symbol	Bit Name	Description	R/W
b0	BUFEN	Data Storage Buffer Enable	0: The data storage buffer is not used. 1: The data storage buffer is used.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADBUFEN register is used to enable the data storage buffer.

BUFEN Bit (Data Storage Buffer Enable)

This bit enables the use of the data storage buffer when using the compare function.
When BUFEN = 1, A/D conversion result (addition result) other than self-diagnosis result is stored in ADBUFn.
Disable the data storage operation (BUFEN = 0) before reading ADBUFn and ADBUFPTR.
Do not use the data storage buffer for data duplexing, continuous scan, or group scan.

40.2.38 A/D Data Storage Buffer Pointer Register (ADBUFPTR)

Address(es): S12AD.ADBUFPTR 0008 90D2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PTROVF	BUFPTR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BUFPTR[3:0]	Data Storage Buffer Pointer	These bits indicate the number of data storage buffer to which the next A/D conversion data is transferred.	R/W
b4	PTROVF	Pointer Overflow Flag	0: The data storage buffer pointer has not overflowed. 1: The data storage buffer pointer has overflowed.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADBUFPTR register is used for the data storage buffer pointer.

BUFPTR[3:0] Bits (Data Storage Buffer Pointer)

These read-only bits indicate the number of data storage buffer to which the next A/D conversion data is transferred. When data has been transferred to data storage buffer 15, the pointer value becomes 0000b and the PTROVF flag is set to 1. When the next data has been transferred, the data in data storage buffer 0 is overwritten.
Writing 00h to this register clears the value of these bits. Writing a value other than 00h is disabled.

PTROVF Flag (Pointer Overflow Flag)

This read-only flag indicates whether the data storage buffer pointer has overflowed. This flag is set to 1 when the pointer value becomes 0000b (overflow).
Writing 00h to this register clears this flag value. Writing a value other than 00h is disabled.

40.2.39 A/D Convert Cycle Control Register (ADCCR)

Address(es): S12AD.ADCCR 0008 907Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CCS	—
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	CCS	Conversion Cycle Select	This bit is used to select the number of conversion cycles. Refer to Table 40.9 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CCS Bit (Conversion Cycle Select)

This bit selects the number of cycles required to convert each bit of the final value.

Table 40.9 shows the number of cycles required for successive conversion.

Table 40.9 Number of Cycles Required for Conversion by Successive Approximation

CCS Bit	ADHSC Bit	Number of Conversion Cycles
0	0	32 cycles
0	1	41 cycles
1	0	22 cycles
1	1	28 cycles

40.3 Operation

40.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. Also, conversion modes are divided into high-speed conversion mode and normal conversion 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 group A and the selected channels of group B are scanned once after starting to be scanned according to the respective synchronous trigger.
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of group A and group B selected by the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers, respectively, starting from the channel with the smallest number n.
When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.
When performing A/D conversion of the temperature sensor output or internal reference voltage, execute scanning individually.
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.

40.3.2 Single Scan Mode

40.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

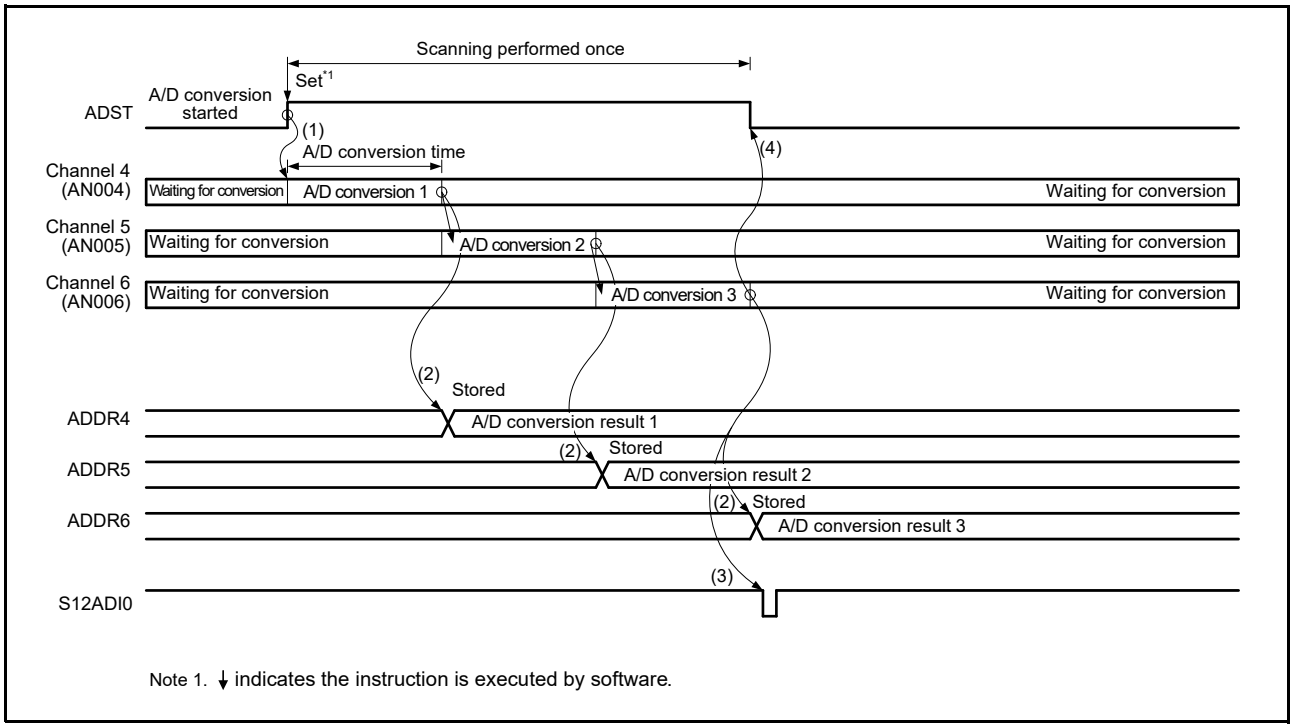


Figure 40.5 Example of Operation in Single Scan Mode (Basic Operation: AN004, AN005, AN006 Selected)

40.3.2.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage VREFH0 supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the 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, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(5) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

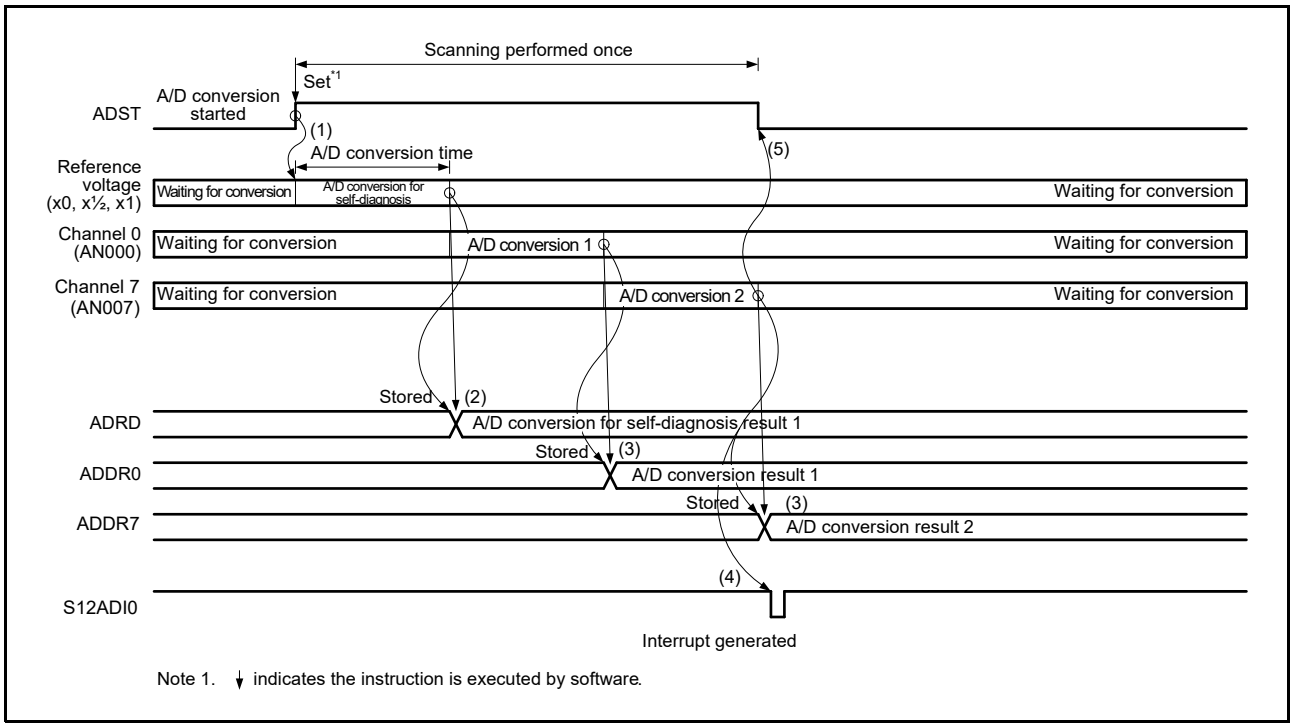


Figure 40.6 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN007 Selected + Self-Diagnosis)

40.3.2.3 A/D Conversion of Temperature Sensor Output/Internal Reference Voltage

A/D conversion of the temperature sensor output and internal reference voltage is performed in single scan mode as below.

All channels should be deselected (by setting the ADANSA0 and ADANSA1 register bits to all 0 and the ADCSR.DBLE bit to 0). When selecting A/D conversion of the temperature sensor output, the A/D conversion select bit for the internal reference voltage (ADEXICR.OCSA) should be set to 0 (deselected). When selecting A/D conversion of the internal reference voltage, the A/D conversion select bit for the temperature sensor output (ADEXICR.TSSA) should be set to 0 (deselected).

(1) Set the sampling time to 5 µs or longer.
(2) After switching to A/D conversion of the internal reference voltage or the temperature sensor output, start A/D conversion by setting the ADST bit to 1.
(3) When A/D conversion is completed, the conversion result is stored into the corresponding A/D temperature sensor data register (ADTSDR) or A/D internal reference voltage data register (ADOCDR). If the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.
(4) The ADST bit remains 1 during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a wait state.

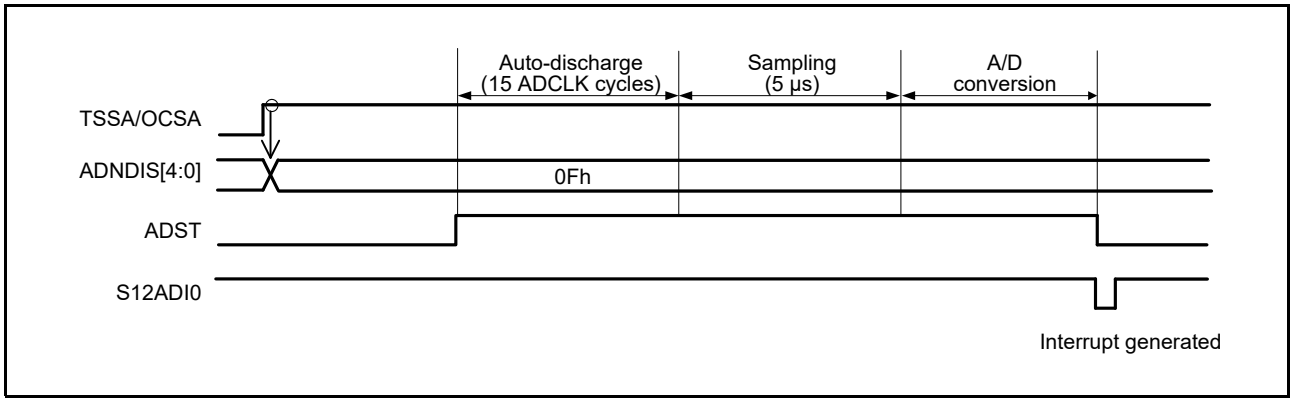


Figure 40.7 Example of Operation in Single Scan Mode (Temperature Sensor Output or Internal Reference Voltage Selected)

40.3.2.4 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

Self-diagnosis should be deselected, and the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set 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 DBLE bit in ADCSR 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 is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) The ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI0 interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (S12ADI0 interrupt upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.
(6) If the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.
(7) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

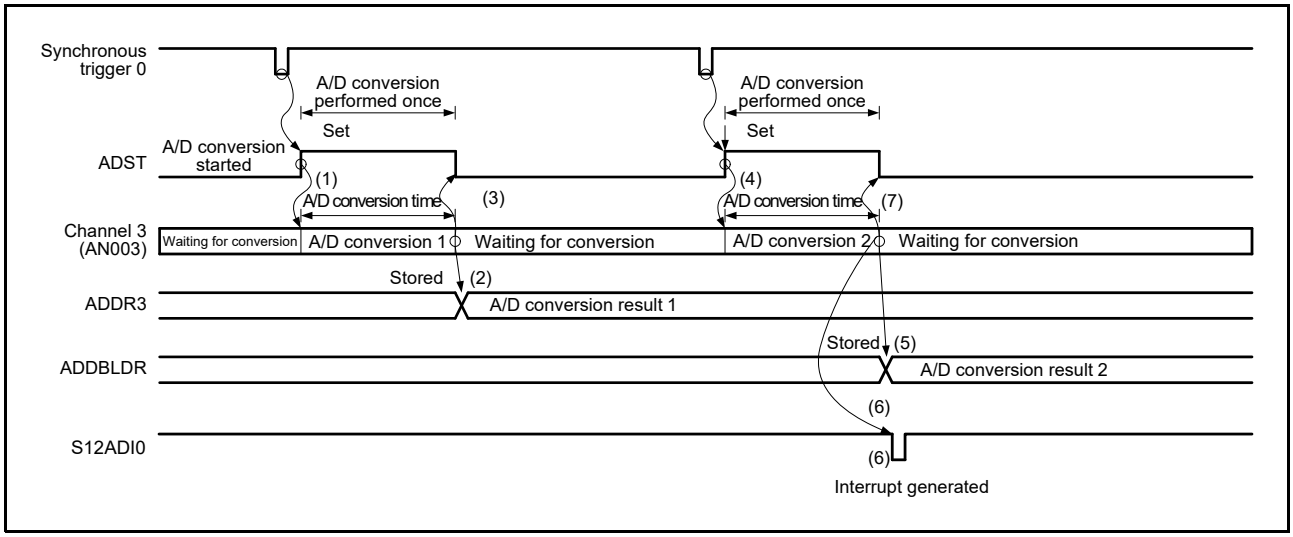


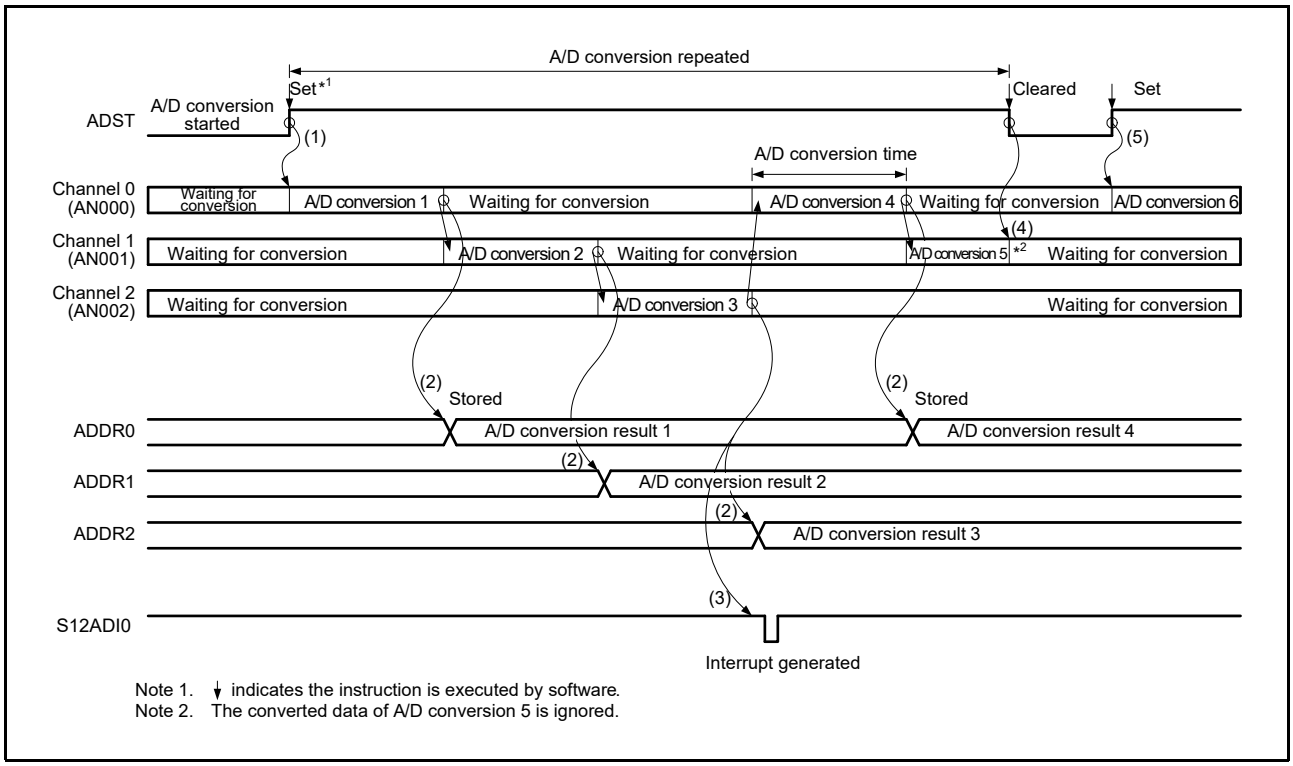
Figure 40.8 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

40.3.3 Continuous Scan Mode

40.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.
In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) The ADCSR.ADST bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



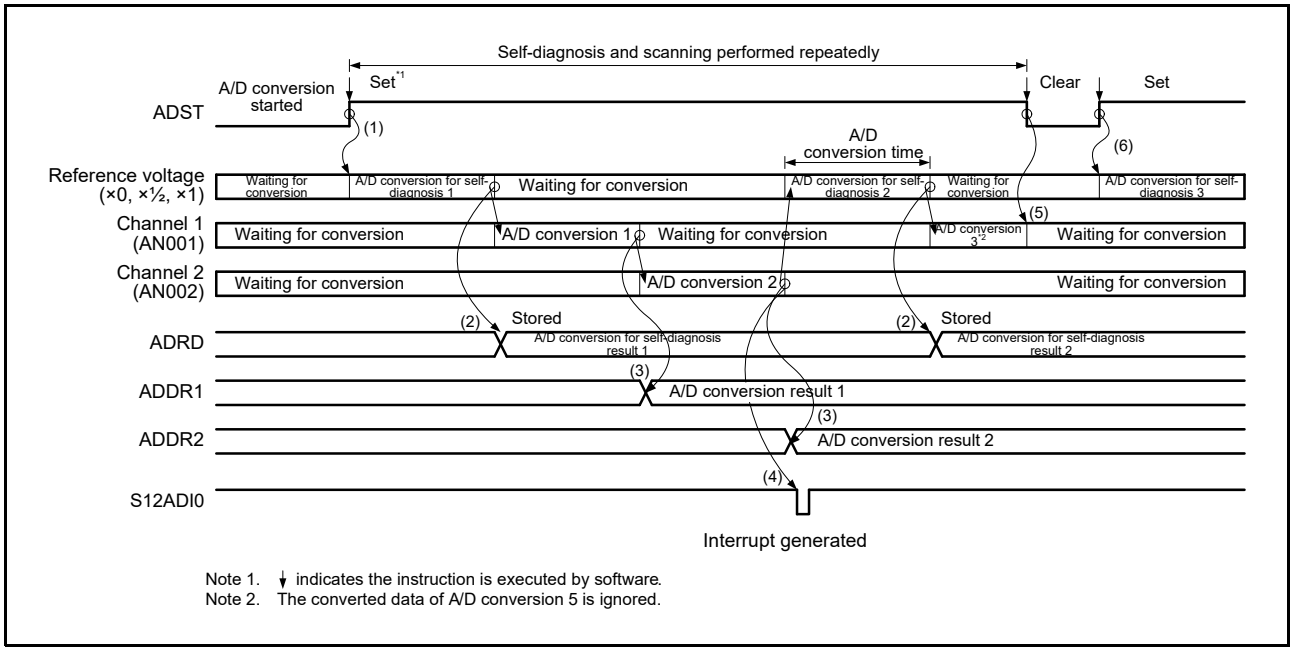
Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 40.9 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

40.3.3.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0 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. In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion for self-diagnosis is started first.
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the 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, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(5) The ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 40.10 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

40.3.4 Group Scan Mode

40.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by a synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.
The synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.
The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. The same channels cannot be selected for both groups.
In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).
When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and group B.
The following describes operation in group scan mode using a trigger from the GPTW. The GTADTRA0N and GTADTRB0N triggers from the GPTW are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group A is started by the GTADTRA0N trigger from the GPTW.
(2) When group A scanning is completed, an S12ADI0 interrupt is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(3) Scanning of group B is started by the GTADTRB0N trigger from the GPTW.
(4) When group B scanning is completed, a GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).

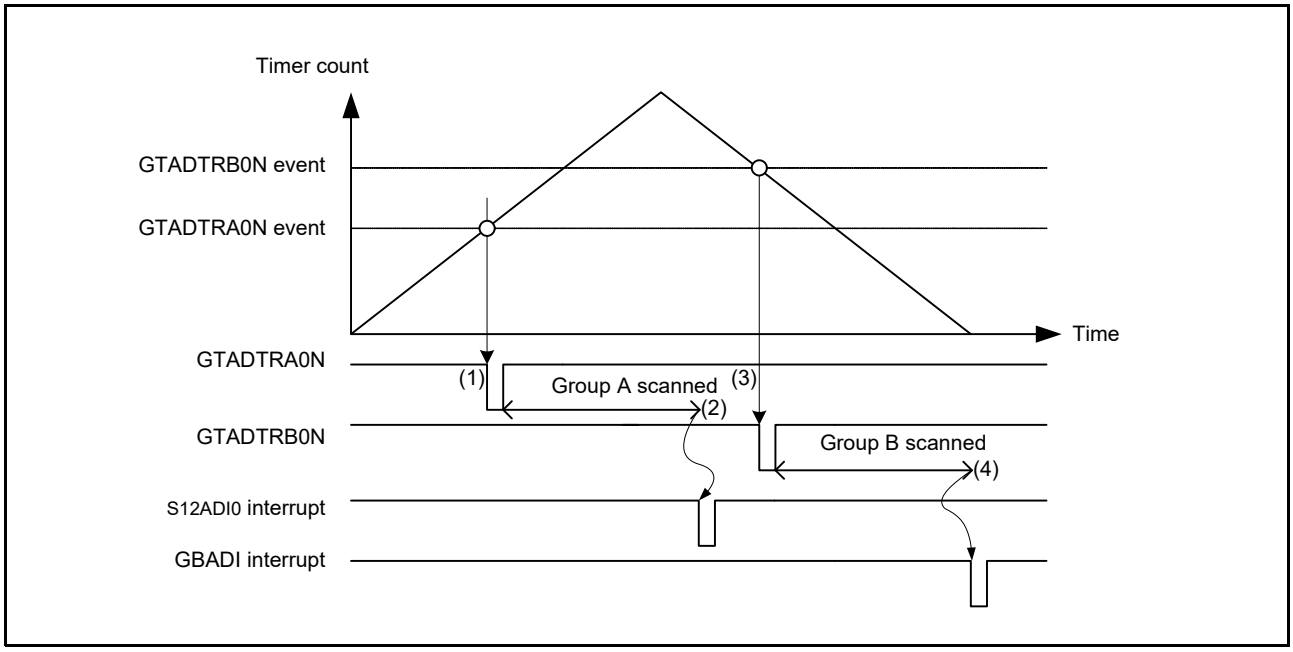


Figure 40.11 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from GPTW Used)

40.3.4.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger is performed once.
In group scan mode, the synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger and asynchronous trigger should not be used.
The group A and group B channels to be A/D-converted are selected using the ADCSR.DBLANS[4:0] bits and the ADANSB0 and ADANSB1 registers, respectively. The same channels cannot be selected for both groups.
In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).
When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the 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 GPTW. The GTADTRA0N and GTADTRB0N triggers from the GPTW are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group B is started by the GTADTRB0N trigger from the GPTW.
(2) When group B scanning is completed, a GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).
(3) The first scanning of group A is started by the first GTADTRA0N trigger from the GPTW.
(4) When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); an S12ADI0 interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(5) The second scanning of group A is started by the second GTADTRA0N trigger from the GPTW.
(6) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. An S12ADI0 interrupt is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

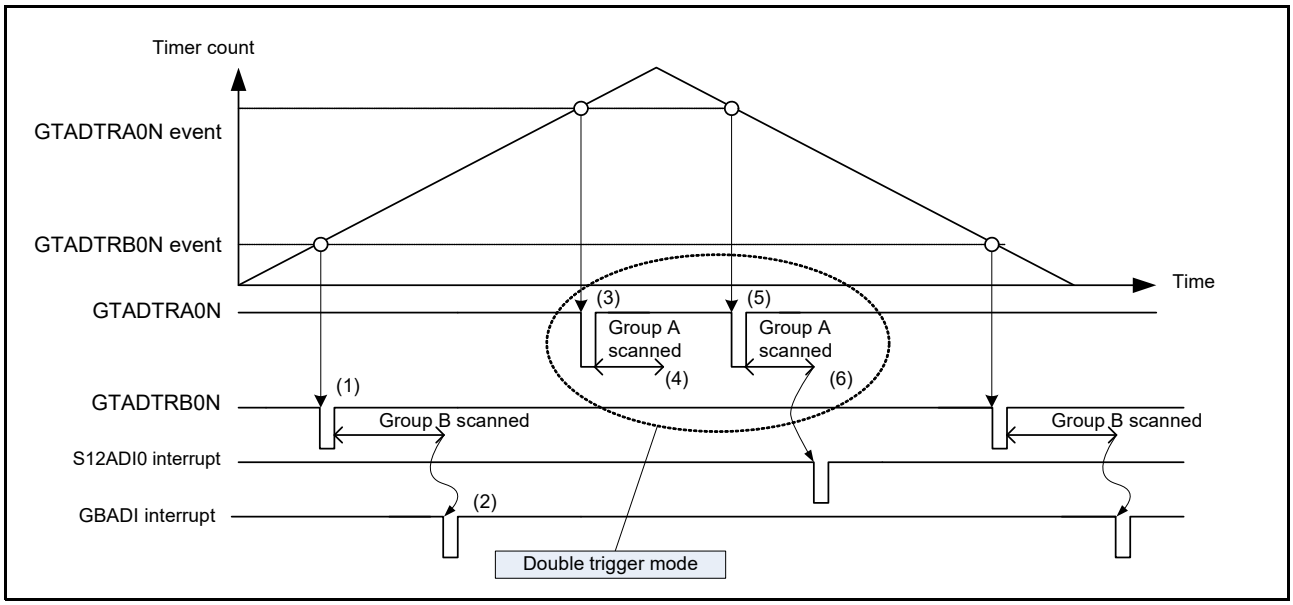


Figure 40.12 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from GPTW Used)

40.3.4.3 Operation under Group-A Priority Control

Setting the PGS bit in the A/D group scan priority control register (ADGSPCR) to 1 in group scan mode makes operation proceed under group-A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 40.13. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.
In operation in basic group scan mode, input of the trigger for the other group during operation for A/D conversion in group A or group B is ignored. Under group-A priority control, if a group-A trigger is input during A/D conversion for group B, A/D conversion for group B is discontinued and A/D conversion for group A proceeds. If the setting of the ADGSPCR.GBRSCN bit is 0, the converter enters a wait state on completion of the A/D conversion for group A. If the setting of the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for group B from the head of the group after completion of the A/D conversion for group A. Table 40.10 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

Scan operations in group A or group B are the same in single scan mode. Furthermore, single scanning continues to proceed if the ADGSPCR.GBRP bit is set to 1 during scanning operations for group B.

For the trigger settings in group scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits and select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[5:0] bits. Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting the ADGSPCR.GBRP bit to 1. Furthermore, as targets for A/D conversion, select channels for group A using the ADANSA0 and ADANSA1 registers, and for group B, select channels different from those for group A using the ADANSB0 and ADANSB1 registers.

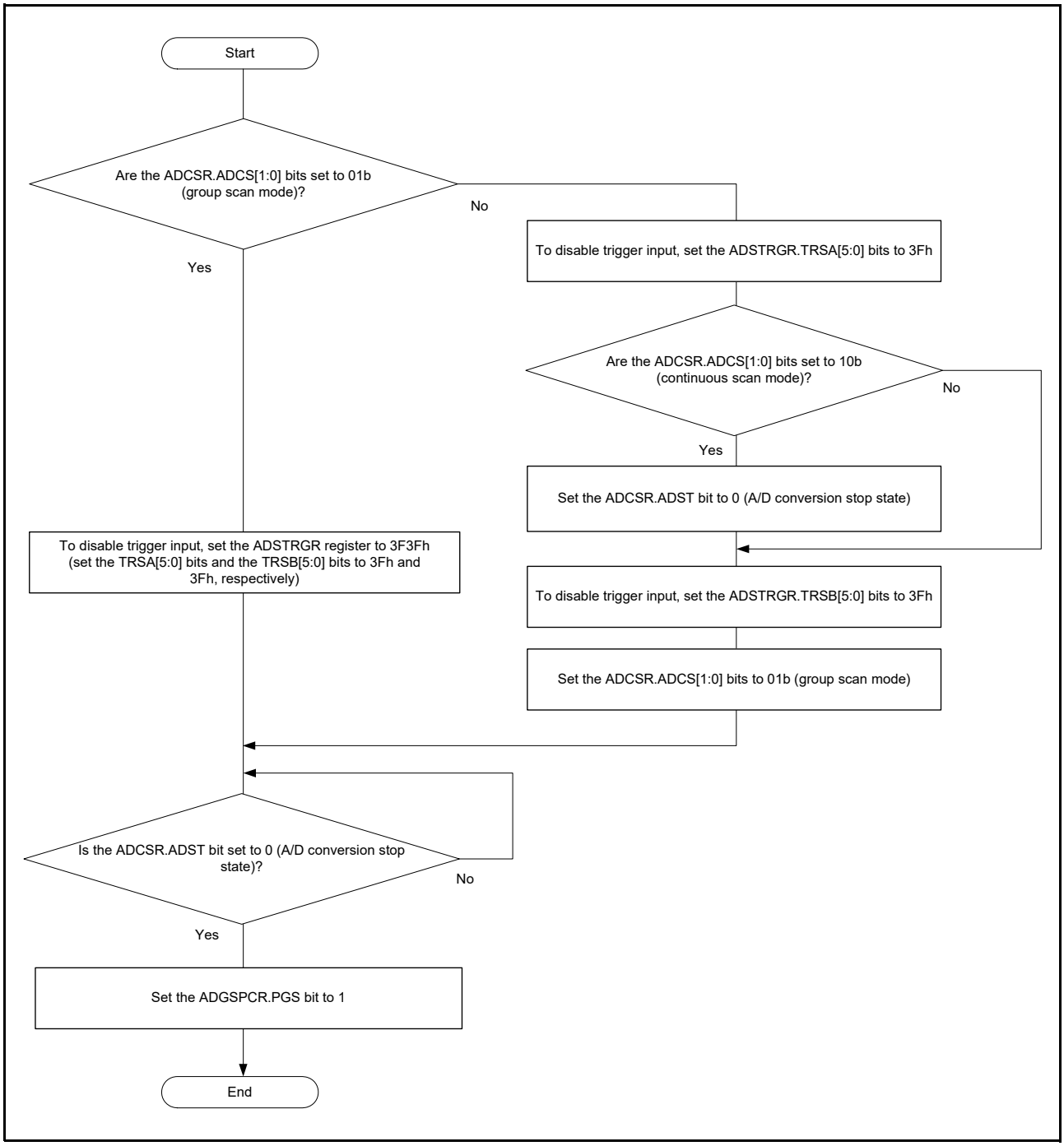


Figure 40.13 Flow of Setting the ADGSPCR.PGS Bit

Table 40.10 Control of A/D Conversion Operations According to the Settings of the ADGSPCR.GBRSCN Bit

A/D Conversion Operation	Trigger Input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group B is discontinued and conversion for group A starts. • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

The following describes the operations in group scan mode under group-A priority control (i.e. ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start), and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(6) After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(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 GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (GBADI interrupt upon group B scanning completion enabled).
(9) The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

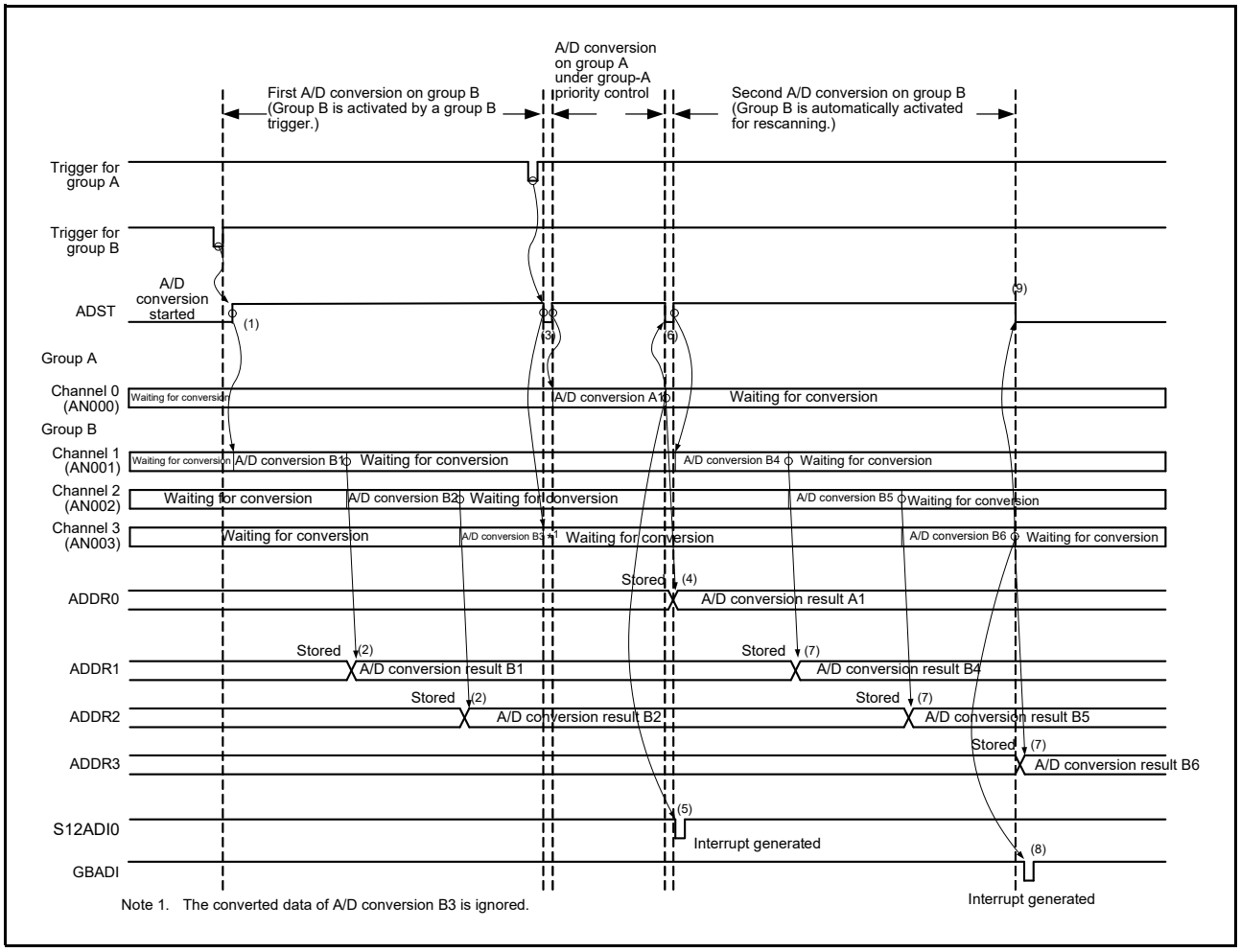


Figure 40.14 Example of Operations under Group-A Priority Control (1) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example when a group A trigger is input again during rescanning operation on group B. In this example, channel 0 is selected for group A and channels 1 to 3 are selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared to 0 (A/D conversion stop) on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued.
(4) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.
(5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(6) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

(7) On completion of A/D conversion on the group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (rescanning operation enabled). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.

(8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(9) If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit is cleared to 0 (A/D conversion stop) and the ongoing A/D conversion on group B is stopped.

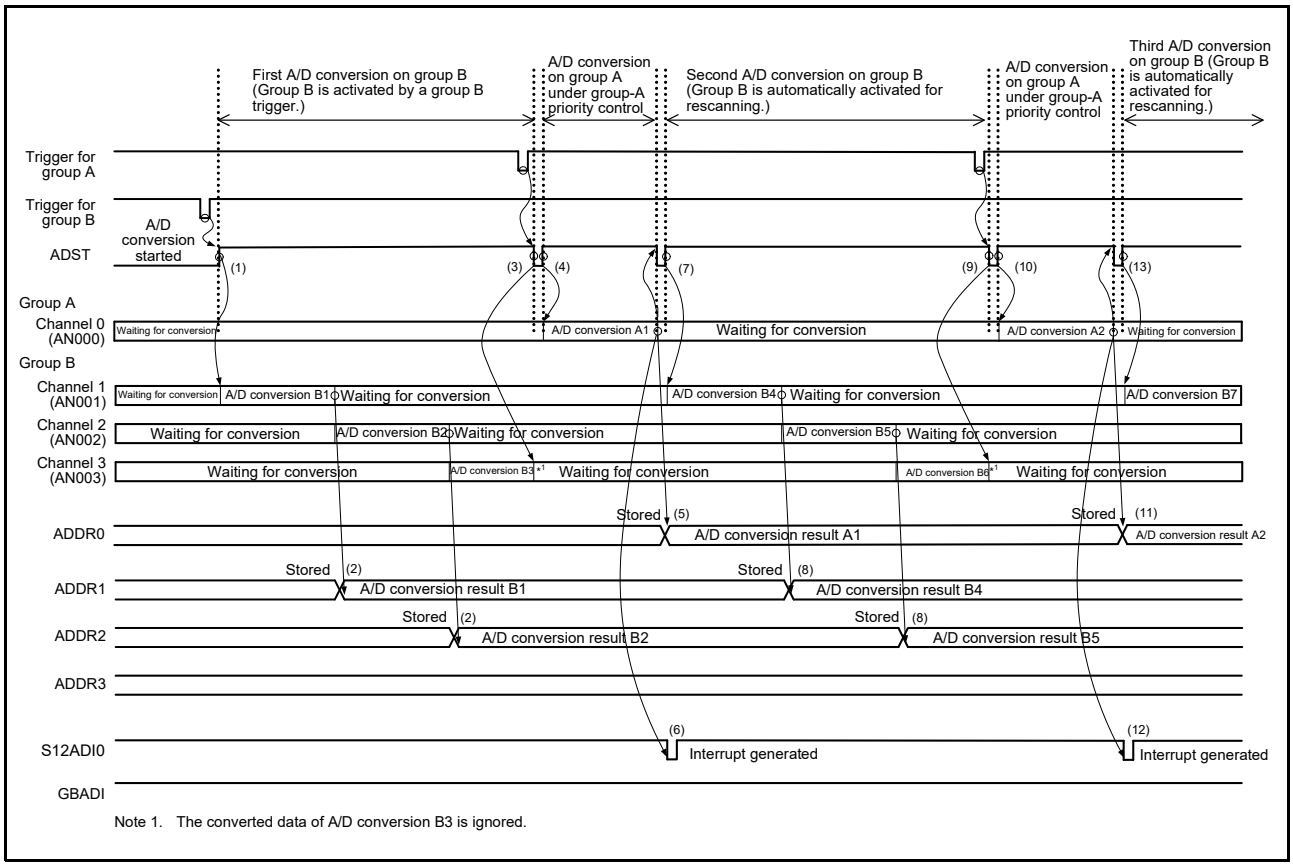
(10) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.

(11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(12) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

(13) On completion of A/D conversion on group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (rescanning operation enabled). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.

(14) If a group A trigger is input during A/D conversion on group B for rescanning, steps 9 to 13 are repeated. If a group A trigger is not input, the ADCSR.ADST bit is cleared automatically on completion of A/D conversion on group B and the 12-bit A/D converter enters a wait state.



Note 1. The converted data of A/D conversion B3 is ignored.

Figure 40.15 Example of Operations under Group-A Priority Control (2) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example of a rescanning operation in which a group B trigger is input during A/D conversion on group A. In this example, channels 1 to 3 are selected for group A and channel 0 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group A sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group B trigger is input during A/D conversion on group A, A/D conversion on group B can be performed after the A/D conversion on group A is completed. (However, if group A triggers are input continuously, the scan operation on group B is canceled by group A and is not performed.)
(4) On completion of the A/D conversion on the group A, an S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(5) On completion of the A/D conversion on the group A, activation of group B for rescanning sets the ADCSR.ADST bit to 1 automatically.
After that, conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(7) On completion of the rescanning operation on the group B, a GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).
(8) The ADST bit retains the value 1 (A/D conversion start) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

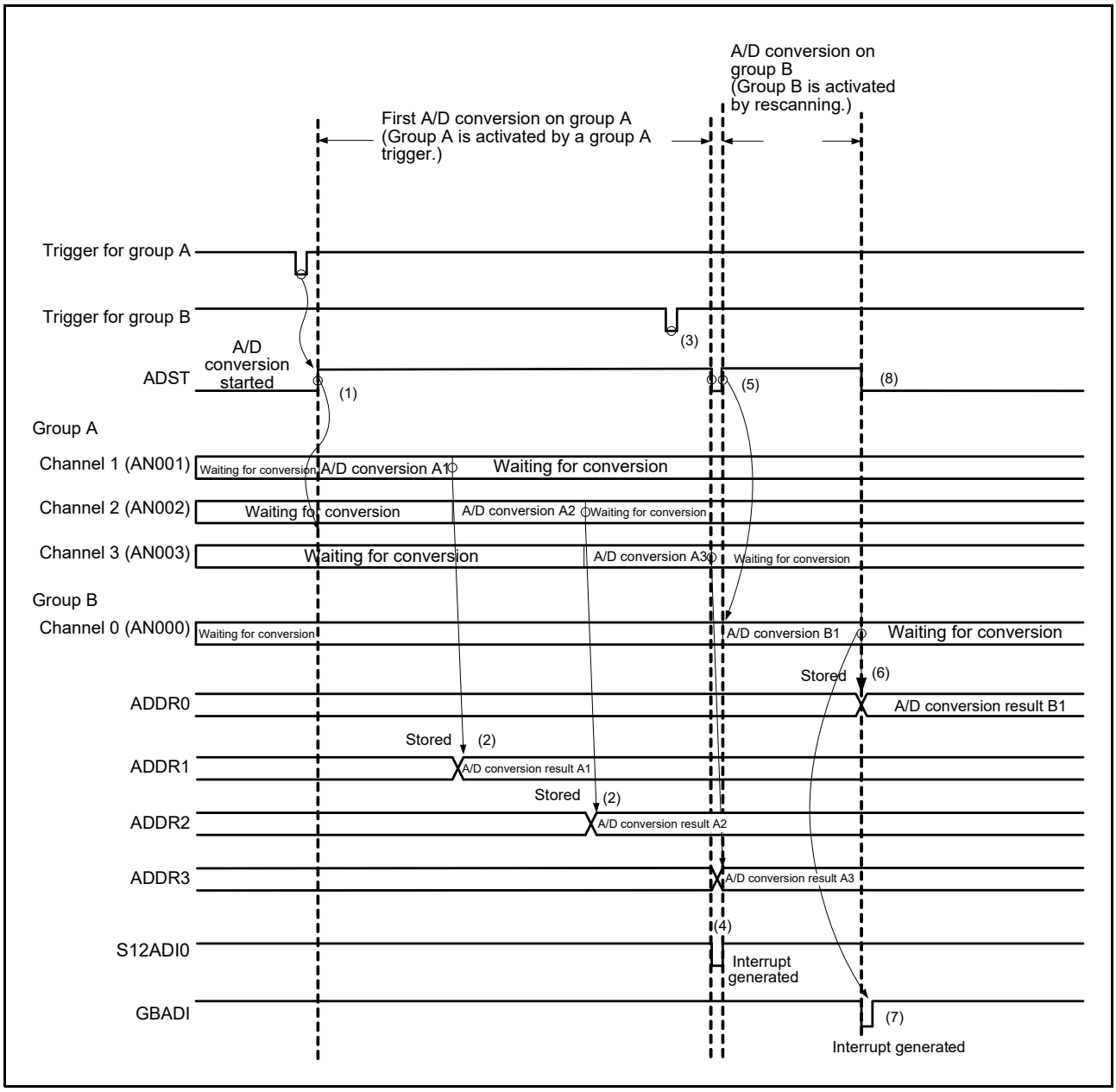


Figure 40.16 Example of Operations under Group-A Priority Control (3) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(6) The ADCSR.ADST bit retains the value 1 (A/D conversion start) during A/D conversion and is cleared on completion of conversion, after which the A/D converter enters a wait state.

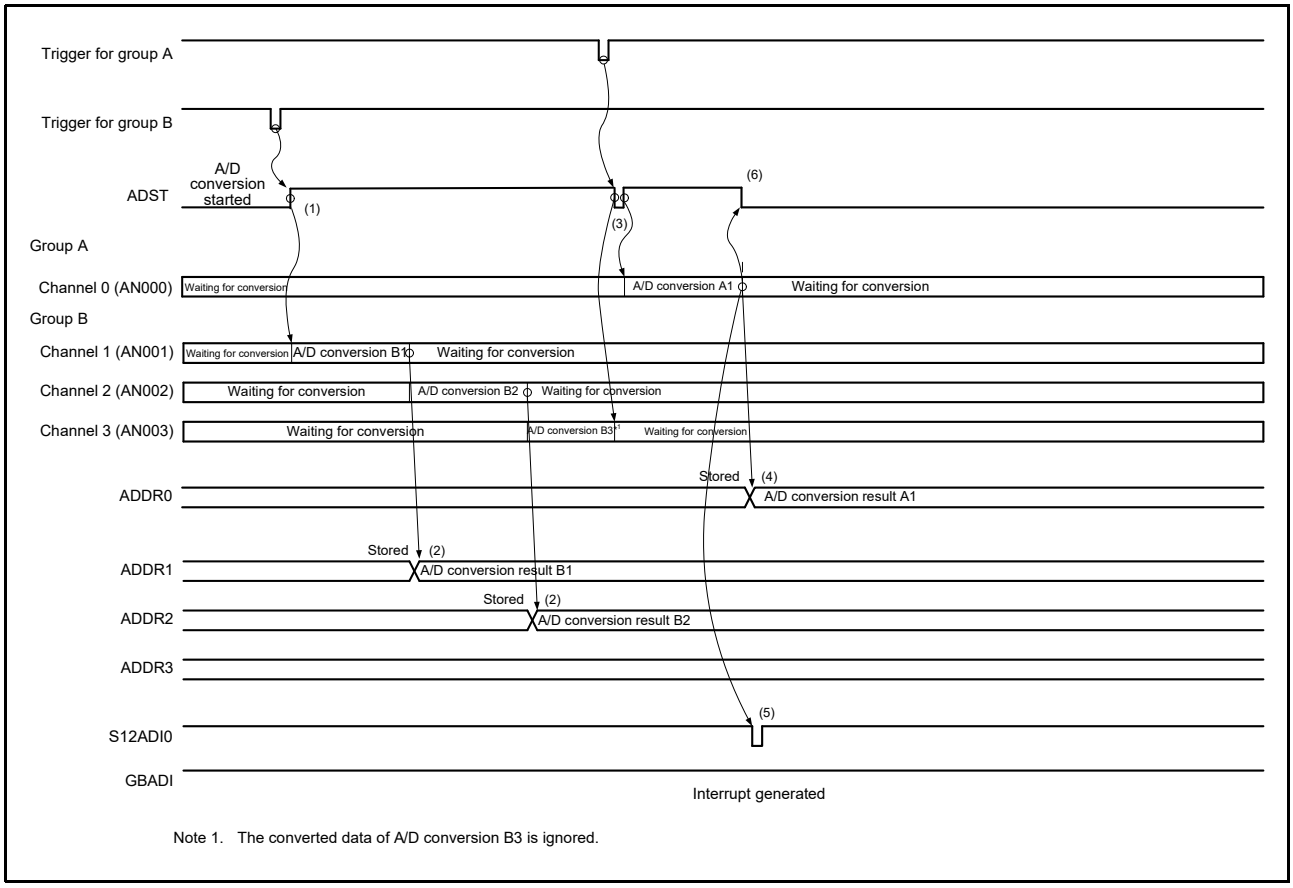
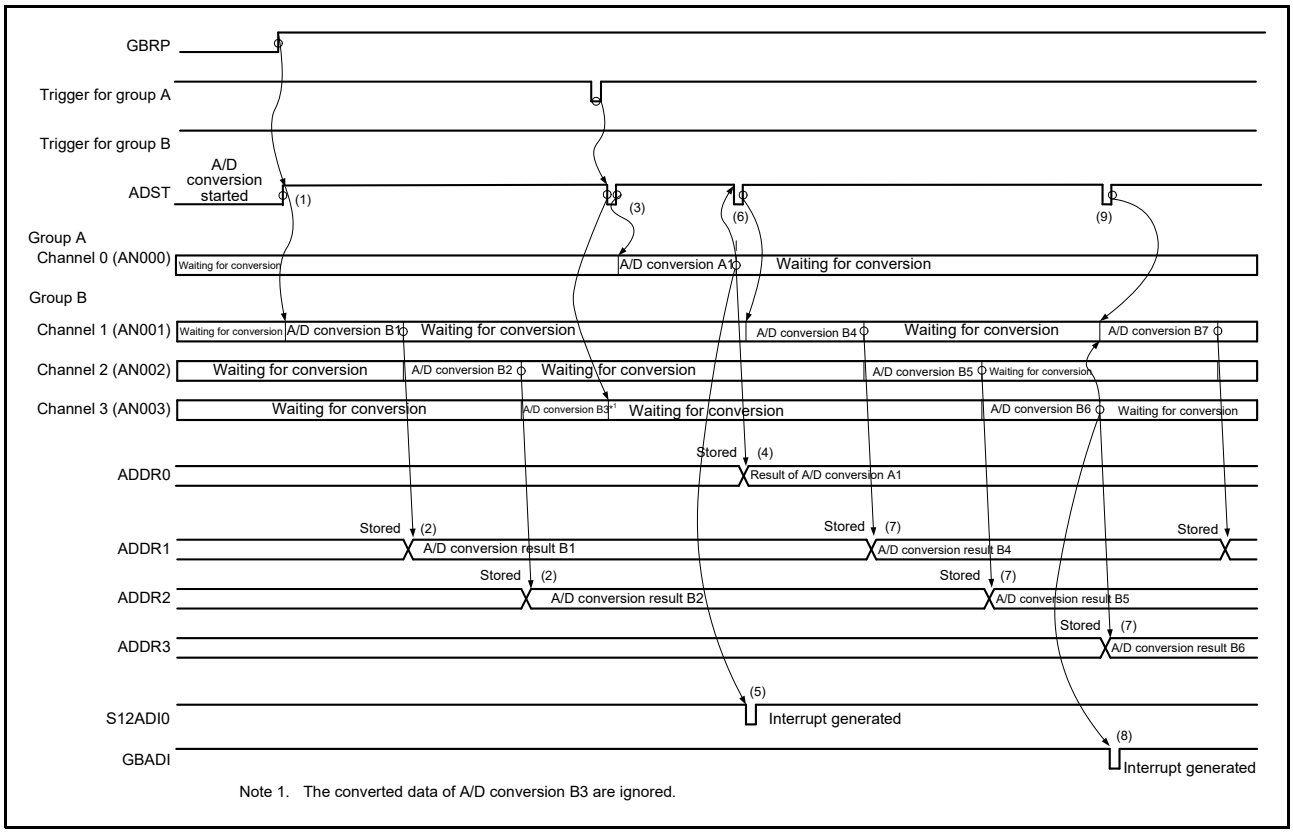


Figure 40.17 Example of Operation under Group-A Priority Control (4) (when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0)

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRP = 1).

(1) The ADCSR.ADST bit is set to 1 (A/D conversion start) when ADGSPCR.GBRP is set to 1, and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI0 interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
(6) After the ADST bit is automatically cleared, again, the ADCSR.ADST bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(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 GBADI interrupt request is generated if the setting of the ADCSR. GBADIE bit is 1.
(9) After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n. 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 A/D conversion when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 40.8.2, Notes on Stopping A/D Conversion.



Note 1. The converted data of A/D conversion B3 are ignored.

Figure 40.18 Example of Operation under Group-A Priority Control (5) (when ADGSPCR.GBRP = 1)

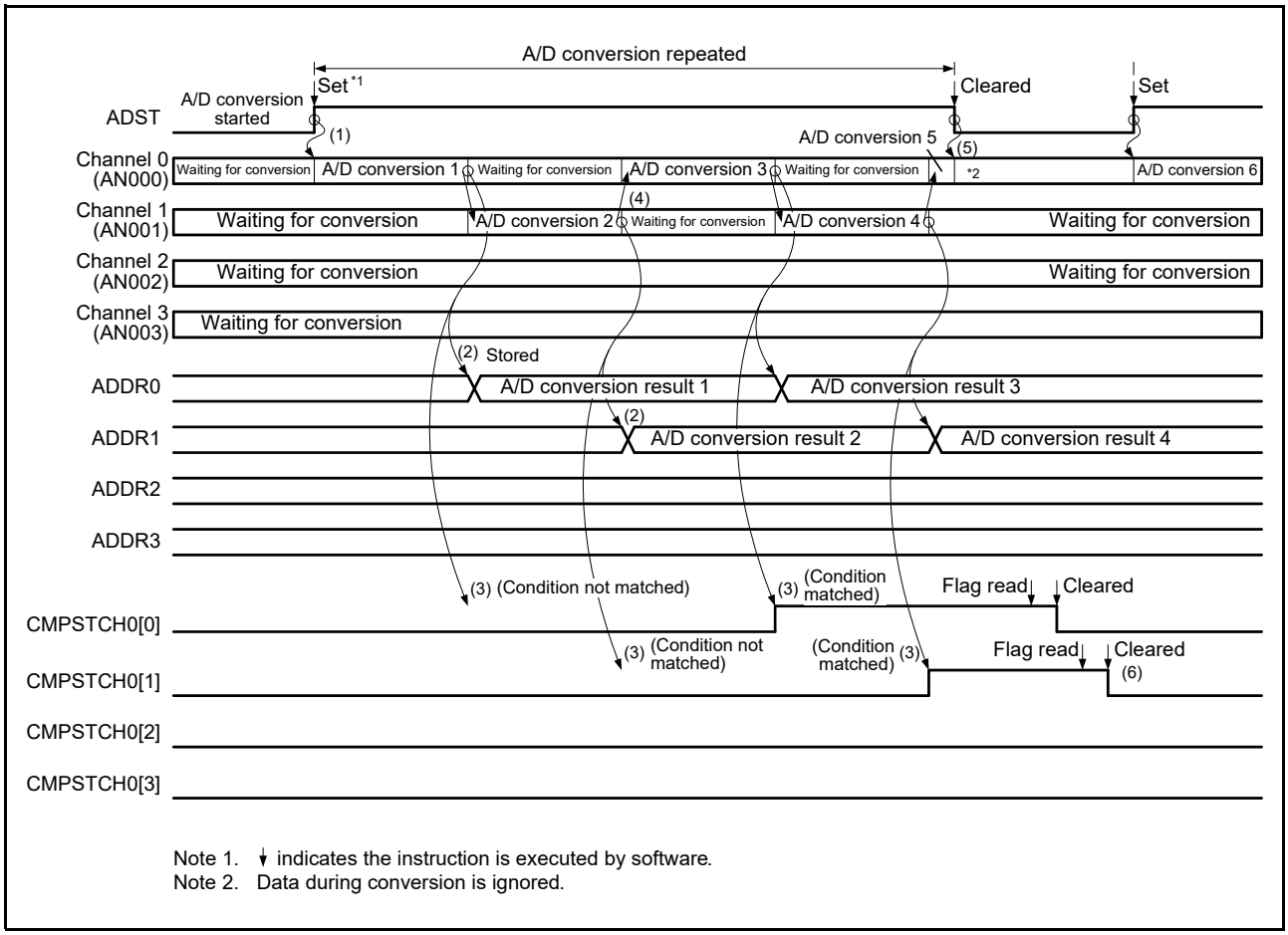
40.3.5 Compare Function (Window A, Window B)

40.3.5.1 Compare Function Window A/B

The compare function compares the reference value set in the register with the A/D conversion result. The reference value can be set for window A and window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. Big differences between window A and window B are different interrupt output signals and that window B can select only one channel.

The following describes operations in combination of continuous scan mode and the compare function.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input A/D conversion starts in the order of the selected channel.
(2) Upon completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE is 1, if the ADCMPANSR0, ADCMPANSR1, or ADCMPANSER register is set for window A, the results of A/D conversion are to be compared with the values set in the ADCMPDR0 and ADCMPDR1 registers. When ADCMPCR.CMPBE is 1, if the ADCMPBNSR register is set for window B, the results of A/D conversion are to be compared with the values set in the ADWINULB and ADWINLLB registers.
(3) As a result of the comparison, when window A meets the condition set in ADCMPDR0, ADCMPDR1 or ADCMPLER, the compare window A flag (ADCMPSR0.CMPSTCHA0n, ADCMPSR1.CMPSTCHA1n, ADCMPSER.CMPSTTSA, or ADCMPSER.CMPSTOCA) is set to 1. In the same way, when window B meets the condition set in ADCMPBNSR.CMPLB, the compare window B flag (ADCMPBSR.CMPSTB) is set to 1.
(4) Upon completion of all selected A/D conversions and comparisons, scan restarts.
(5) Set the ADCSR.ADST bit to 0 (A/D conversion stop), and execute processing for the channel with the compare flag set to 1.
(6) Clear all compare flags after processing is competed. To perform comparison again, restart A/D conversion.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data during conversion is ignored.

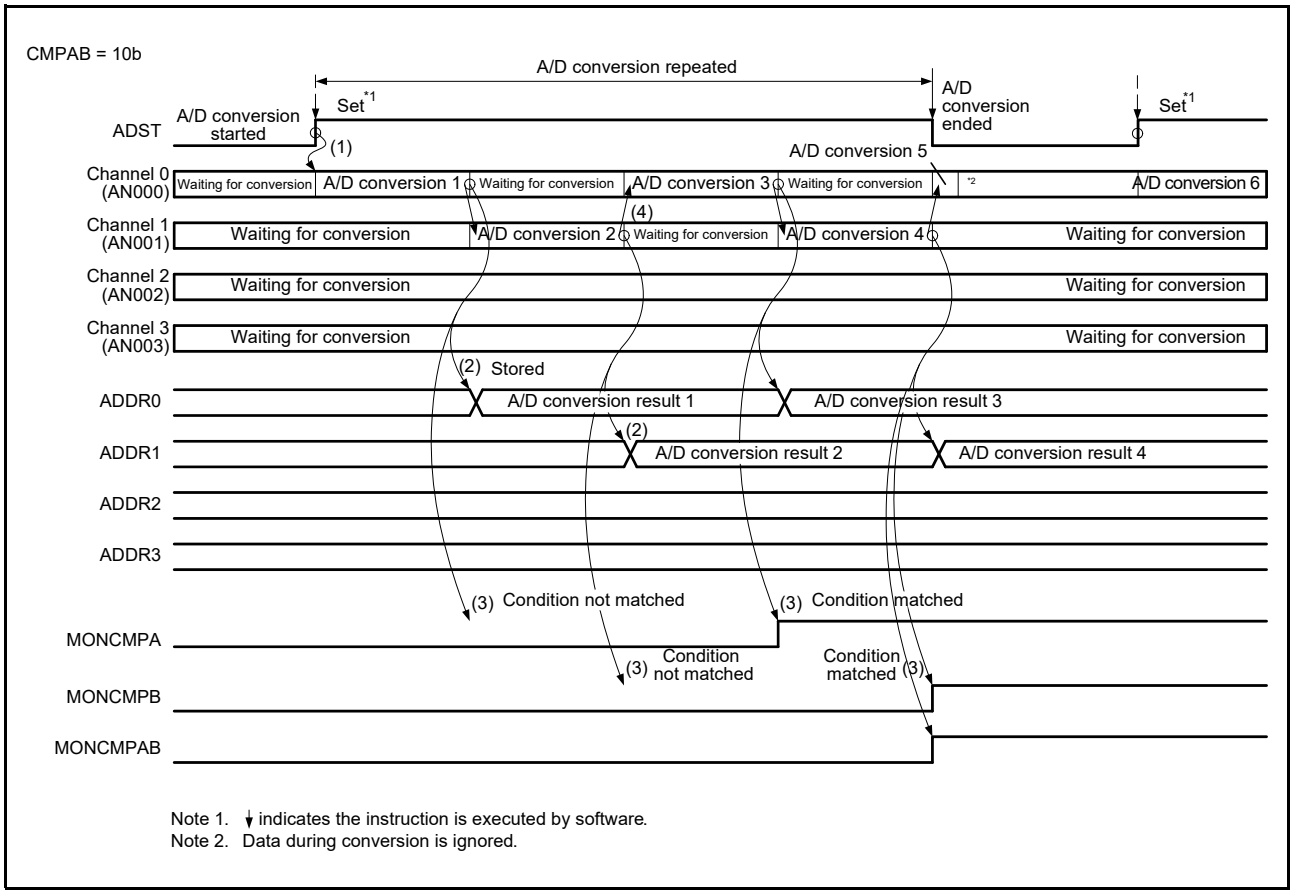
Figure 40.19 Operation Example of Comparison (AN000 to AN003 Compared)

40.3.5.2 ELC Output of Compare Function

The ELC output of the compare function is used to specify the high-side reference value and the low-side reference value for window A and window B respectively, and to compare the A/D converted value of the selected channel with the high/low-side reference value. Depending on whether the comparison conditions for window A and window B are met or not met, the ELC event (S12ADWMELC/S12ADWUMELC) is output according to the event conditions (A or B, A and B, A exor B).
If multiple channels are selected for window A, when the comparison conditions for any of the channels are met, it is recognized that the comparison conditions for window A are met.

When using this function, perform A/D conversion in single scan mode.
Any channels from AN000 to AN008, AN016 to AN031, temperature sensor output, and internal reference voltage are selectable for window A.
However, when selecting the internal reference voltage or the temperature sensor output, it cannot be selected together with any other channel. Any channels from AN000 to AN008, AN016 to AN031, temperature sensor output, and internal reference voltage are selectable for window B.
The setting procedure is as follows when this function is to be used. The setting procedure required for normal A/D conversion in single scan mode is omitted.

(1) Confirm that the value of the ADCSR.ADCS[1:0] bits is 00b (single scan mode).
(2) Select channels (from among AN000 to AN008, AN016 to AN031, temperature sensor, and internal reference voltage) in the ADCMPANSR0, ADCMPANSR1, or ADCMPANSER register (for window A) and in the ADCMPBNSR register (for window B).
(3) Set window comparison conditions in the ADCMPLR0, ADCMPLR1, ADCMPLER, and ADCMPBNSR registers, and set the upper-limit and lower-limit reference values in the ADCMPDR0, ADCMPDR1, ADWINULB, and ADWINLLB registers.
(4) Set composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register. A scan end event (S12ADELC) is output to the ELC at the end of each single scan. In addition, a match or mismatch event (S12ADWMELC or S12ADWUMELC) is output with a delay of one PCLKB cycle depending on the ADCMPCR.CMPAB[1:0] setting.
Since match and mismatch events are mutually exclusive, these are not output at the same time.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data during conversion is ignored.

Figure 40.20 Example of Window Compare Function Operation (AN000 to AN003 Compared)

40.3.5.3 Using Data Buffers

This S12ADE is provided with a ring buffer function consisting of 16 A/D data buffers. This function sequentially stores A/D conversion results other than self-diagnosis result (including addition/average results) in data buffers (ADBUFn, n = 0 to 15) when the compare function is used.

Each conversion result is stored at the timing when the A/D conversion result is stored in the data register, and most recent 16 conversion result data are retained.

The following shows the schematic of data buffers, pointer, and overflow flag operations. When the BUFEN bit is set to1, the A/D conversion result is transferred at each end of A/D conversion. The pointer indicates the number of data buffer to which the next transferred data is to be written. When data is written to up to buffer 15, the pointer is reset to 0000b and the overflow flag is set to 1. Subsequently transferred data overwrites the previously written data. The pointer and overflow flag are reset to the initial value by writing 00h to the ADBUFPTR register.

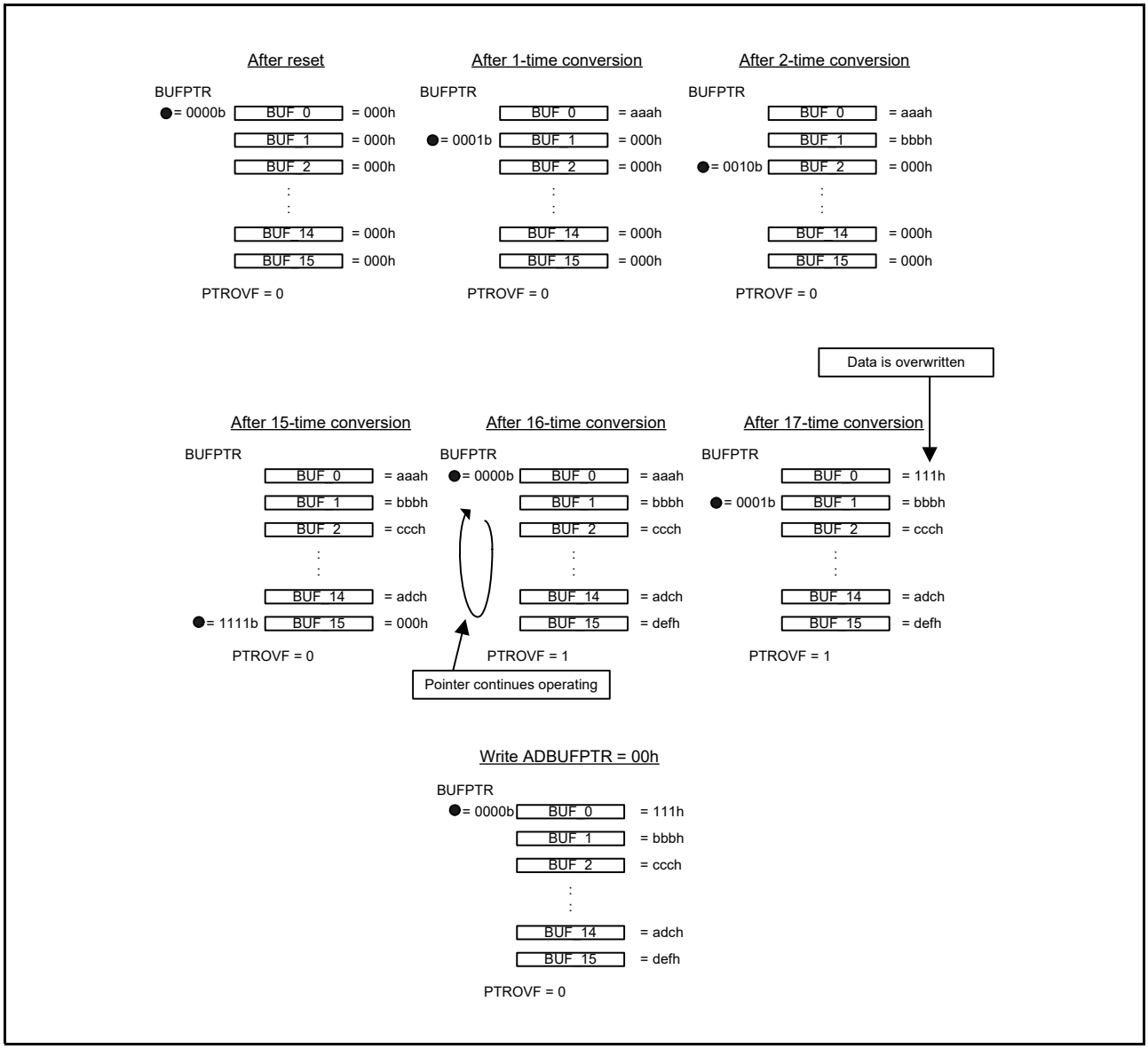


Figure 40.21 Schematic of Data Buffers, Pointer, and Overflow Flag Operations

40.3.5.4 Restrictions for Compare Function

The following restrictions are provided for the compare function.

1. The compare function must not be used together the self-diagnosis function or double trigger mode. (The compare function is not available for the ADRD register and the ADDBLDR register.)
2. Specify single scan mode when using match/mismatch event outputs.
3. When temperature sensor or internal reference voltage is selected for window A, window B operations are disabled.
4. When temperature sensor or internal reference voltage is selected for window B, window A operations are disabled.
5. It is prohibited to set the same channel for window A and window B.
6. When using the buffer function, specify single scan mode. (It is also prohibited to use double trigger mode together.)
7. Set the reference voltage values so that the high-side reference voltage value is equal to or larger than the low-side reference voltage value.

40.3.6 Analog Input Sampling Time and Scan Conversion Time

Scan conversion can be activated either by software, synchronous trigger, or asynchronous trigger input. After the start-of-scanning-delay time (t_D) has elapsed, processing for disconnection detection assistance and processing of conversion for self-diagnosis proceed, and this is followed by processing for A/D conversion.
Figure 40.22 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software or a synchronous trigger. Figure 40.23 shows the scan conversion timing in single scan mode, in which scan conversion is activated by an asynchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), disconnection detection assistance processing time (t_{DIS})*1, self-diagnosis A/D conversion processing time (t_{DIAG})*2, A/D conversion processing time (t_{CONV}), and end-of-scanning-delay time (t_{ED}).
The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.
The time for conversion by successive approximation (t_{SAM}) is at 32 ADCLK states during high-speed conversion operation, and 41 ADCLK states during low-current conversion operation. Table 40.11 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{*3} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED}.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $(t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)$.

Note 1. When disconnection detection assistance is not selected, $t_{DIS} = 0$. The auto-discharge period of 15 ADCLK states is inserted only when the temperature sensor or internal reference voltage is A/D-converted.
Note 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.
Note 3. $t_{CONV} \times n$ when the sampling time (t_{SPL}) of selected channels is the same, but it is the total of the sampling time of each channel and time for conversion by successive approximation (t_{SAM}).

Table 40.11 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLKB)

Item			Symbol		Type/Conditions: Synchronous Trigger *5	Type/Conditions: Asynchronous Trigger	Type/Conditions: Software Trigger	Unit
Scan start processing time*1, *2	A/D conversion on group A under group-A priority control.	Group B is to be stopped. (Group A is activated after group B is stopped due to an A/D conversion source of group A.)	t_D		3 PCLKB + 6 ADCLK	—	—	Cycle
		Group B is not to be stopped. (Activation by an A/D conversion source of group A.)			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*3			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Dh) × ADCLK*4			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	32 ADCLK (during high-speed conversion operation and Conversion Cycle Select bit is 0)			
					41 ADCLK (during low-current conversion operation and Conversion Cycle Select bit is 0)			
					22 ADCLK (during high-speed conversion operation and Conversion Cycle Select bit is 1)			
					28 ADCLK (during low-current conversion operation and Conversion Cycle Select bit is 0)			
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.			t_{DED}	2 ADCLK			
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.			t_{DSD}	2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 8, L, T, O) (initial value = 0Dh) × ADCLK*4			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	32 ADCLK (during high-speed conversion operation and Conversion Cycle Select bit is 0)			
					41 ADCLK (during low-current conversion operation and Conversion Cycle Select bit is 0)			
					22 ADCLK (during high-speed conversion operation and Conversion Cycle Select bit is 1)			
					28 ADCLK (during low-current conversion operation and Conversion Cycle Select bit is 0)			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK*6			

Note 1. For t_D, t_{DIAG}, t_{CONV}, and t_{ED}, see Figure 40.22 and Figure 40.23.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The value is fixed to 0Fh (15 ADCLK) when the temperature sensor output or internal reference voltage is A/D-converted.
Note 4. The required sampling time (ns) is specified according to the voltage conditions. See section 47.7, A/D Conversion Characteristics.
Note 5. This does not include the time consumed in the path from timer output to trigger input.
Note 6. 2 PCLKB + 2 ADCLK when ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2).

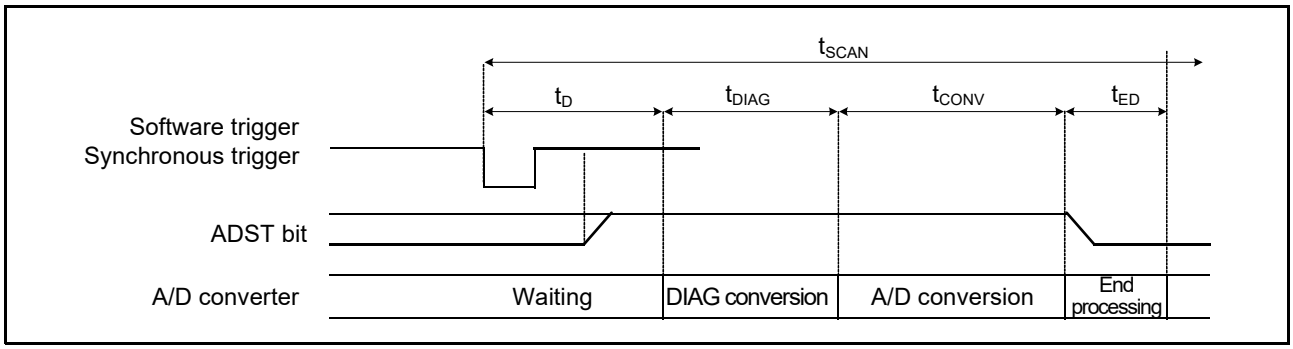


Figure 40.22 Scan Conversion Timing (Activated by Software or Synchronous Trigger)

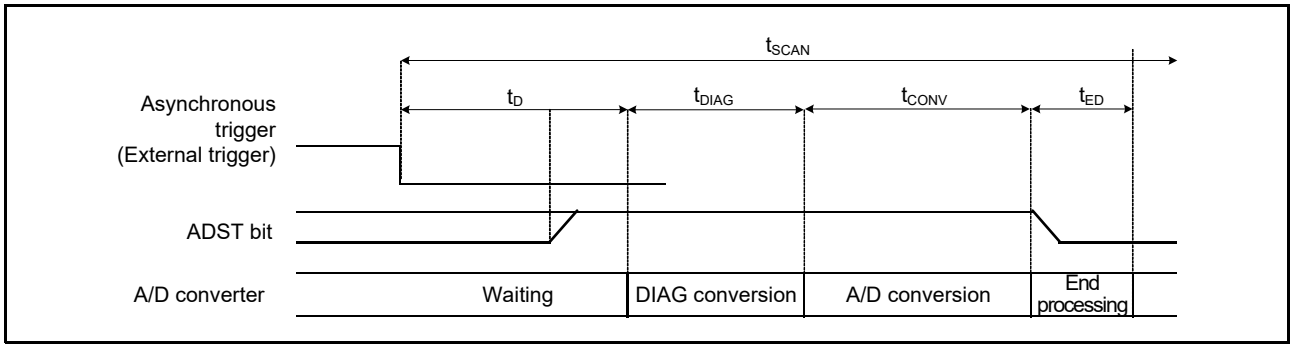


Figure 40.23 Scan Conversion Timing (Activated by Asynchronous Trigger)

40.3.7 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) to 0000h when the A/D data registers are read by the CPU, DTC, or DMAC.

The ring buffer (ADBUFn: n = 0 to 15) is not subject to auto-clearing.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

40.3.8 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 or average mode can be specified when A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage is selected.

40.3.9 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 40.24 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 40.25 shows an example of disconnection detection when precharge is selected. Figure 40.26 shows an example of disconnection detection when discharge is selected.

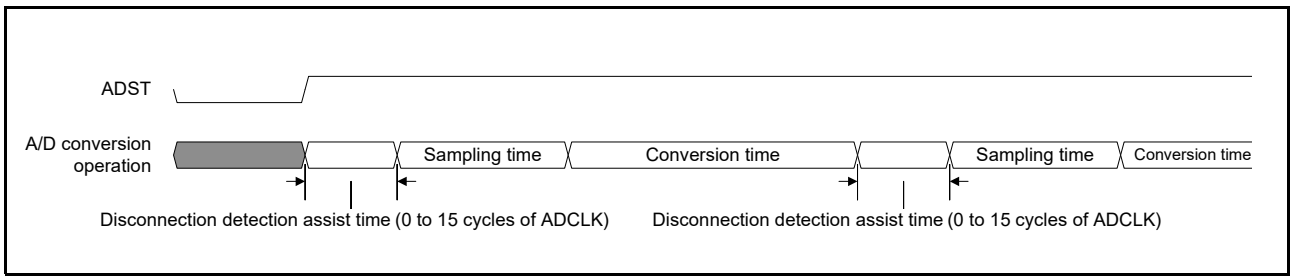
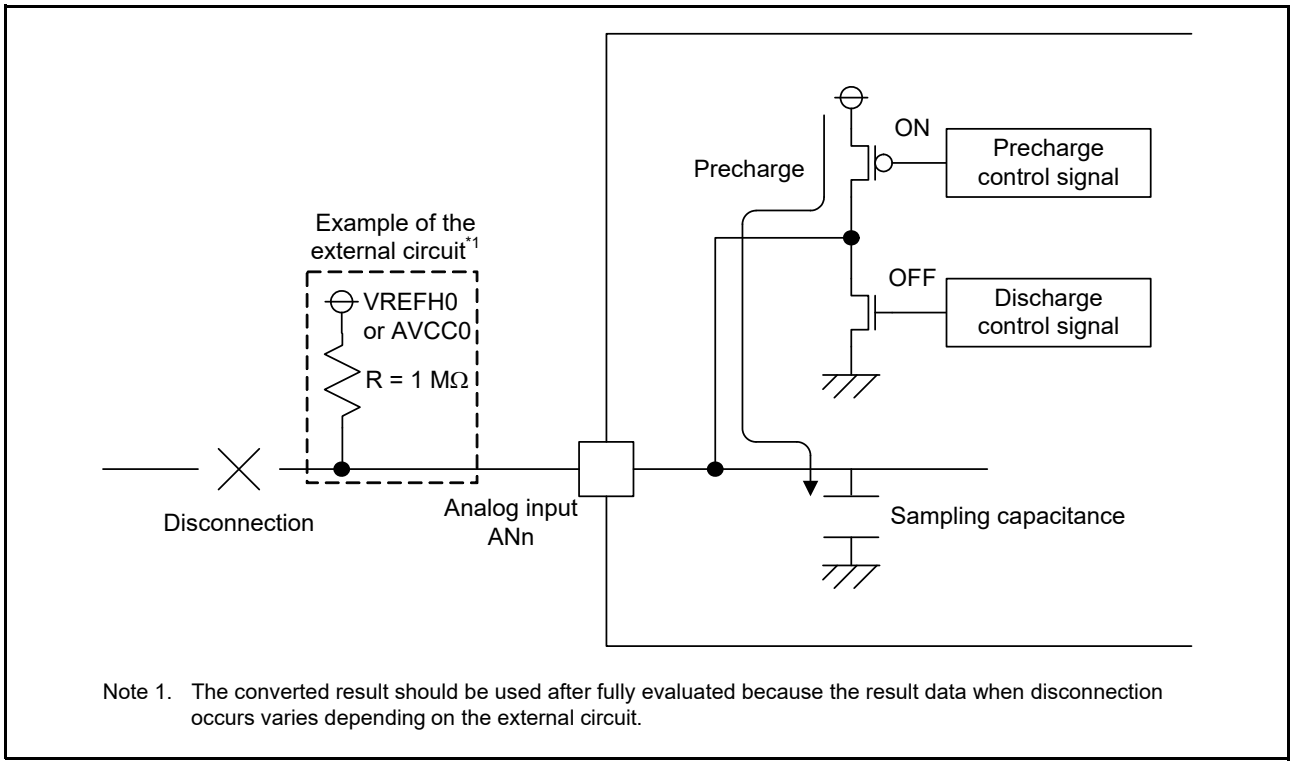
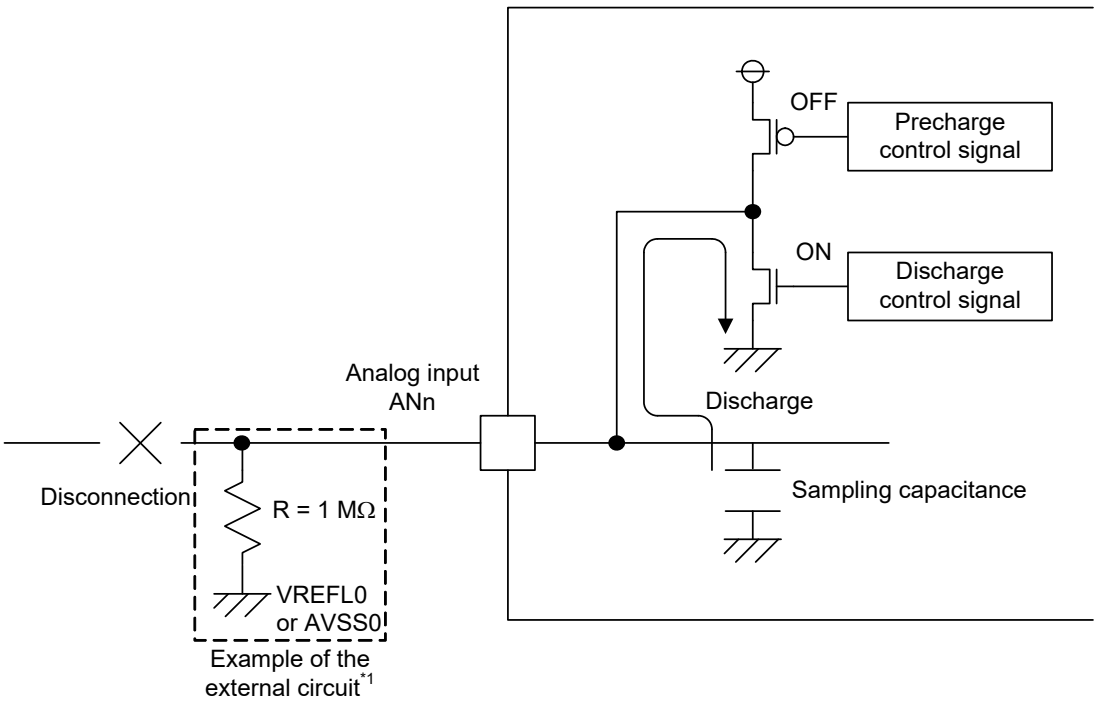


Figure 40.24 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 40.25 Example of Disconnection Detection When Precharge is Selected



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 40.26 Example of Disconnection Detection When Discharge is Selected

40.3.10 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 40.27 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 40.8.3, A/D Conversion Restarting Timing and Termination Timing. An asynchronous trigger cannot be selected for group B used in group scan mode.

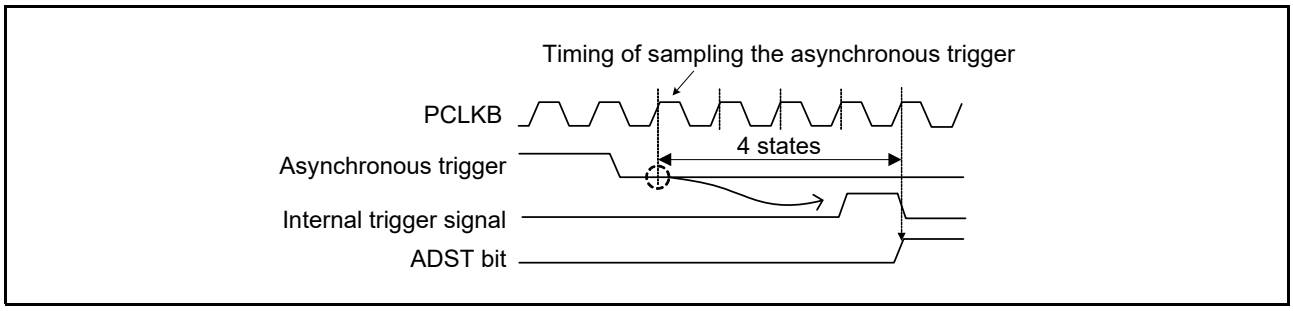


Figure 40.27 Timing of Sampling Asynchronous Trigger

40.3.11 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.

40.4 Interrupt Sources and DTC/DMA Transfer Requests

40.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests S12ADI0 and GBADI to the CPU.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI0 interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a GBADI interrupt, respectively.
In addition, the DTC or DMAC can be activated when an S12ADI0 or a GBADI interrupt is generated. Using an S12ADI0 or a GBADI interrupt to allow the DTC or DMAC to read the converted data enables continuous conversion without burden on software.
For details on DTC settings, see section 18, Data Transfer Controller (DTCb), and for details on DMAC settings, see section 17, DMA Controller (DMACA).

40.5 Event Link Function

40.5.1 Event Output to the ELC

The ELC uses the S12ADI0 interrupt request signal as an event signal (S12ADELC), enabling link operation for the preset module. An event signal is generated under the conditions set by the event link control bits (ADELCCR.ELCC[1:0] bits).
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit.
The 12-bit A/D converter outputs the A/D conversion end event (S12ADELC), window function compare match event (S12ADWMELC), and mismatch event (S12ADWUMELC).
The scan end event (S12ADELC) is output to the ELC at the same time as the interrupt output (S12ADI0) regardless of the ADCSR.ADIE setting.
The compare match/mismatch event (S12ADWMELC/S12ADWUMELC) is output to the ELC with a delay of one PCLKB cycle from the interrupt output (S12ADI0) regardless of the ADCSR.ADIE setting.
When using compare match/mismatch events (S12ADWMELC/S12ADWUMELC) to the ELC, specify single scan mode.

40.5.2 12-Bit A/D Converter Operation by Event from the ELC

The 12-bit A/D converter can be started by the predetermined event by setting the ELSRn register of the ELC.

40.5.3 Note on 12-Bit A/D Converter When an Event Is Input from the ELC

If an event occurs during A/D conversion, the event is disabled.

40.6 Selecting Reference Voltage

For the A/D converter, the high-potential reference voltage can be selected from VREFH0 and AVCC0, and the low-potential reference voltage can be selected from VREFL0 and AVSS0, respectively. Set these before starting A/D conversion. For details of this setting, refer to section 40.2.30, A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT).

40.7 Allowable Impedance of Signal Source

Figure 40.28 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.

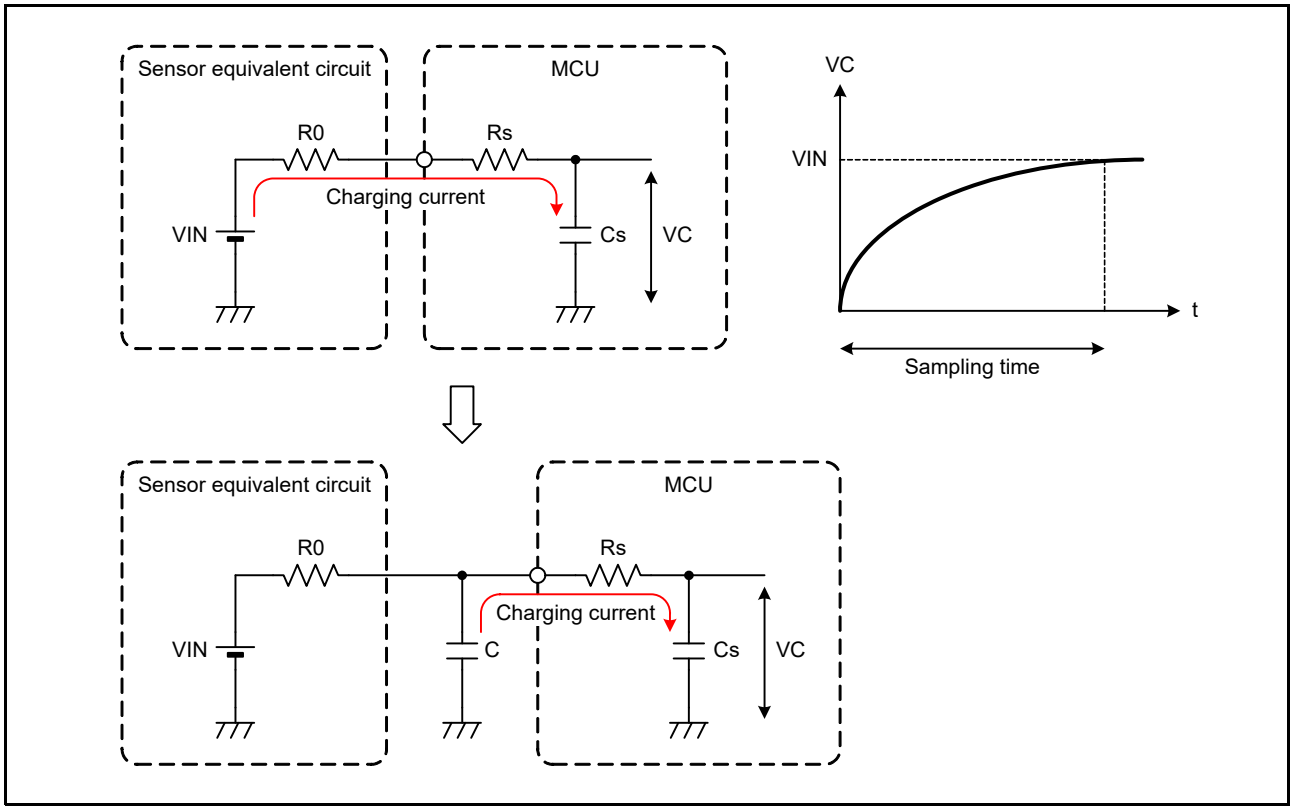


Figure 40.28 Equivalent Circuit of Analog Input Pin and External Sensor

40.8 Usage Notes

40.8.1 Notes on Reading Data Registers

Read the A/D data registers, A/D data duplication register, 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 in word units. If a register is read twice in byte 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 byte units.

40.8.2 Notes on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in Figure 40.29.

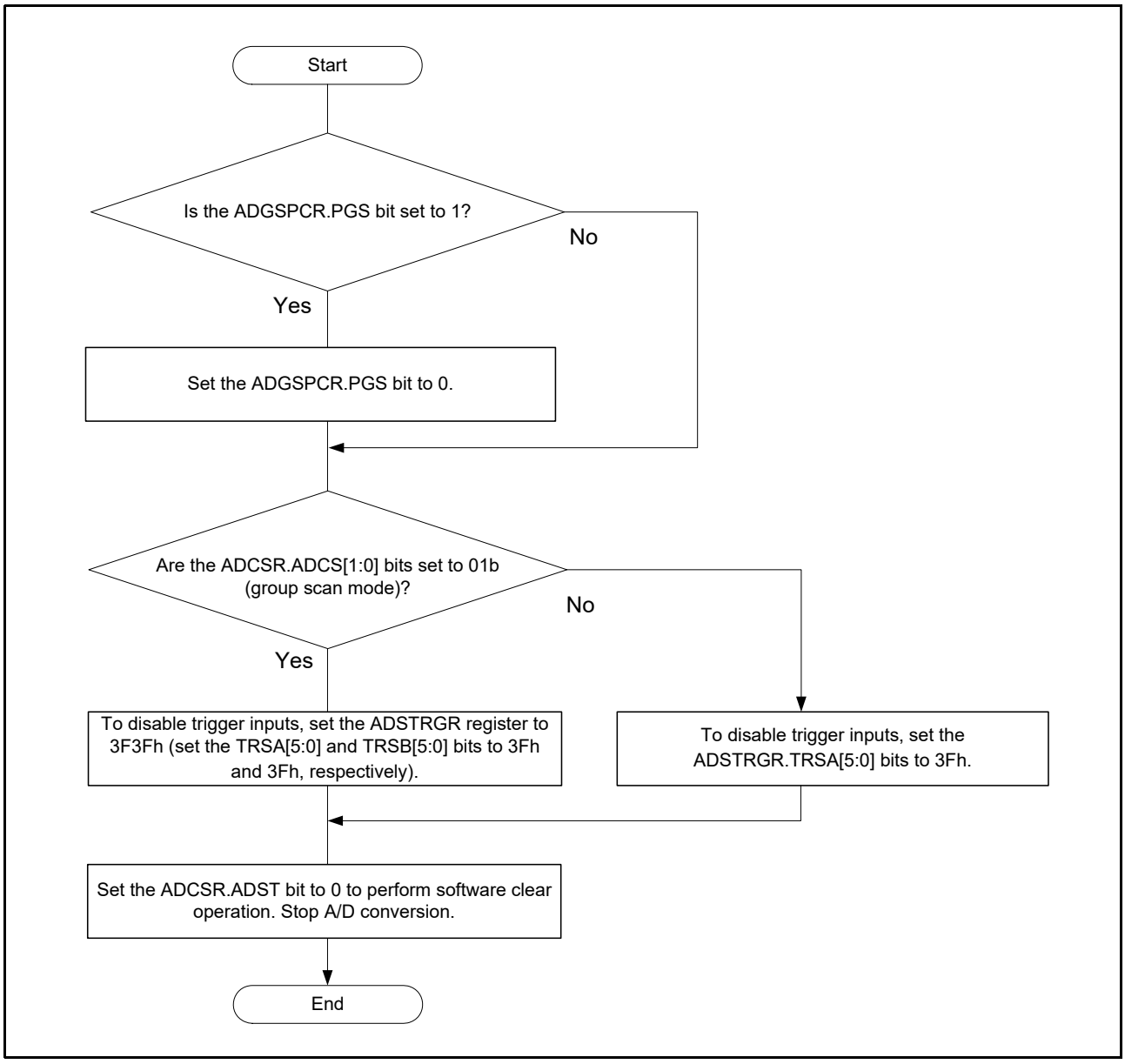


Figure 40.29 Procedure for Clear Operation by Software through the ADCSR.ADST Bit

40.8.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of three ADCLK cycles (1 PCLKB + 2 ADCLK when ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2)) for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

40.8.4 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.

40.8.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register A (MSTPCRA). The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the module stop state.
After the module stop state is released, wait for 1 µs to start A/D conversion. For details, refer to section 11, Low Power Consumption.

40.8.6 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.
Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in Figure 40.29. After that, wait for two clock cycles of ADCLK before entering the peripheral module stop state or software standby mode.

40.8.7 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1 µs before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

40.8.8 Pin Setting when the 12-bit A/D Converter is Used

When the 12-bit A/D converter is used, do not set any pin of port 4 as output.
Output from any of the pins may affect on A/D conversion accuracy because analog power supply is used in the part of the port 4 circuit.

40.8.9 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

40.8.10 ADHSC Bit Rewriting Procedure

Before rewriting the A/D conversion select bit (ADCSR.ADHSC) from 0 to 1 or from 1 to 0, the 12-bit A/D converter must be in the standby state. Carry out steps 1 to 3 below to modify the ADCSR.ADHSC bit. After the sleep bit (ADHVREFCNT.ADSLP) is cleared to 0, wait for at least 1 µs and then start A/D conversion.

ADHSC Bit Rewriting Procedure:

1. Set the sleep bit (ADHVREFCNT.ADSLP) to 1.
2. Wait for at least 0.2 µs, and then modify the A/D conversion select bit (ADCSR.ADHSC).
3. Wait for at least 4.8 µs, and then clear the sleep bit (ADHVREFCNT.ADSLP) to 0.

Note: It is prohibited to set the ADHVREFCNT.ADSLP bit to 1 except for modifying the A/D conversion select bit (ADCSR.ADHSC).

40.8.11 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage applied to analog input pins ANn: AVSS0 ≤ VAN ≤ AVCC0
  Reference voltage range applied to pins VREFH0 and VREFL0: VREFH0 ≤ AVCC0, VREFL0 = AVSS0
  Conversion will not succeed if the voltage applied to analog input pins ANn is greater than VREFH0 (see Figure 40.30).

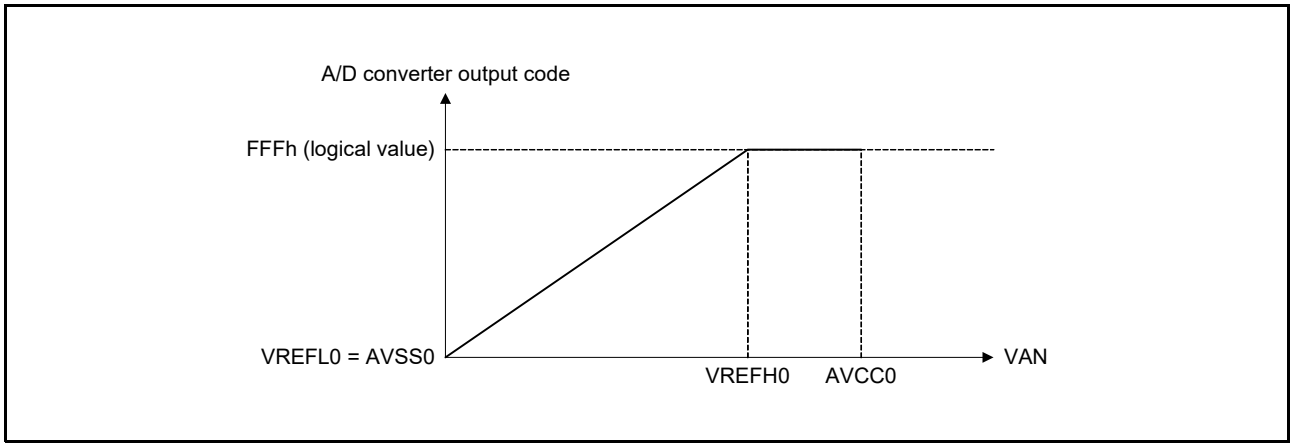


Figure 40.30 Relationship Between Voltage Applied to Analog Input Pins and Output Code

- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)
  The following condition should be satisfied: AVSS0 = VSS. When performing A/D conversion of analog input pins AN016 to AN031, the following condition should be satisfied: AVCC0 = 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 40.31, and connection should be made so that the following conditions are satisfied at the supply side.
  VREFL0 = AVSS0 = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied.
  AVCC0 = VCC and AVSS0 = VSS

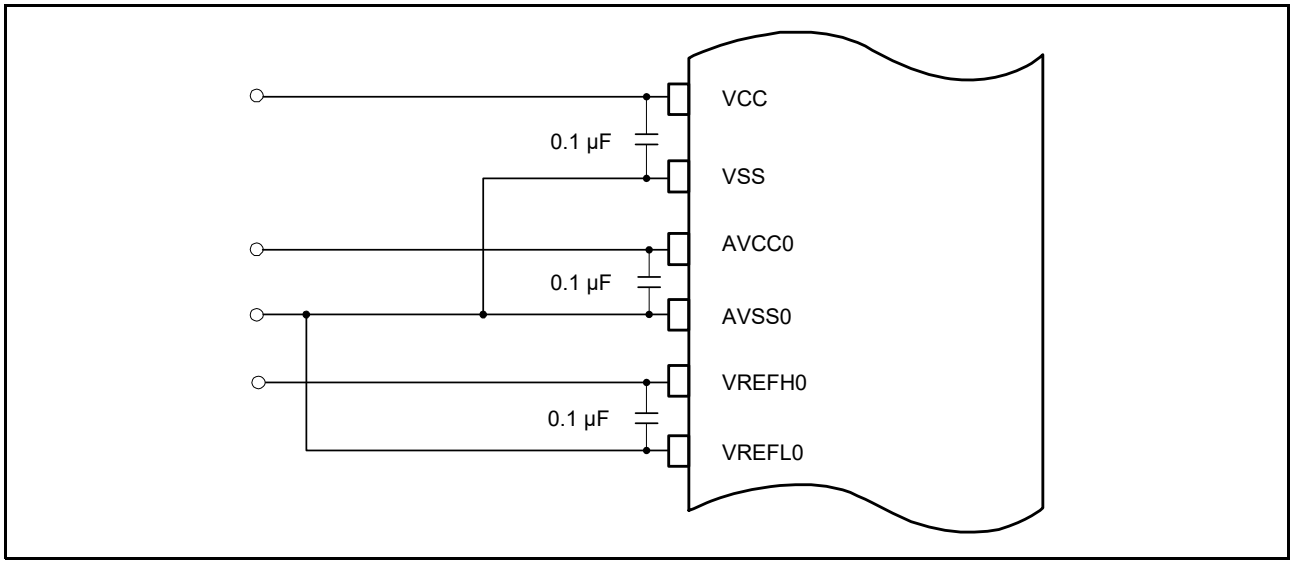


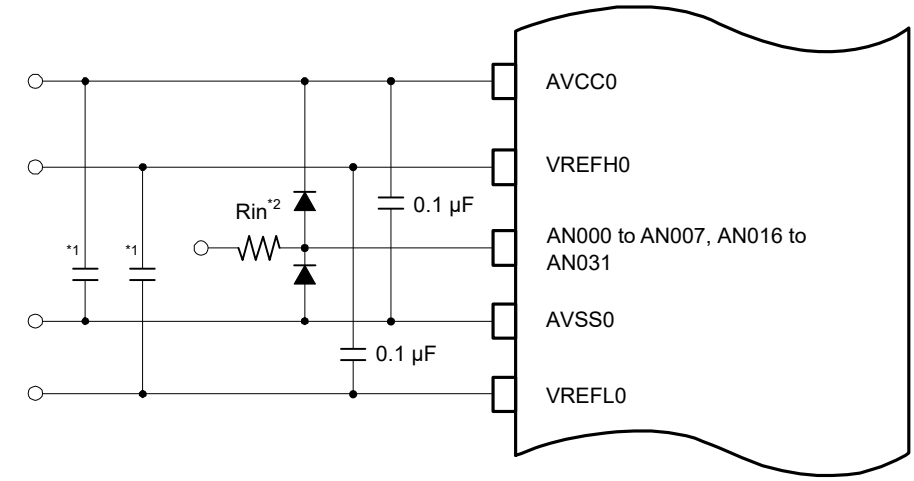
Figure 40.31 Power Supply Pin Connection Example

40.8.12 Notes on Board Design

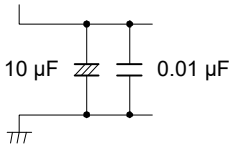
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN007, AN016 to AN031), 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).

40.8.13 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN007, AN016 to AN031) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and a protection circuit should be connected to protect the analog input pins (AN000 to AN007, AN016 to AN031) as shown Figure 40.32.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance

Figure 40.32 Sample Protection Circuit for Analog Inputs

41. D/A Converter (DAa)

41.1 Overview

This MCU includes two channels of 8-bit D/A converter.
Table 41.1 lists the specifications of the 8-bit D/A converter and Figure 41.1 shows a block diagram of the 8-bit D/A converter.

Table 41.1 Specifications of 8-Bit D/A Converter

Item	Specifications
Resolution	8 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 8-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.

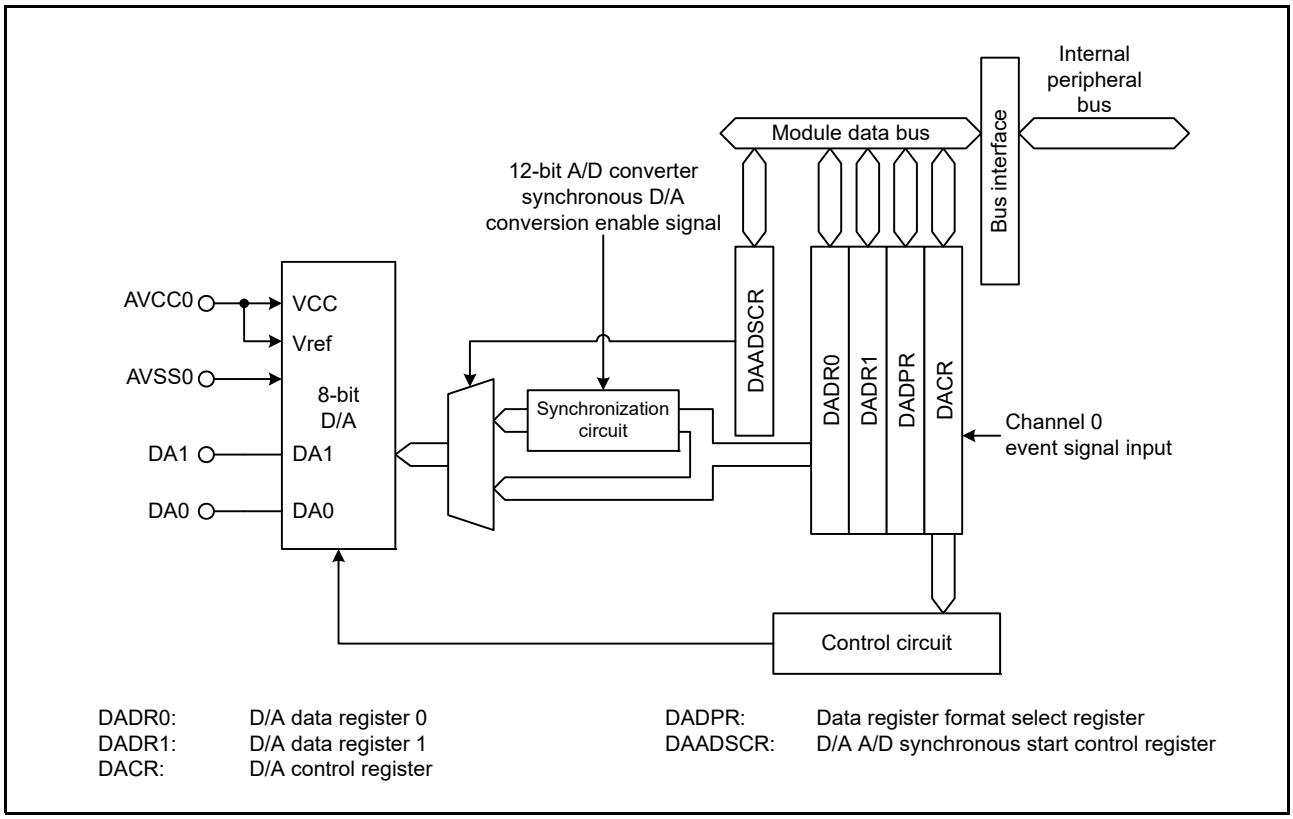


Figure 41.1 Block Diagram of 8-Bit D/A Converter

Table 41.2 lists the pin configuration of the 8-bit D/A converter.

Table 41.2 Pin Configuration of 8-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

41.2 Register Descriptions

41.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DA.DADR0 0008 80C0h, DA.DADR1 0008 80C2h

• 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.

8-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits "—" are read as 0. The write value should be 0.

41.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 80C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	—	—	—	—	—	—
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	—	Reserved	This bit is read as 0. The write value should be 0.	R
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

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.

DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion and analog output.

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.

DAOE1 Bit (D/A Output Enable 1)

The DAOE1 bit controls the D/A conversion and analog output.

41.2.3 Data Register Format Select Register (DADPR)

Address(es): DA.DADPR 0008 80C5h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are 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

41.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DA.DAADSCR 0008 80C6h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: 8-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: 8-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 8-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 8-bit D/A converter.

41.3 Operation

The 8-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 41.2 shows the timing of this operation.

(1) Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.

(2) Set the DACR.DAOE0 bit to 1 to start D/A conversion. The 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:

$$\frac{\text{Value of DADRm register}}{256} \times \text{AVCC0}$$

(3) 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.

(4) When the DAOE0 bit is set to 0, analog output is disabled.

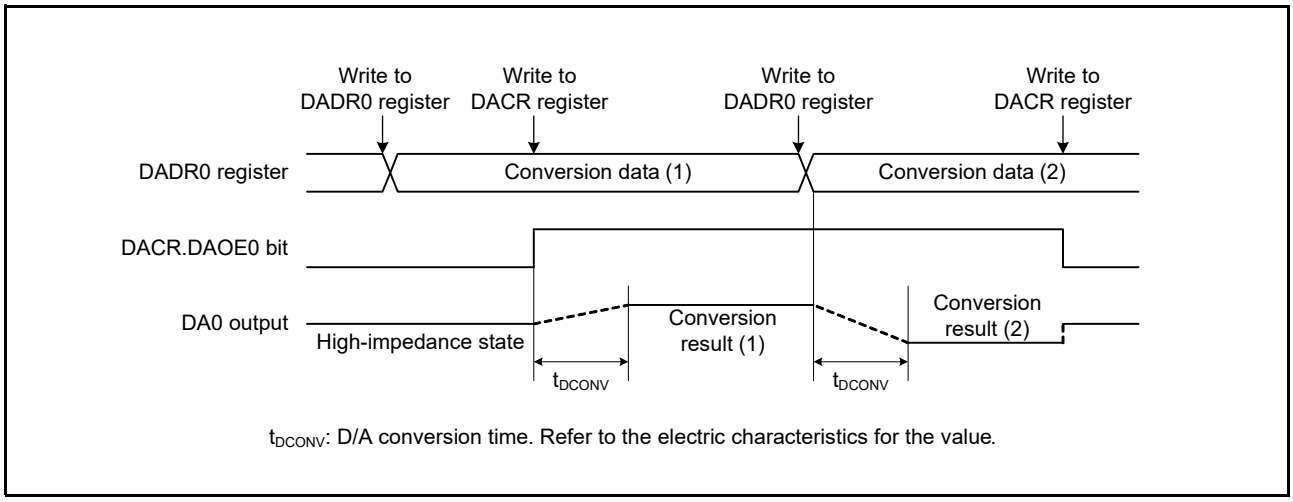


Figure 41.2 Example of 8-Bit D/A Converter Operation

41.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, an inrush current occurs to the 8-bit D/A converter. Since the same analog power supply is shared by the 8-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 41.3 shows an example of channel 0 D/A conversion, in which the 8-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.

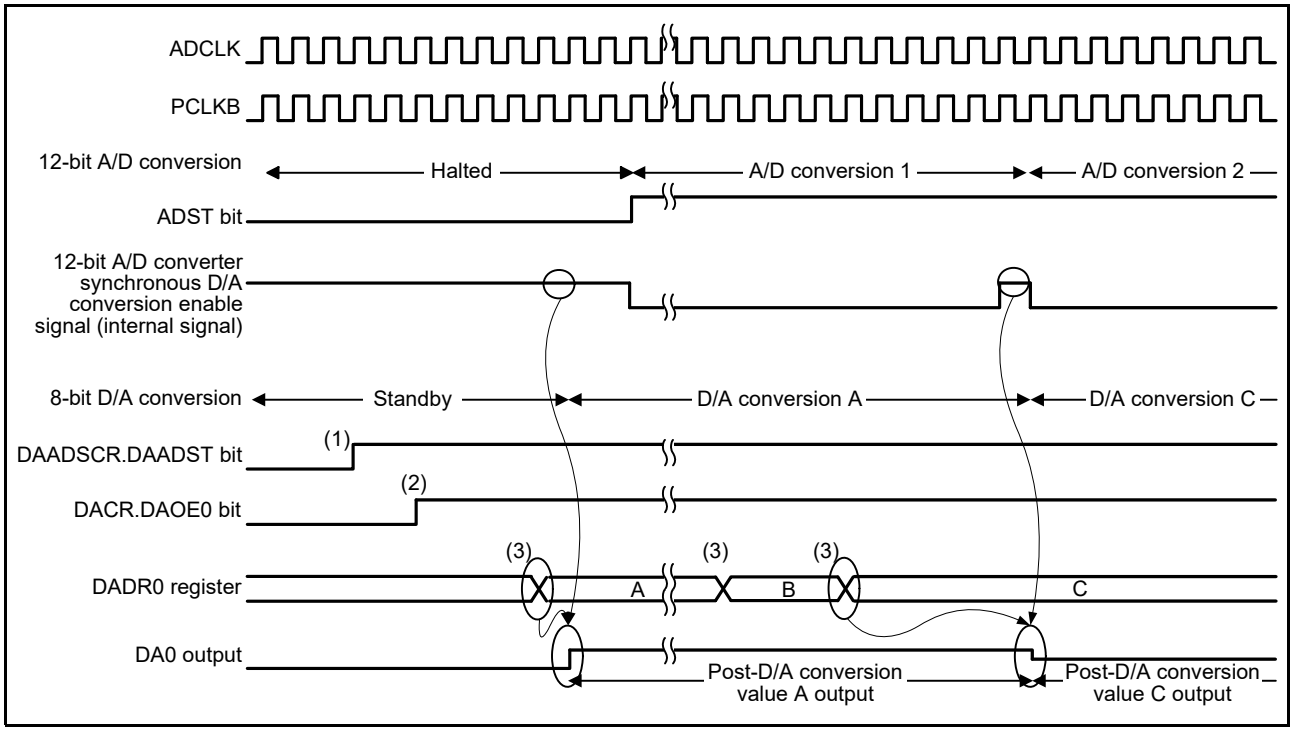


Figure 41.3 Example of Conversion When the 8-Bit D/A Converter is Synchronized with the 12-Bit A/D Converter

When ADCLK is faster than PCLKB, the 8-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 41.4 shows example when the 8-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.

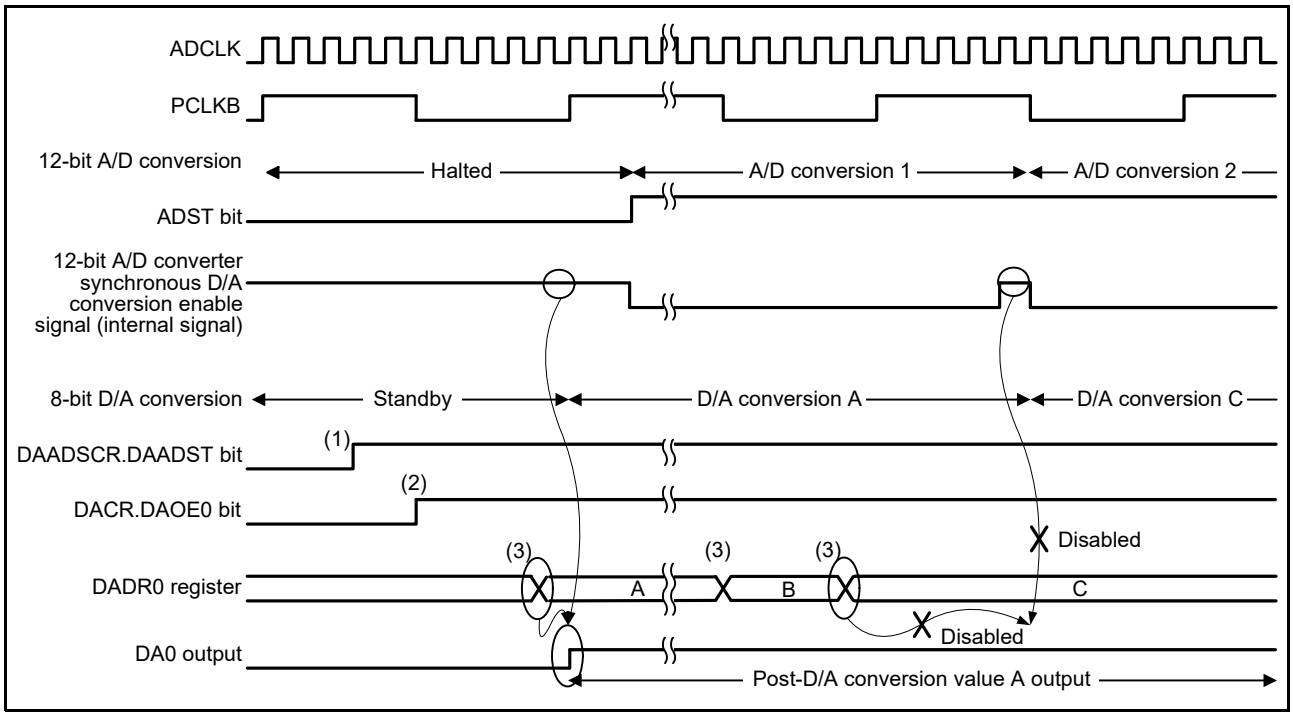


Figure 41.4 Example When the 8-Bit D/A Converter Cannot Capture the 12-Bit A/D Converter Synchronous D/A Conversion Enable Signal

41.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

(1) Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
(2) Set the bit value of the ELSR16 setting event signal to link the ELSR16 register of the ELC.
(3) Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
(4) Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
(5) Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop event link operation of 8-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

41.5 Usage Notes on Event Link Operation

(1) 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.
(2) 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.

41.6 Usage Notes

41.6.1 Module Stop Function Setting

Operation of the 8-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 8-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.

41.6.2 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, and DAOE0 bits to 0.

41.6.3 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, and DAOE0 bits to 0.

41.6.4 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.

42. Temperature Sensor (TEMPSA)

42.1 Overview

This MCU has a built-in temperature sensor. The temperature sensor outputs a voltage proportional to temperature. By converting the output voltage of the temperature sensor into a digital value with a 12-bit A/D converter and converting it into temperature, the internal temperature of the MCU can be obtained.
Table 42.1 lists the specifications of the temperature sensor. Figure 42.1 shows a overall block diagram of the temperature sensor system.

Table 42.1 Temperature Sensor Specifications

Item	Description
Temperature sensor voltage output	The temperature sensor voltage is output to the 12-bit A/D converter.

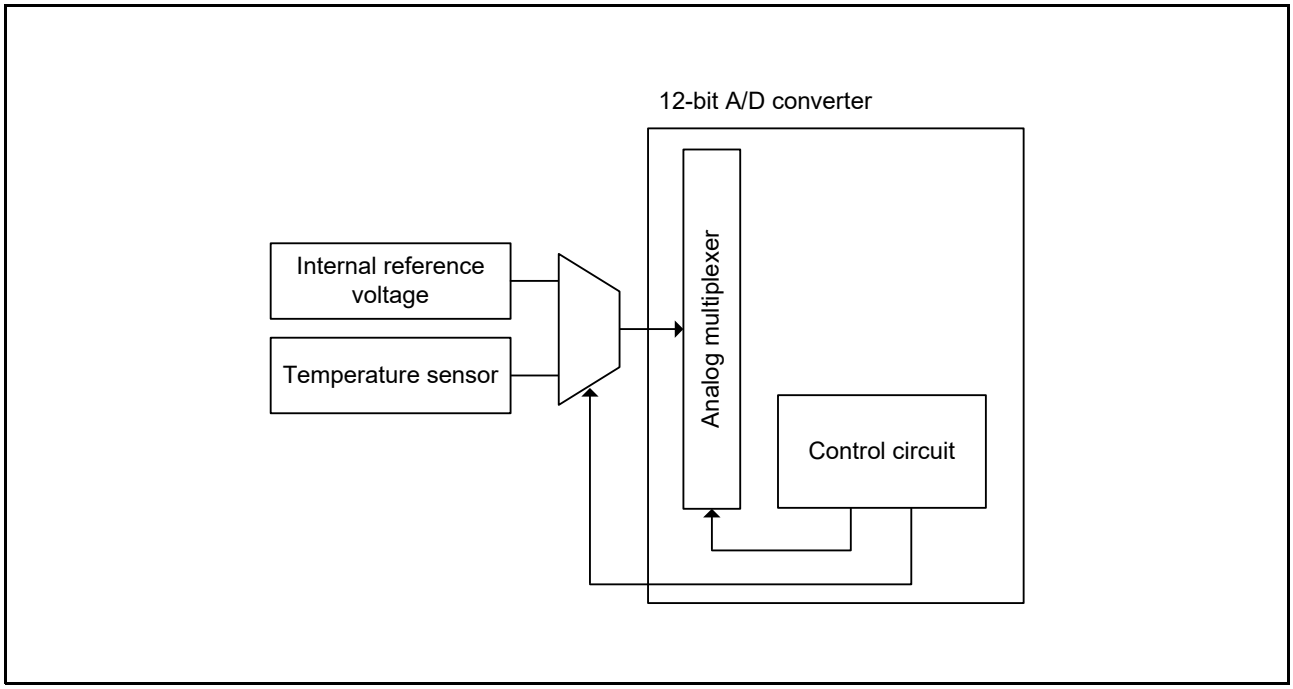


Figure 42.1 Block Diagram of Temperature Sensor System

42.2 Register Descriptions

42.2.1 Temperature Sensor Calibration Data Register (TSCDR)

Address(es): TEMPS.TSCDR 007F C228h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip															

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment. Temperature sensor calibration data is a value (CAL_{125}) obtained by using the 12-bit A/D converter to convert the voltage output by the temperature sensor under the condition of Tj = 125°C and AVCC0 = VREFH0 = 3.3 V.

The voltage V1 output by the temperature sensor under the condition of Tj = 125°C can be calculated from the converted value CAL_{125} according to the formula below.

$V1 = 3.3 \times CAL_{125}/4096 \quad (V)$

Note that voltage V1 is independent of both the AVCC0 and VREFH0 voltages.

42.3 Using the Temperature Sensor

The temperature sensor outputs a voltage proportional to temperature. By converting this voltage into a digital value with a 12-bit A/D converter and converting it into temperature, the internal temperature of the MCU can be obtained.

42.3.1 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)

Since there are individual differences in temperature sensors, it is recommended to prepare a temperature slope by performing trial measurements at two different temperatures as shown below.
Use the 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
Again, using the 12-bit A/D converter, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature 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 section 47, Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this method produces less accurate temperatures than measurement at two points.

T = (Vs – V1)/Slope + T1

In this MCU, the TSCDR register stores the temperature value (CAL_{125}) of the temperature sensor measured under the condition of Tj = 125°C and AVCC0 = VREFH0 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.
If V1 is calculated from CAL_{125},

$V1 = 3.3 \times CAL_{125}/4096$ (V)

Using this, the measured temperature can be calculated according to the formula below.

T = (Vs – V1)/Slope + 125 (°C)

T: Measured temperature (internal temperature of the MCU) (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
V1: Voltage output by the temperature sensor when Tj = 125°C and AVCC0 = VREFH0 = 3.3 V (V)
Slope: Temperature slope listed in Table 47.74 ÷ 1000 (V/°C)

Error in the measured temperature (the range of variation is 3σ) is shown in Figure 42.2.

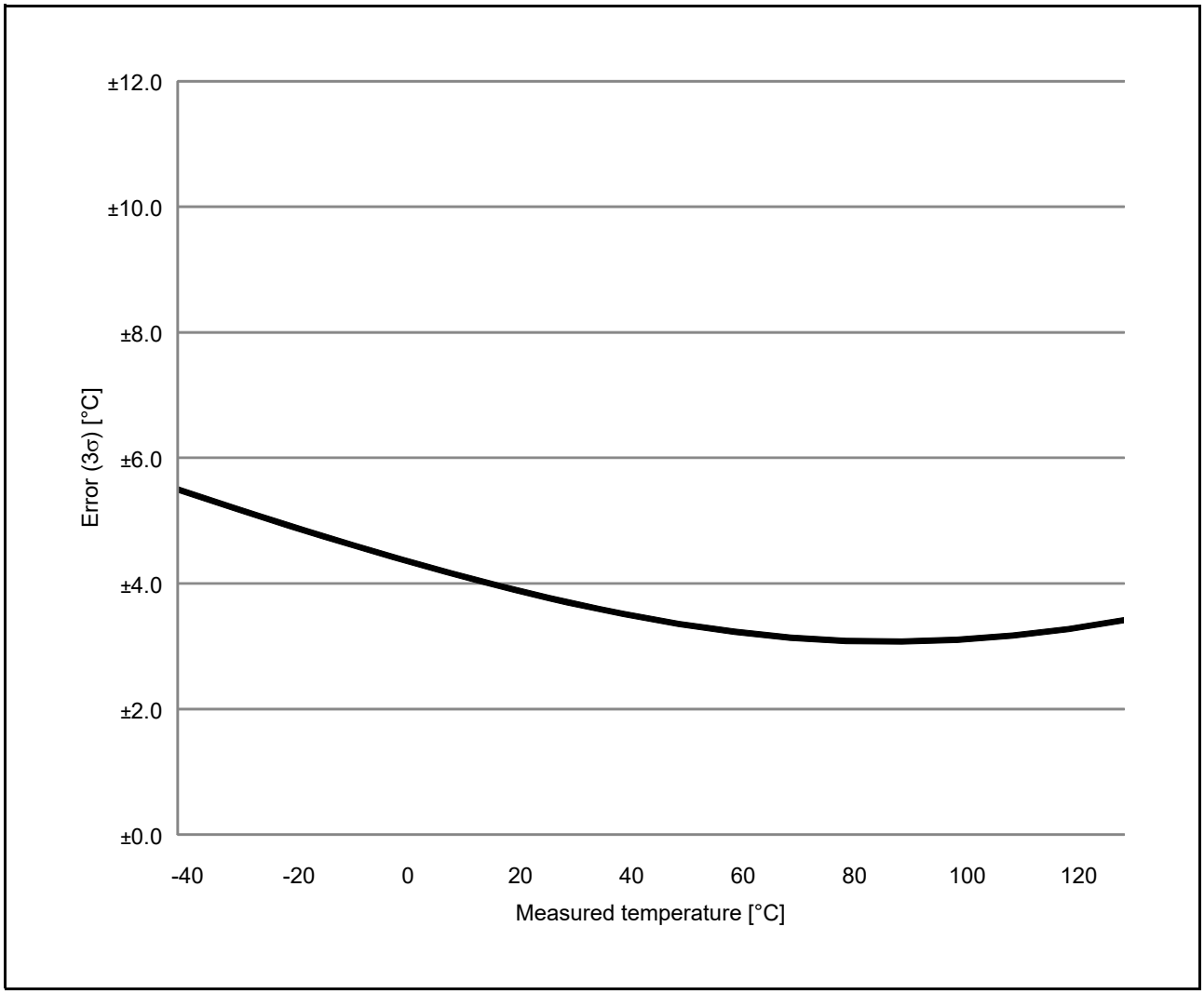


Figure 42.2 Error in the Measured Temperature (Designed Value)

42.3.2 Setting the 12-Bit A/D Converter

For details, refer to section 40, 12-Bit A/D Converter (S12ADE).

43. Comparator B (CMPBa)

Comparator B compares a reference input voltage and an analog input voltage. Comparator B0 and comparator B1 operate independently.
In this section, “PCLK” is used to refer to PCLKB.

43.1 Overview

The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference input voltage can be selected from either an input to the CVREFBn pin (n = 0, 1) or the internal reference voltage generated internally in the MCU.
The comparator B response speed can be set before starting an operation. Setting high-speed mode decreases the response delay time, but increases current consumption. Setting low-speed mode increases the response delay time, but decreases current consumption.
Table 43.1 lists the specifications of comparator B, Figure 43.1 shows a block diagram of comparators B0 and B1 when the window function is disabled, and Figure 43.2 shows a block diagram of comparators B0 and B1 when the window function is enabled. Table 43.2 lists the I/O pins of comparator B.

Table 43.1 Comparator B Specifications (n = 0, 1)

Item	Specification
Analog input voltage	Input voltage to the CMPBn pin
Reference input voltage	Input voltage to the CVREFBn pin or internal reference voltage
Comparison result	Read from the CPBFLG.CPBnOUT flag The comparison result can be output to the CMPOBn pin.
Interrupt request generation timing	When comparator B0 comparison result changes When comparator B1 comparison result changes
Event generation timing to ELC	When comparator B0 comparison result changes When comparator B0 or comparator B1 comparison result changes
POE source output timing	When comparator B0 comparison result changes When comparator B1 comparison result changes
Selectable function	• Digital filter function Whether the digital filter is applied or not, and the sampling frequency can be selected. • Window function Whether the window function is enabled or disabled (VRFL < CMPBn < VRFH)*1 can be selected. • Reference input voltage CVREFBn pin input or internal reference voltage (generated internally) can be selected. • Comparator B response speed High-speed mode/low-speed mode can be selected.
Low power consumption function	Module stop state can be set.

Note 1. VRFL: low-side reference voltage, VRFH: high-side reference voltage

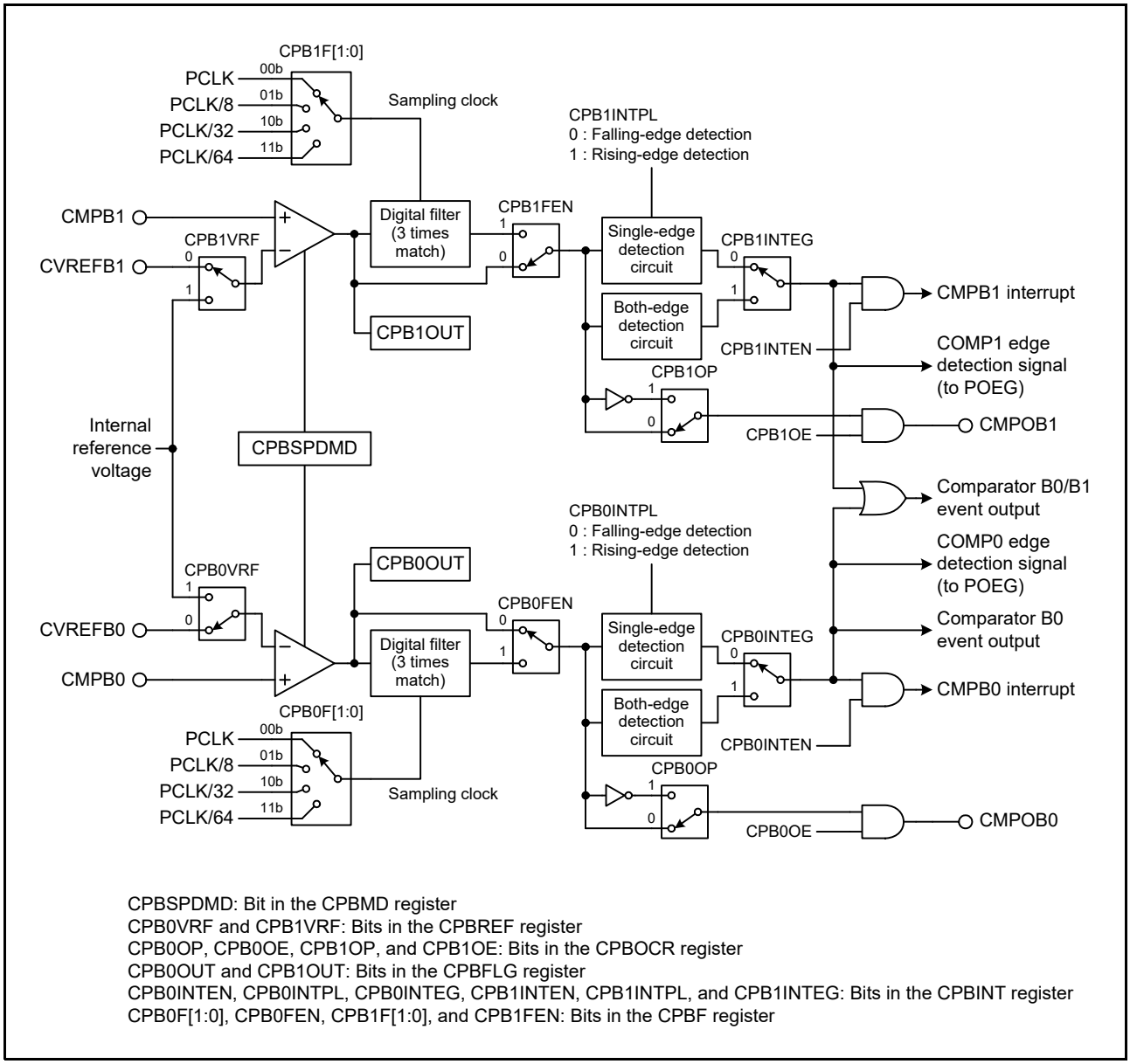


Figure 43.1 Block Diagram of Comparators B0 and B1 When Window Function is Disabled

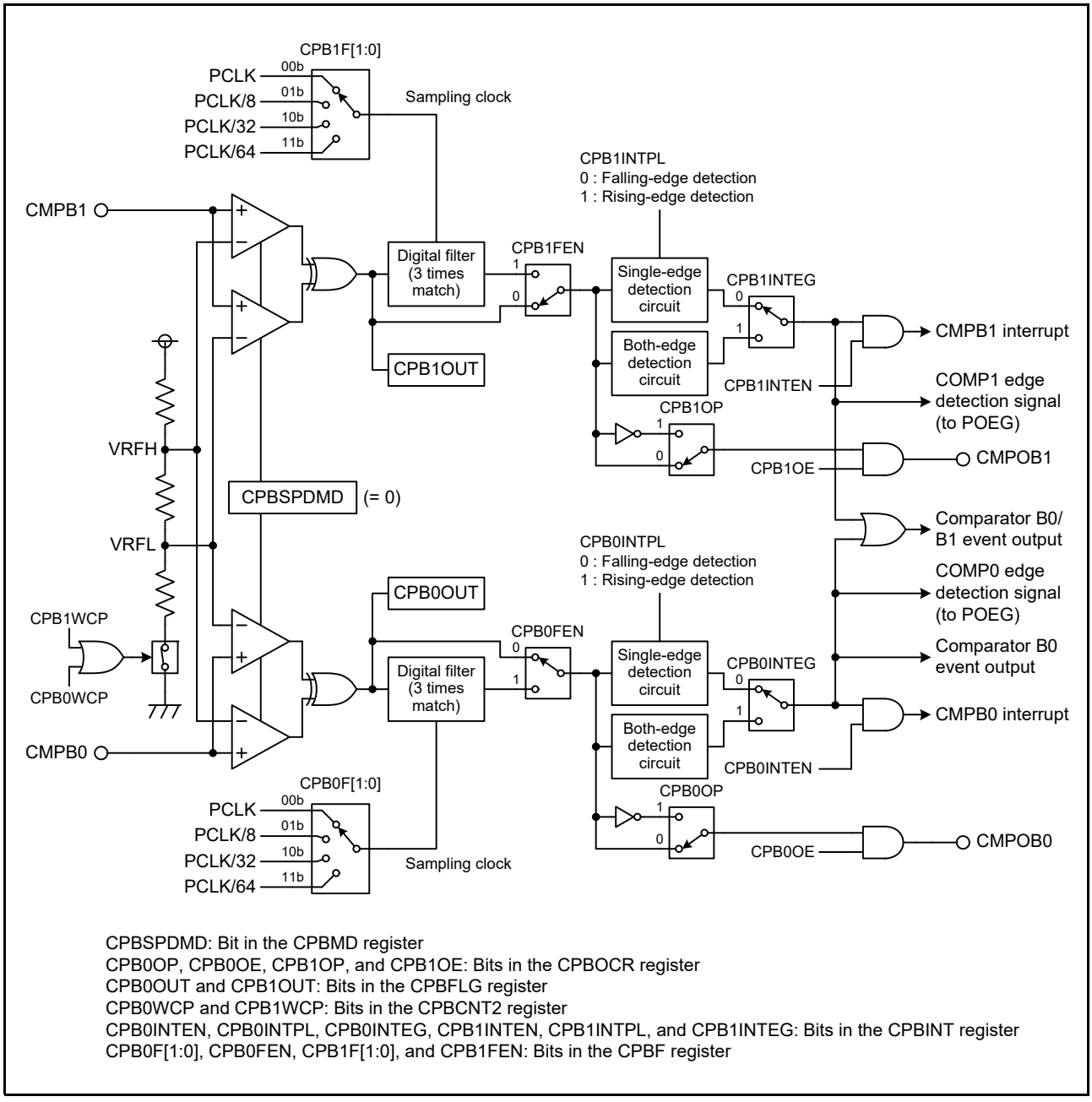


Figure 43.2 Block Diagram of Comparators B0 and B1 When Window Function is Enabled

Table 43.2 I/O Pins of Comparator B

Pin Name	I/O	Function
CMPB0	Input	Comparator B0 analog pin
CVREFB0	Input	Comparator B0 reference input voltage pin
CMPB1	Input	Comparator B1 analog pin
CVREFB1	Input	Comparator B1 reference input voltage pin
CMPOB0	Output	Comparator B0 output
CMPOB1	Output	Comparator B1 output

43.2 Register Descriptions

43.2.1 Comparator B Control Register 1 (CPBCNT1)

Address: CMPB.CPBCNT1 0008 C580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CPB1INI	—	—	—	CPB0INI
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0INI	Comparator B0 Power Enable	0: Disabled 1: Enabled (comparator powered on)	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB1INI	Comparator B1 Power Enable	0: Disabled 1: Enabled (comparator powered on)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

43.2.2 Comparator B Control Register 2 (CPBCNT2)

Address: CMPB.CPBCNT2 0008 C581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CPB1WCP	—	—	—	CPB0WCP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0WCP	Comparator B0 Window Function Enable	0: Disabled 1: Enabled*1	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB1WCP	Comparator B1 Window Function Enable	0: Disabled 1: Enabled*1	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set the CPBSPDMD bit to 0 before enabling the window function.

43.2.3 Comparator B Flag Register (CPBFLG)

Address: CMPB.CPBFLG 0008 C582h

	b7	b6	b5	b4	b3	b2	b1	b0
	CPB1OUT	—	—	—	CPB0OUT	—	—	—
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	CPB0OUT	Comparator B0 Monitor Flag	When the window function is disabled 0: CMPB0 < CVREFB0, CMPB0 < internal reference voltage, or comparator B0 operation disabled 1: CMPB0 > CVREFB0, or CMPB0 > internal reference voltage When the window function is enabled 0: CMPB0 < VRFL*1, CMPB0 > VRFH*1, or comparator B0 operation disabled 1: VRFL < CMPB0 < VRFH	R
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CPB1OUT	Comparator B1 Monitor Flag	When the window function is disabled 0: CMPB1 < CVREFB1, CMPB1 < internal reference voltage, or comparator B1 operation disabled 1: CMPB1 > CVREFB1, or CMPB1 > internal reference voltage When the window function is enabled 0: CMPB1 < VRFL*1, CMPB1 > VRFH*1, or comparator B1 operation disabled 1: VRFL < CMPB1 < VRFH	R

Note 1. VRFL: low-side reference voltage, VRFH: high-side reference voltage

43.2.4 Comparator B Interrupt Control Register (CPBINT)

Address: CMPB.CPBINT 0008 C583h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CPB1INTPL	CPB1INTEG	CPB1INTEN	—	CPB0INTPL	CPB0INTEG	CPB0INTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0INTEN	Comparator B0 Interrupt Enable	0: Disabled 1: Enabled	R/W
b1	CPB0INTEG	Comparator B0 Interrupt/Event Edge Select*1	0: Single edge 1: Both edges	R/W
b2	CPB0INTPL	Comparator B0 Interrupt/Event Edge Polarity Select*2	0: Falling edge 1: Rising edge	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CPB1INTEN	Comparator B1 Interrupt Enable	0: Disabled 1: Enabled	R/W
b5	CPB1INTEG	Comparator B1 Interrupt/Event Edge Select*1	0: Single edge 1: Both edges	R/W
b6	CPB1INTPL	Comparator B1 Interrupt/Event Edge Polarity Select*2	0: Falling edge 1: Rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The IR058.IR flag may become 1 (interrupt request is generated) when the CPB0INTPL bit is modified, and the IR059.IR flag may become 1 (interrupt request is generated) when the CPB1INTPL bit is modified. For details, refer to section 14, Interrupt Controller (ICUb).

Note 2. The CPBnINTPL bit setting is valid only when the CPBnINTEG bit is 0 (single edge).

43.2.5 Comparator B Filter Select Register (CPBF)

Address: CMPB.CPBF 0008 C584h

	b7	b6	b5	b4	b3	b2	b1	b0
	CPB1F[1:0]		—	CPB1FEN	CPB0F[1:0]		—	CPB0FEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0FEN	Comparator B0 Filter Enable/Disable Select*1	0: Filter is disabled. 1: Filter is enabled.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	CPB0F[1:0]	Comparator B0 Filter Select*1	b3 b2 0 0: Sampling at PCLK 0 1: Sampling at PCLK/8 1 0: Sampling at PCLK/32 1 1: Sampling at PCLK/64	R/W
b4	CPB1FEN	Comparator B1 Filter Enable/Disable Select*1	0: Filter is disabled. 1: Filter is enabled.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	CPB1F[1:0]	Comparator B1 Filter Select*1	b7 b6 0 0: Sampling at PCLK 0 1: Sampling at PCLK/8 1 0: Sampling at PCLK/32 1 1: Sampling at PCLK/64	R/W

Note 1. The CPBnF[1:0] bits are enabled only when the CPBnFEN bit = 1 (filter is enabled).

43.2.6 Comparator B Mode Select Register (CPBMD)

Address: CMPB.CPBMD 0008 C585h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CPBSPDMD	Comparator B Speed Select*1	0: High-speed mode 1: Low-speed mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When rewriting the CPBSPDMD bit, be sure to set the CPBnINI bit (n = 0, 1) in the CPBCNT1 register to 0 in advance.

43.2.7 Comparator B Reference Input Voltage Select Register (CPBREF)

Address: CMPB.CPBREF 0008 C586h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CPB1VRF	—	—	—	CPB0VRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0VRF	Comparator B0 Reference Input Voltage Select	0: Comparator B0 reference input voltage is CVREFB0 input 1: Comparator B0 reference input voltage is internal reference voltage*1, *2, *3	R/W*4
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB1VRF	Comparator B1 Reference Input Voltage Select	0: Comparator B1 reference input voltage is CVREFB1 input 1: Comparator B1 reference input voltage is internal reference voltage*1, *2, *3	R/W*4
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Enabled only when the window function is disabled. When the window function is enabled, the internal reference voltage of comparator B is selected regardless of the setting of this bit.
Note 2. When the internal reference voltage is selected, the temperature sensor output cannot be selected for the A/D converter.
Note 3. When the internal reference voltage is selected, the voltage generation circuit operates and current increases by about 75 µA. This circuit is not automatically turned off even if the MCU enters software standby mode with the internal reference voltage selected.
Note 4. Do not rewrite the CPBnVRF bit when CPBCNT2.CPBnWCP = 0.

Notes on changing the reference input voltage

- When changing the reference input voltage from CVREFBn (n = 0, 1) to the internal reference voltage, use the following procedure.

1. Set the CPBCNT1.CPBnINI bit to 1.
2. Set the CPBCNT2.CPBnWCP bit to 1.
3. Set the CPBREF.CPBnVRF bit to 1 to select the internal reference voltage.
4. Set the analog select bit (ASEL) in the pin function control register of the port that is used as the CVREFBn pin to 0.
5. Wait for the comparator operation to stabilize (operation stabilization wait time (Tcmp)*1).
6. Set the CPBCNT2.CPBnWCP bit to 0.

- When changing the reference input voltage from the internal reference voltage to CVREFBn (n = 0, 1), use the following procedure.

1. Set the CPBCNT1.CPBnINI bit to 1.
2. Set the CPBCNT2.CPBnWCP bit to 1.
3. Set the CPBREF.CPBnVRF bit to 0 to select the CVREFBn pin input.
4. Set the analog select bit (ASEL) in the pin function control register of the port that is used as the CVREFBn pin to 1.
5. Wait for the comparator operation to stabilize (operation stabilization wait time (Tcmp)*1).
6. Set the CPBCNT2.CPBnWCP bit to 0.

Note 1. For operation stabilization wait time (Tcmp), refer to the Electrical Characteristics chapter.

43.2.8 Comparator B Output Control Register (CPBOCR)

Address: CMPB.CPBOCR 0008 C587h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CPB1OP	CPB1OE	—	—	CPB0OP	CPB0OE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0OE	CMPOB0 Pin Output Enable	0: Comparator B0 CMPOB0 pin output disabled*1 1: Comparator B0 CMPOB0 pin output enabled	R/W
b1	CPB0OP	CMPOB0 Output Polarity Select	0: Comparator B0 output is output to CMPOB0 pin 1: Inverted comparator B0 output is output to CMPOB0 pin	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB1OE	CMPOB1 Pin Output Enable	0: Comparator B1 CMPOB1 pin output disabled*1 1: Comparator B1 CMPOB1 pin output enabled	R/W
b5	CPB1OP	CMPOB1 Output Polarity Select	0: Comparator B1 output is output to CMPOB1 pin 1: Inverted comparator B1 output is output to CMPOB1 pin	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the CPBnOE bit (n = 0, 1) is set to 0 to disable the CMPOBn pin output, low is output to the CMPOBn pin regardless of the value of the CPBnOP bit.

43.3 Operation

Comparator B0 and comparator B1 operate independently, and their operations are the same. Operation is not guaranteed when the values of registers are changed during comparator operation.

43.3.1 Setting Procedure

Table 43.3 shows the procedure of setting comparator B associated registers when the window function is disabled. Table 43.4 shows the procedure of setting comparator B associated registers when the window function is enabled.

Table 43.3 Setting Procedure for Comparator B Related Registers When Window Function is Disabled (n = 0, 1)

Step	Register	Bit	Setting	
1	PijPFS of the port to which the CMPBn pin is assigned	ASEL	1	
2	CPBMD	CPBSPDMD	Select the comparator response speed (0: High-speed mode/1: Low-speed mode)	
3	CPBCNT1	CPBnINI	Powered on: 1	
4	CPBCNT2	CPBnWCP	1*1	
5	CPBREF	CPBnVRF	0: Reference input voltage = CVREFBn input*1	1: Reference input voltage = Internal reference voltage
6	PijPFS of the port to which the CVREFBn pin is assigned	ASEL	1	0
7	Waiting for the comparator operation to stabilize*1 (operation stabilization wait time (Tcmp)*2)			
8	CPBCNT2	CPBnWCP	0*1	
9	CPBF	Select whether to enable or disable the filter and select the sampling clock.		
10	Waiting for the comparator operation to stabilize (operation stabilization wait time (Tcmp)*2)			
11	CPBOCR	CPBnOP, CPBnOE	Set the CMPOBn output (select the polarity and set output enabled or disabled).	
12	CPBINT	CPBnINTEN	When using an interrupt: 1 (interrupt enabled)	
		CPBnINTEG	When using an interrupt, event output, or POE source output: Select the edge (1 = both edges or 0 = single edge).	
		CPBnINTPL	When using an interrupt, event output, or POE source output: For CPBnINTEG = 0 (single edge selected), select the edge polarity (1 = rising edge or 0 = falling edge).	
13	IPR058 (comparator B0), IPR059 (comparator B1)	IPR[3:0]	When using an interrupt: Select the interrupt priority level.	
	IR058 (comparator B0), IR059 (comparator B1)	IR	When using an interrupt: 0 (no interrupt requested: initialization)	
	IER07	IEN2 (comparator B0), IEN3 (comparator B1)	When using an interrupt: 1 (interrupt is enabled on the interrupt controller (ICU) side)	

Note 1. This setting is necessary when changing the reference input voltage from the CVREFBn input to the internal reference voltage or from the internal reference voltage to the CVREFBn input. When selecting the CVREFBn input after the reset is released, steps 4, 5, 7, and 8 are not necessary because the initial value of the CPBREF.CPBnVRF bit is 0.

Note 2. For operation stabilization wait time (Tcmp), refer to the Electrical Characteristics chapter.

Table 43.4 Setting Procedure for Comparator B Related Registers When Window Function is Enabled (n = 0, 1)

Step	Register	Bit	Setting
1	PijPFS of the port to which the CMPBn pin is assigned	ASEL	1
2	CPBMD	CPBSPDMD	0 (always specify high-speed mode)
3	CPBCNT1	CPBnINI	Powered on: 1
4	CPBF	Select whether to enable or disable the filter and select the sampling clock.	
5	CPBCNT2	CPBnWCP	1 (operation enabled)
6	Waiting for the comparator operation to stabilize (operation stabilization wait time (Tcmp)*1)		
7	CPBOCR	CPBnOP, CPBnOE	Set the CMPOBn output (select the polarity and set output enabled or disabled).
8	CPBINT	CPBnINTEN	When using an interrupt: 1 (interrupt enabled)
		CPBnINTEG	When using an interrupt, event output, or POE source output: Select the edge (1 = both edges or 0 = single edge).
		CPBnINTPL	When using an interrupt, event output, or POE source output: For CPBnINTEG = 0 (single edge selected), select the edge polarity (1 = rising edge or 0 = falling edge).
9	IPR058 (comparator B0), IPR059 (comparator B1)	IPR[3:0]	When using an interrupt: Select the interrupt priority level.
	IR058 (comparator B0), IR059 (comparator B1)	IR	When using an interrupt: 0 (no interrupt requested: initialization)
	IER07	IEN2 (comparator B0), IEN3 (comparator B1)	When using an interrupt: 1 (interrupt enabled)

Note 1. For operation stabilization wait time (Tcmp), refer to the Electrical Characteristics chapter.

43.3.2 Examples of Operation

Figure 43.3 shows an operation example of comparator Bn when window function is disabled (n = 0, 1).
The reference input voltage (CVREFB0/CVREFB1 or internal reference voltage) and the analog input voltage (CMPBn) are compared. If the analog input voltage is higher than the reference input voltage, the CPBFLG.CPBnOUT bit is set to 1. If the analog input voltage is lower than the reference input voltage, the CPBnOUT bit is set to 0.
To use the comparator Bn interrupt, set the CPBINT.CPBnINTEN bit to 1 (interrupt enabled). If the comparison result changes at this time, a comparator Bn interrupt request is generated. For details on interrupts, refer to section 43.4, Interrupts.
Comparators B0 and B1 output event signals to the ELC to activate other modules. For details on the event signals, refer to section 43.5, Event Link Output.
Also, comparators B0 and B1 output COMPn edge detection signals to the POEG as POE sources. For details on the POE source output, refer to section 43.6, POE Source Output to the POEG.
The values of the registers should not be changed during comparison.

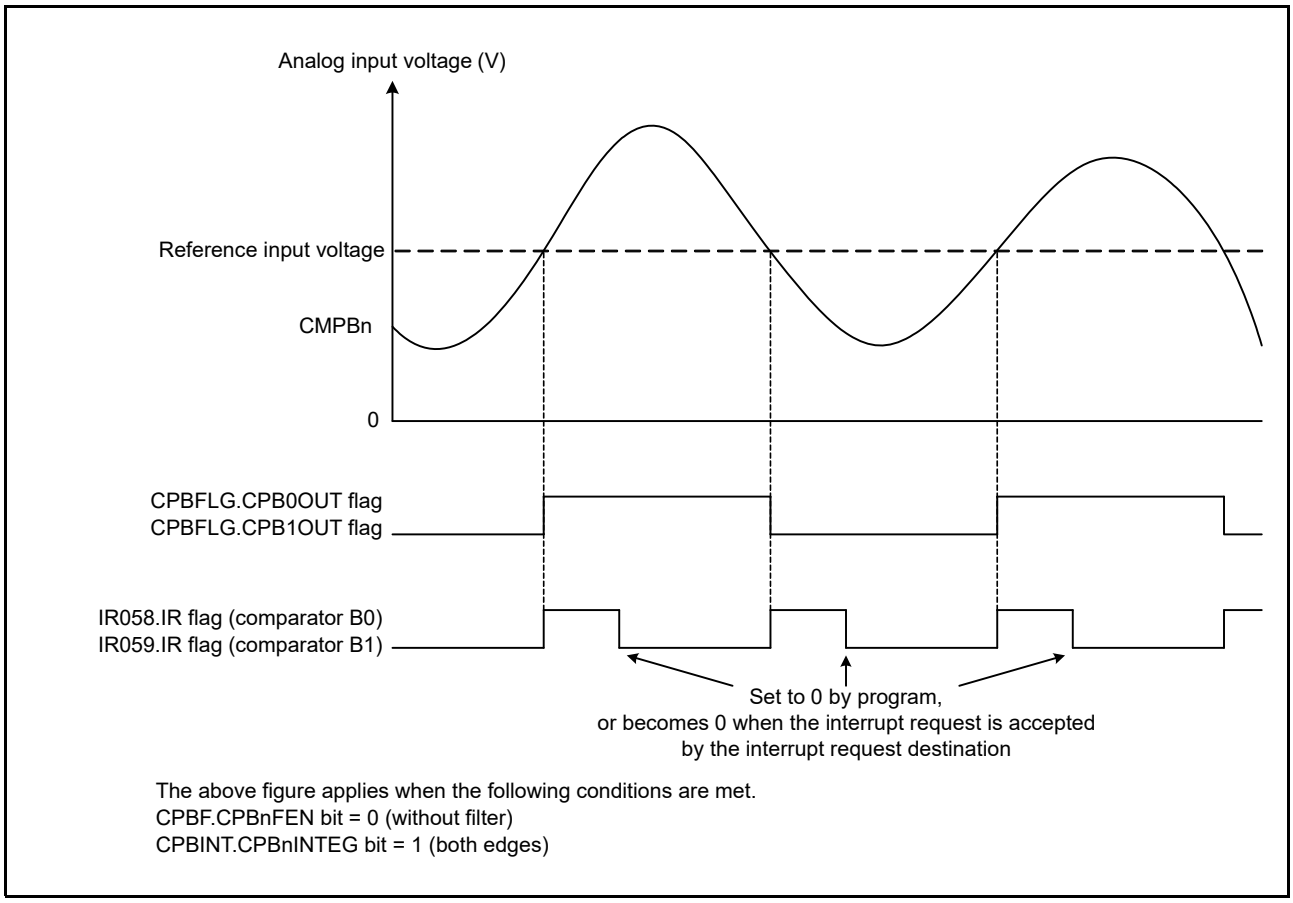


Figure 43.3 Operation Example of Comparator Bn When Window Function is Disabled (n = 0, 1)

Figure 43.4 shows an operation example of comparator Bn when the window function is enabled (n = 0, 1).
The internal reference voltage (VRFH/VRFL) for the window function and the analog input voltage (CMPBn) are compared. The CPBnOUT bit is set to 1 when VRFL < the analog input voltage < VRFH, and the CPBnOUT bit is set to 0 when the analog input voltage < VRFL, or VRFH < the analog input voltage.
To use the comparator Bn interrupt, set the CPBINT.CPBnINTEN bit to 1 (interrupt enabled). If the comparison result changes at this time, a comparator Bn interrupt request is generated. For details on interrupts, refer to section 43.4, Interrupts.
Comparators B0 and B1 output event signals to the ELC to activate other modules. For details on the event signals, refer to section 43.5, Event Link Output.
Also, comparators B0 and B1 output COMPn edge detection signals to the POEG as POE sources. For details on the POE source output, refer to section 43.6, POE Source Output to the POEG.
The values of the registers should not be changed during comparison.

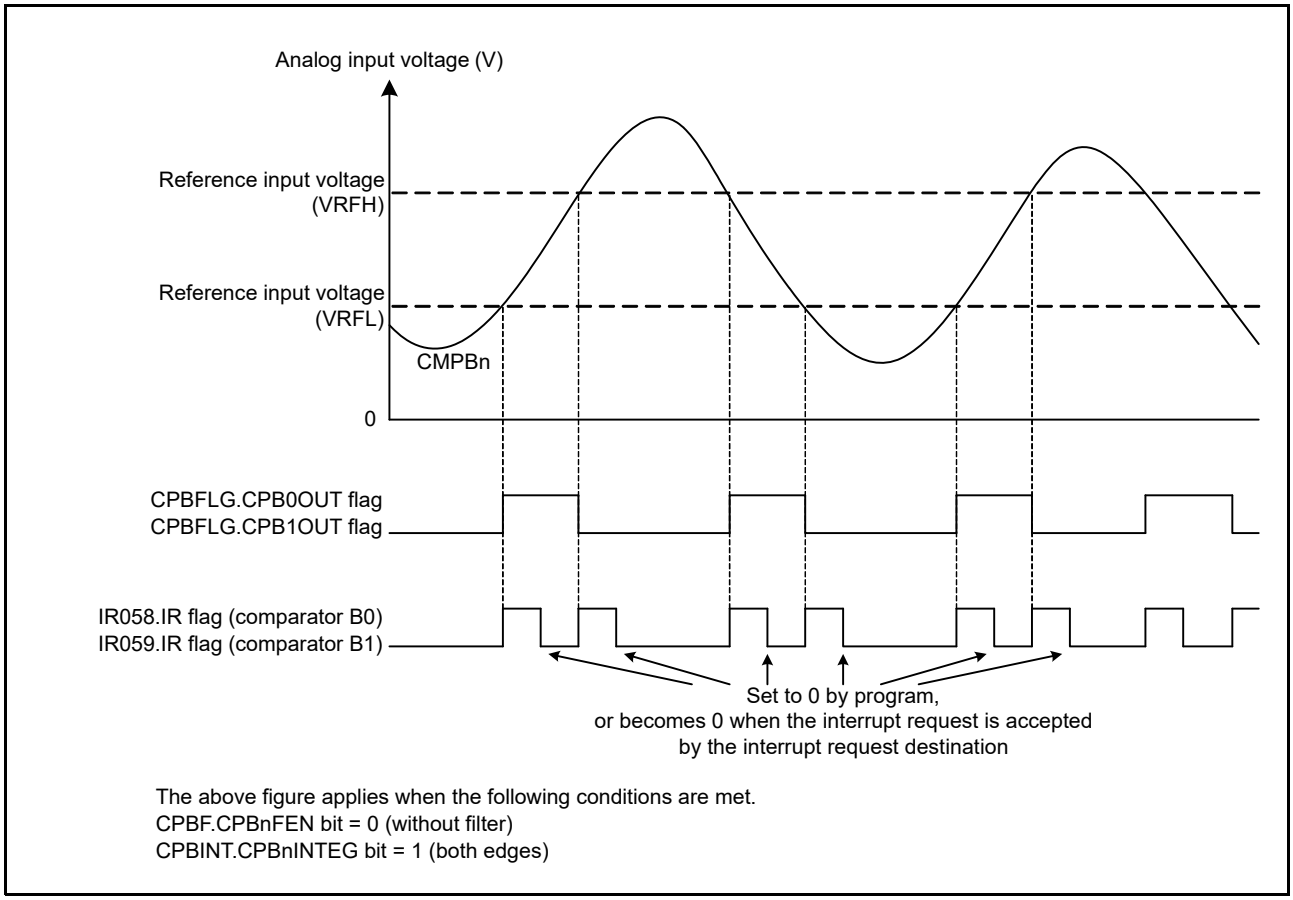


Figure 43.4 Operation Example of Comparator Bn When Window Function is Enabled (n = 0, 1)

43.3.3 Comparator Bn Digital Filter (n = 0, 1)

The sampling clock can be selected by the CPBF.CPBnF[1:0] bits. The CPBnOUT signal (internal signal) output from comparator Bn is sampled at every sampling clock cycle. At the next clock timing after the level matches three times, the IR058.IR flag (when comparator B0 selected) or IR059.IR flag (when comparator B1 selected) is set to 1 (interrupt requested) and an event signal to the ELC and an edge detection signal to the POEG are output.
Figure 43.5 shows the configuration of the comparator Bn digital filter, and Figure 43.6 shows an operation example of the comparator Bn digital filter.

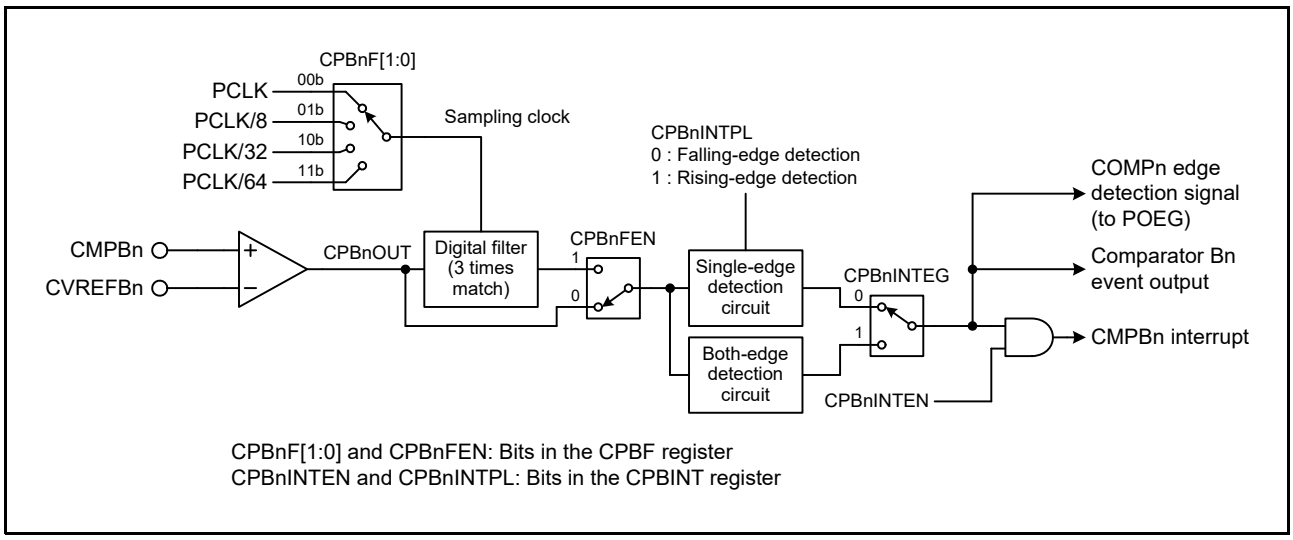


Figure 43.5 Configuration of Comparator Bn Digital Filter (n = 0, 1)

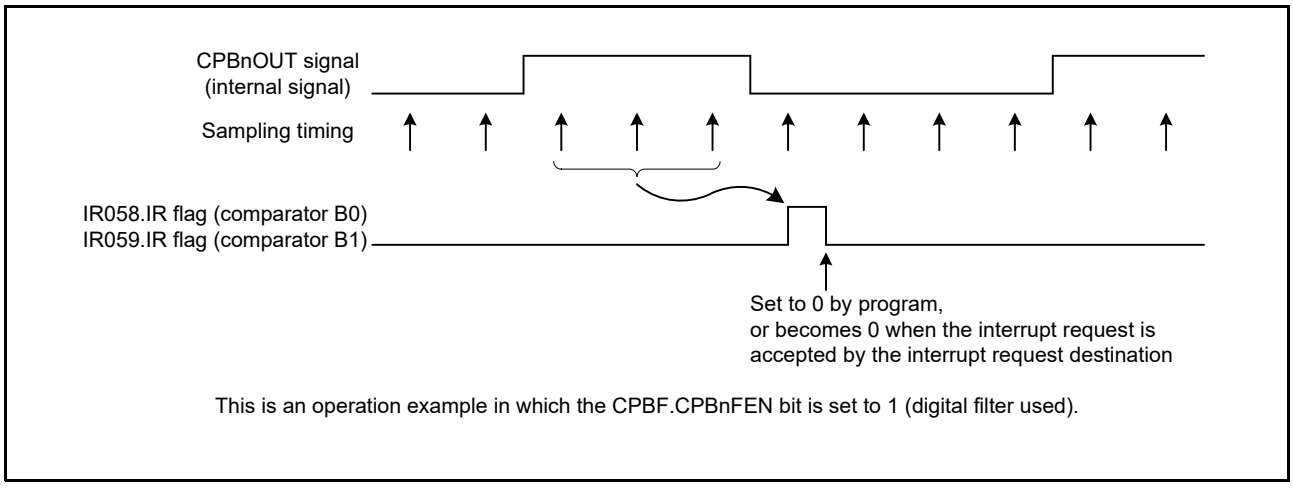


Figure 43.6 Operation Example of Comparator Bn Digital Filter (n = 0, 1)

43.3.4 Comparator Bn Output Function (n = 0, 1)

The comparison result from comparator B can be output to external pins. The CPBOCR.CPBnOP and CPBOCR.CPBnOE bits can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled. For the register settings and corresponding comparator output, refer to section 43.2.8, Comparator B Output Control Register (CPBOCR).

To output the comparator B comparison result to the CMPOB0 or CMPOB1 output pin, use the following procedure to make port settings. Note that the ports are set to input after a reset.

(1) Set the mode and input for comparator B (steps 1 to 10 listed in Table 43.3 and steps 1 to 6 listed in Table 43.4).
(2) Select the polarity of the CMPOB0 or CMPOB1 output and enable the output (set the CPBOCR.CPBnOP and CPBOCR.CPBnOE bits).
(3) Set the port mode register and pin function control register corresponding to the CMPOB0 or CMPOB1 output pin (start outputting from the pin).

43.3.5 Example of Using Comparator B to Exit Software Standby Mode

The following shows an example of using comparator B1 output to exit software standby mode.
In this example, it is assumed that the reference input voltage (CVREFB1) > analog input voltage (CMPB1).

Set the following steps (1) to (3) before entering software standby mode.

(1) Set the registers associated with comparator B1 according to section 43.3, Operation.
However, set the CPBF.CPB1FEN bit to ‘filter is disabled’, the CPBOCR.CPB1OE bit to ‘output enabled’, and the CPBOCR.CPB1OP bit to ‘comparator B1 output is output to the CMPOB1 pin’.
(2) Make the IRQ4 interrupt settings according to section 14.4.8, External Pin Interrupts.
However, set the IRQFLTE0.FLTEN4 bit to 0 (digital filter disabled) and set the IRQCR4.IRQMD[1:0] bits to the same polarity as that of comparator B1 output.
In this example, a rising edge is selected.
(3) Set the multi-function pin controller to select the CMPOB1 function and enable IRQ4.

When exiting software standby mode, input a voltage from the comparator B1 analog pin (CMPB1) so that the reference input voltage (CVREFB1) is less than the analog input voltage (CMPB1). This allows the IRQ4 interrupt to be generated through the comparator B1 output pin (CMPOB1) and the MCU exits software standby mode.

43.4 Interrupts

Comparator B generates two interrupt requests from sources, comparator B0 interrupt (CMPB0 interrupt) and comparator B1 interrupt (CMPB1 interrupt). The CMPBn interrupt (n = 0, 1) uses the IR058.IR flag, IR059.IR flag, IPR058.IPR[3:0] bits, IPR059.IPR[3:0] bits, and the respective single interrupt vector.
To use the CMPBn interrupt, set the CPBINT.CPBnINTEN bit to 1 (interrupt enabled). In addition, select either single-edge detection or both-edge detection using the CPBINT.CPBnINTEG bit. When single-edge detection is selected, select the polarity using the CPBINT.CPBnINTPL bit.
Inputs can also be passed through the digital filter with four different sampling clocks.

43.5 Event Link Output

Comparator B outputs the following events to the event link controller (ELC).

(1) Comparison result of comparator B0 is changed.
(2) Comparison result of comparator B0 or B1 is changed.*1

Note 1. If the comparison results of comparators B0 and B1 are output simultaneously or offset by one clock cycle, they are output as a single event.

43.5.1 Relationship between Interrupt Handling and Event Linking

Comparators B0 and B1 output event signals to the event link controller (ELC) to initiate operations of other modules selected in advance. Event signals to the event link controller (ELC) are output independent of the CPBnINTEN bit value.
Similar to the interrupt request signals, the condition for the generation of the event signal output from comparator Bn to the ELC can be selected between single-edge detection and both-edge detection by setting the CPBINT.CPBnINTEG bit. When the single-edge detection is selected, the polarity can be selected by the CPBINT.CPBnINTPL bit.

43.6 POE Source Output to the POEG

When the comparison result of comparator B0 or B1 changes, an edge detection signal is output to the POEG. The same settings as for the interrupt request signal can be selected for the edge detection signal, but the output cannot be enabled or disabled.
When using the edge detection signal as a POE source, first set the comparator B and then set the POEG.

43.7 Usage Note

43.7.1 Module Stop Function Setting

Operation of comparator B can be enabled or disabled by setting a bit in the module stop control register B (MSTPCRB). Comparator B is initially disabled after a reset. Registers only become accessible after it has been released from the module stop state. For details, refer to section 11, Low Power Consumption.

44. Data Operation Circuit (DOC)

44.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16-bit values.
Table 44.1 lists the specifications of the DOC and Figure 44.1 is a block diagram of the DOC.
An interrupt can be generated if the result of 16-bit comparison meets one of the set interrupt conditions.

Table 44.1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
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, which is an overflow. • The result of data subtraction is less than 0000h, 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, which is an overflow. • The result of data subtraction is less than 0000h, which is an underflow.

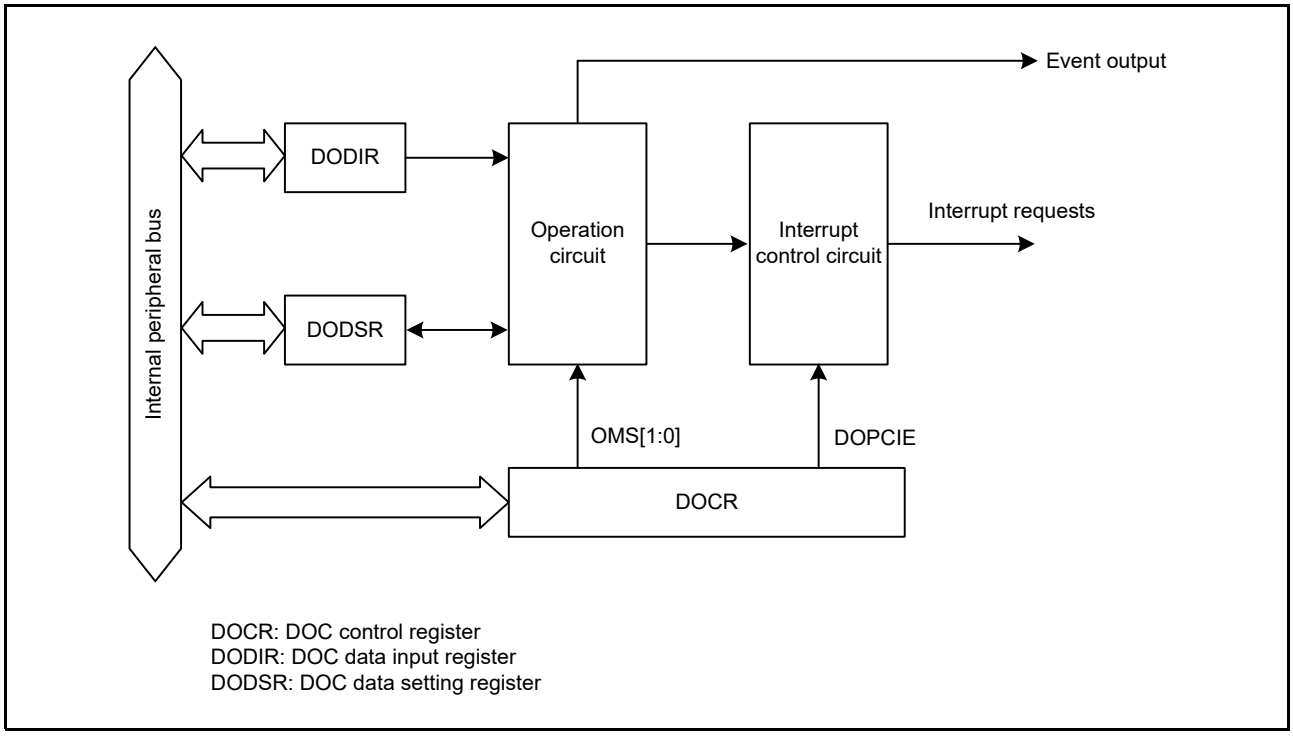


Figure 44.1 DOC Block Diagram

44.2 Register Descriptions

44.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	DCSEL	Detection Condition Select*1	0: ‘Not equal to’ is to be detected. 1: ‘Equal to’ is to be detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b5	DOPCF	Data Operation Result Flag	Indicates the result of an operation.	R
b6	DOPCFCL	Data Operation Result Clear	0: Retain the value of the DOPCF flag. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

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.

DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.
This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the DOC.

DOPCF Flag (Data Operation Result Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL Bit (Data Operation Result Clear)

Writing 1 to this bit clears the DOPCF flag.
This bit is read as 0.

44.2.2 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 0008 B082h

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

The DODIR register is a readable and writable register that holds values for use in operations.

44.2.3 DOC Data Setting Register (DODSR)

Address(es): DOC.DODSR 0008 B084h

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

The DODSR register is a readable and writable register that holds values for use in comparison or the results of other operations.
In data comparison mode, store the standard value for use in comparison in this register.
In data addition or data subtraction mode, this register holds the results of operations.

44.3 Operation

44.3.1 Data Comparison Mode

Figure 44.2 shows an example of the steps involved in data comparison mode operation by the DOC.
An example of operation when DCSEL is set to 0 (‘not equal to’ is to be detected as the result of data comparison) is shown below.

(1) Writing 00b to the DOCR.OMS[1:0] bits places the DOC in the data comparison mode.
(2) Specify the standard value for comparison in the DODSR register.
(3) Write the value for comparison in the DODIR register.
(4) Write all values for use in comparison to the DODIR register.
(5) If the value written to the DODIR register is not equal to the value set in the DODSR register, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

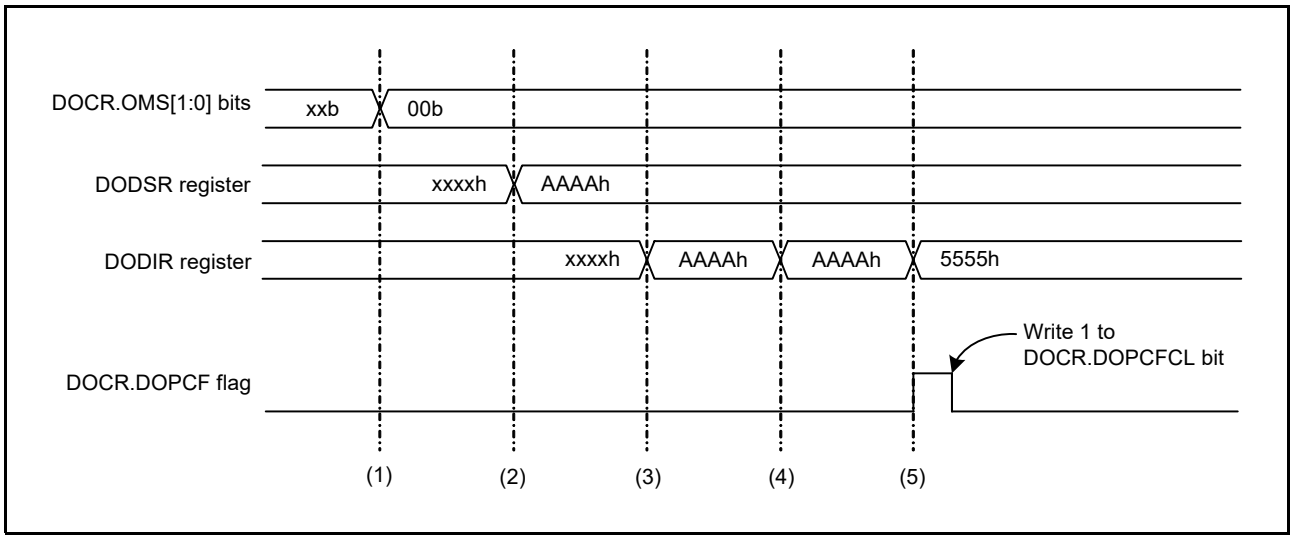


Figure 44.2 Example of Operation in Data Comparison Mode

44.3.2 Data Addition Mode

Figure 44.3 shows an example of the steps involved in data addition mode operation by the DOC.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) Set the initial value in the DODSR register.
(3) Write the value for addition in the DODIR register. The result of the operation is stored in DODSR.
(4) Write all values for use in addition to the DODIR register.
(5) If the result of the operation is greater than FFFFh, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

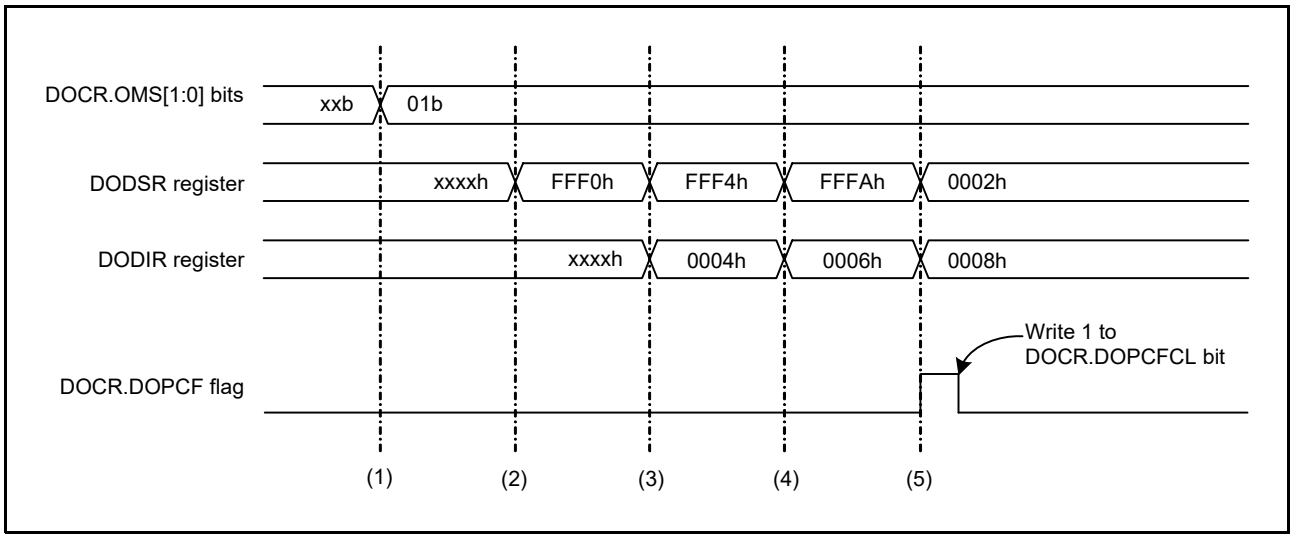


Figure 44.3 Example of Operation in Data Addition Mode

44.3.3 Data Subtraction Mode

Figure 44.4 shows an example of the steps involved in data subtraction mode operation by the DOC.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) Set the initial value in the DODSR register.
(3) Write the value for subtraction in the DODIR register. The result of the operation is stored in DODSR.
(4) Write all values for use in subtraction to the DODIR register.
(5) If the result of the operation is less than 0000h, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

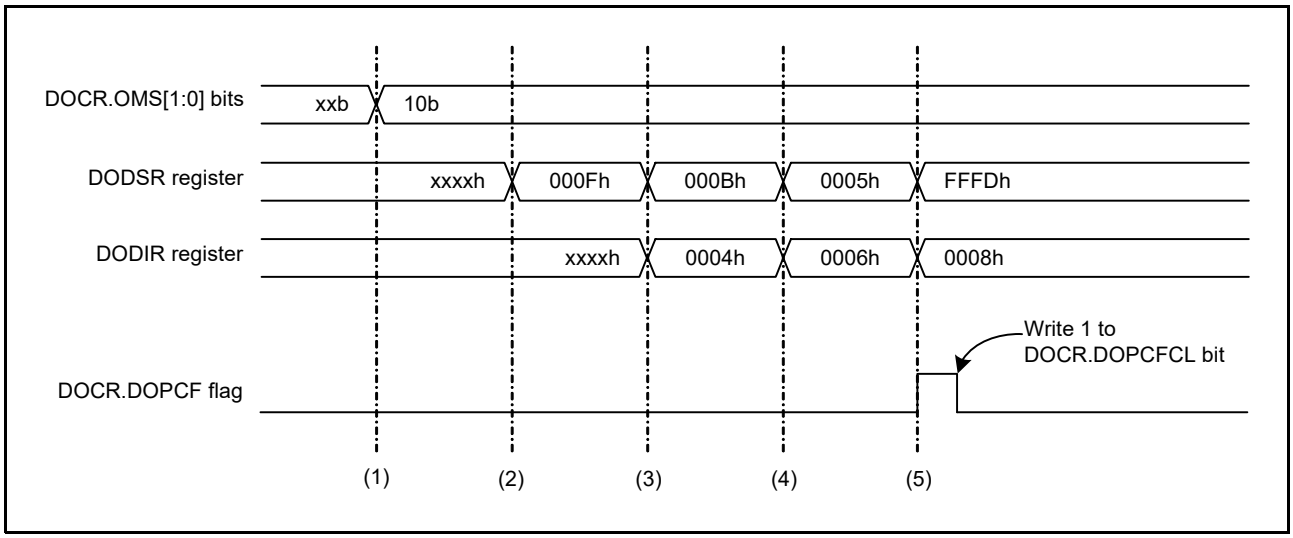


Figure 44.4 Example of Operation in Data Subtraction Mode

44.4 Interrupt Requests

The data operation circuit interrupt (DOPCI) is the interrupt request generated by the DOC. The DOCR.DOPCF flag becomes 1 when the interrupt source condition is satisfied.
Table 44.2 lists the details of the interrupt request.

Table 44.2 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. • The result of data subtraction is less than 0000h.

44.5 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.
- The result of data subtraction is less than 0000h.

44.5.1 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.

44.6 Usage Note

44.6.1 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.

45. RAM

This MCU has a 128-Kbyte high-speed static RAM (RAM), which runs in the no-wait state.

45.1 Overview

Table 45.1 lists the specifications of the RAM.

Table 45.1 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
Low power consumption function	Transitions to the module stop state are possible.
Error checking	• Parity error detection • A non-maskable interrupt or interrupt is generated in response to an error.

Note 1. When accessing across the 8-byte boundary, the number of cycles is doubled.
Note 2. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.2, System Control Register 1 (SYSCR1).

45.2 Register Descriptions

45.2.1 RAM Operating Mode Control Register (RAMMODE)

Address(es): RAM.RAMMODE 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.

45.2.2 RAM Error Status Register (RAMSTS)

Address(es): RAM.RAMSTS 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.

45.2.3 RAM Error Address Capture Register (RAMECAD)

Address(es): RAM.RAMECAD 0008 1208h

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

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0.	R
b18 to b3	READ	Error Address	The address where an error is found is read.	R
b31 to b19	—	Reserved	These bits are read as 0.	R

When parity checking is enabled, this register will hold the address where a parity check error was found.
The address of the 8-byte boundary below the location where the error was found is stored in this register at the same time the RAMSTS.RAMERR flag is set to 1.
The error address is not updated when the RAMERR flag is 1 (error has occurred). Its value does not change when parity checking is disabled because no parity check error is detected.
The RAMECAD register is initialized only by a reset.

45.2.4 RAM Protection Register (RAMPRCR)

Address(es): RAM.RAMPRCR 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.

45.3 Operation

45.3.1 Parity Checking

Enabling and disabling of parity checking can be selected through the RAMMODE register setting. In the initial state, parity checking is disabled. Even parity checking is used in this device.
1-bit parity check code is added to each 1-byte data for writing, and the parity is checked for reading.
If a 1-bit error is detected in the 1 byte when the parity is checked for reading, a RAM error interrupt can be generated. If a 2-bit error or more is detected in the 1 byte, errors cannot be correctly detected.
After power-on, parity check code is undefined until data is written. To use parity checking, write the initial value to all areas while parity checking is enabled before accessing to the RAM immediately after a reset.
Operation cannot be guaranteed if access is made to an area where the initial value is not written.

45.3.2 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.

45.3.3 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, see section 14, Interrupt Controller (ICUb).

Table 45.2 RAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
RAMERR	RAM error	Not possible	Not possible

45.4 Usage Notes

45.4.1 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.

45.4.2 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.

45.4.3 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).

46. Flash Memory (FLASH)

This MCU has packages with 256, 384, and 512 Kbyte flash memory (ROM) for storing code and 8-Kbyte flash memory (E2 DataFlash) for storing data.
In this section, “PCLK” is used to refer to PCLKB.

46.1 Overview

Table 46.1 lists the flash memory specifications.
Table 46.8 lists the I/O pins used in boot mode.

Table 46.1 Flash Memory Specifications

Item	Description
Memory space	• User area: Up to 512 Kbytes • Data area: 8 Kbytes • Extra area: Stores the start-up area information, access window information, and unique ID
Operating clock	• FCLK: 1 to 64 MHz (for ROM P/E mode and E2 DataFlash P/E mode) up to 64 MHz (for E2 DataFlash read mode) • HOCO clock: 24 MHz, 32 MHz, 48 MHz, or 64 MHz (for ROM P/E mode and E2 DataFlash P/E mode)
Software commands	• The following commands are implemented: Program, blank check, block erase, and all-block erase • The following commands are implemented for programming the extra area: Start-up area information program, access window protection, and access window information program
Value after erasure	• ROM: FFh • E2 DataFlash: FFh
Interrupt	An interrupt (FRDYI) is generated upon completion of software command processing or forced stop processing.
On-board programming	Boot mode (SCI Interface)*1 • Channel 1 of the serial communications interface (SCI1) is used for asynchronous serial communication. • The user area and data area are rewritable. Boot mode (FINE interface)*1 • The FINE is used. • The user area and data area are rewritable. Boot mode (USB interface)*1 • Channel 0 of the USB 2.0 function (USB0) module is used. • The user area and data area are rewritable. • The flash memory can be rewritable in self-powered or bus-powered mode. • A personal computer can be connected using only a USB cable. Self-programming in single-chip mode • The user area and data area are rewritable using the flash rewrite routine in the user program.
Off-board programming	The user area and data area are rewritable using a flash programmer compatible with this MCU.
ID code protection	• Connection with the serial programmer can be enabled or disabled using ID codes in boot mode. • Connection with the on-chip debugging emulator can be enabled or disabled using ID codes.
Start-up program protection	This function is used to safely rewrite block 0 to block 7.
Area protection	This function enables rewriting only the selected blocks in the user area and disables the other blocks during self-programming.
Background Operation (BGO)	Programs on the ROM can be executed while rewriting the E2 DataFlash.

Note 1. Refer to the manual of each serial programmer and “Renesas Flash Programmer Flash memory programming software User’s Manual” for more details.

46.2 ROM Area and Block Configuration

The maximum ROM size of this MCU is 512 Kbytes. The ROM area is divided into blocks. A block is 2-Kbyte area. When executing the block erase command, the memory is erased by the block. Figure 46.1 shows the ROM area and block configuration.

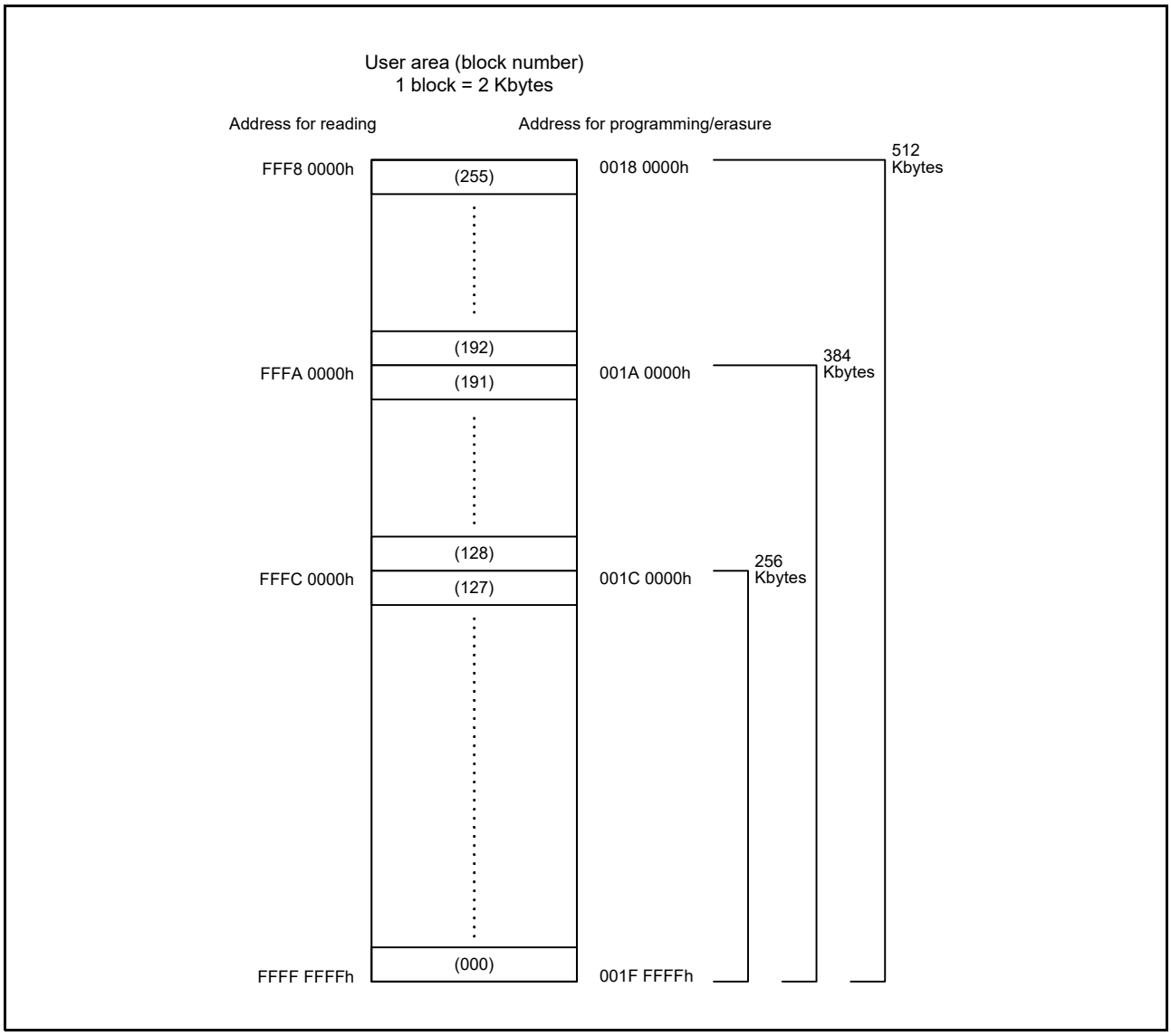


Figure 46.1 ROM Area and Block Configuration

Table 46.2 Correspondence Between ROM Capacity and Addresses for Reading

ROM Capacity	Addresses for Reading
512 Kbytes	FFF8 0000h to FFFF FFFFh
384 Kbytes	FFFA 0000h to FFFF FFFFh
256 Kbytes	FFFC 0000h to FFFF FFFFh

46.3 E2 DataFlash Area and Block Configuration

The E2 DataFlash is 8 Kbytes in the MCU. The E2 DataFlash is divided into blocks and erased in block units. Figure 46.2 shows the E2 DataFlash area and the block configuration.

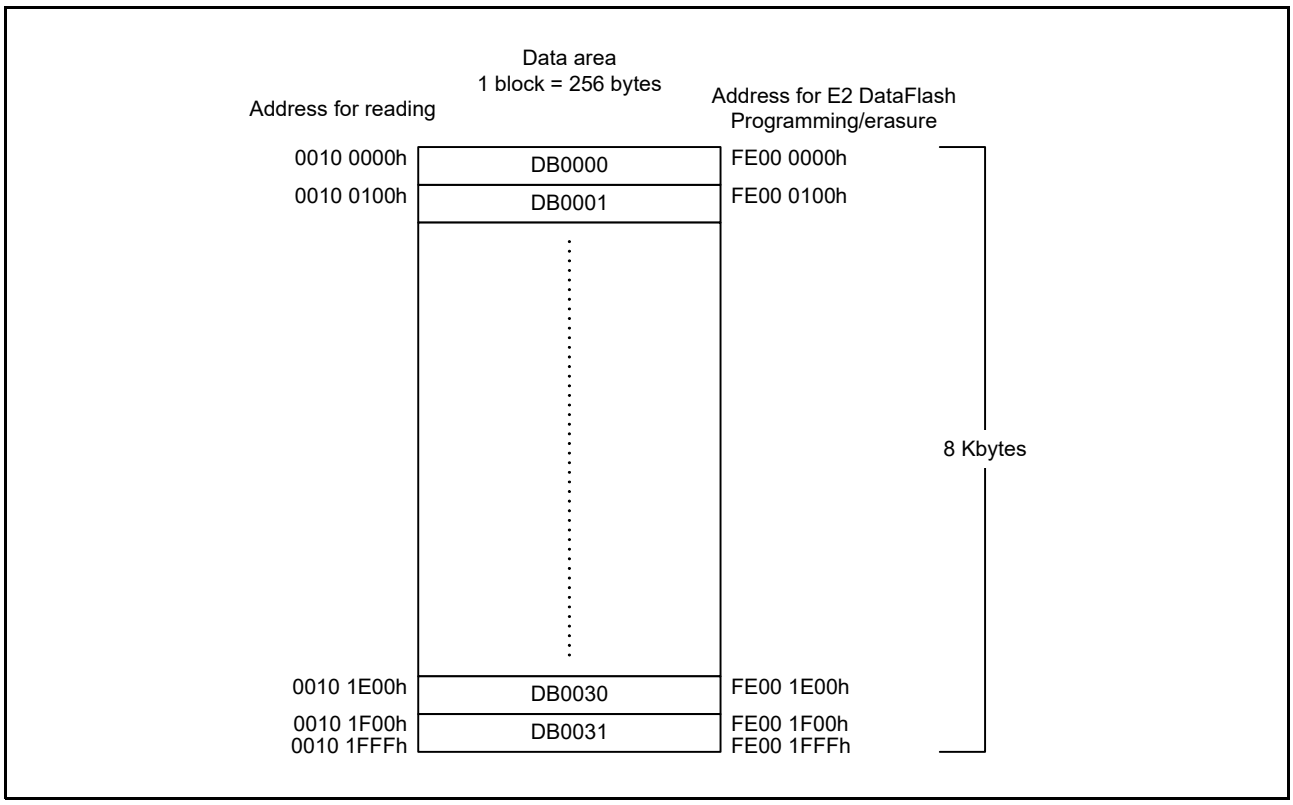


Figure 46.2 E2 DataFlash Area and Block Configuration

46.4 Register Descriptions

46.4.1 E2 DataFlash Control Register (DFLCTL)

Address(es): FLASH.DFLCTL 007F C090h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DFLEN	E2 DataFlash Access Enable	0: Access to E2 DataFlash and access to the extra area in P/E mode*1 disabled 1: Access to E2 DataFlash and access to the extra area in P/E mode*1 enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Start-up area information program, access window protection, and access window information program

The DFLCTL register is used to enable or disable access (read, program, and erase) to the E2 DataFlash and access (start-up area information program, access window protection, and access window information program) to the extra area in P/E mode.

When reading, programming, and erasing the E2 DataFlash, set the DFLCTL.DFLEN bit to 1 and wait for the E2 DataFlash STOP recovery time (tDSTOP) to elapse before reading the E2 DataFlash and entering E2 DataFlash P/E mode. Do not read the E2 DataFlash or enter E2 DataFlash P/E mode until tDSTOP has elapsed.

Refer to section 46.7.1, Sequencer Modes for details on E2 DataFlash P/E mode. Refer to section 47, Electrical Characteristics for E2 DataFlash STOP recovery time (tDSTOP).

46.4.2 Flash P/E Mode Entry Register (FENTRYR)

Address(es): FLASH.FENTRYR 007F FFB0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRY0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FENTRY0	ROM P/E Mode Entry 0	0: ROM is in read mode. 1: ROM can be placed in P/E mode.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	0: E2 DataFlash is in read mode. 1: E2 DataFlash can be placed in P/E mode.	R/W
b15 to b8	FEKEY[7:0]	Key Code	The FEKEY[7:0] bits are used to control rewriting of the FENTRYR register. When rewriting the value of the lower 8 bits, set the FEKEY[7:0] bits to AAh at the same time (write this register in 16 bits). The FEKEY[7:0] bits are read as 00h.	R/W

To rewrite the ROM or E2 DataFlash, the FENTRYD or FENTRY0 bit must be set to 1 after oscillating HOCO to place the ROM or E2 DataFlash in P/E mode.
When returning to read mode, set the FENTRYR register and confirm that its value has been rewritten before reading the ROM or E2 DataFlash.
Refer to section 46.7.1, Sequencer Modes for details on P/E mode and read mode.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place the ROM in P/E mode.
[Setting condition]

- AA01h is written to the FENTRYR register when the FENTRYR register is 0000h.

Note: When entering ROM P/E mode, the instruction fetch address must be transferred to an area other than the ROM so that instruction fetching is not executed to the ROM. Copy necessary instruction code to the internal RAM and jump to the RAM. Note that E2 DataFlash can be rewritten by a program in the ROM.

[Clearing condition]

- AA00h is written to the FENTRYR register.

FENTRYD Bit (E2 DataFlash P/E Mode Entry)

This bit is used to place the E2 DataFlash in P/E mode.
[Setting condition]

- AA80h is written to the FENTRYR register when the FENTRYR register is 0000h.

[Clearing condition]

- AA00h is written to the FENTRYR register.

46.4.3 Memory Wait Cycle Setting Register (MEMWAITR)

Address(es): FLASH.MEMWAITR 007F FFC0h

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

Bit	Symbol	Bit Name	Description	R/W
b0	MEMWAIT	Memory Wait Cycle Setting	0: No wait states 1: Wait states	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	MEKEY[7:0]	Key Code	These bits are used to control rewriting of the MEMWAITR register. When rewriting the value of the lower 8 bits, set these bits to AAh at the same time (write this register in 16-bit units). These bits are read as 00h.	R/W

The MEMWAITR register is used to control the wait cycle of the ROM.

MEMWAIT Bit (Memory Wait Cycle Setting)

This bit is used to specify the wait cycle of the ROM.

When setting the system clock (ICLK) frequency higher than 32 MHz, set the MEMWAIT bit to 1. Set the MEMWAIT bit to 0 in middle-speed operating mode, middle-speed operating mode 2, and low-speed operating mode.

Change the value of the MEMWAIT bit when the MCU is in high-speed operating mode and the ICLK frequency is 32 MHz or less.

Table 46.3 Restrictions on Setting of the MEMWAIT bit

	High-Speed Operating Mode		Middle-Speed Operating Mode Middle-Speed Operating Mode 2 Low-Speed Operating Mode
MEMWAIT bit	ICLK > 32 MHz	ICLK ≤ 32 MHz	
0	Setting prohibited	Setting possible	Setting possible
1	Setting possible	Setting possible	Setting prohibited

46.4.4 Protection Unlock Register (FPR)

Address(es): FLASH.FPR 007F C180h

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

x: Undefined

This write-only register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control. Writing to the FPMCR register is enabled only when the following procedure is used to access the register.

Procedure to unlock protection

(1) Write A5h to the FPR register.
(2) Write a set value to the FPMCR register.
(3) Write the inverted set value to the FPMCR register.
(4) Write a set value to the FPMCR register again.

When a procedure other than the above is used to write data, the FPSR.PERR flag is set to 1.

46.4.5 Protection Unlock Status Register (FPSR)

Address(es): FLASH.FPSR 007F C184h

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

Bit	Symbol	Bit Name	Description	R/W
b0	PERR	Protect Error Flag	0: No error 1: An error occurs.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

PERR Flag (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection.

[Clearing condition]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 46.4.4, Protection Unlock Register (FPR).

46.4.6 Flash P/E Mode Control Register (FPMCR)

Address(es): FLASH.FPMCR 007F C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FMS1	RPDIS	—	FMS0	—
Value after reset:	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	FMS0	Flash Operating Mode Select 0	FMS1 FMS0 0 0: ROM/E2 DataFlash read mode 0 1: ROM P/E mode 1 0: E2 DataFlash P/E mode 1 1: Setting prohibited.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RPDIS	ROM P/E Disable	0: ROM programming/erasure enabled 1: ROM programming/erasure disabled	R/W
b4	FMS1	Flash Operating Mode Select 1	See the FMS0 bit.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The FPMCR register is used to set the operating mode of the flash memory.
This register is protected. Set its value using the procedure to unlock protection. For details, refer to section 46.4.4, Protection Unlock Register (FPR).
When entering ROM P/E mode or during ROM P/E mode, an instruction must be executed on the RAM.

FMS0 and FMS1 Bits (Flash Operating Mode Select 0 and Flash Operating Mode Select 1)

These bits are used to set the operating mode of the flash memory.

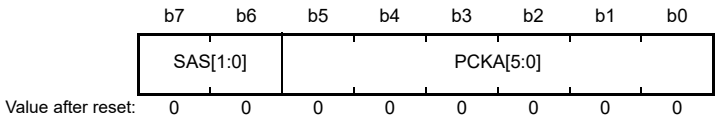
- Transition from read mode to ROM P/E mode
  Set the FMS1 bit = 0, the FMS0 bit = 1, and the RPDIS bit = 0.
- Transition from ROM P/E mode to read mode
  Set the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.
  Wait for ROM mode transition wait time (tMS, refer to section 47, Electrical Characteristics).
- Transition from read mode to E2 DataFlash P/E mode
  Set the FMS1 bit = 1, the FMS0 bit = 0, and the RPDIS bit = 0.
- Transition from E2 DataFlash P/E mode to read mode
  Set the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.
  Wait for ROM mode transition wait time (tMS, refer to section 47, Electrical Characteristics).

RPDIS Bit (ROM P/E Disable)

This bit is used to disable the execution of ROM programming/erasure with software.

46.4.7 Flash Initial Setting Register (FISR)

Address(es): FLASH.FISR 007F C1D8h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PCKA[5:0]	Peripheral Clock Notification	These bits are used to set the frequency of the FlashIF clock (FCLK).	R/W
b7, b6	SAS[1:0]	Start-Up Area Select	b7 b6 0 x: The start-up area is selected according to the start-up area settings of the extra area. 1 0: The start-up area is switched to the default area temporarily. 1 1: The start-up area is switched to the alternate area temporarily.	R/W

x: Don’t care

Data can be written to the FISR register in ROM P/E mode or E2 DataFlash P/E mode.

PCKA[5:0] Bits (Peripheral Clock Notification)

These bits are used to set the frequency of the FlashIF clock (FCLK) when programming/erasing the ROM/E2 DataFlash.

Set the FCLK frequency in the PCKA[5:0] bits before programming/erasure. Do not change the frequency during programming/erasure of the ROM/E2 DataFlash.

- When FCLK is higher than 4 MHz
  Set a rounded-up value for a non-integer frequency.
  For example, set 32 MHz (PCKA[5:0] bits = 011111b) when the frequency is 31.5 MHz.
- When FCLK is 4 MHz or lower
  Do not use a non-integer frequency.
  Use the FCLK at a frequency of 1, 2, 3, or 4 MHz.

Note: When the PCKA[5:0] bits are set to a frequency different from the FCLK, the data in the ROM/E2 DataFlash may be damaged.

Table 46.4 Example of FlashIF Clock Frequency Settings

FlashIF Clock Frequency (MHz)	PCKA[5:0] Bit Setting	FlashIF Clock Frequency (MHz)	PCKA[5:0] Bit Setting	FlashIF Clock Frequency (MHz)	PCKA[5:0] Bit Setting
64	101111b	62	101110b	60	101101b
58	101100b	56	101011b	54	101010b
52	101001b	50	101000b	48	100111b
46	100110b	44	100101b	42	100100b
40	100011b	38	100010b	36	100001b
34	100000b	32	011111b	31	011110b
30	011101b	29	011100b	28	011011b
27	011010b	26	011001b	25	011000b
24	010111b	23	010110b	22	010101b
21	010100b	20	010011b	19	010010b
18	010001b	17	010000b	16	001111b
15	001110b	14	001101b	13	001100b
12	001011b	11	001010b	10	001001b
9	001000b	8	000111b	7	000110b
6	000101b	5	000100b	4	000011b
3	000010b	2	000001b	1	000000b

SAS[1:0] Bits (Start-Up Area Select)

These bits are used to select the start-up area. To change the start-up area, the following three methods can be used.

- When selecting the start-up area according to the start-up area settings of the extra area
  With the SAS[1:0] bits set to 00b or 01b, the start-up area is selected according to the start-up area settings of the extra area. The settings are enabled after a reset is released.
- When switching the start-up area to the default area temporarily
  When 10b is written to the SAS[1:0] bits, the start-up area is switched to the default area immediately after data is written to the register, regardless of the start-up area settings of the extra area.
  When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.
- When switching the start-up area to the alternative area temporarily
  When 11b is written to the SAS[1:0] bits, the start-up area is switched to the alternative area, regardless of the start-up area settings of the extra area.
  When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.

46.4.8 Flash Reset Register (FRESETR)

Address(es): FLASH.FRESETR 007F C124h

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

Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: Flash control circuit reset is released. 1: Flash control circuit is reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FRESET Bit (Flash Reset)

When this bit is set to 1, registers FASR, FSARH, FSARL, FEARH, FEARL, FWB0, FWB1, FWB2, FWB3, FCR, and FEXCR are reset. Also, the values of registers FEAMH and FEAML are undefined. Do not access these registers during a reset. To release the reset, set this bit to 0.

Do not write to this register while executing a software command or rewriting the extra area.

46.4.9 Flash Area Select Register (FASR)

Address(es): FLASH.FASR 007F C104h

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

Bit	Symbol	Bit Name	Description	R/W
b0	EXS	Extra Area Select	0: User area or data area 1: Extra area	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Data can be written to the FASR register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1.

Data cannot be written to this register while the FRESETR.FRESET bit is 1.

EXS Bit (Extra Area Select)

Set this bit to 1 before issuing a software command (start-up area information program, access window protection, or access window information program) for the extra area. Set this bit to 0 before issuing a software command (program, blank check, block erase, or all-block erase) for the user area.

After issuing a software command, do not change the value until changing it for issuing the next software command.

46.4.10 Flash Control Register (FCR)

Address(es): FLASH.FCR 007F C114h

	b7	b6	b5	b4	b3	b2	b1	b0
	OPST	STOP	—	—	CMD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMD[3:0]	Software Command Setting	b3 b0 0 0 0 1: Program 0 0 1 1: Blank check 0 1 0 0: Block erase 0 1 1 0: All-block erase Settings other than above are prohibited.*1	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	STOP	Forced Processing Stop	When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FCR register to 00h when the FSTATR1.FRDY flag is 1.

Data can be written to the FCR register when in ROM P/E mode and the ROM can be programmed/erased or in E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[3:0] Bits (Software Command Setting)

These bits are used to set a software command (program, blank check, block erase, or all-block erase).
The function of each command is described below.

- Program
  Write the value set in registers FWB0, FWB1, FWB2, and FWB3 to the address set in registers FSARH and FSARL.
- Blank check
  Check whether there is data in the area from the address set in registers FSARH and FSARL to the address set in registers FEARH and FEARL. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
- Block erase
  Erase consecutive areas specified in the flash memory by the blocks. Set the beginning address of the block in registers FSARH and FSARL and the end address in registers FEARH and FEARL.
- All-block erase
  Erase all blocks in the ROM or E2 DataFlash.
  All-block erase requires less time to erase the memory compared to block erase. When erasing the whole of the ROM area, set the beginning address of the ROM area in registers FSARH and FSARL, and the end address in registers FEARH and FEARL. Table 46.5 lists the setting address for all-block erase.

Table 46.5 Setting Address for All-Block Erase

Target	Memory Size	FSARH/FSARL	FEARH/FEARL
ROM	512 Kbytes	0018 0000h	001F FFF8h
	384 Kbytes	001A 0000h	001F FFF8h
	256 Kbytes	001C 0000h	001F FFF8h
E2 DataFlash	8 Kbytes	FE00 0000h	FE00 1FFFh

STOP Bit (Forced Processing Stop)

This bit is used to forcibly stop the processing (blank check, block erase, or all-block erase) being executed.
After setting this bit to 1, wait until the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[3:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FRDY flag is 0 before executing the next processing.

46.4.11 Flash Extra Area Control Register (FEXCR)

Address(es): FLASH.FEXCR 007F C1DCh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Software Command Setting	b2 b0 0 0 1: Start-up area information program/access window protection*1 0 1 0: Access window information program Settings other than above are prohibited.*2	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. Once protected, the protection cannot be removed.
Note 2. This does not include set the FEXCR register to 00h when the FSTATR1.EXRDY flag is 1.

Data can be written to the FEXCR register when in ROM P/E mode and the ROM can be programmed/erased.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[2:0] Bits (Software Command Setting)

These bits are used to set a software command (start-up area information program, access window protection, or access window information program).
The details of each command are described below.

- Start-up area information program
  This command is used to switch the start-up area used for start-up program protection.
  When setting the start-up area to the default area, set the FWB1 and FWB0 registers to FFFFh, and execute this command.
  When setting the start-up area to the alternative area, set the FWB1 register to FFFFh, set the FWB0 register to FEFFh, and execute this command.
  When the FWB1 and FWB0 registers are set to values other than the above, do not execute the start-up area information program.
- Access window protection
  This command is used to protect the settings for the access window.
  When the access window protection command is executed after setting the FWB1 register to FFFFh, bit 14 of the FWB0 register to 0, bit 8 to the same value as the FSCMR.SASMF flag, and the other bits to 1, the settings for the access window are protected. Once protected, the protection cannot be removed.
- Access window information program
  This command is used to set the access window used for area protection.
  Set the access window in block units.
  Specify the access window start address, which is the beginning address of the access window in the FWB0 register, specify the access window end address, which is the next address of the last address of the access window in the FWB1 register, and issue this command. Set bit 21 to bit 11 of the address for programming/erasure in each register.
  If the same value is set as the start address and end address, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[2:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.EXRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FSTATR1.EXRDY flag is 0 before executing the next processing.
Writing to the extra area is started by writing 1 to the OPST bit. Do not write to the CMD[2:0] bits while a software command is being executed.

46.4.12 Flash Processing Start Address Register H (FSARH)

Address(es): FLASH.FSARH 007F C110h

	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 FSARH register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 31 to bit 25 and bit 20 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 46.1 and Figure 46.2 for details on the addresses of the flash memory.

46.4.13 Flash Processing Start Address Register L (FSARL)

Address(es): FLASH.FSARL 007F C108h

	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 FSARL register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
When the target is the ROM, set bit 2 to bit 0 to 000b.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
This register is incremented by 8h if the ROM is specified and 1h if the E2 DataFlash is specified after a program command is executed. Therefore, it is not necessary to set the target address to be written to this register when executing a program command sequentially.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 46.1 and Figure 46.2 for details on the addresses of the flash memory.

46.4.14 Flash Processing End Address Register H (FEARH)

Address(es): FLASH.FEARH 007F C120h

	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 FEARH register is used to set the end address of the target processing range in the flash memory when a software command is executed.
Set bit 31 to bit 25 and bit 20 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 46.1 and Figure 46.2 for details on the addresses of the flash memory.

46.4.15 Flash Processing End Address Register L (FEARL)

Address(es): FLASH.FEARL 007F C118h

	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 FEARH register is used to set the end address of the target range for processing when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
When the target is the ROM, set bit 2 to bit 0 to 000b.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 46.1 and Figure 46.2 for details on the addresses of the flash memory.

46.4.16 Flash Write Buffer Register n (FWBn) (n = 0 to 3)

Address(es): FLASH.FWB0 007F C130h, FLASH.FWB1 007F C138h, FLASH.FWB2 007F C140h, FLASH.FWB3 007F C144h

	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 programming the ROM, E2 DataFlash, or extra area. The data can be written in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

The read value of this register is undefined while executing a software command set by the FCR register or the FEXCR register.

When programming the extra area, set the 4-byte data for programming in registers FWB0 and FWB1.

When programming the E2 DataFlash, set the data for programming in the lower 8 bits in the FWB0 register.

When programming the ROM, set the 8-byte data for programming in registers FWB0 to FWB3. Figure 46.3 shows the relationship between the addresses indicated by registers FSARH and FSARL and the data set in the FWBn register.

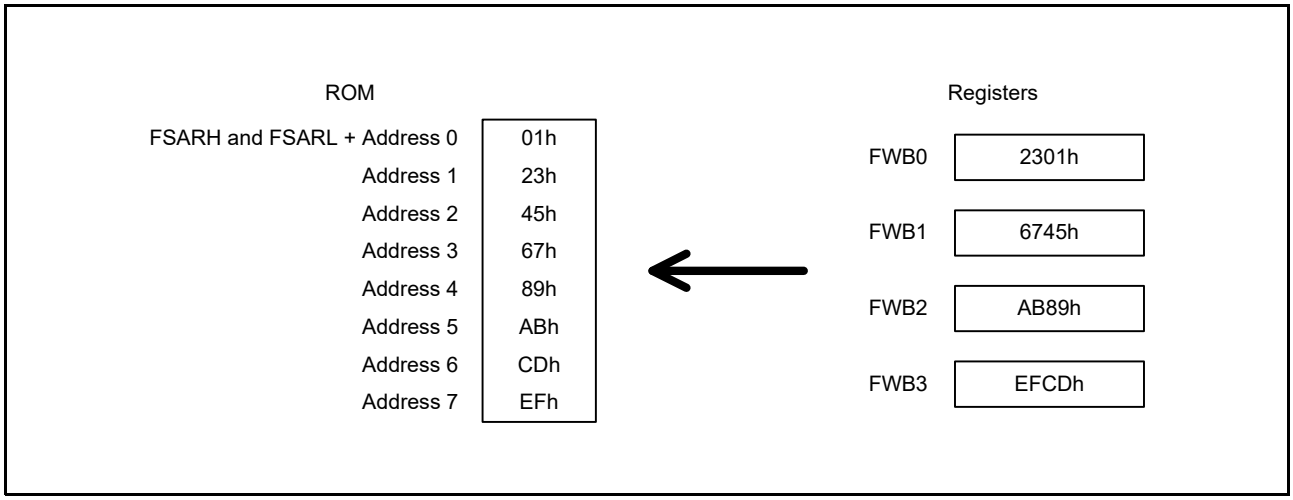


Figure 46.3 FWBn Register Setting Values and Data Allocation in the ROM

46.4.17 Flash Status Register 0 (FSTATR0)

Address(es): FLASH.FSTATR0 007F C128h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EILGLERR	ILGLERR	BCERR	—	PRGERR	ERERR
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ERERR	Erase Error Flag	0: Erasure terminates normally. 1: An error occurs during erasure.	R
b1	PRGERR	Program Error Flag	0: Programming terminates normally. 1: An error occurs during programming.	R
b2	—	Reserved	The read value is undefined.	R
b3	BCERR	Blank Check Error Flag	0: Blank checking terminates normally. 1: An error occurs during blank checking.	R
b4	ILGLERR	Illegal Command Error Flag	0: No illegal software command or illegal access is detected. 1: An illegal command or illegal access is detected.	R
b5	EILGLERR	Extra Area Illegal Command Error Flag	0: No illegal command or illegal access to the extra area is detected. 1: An illegal command or illegal access to the extra area is detected.	R
b7, b6	—	Reserved	The read value is undefined.	R

This register is a status register used to confirm the result of executing a software command. Each error flag is set to 0 when the next software command is executed.

ERERR Flag (Erase Error Flag)

This flag indicates the result of the erase processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during erasure.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR Flag (Program Error Flag)

This flag indicates the result of the program processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during programming.

[Clearing condition]

- The next software command is executed.

BCERR Flag (Blank Check Error Flag)

This flag indicates the result of the blank check processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during blank checking.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during blank checking.

ILGLERR Flag (Illegal Command Error Flag)

This flag indicates the result of executing a software command.
[Setting conditions]

- Programming/erasure is executed to an area other than the access window range.
- A blank check or block erase command is executed when the set value of registers FSARH and FSARL is larger than the set value of registers FEARH and FEARL.
- Program and block erase commands are executed when the FASR.EXS bit is 1.
- An all-block erase command is executed while the access window is set.
- An all-block erase command is executed without setting registers FSARH and FSARL and registers FEARH and FEARL properly.
- The E2 DataFlash address is set in registers FSARH and FSARL and a software command is executed when the ROM is in P/E mode.
- The ROM address is set in registers FSARH and FSARL and a software command is executed when the E2 DataFlash is in P/E mode.
- The ROM and E2 DataFlash are set to P/E mode and a software command is executed.

[Clearing condition]

- The next software command is executed.

EILGLERR Flag (Extra Area Illegal Command Error Flag)

This flag indicates the result of executing a software command for the extra area.
[Setting condition]

- A software command for the extra area is executed when the FASR.EXS bit is 0.

[Clearing condition]

- The next software command is executed.

46.4.18 Flash Status Register 1 (FSTATR1)

Address(es): FLASH.FSTATR1 007F C12Ch

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0.	R
b2	—	Reserved	This bit is read as 1.	R
b5 to b3	—	Reserved	These bits are read as 0.	R
b6	FRDY	Flash Ready Flag	0: Other than below 1: 00h can be written to the FCR register (processing to complete the software command).	R
b7	EXRDY	Extra Area Ready Flag	0: Other than below 1: 00h can be written to the FEXCR register (processing to complete the software command).	R

This register is a status register used to confirm the result of executing a software command. Each flag is set to 0 when the next software command is executed.

FRDY Flag (Flash Ready Flag)

This flag is used to confirm whether a software command is executed.
This flag becomes 1 when processing of the executed software command or the forced stop processing is completed, and this flag becomes 0 when setting the FCR.OPST bit to 0.
Also, an interrupt (FRDYI) is generated when this flag becomes 1.

EXRDY Flag (Extra Area Ready Flag)

This flag is used to confirm whether a software command for the extra area is executed.
This flag is set to 1 when processing of the executed software command is completed, and 0 when the FEXCR.OPST bit is set to 0.
Also, an interrupt (FRDYI) is generated when this flag becomes 1.

46.4.19 Flash Error Address Monitor Register H (FEAMH)

Address(es): FLASH.FEAMH 007F C1E8h

	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 check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 31 to bit 25 and bit 20 to bit 16 of the address where the error has occurred for the program command or blank check command, or it stores bit 31 to bit 25 and bit 20 to bit 16 of the beginning address of the area where the error has occurred for the block erase command or all-block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
If the software command terminates normally, this register stores bit 31 to bit 25 and bit 20 to bit 16 of the end address at execution of the command.
Refer to Figure 46.1 and Figure 46.2 for details on the addresses of the flash memory.

46.4.20 Flash Error Address Monitor Register L (FEAML)

Address(es): FLASH.FEAML 007F C1E0h

	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 check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the address where the error has occurred for the program command or blank check command, or it stores bit 15 to bit 0 of the beginning address of the area where the error has occurred for the block erase command or all-block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
When the software command is normally completed, this register stores bit 15 to bit 0 of the last address at execution of the command.
When executing a software command for the ROM, lower 3 bits become 000b.
Refer to Figure 46.1 and Figure 46.2 for details on the addresses of the flash memory.

46.4.21 Flash Start-Up Setting Monitor Register (FSCMR)

Address(es): FLASH.FSCMR 007F C1C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	AWPR	—	—	—	—	—	SASMF	—	—	—	—	—	—	—	—
Value after reset:	0	Value set by user*1	1	1	0	1	1	Value set by user*2	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.	R
b8	SASMF	Start-Up Area Setting Monitor Flag	0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
b10, b9	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b11	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b13, b12	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b14	AWPR	Access Window Protection Flag	0: The settings for the access window cannot be rewritten. 1: The settings for the access window can be rewritten.	R
b15	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bit 14 in the FWB0 register after the access window protection command is executed.

Note 2. The value of the blank product is 1. It is set to the same value set in bit 8 in the FWB0 register after the start-up area information program command is executed.

SASMF Flag (Start-Up Area Setting Monitor Flag)

This flag is used to confirm the settings of the start-up area.

When this flag is 0, the user program is set to start up using the alternative area.

When this flag is 1, the user program is set to start up using the default area.

AWPR Flag (Access Window Protection Flag)

This flag is used to confirm whether the settings of the access window are protected or not.

When this flag is 0, the access window start and end addresses cannot be rewritten.

When this flag is 1, the access window start and end addresses can be rewritten.

46.4.22 Flash Access Window Start Address Monitor Register (FAWSMR)

Address(es): FLASH.FAWSMR 007F C1C8h

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

Note 1. The value of the blank product is 1. It is set to the same value set in bit 10 to bit 0 the FWB0 register after the access window information program command is executed.

This register is used to confirm the set value of the access window start address used for area protection.

46.4.23 Flash Access Window End Address Monitor Register (FAWEMR)

Address(es): FLASH.FAWEMR 007F C1D0h

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

Note 1. The value of the blank product is 1. It is set to the same value set in bit 10 to bit 0 in the FWB1 register after the access window information program command is executed.

This register is used to confirm the set value of the access window end address used for area protection.

46.4.24 Unique ID Register n (UIDRn) (n = 0 to 3)

Address(es): FLASH.UIDR0 007F C350h, FLASH.UIDR1 007F C354h, FLASH.UIDR2 007F C358h, FLASH.UIDR3 007F C35Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each chip															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip															

The UIDRn register stores a 16-byte ID code (unique ID) for identifying the individual MCU.
The unique ID is stored in the extra area of the flash memory and cannot be rewritten by the user.

46.5 Start-Up Program Protection

When rewriting the start-up program*1 by self-programming, if the rewrite operation is interrupted due to temporary blackout, the start-up program may not be successfully programmed and the user program may not start properly. This problem can be avoided by rewriting the start-up program without erasing the existing start-up program using the start-up program protection. This function is available in products with a 32-Kbyte or larger ROM.
Figure 46.4 shows the overview of the start-up program protection. In this figure, the default area indicates block 0 to block 7, and the alternate area indicates block 8 to block 15.

Note 1. Program to perform operation to start the user program. It includes the fixed vector table.

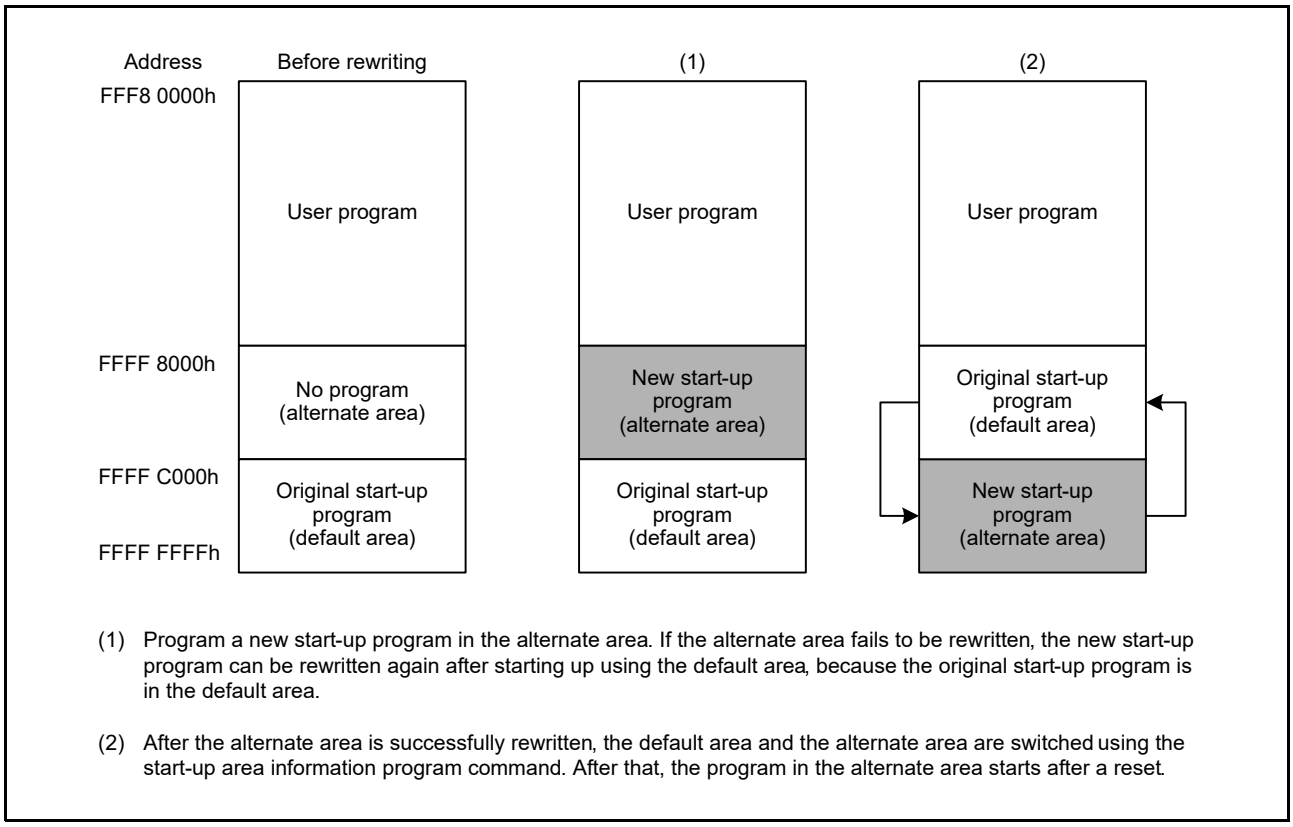


Figure 46.4 Overview of the Start-Up Program Protection

46.6 Area Protection

Area protection enables rewriting only the selected blocks (access window) in the user area and disables rewriting the other blocks during self-programming. The access window cannot be set in the data area.
Specify the start address and end address to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.
Figure 46.5 shows the overview of the area protection.

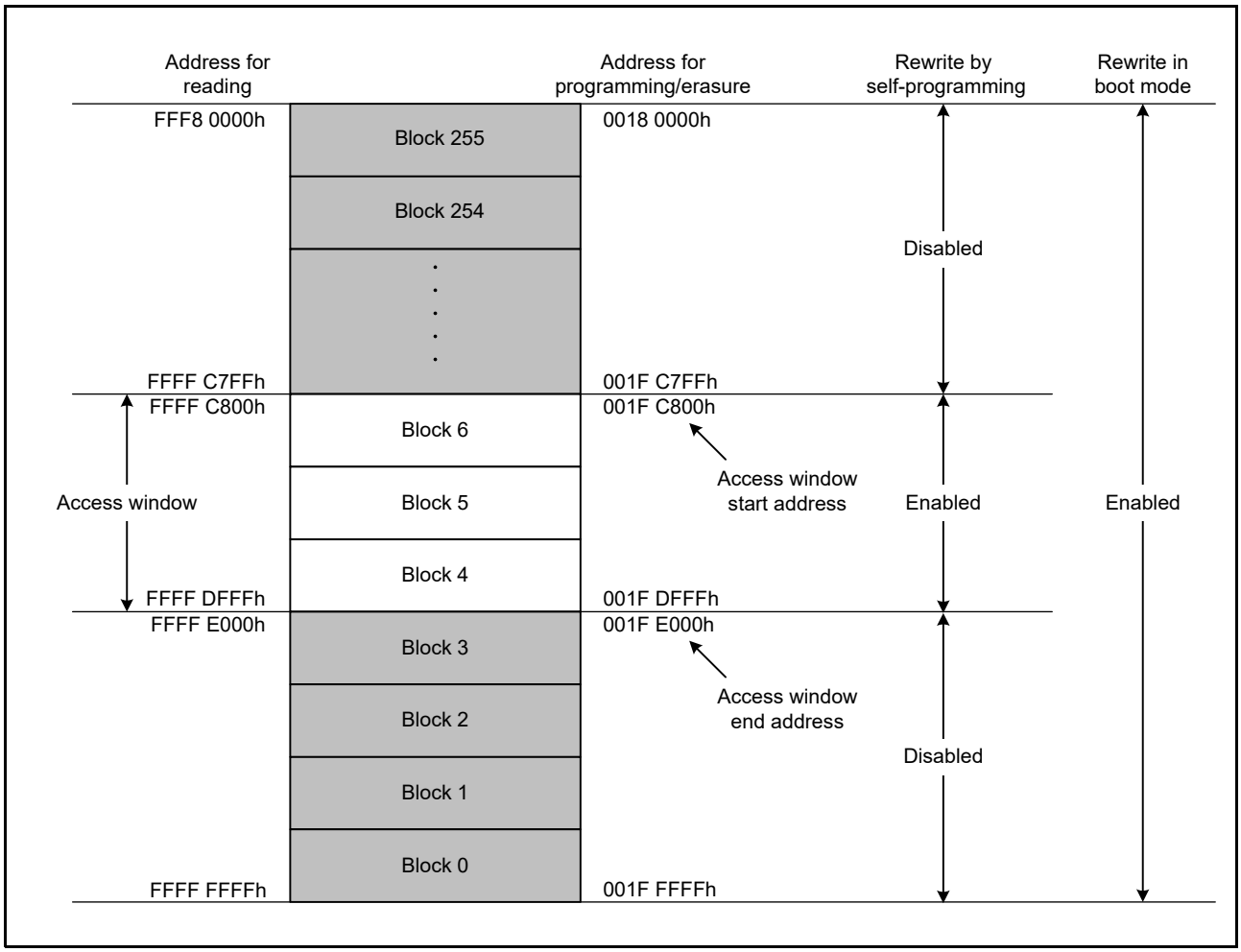


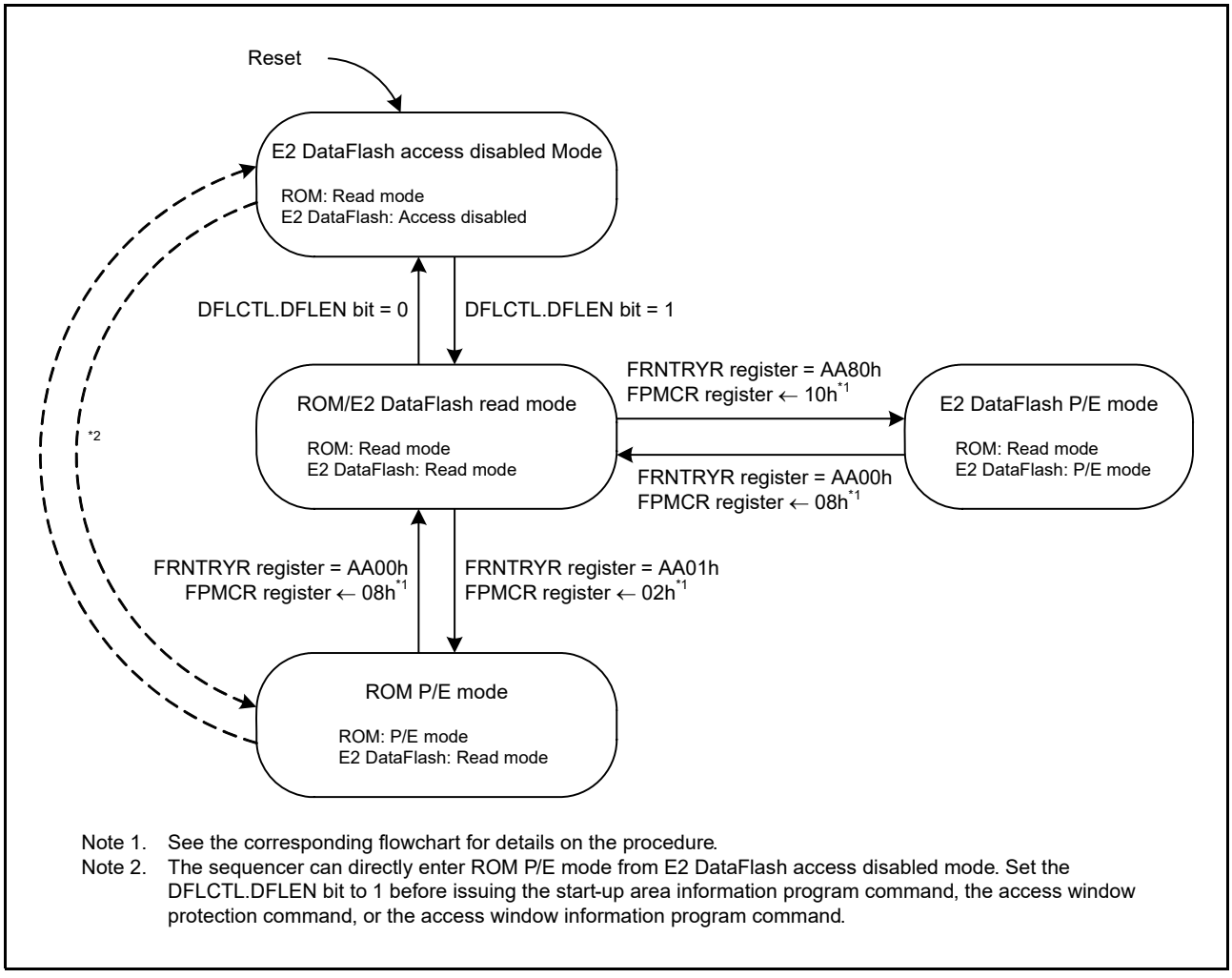
Figure 46.5 Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 512-Kbyte ROM)

46.7 Programming and Erasure

The ROM and E2 DataFlash can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.
The mode transitions and commands required to program or erase the ROM and E2 DataFlash are described below. The descriptions apply in common to boot mode and single-chip mode.

46.7.1 Sequencer Modes

The sequencer has four modes. Transitions between modes are caused by writing to the DFLCTL and FENTRYR registers and setting the FPMCR register. Figure 46.6 is a diagram of mode transitions of the flash memory.



Note 1. See the corresponding flowchart for details on the procedure.
Note 2. The sequencer can directly enter ROM P/E mode from E2 DataFlash access disabled mode. Set the DFLCTL.DFLEN bit to 1 before issuing the start-up area information program command, the access window protection command, or the access window information program command.

Figure 46.6 Mode Transitions of the Flash Memory

46.7.1.1 E2 DataFlash Access Disabled Mode

In E2 DataFlash access disabled mode, access to the E2 DataFlash is disabled. After a reset, the sequencer enters this mode.
When setting the DFLCTL.DFLEN bit to 1, the E2 DataFlash is placed in read mode.

46.7.1.2 Read Mode

Read mode is for high-speed reading of the ROM/E2 DataFlash. Reading from a ROM address for reading can be accomplished in one ICLK clock.

(1) ROM/E2 DataFlash Read Mode

In this mode, both the ROM and E2 DataFlash are in read mode. The sequencer enters this mode from P/E mode when setting the FPMCR register to 08h, setting the FENTRYR.FENTRYD bit to 0, and setting the FENTRYR.FENTRY0 bit to 0.

46.7.1.3 P/E Modes

The P/E mode is for programming and erasure of the ROM/E2 DataFlash.

(1) ROM P/E Mode

In this mode, the ROM is in P/E mode, and the E2 DataFlash is in read mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD to 0, setting the FENTRYR.FENTRY0 bit to 1, and setting the FPMCR register 02h.

(2) E2 DataFlash P/E Mode

In this mode, the ROM is in read mode, and the E2 DataFlash is in P/E mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 1, setting the FENTRYR.FENTRY0 bit to 0, and setting the FPMCR register 10h.

46.7.2 Mode Transitions

46.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode

Reading of the E2 DataFlash requires switching from E2 DataFlash access disabled mode to ROM/E2 DataFlash read mode.

Set the DFLCTL.DFLEN bit to 1 to switch to ROM/E2 DataFlash read mode.

Figure 46.7 shows the procedure for transition from E2 DataFlash access disabled mode to ROM/E2 DataFlash read mode.

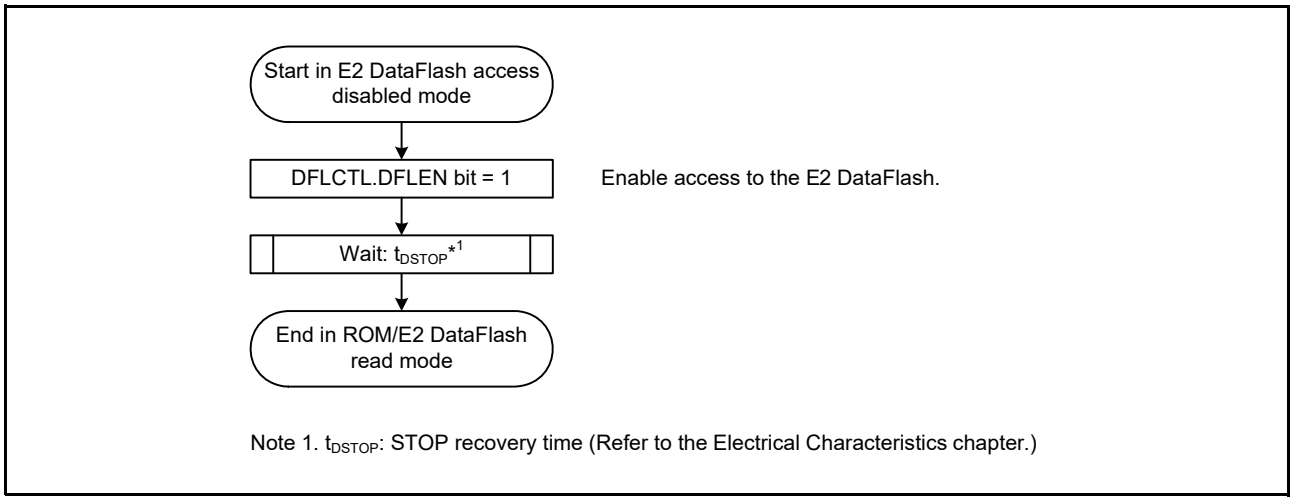


Figure 46.7 Procedure for Transition from E2 DataFlash Access Disabled Mode to ROM/E2 DataFlash Read Mode

46.7.2.2 Transition from Read Mode to P/E Mode

Switching to ROM P/E mode is required before executing a software command for the ROM.
Figure 46.8 shows the procedure for transition from ROM/E2 DataFlash read mode to ROM P/E mode. Figure 46.9 shows the procedure for transition from ROM/E2 DataFlash read mode to E2 DataFlash P/E mode.

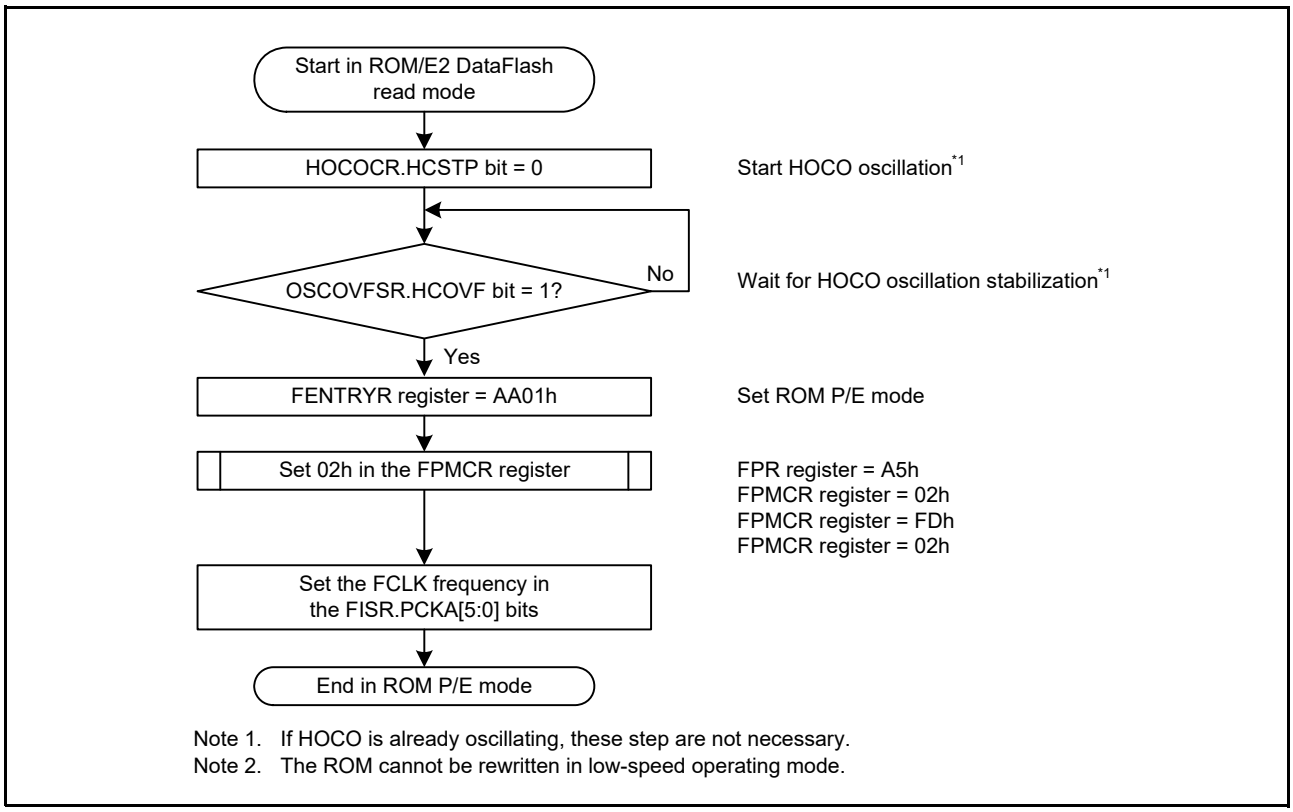
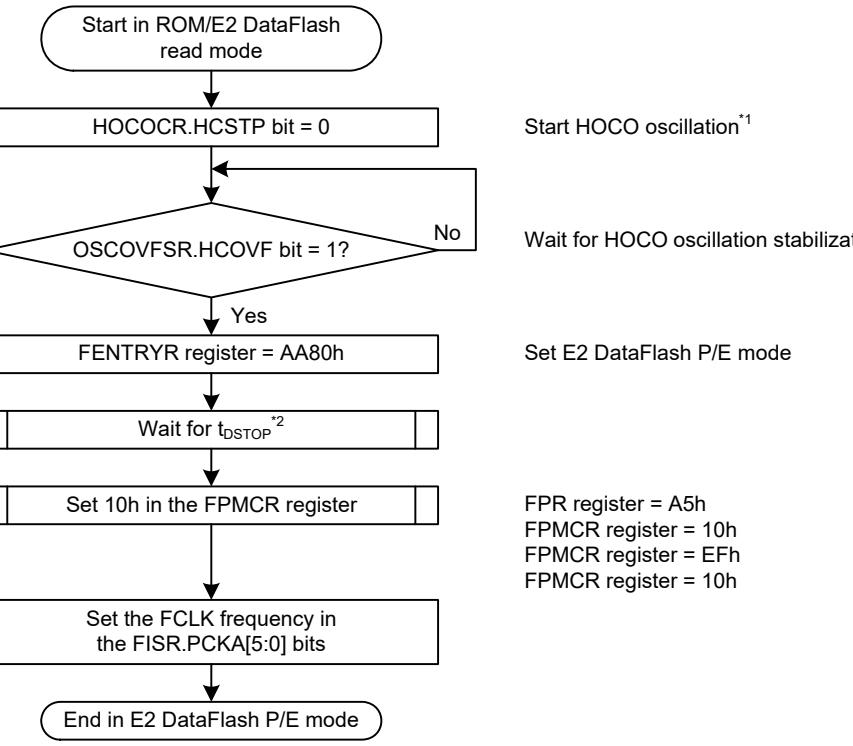


Figure 46.8 Procedure for Transition from ROM/E2 DataFlash Read Mode to ROM P/E Mode

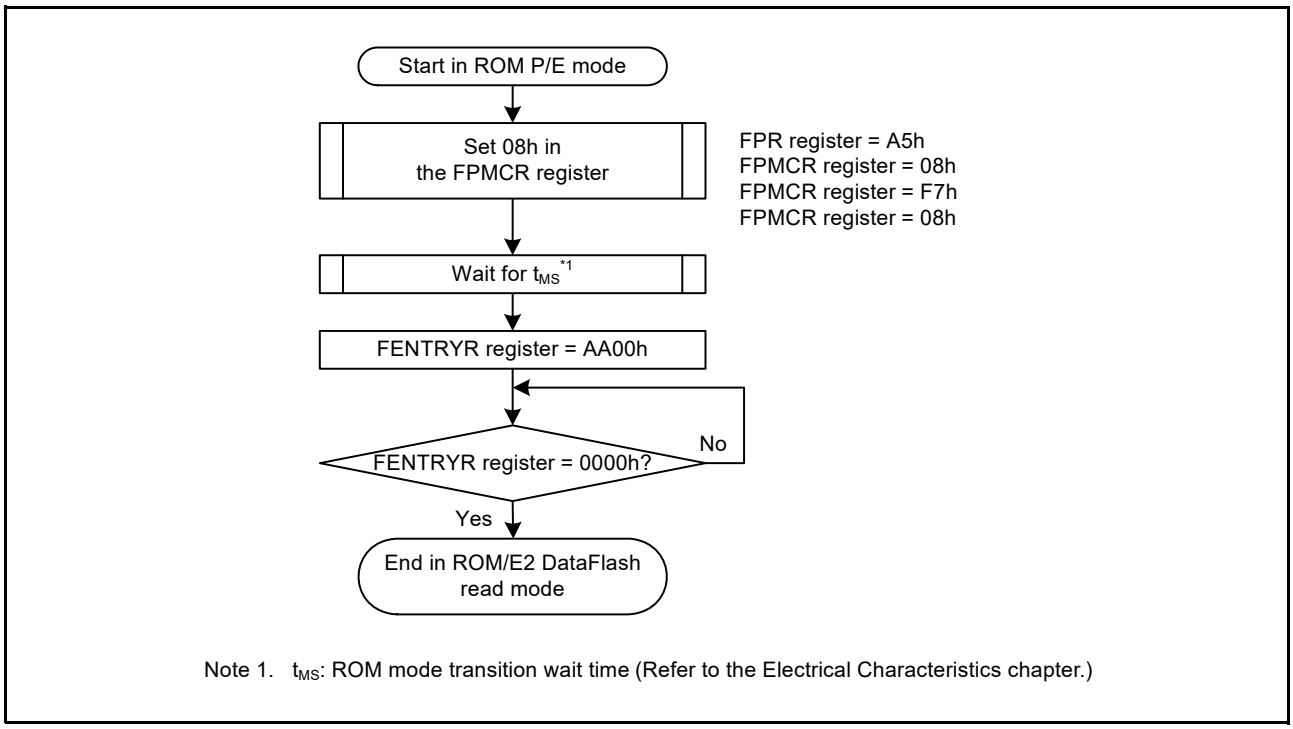


Note 1. If HOCO is already oscillating, these step are not necessary.
Note 2. t_{DSTOP}: STOP recovery time (Refer to the Electrical Characteristics chapter.)
Note 3. The ROM cannot be rewritten in low-speed operating mode.

Figure 46.9 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode

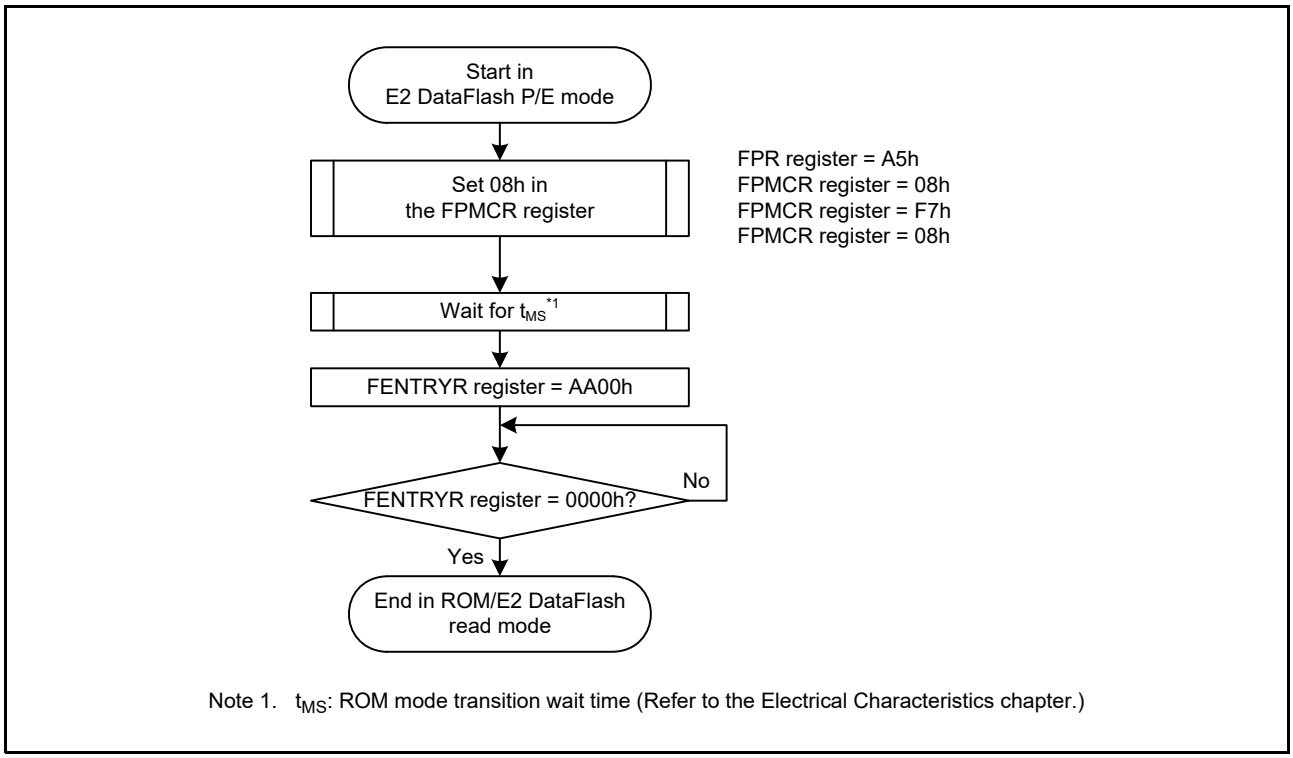
46.7.2.3 Transition from P/E Mode to Read Mode

High-speed reading of the ROM requires switching to ROM/E2 DataFlash read mode.
Figure 46.10 shows the procedure for transition from ROM P/E mode to ROM/E2 DataFlash read mode. Figure 46.11 shows the procedure for transition from E2 DataFlash P/E mode to ROM/E2 DataFlash read mode.



Note 1. t_{MS}: ROM mode transition wait time (Refer to the Electrical Characteristics chapter.)

Figure 46.10 Procedure for Transition from ROM P/E Mode to ROM/E2 DataFlash Read Mode



Note 1. t_{MS}: ROM mode transition wait time (Refer to the Electrical Characteristics chapter.)

Figure 46.11 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode

46.7.3 Software Commands

Software commands consist of commands for programming and erasure and commands for programming start-up program area information and access window information. Table 46.6 lists the software commands for use with the flash memory.

Table 46.6 Software Commands

Command	Function
Program	• ROM programming (8 bytes) • E2 DataFlash programming (1 byte)
Block erase	ROM/E2 DataFlash erasure
All-block erase	Erasure of all blocks in the ROM/E2 DataFlash
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Start-up area information program	Rewrite the start-up area switching information used for start-up program protection.
Access window protection	Protect the settings for the access window used for area protection from being rewritten.
Access window information program	Set the access window used for area protection.

46.7.4 Software Command Usage

This section describes how to use each software command, using flowcharts.

46.7.4.1 Program

Figure 46.12 and Figure 46.13 show the procedure to issue the program command.

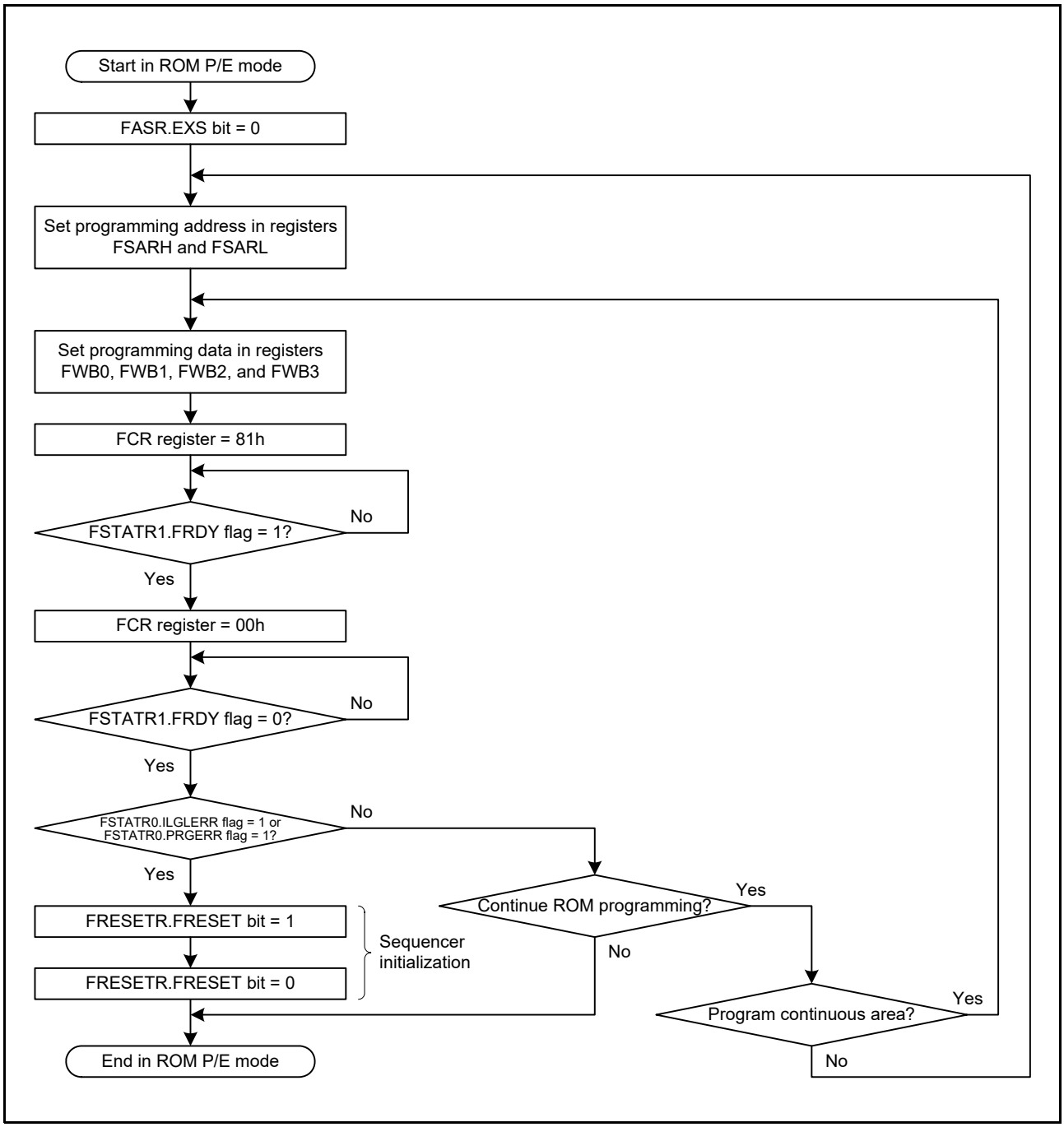


Figure 46.12 Procedure to Issue the Program Command for the ROM

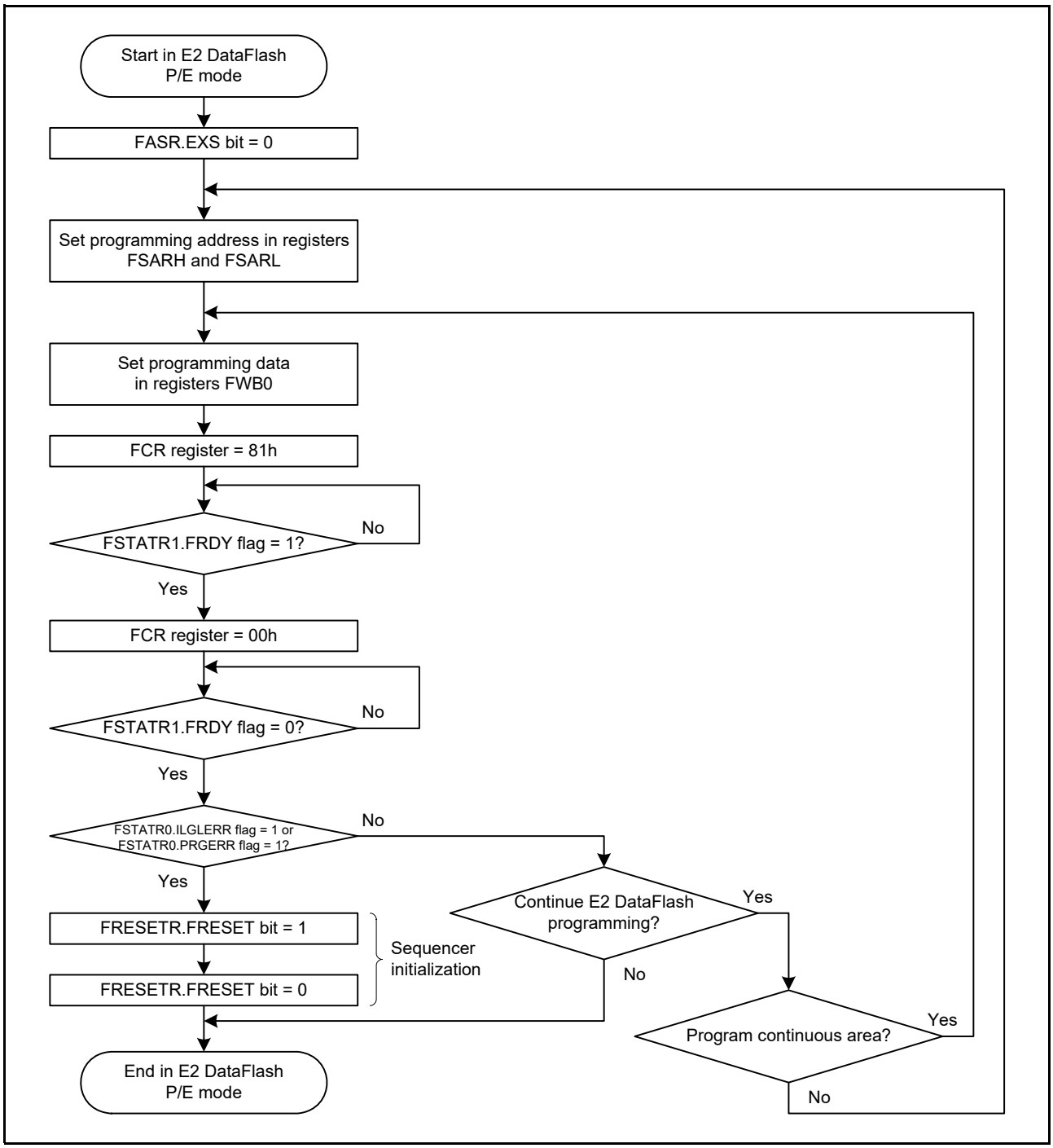


Figure 46.13 Procedure to Issue the Program Command for the E2 DataFlash

46.7.4.2 Block Erase

Figure 46.14 and Figure 46.15 show the procedure to issue the block erase command.

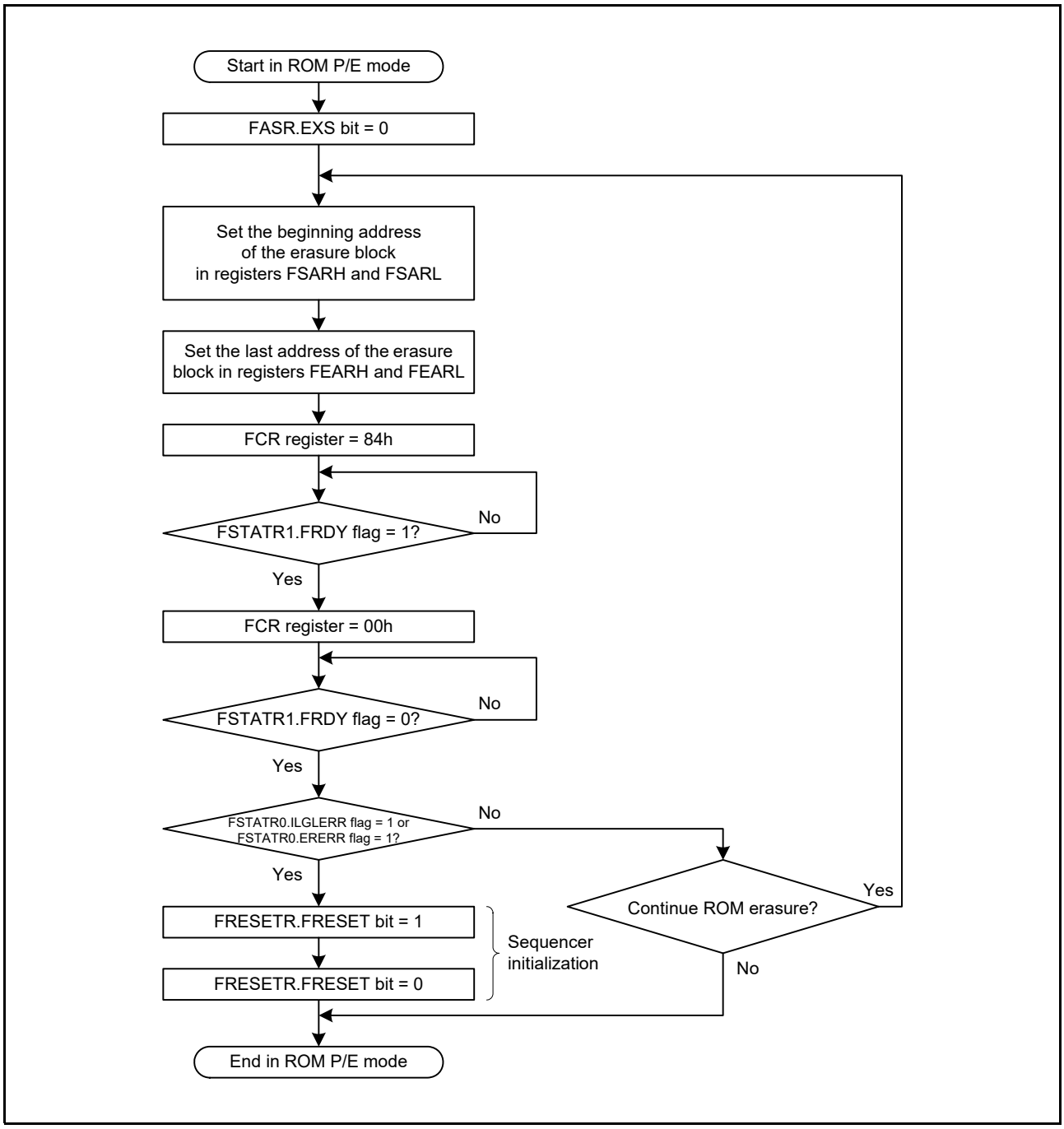


Figure 46.14 Procedure to Issue the Block Erase Command for the ROM

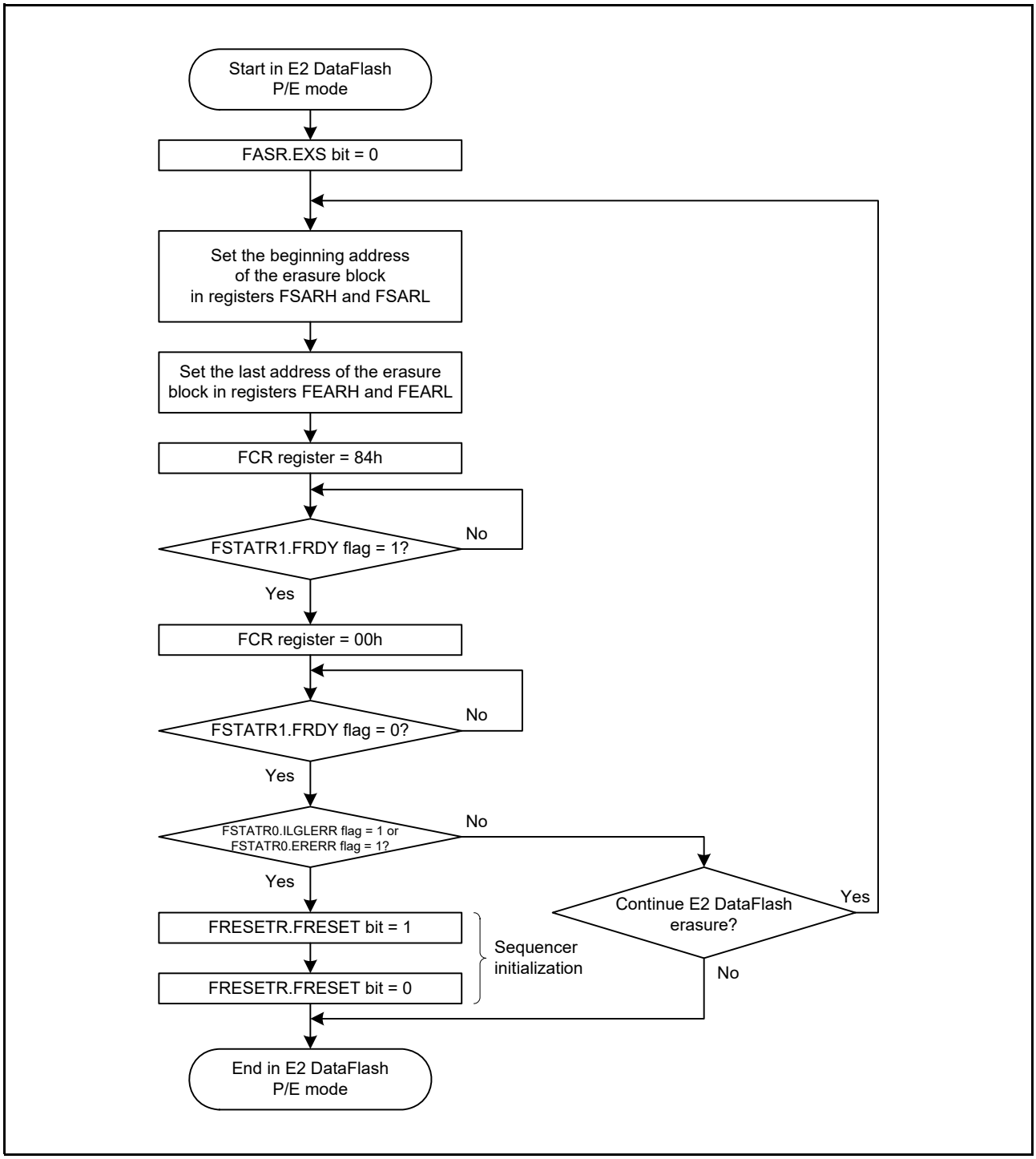


Figure 46.15 Procedure to Issue the Block Erase Command for the E2 DataFlash

46.7.4.3 All-Block Erase

Figure 46.16 and Figure 46.17 show the procedure to issue the all-block erase command.

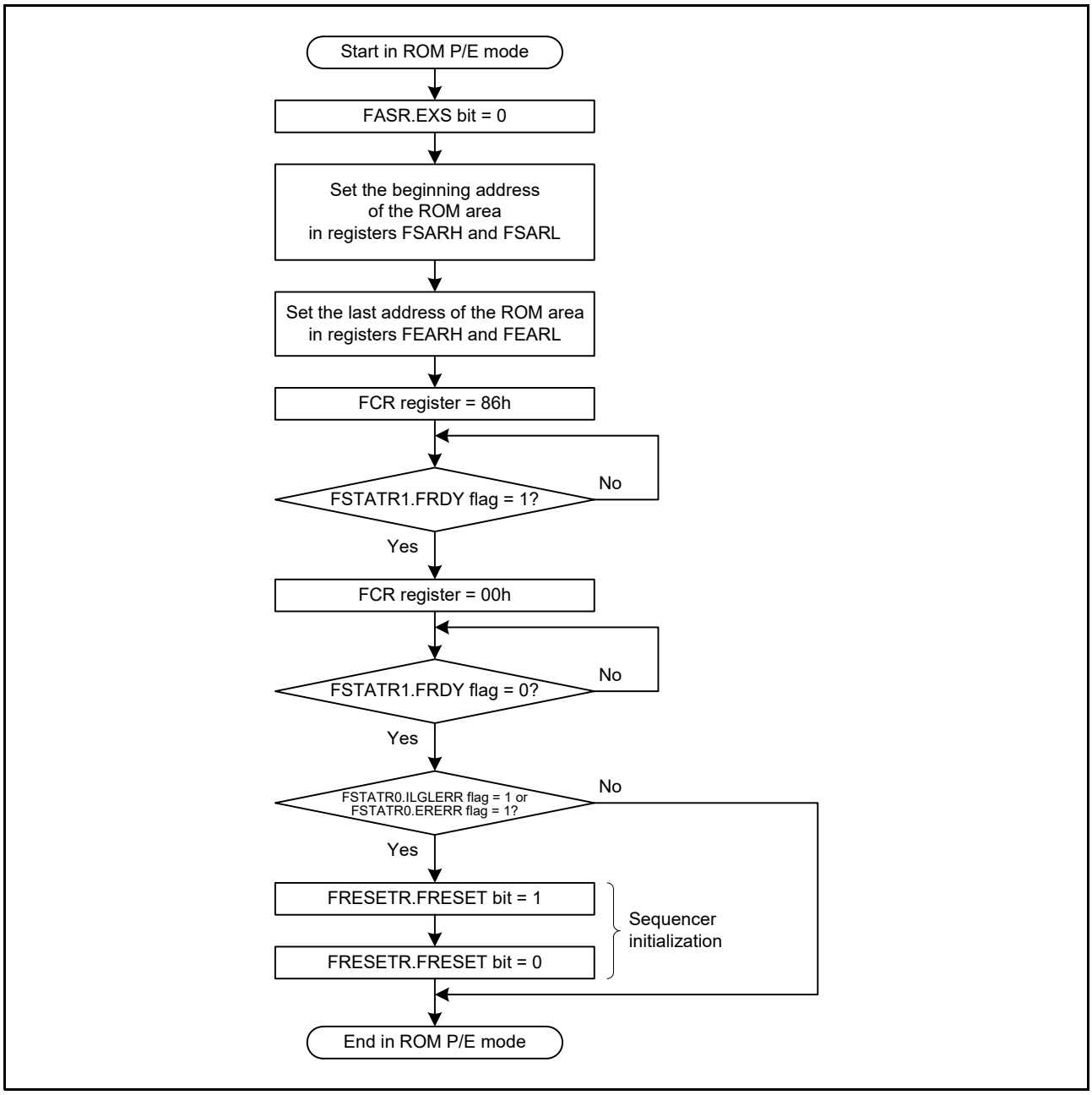


Figure 46.16 Procedure to Issue the All-Block Erase Command for the ROM

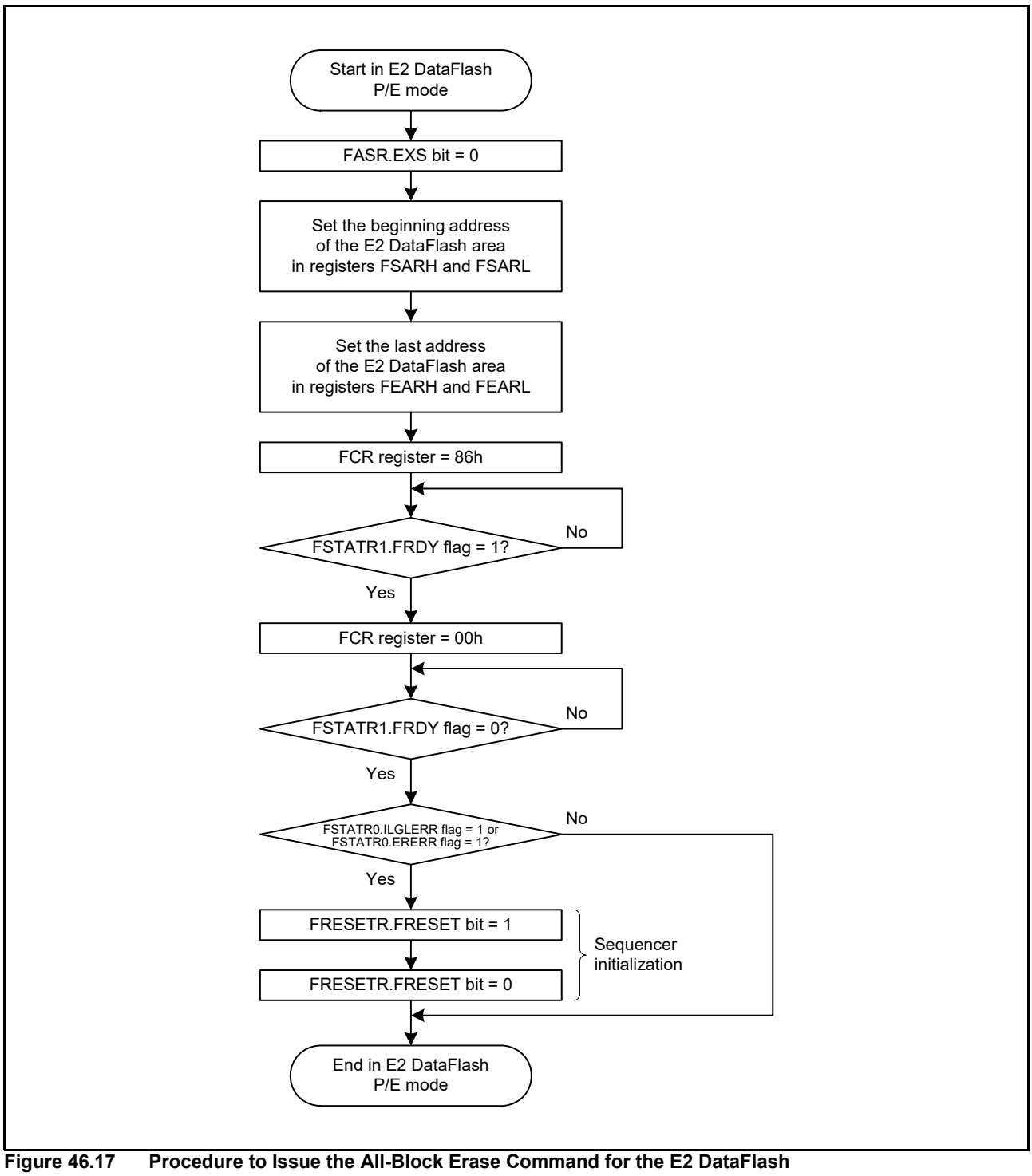


Figure 46.17 Procedure to Issue the All-Block Erase Command for the E2 DataFlash

46.7.4.4 Blank Check

Figure 46.18 and Figure 46.19 show the procedure to issue the blank check command.

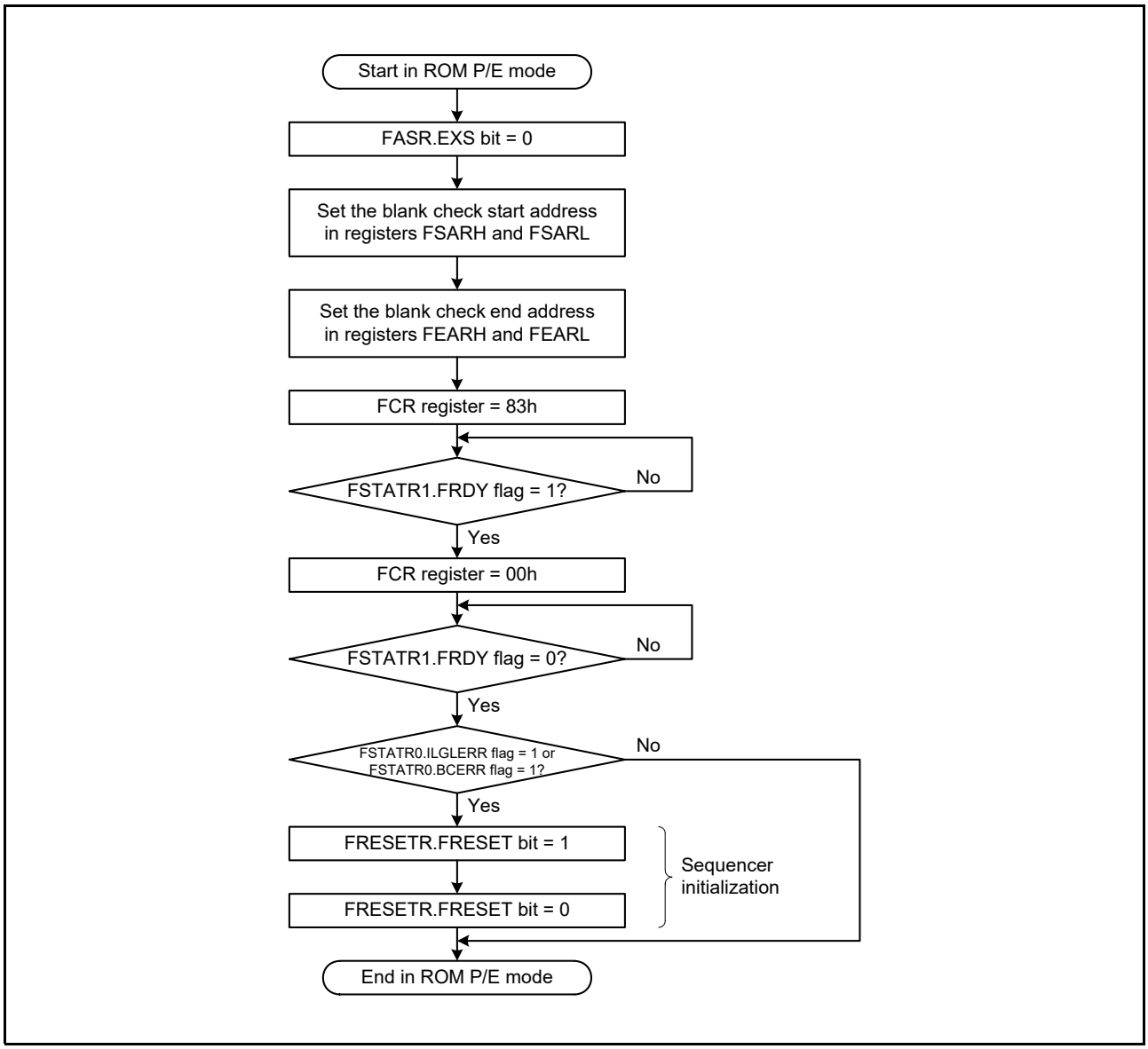


Figure 46.18 Procedure to Issue the Blank Check Command for the ROM

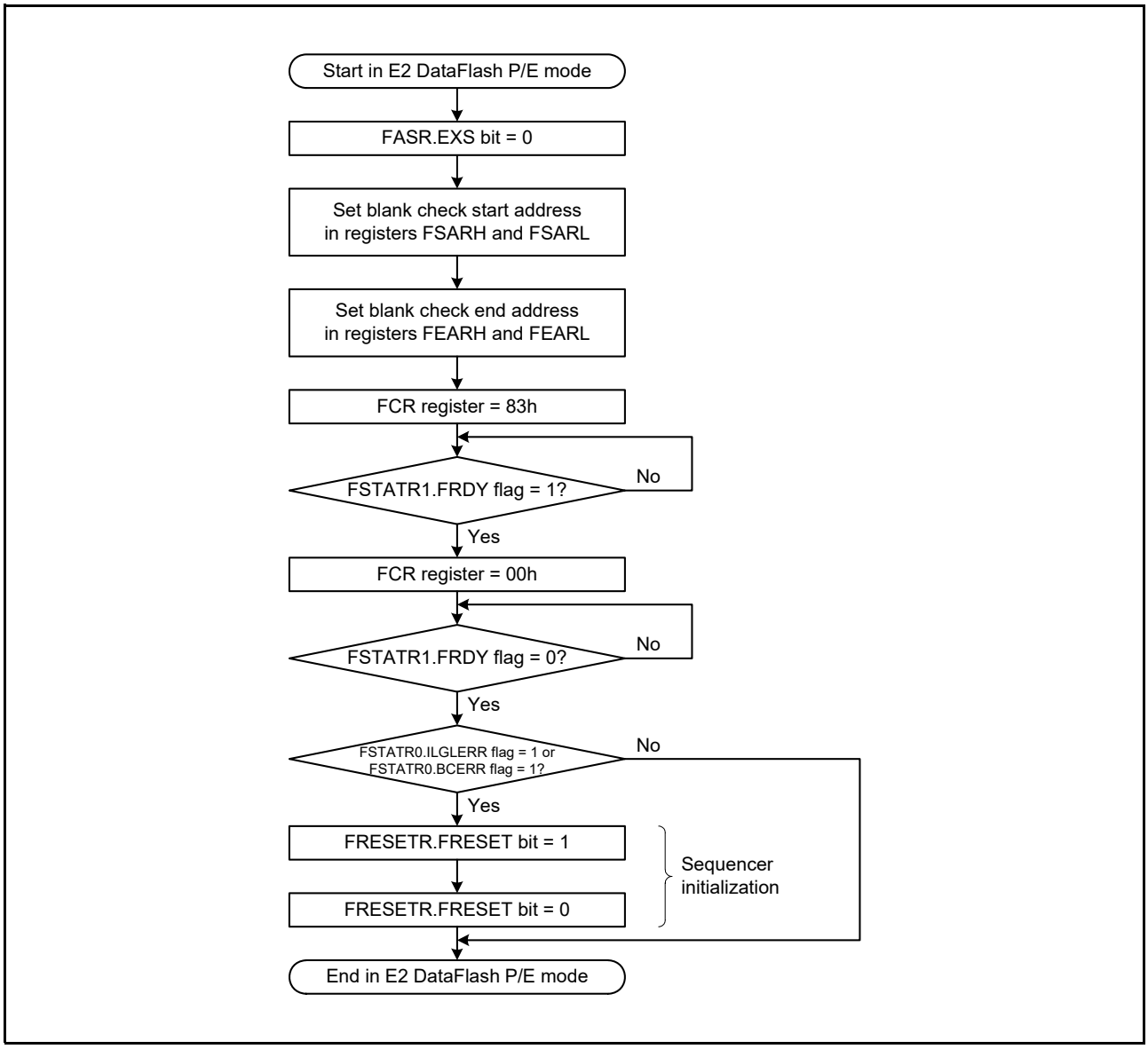


Figure 46.19 Procedure to Issue the Blank Check Command for the E2 DataFlash

46.7.4.5 Start-Up Area Information Program/Access Window Protection/Access Window Information Program

Figure 46.20 shows the procedure to issue the start-up area information program command, access window protection command, and access window information program command.
When the sequencer has directly entered ROM/PE mode from E2 DataFlash access disabled mode, set the DFLCTL.DFLEN bit to 1 at the beginning of the procedure.

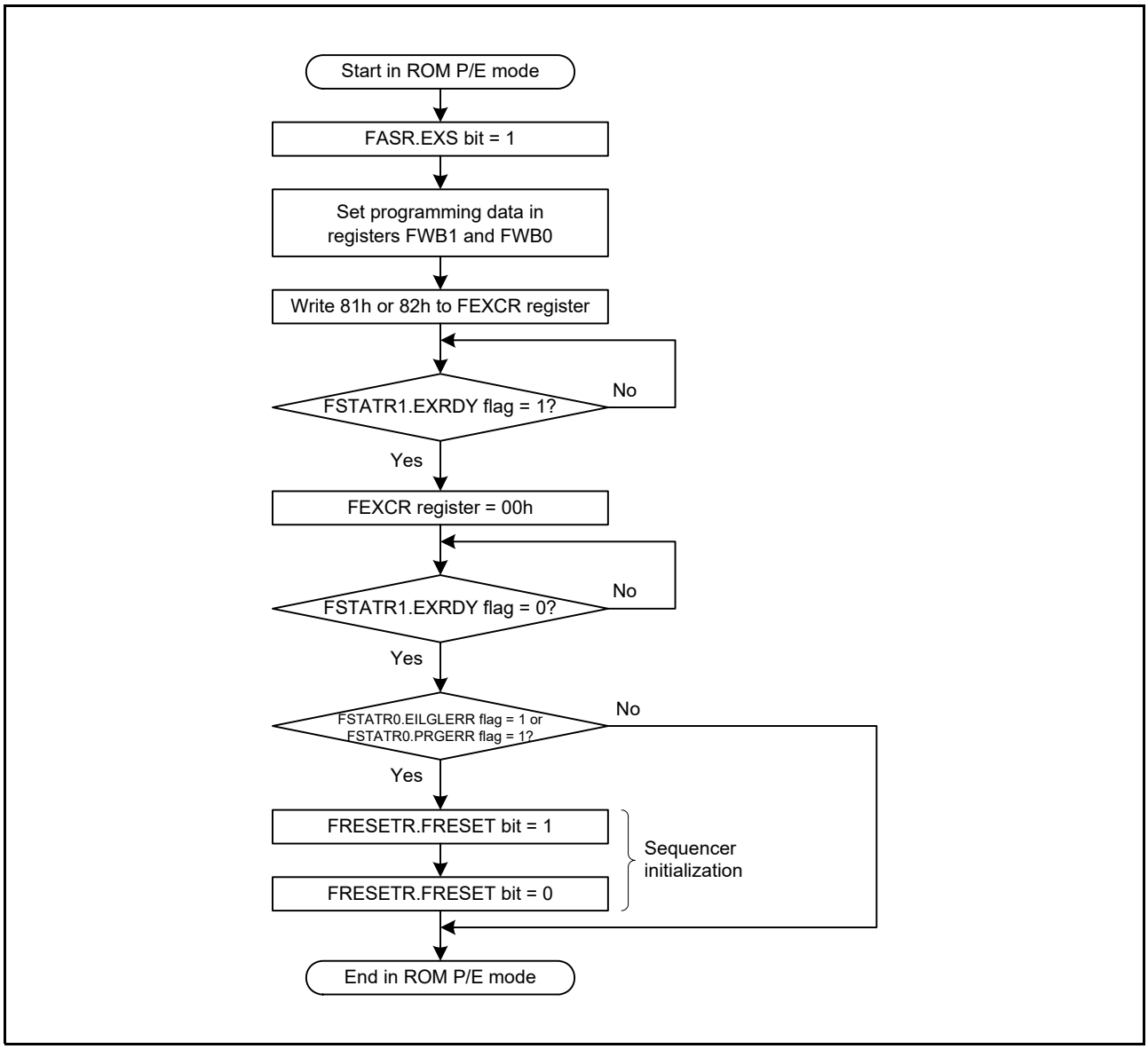


Figure 46.20 Procedure to Issue the Start-Up Area Information Program Command/Access Window Protection Command/Access Window Information Program Command

46.7.4.6 Forced Stop of Software Commands

Perform the procedure shown in Figure 46.21 to forcibly stop the blank check command or block erase command. When the command processing is forcibly stopped, registers FEAMH and FEAML store the address at the time of the forced stop. For blank check, the stopped processing can be continued by copying the FEAMH and FEAML register values to registers FSARH and FSARL.

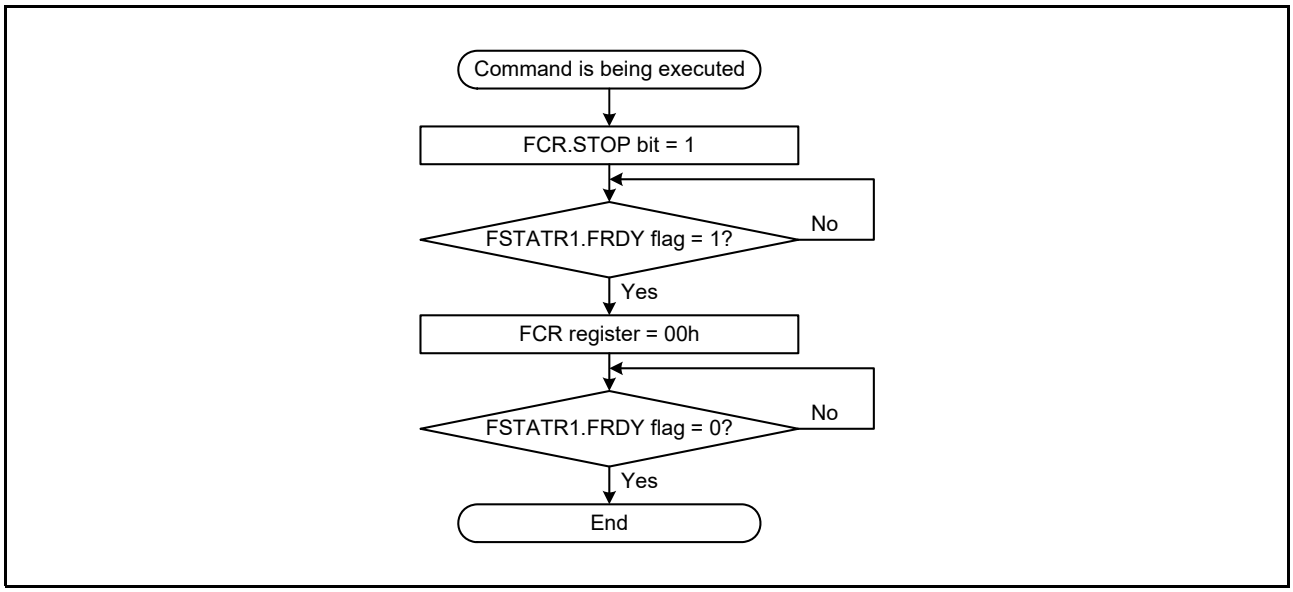


Figure 46.21 Procedure for Forced Stop of Software Commands

46.7.5 Interrupt

When software command processing or forced stop processing is completed, an interrupt (FRDYI) is generated.
When the FSTATR1.FRDY flag becomes 0 by setting the FCR.OPST bit to 0 and the FSTATR1.EXRDY flag becomes 0 by setting the FEXCR.OPST bit to 0, the next interrupt (FRDYI) can be accepted.
Clear the IRn.IR flag before setting the IERm.IENj bit of the ICU corresponding to this interrupt.

46.8 Boot Mode

The USB interface, SCI, or FINE interface is used in boot mode.
Table 46.7 lists the programmable and erasable areas and peripheral modules used in boot mode. Table 46.8 lists the I/O pins used in boot mode.

Table 46.7 Programmable and Erasable Areas and Peripheral Modules Used in Boot Mode

	Boot Mode		
Item	**USB Interface**	**SCI Interface**	**FINE Interface**
Programmable and erasable areas	User area Data area	User area Data area	User area Data area
Peripheral module	USB0	SCI1 (asynchronous serial communication)	FINE

Table 46.8 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode	Description
PC7/UB	Input	Boot mode	Select operating mode (refer to section 3, Operating Modes).
MD	Input		Select operating mode (refer to section 3, Operating Modes).
MD/FINED	I/O	Boot mode (FINE interface)	Select operating mode/FINE data I/O
USB0_DP, USB0_DM	I/O	Boot mode (USB interface)	USB data I/O
P16/USB0_VBUS	Input		Detect USB cable connection/disconnection
P35/UPSEL	Input		Set bus-powered mode or self-powered mode
P30/RXD1	Input	Boot mode (SCI interface)	Receive data*1
P26/TXD1	Output		Transmit data*1

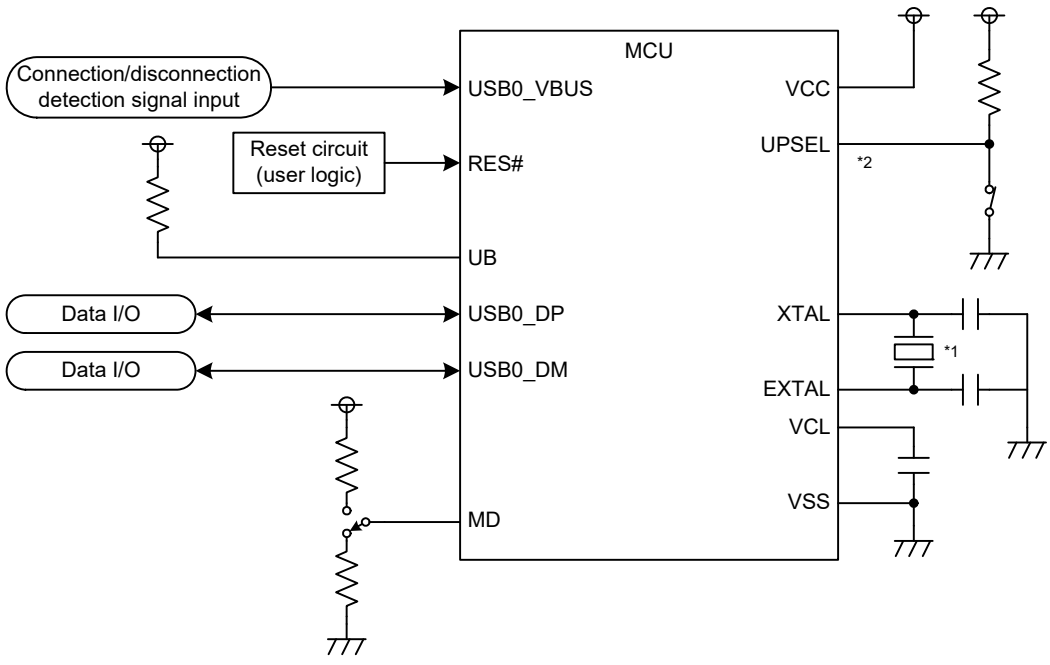
Note 1. When using the SCI, connect (pull up) this pin to VCC via a resistor.

46.8.1 Boot Mode (USB Interface)

The flash memory can be programmed and erased using the USB interface in boot mode (USB interface). The user area and data area can be rewritten.
When a reset is released while the MD pin is low and the UB pin is high, the MCU starts in boot mode (USB interface). Self-powered or bus-powered can be selected in accordance with the state of the UPSEL pin. When a reset is released while the UPSEL pin is low, self-powered mode is selected. When a reset is released while the UPSEL pin is high, bus-powered mode is selected.
Contact the manufacturer for details on the serial programmer (USB programmer).

46.8.1.1 Operating Conditions in Boot Mode (USB Interface)

USB0 is used for communication with the serial programmer in boot mode (USB interface).
4, 6, 8, 12, or 16 MHz can be used as the frequency input to the main clock oscillator. The operating voltage range is between 3.0 V and 3.6 V.
Connect the UB pin to VCC directly or VCC via a resistor (pull up).
Figure 46.22 shows an example of pin connections in boot mode (USB interface) when self-powered. Table 46.9 lists pin handling in boot mode (USB interface) when self-powered. Figure 46.23 shows an example of pin connections in boot mode (USB interface) when bus-powered. Table 46.10 lists pin handling in boot mode (USB interface) in bus-powered.
Examples of pin connections shown in Figure 46.22 and Figure 46.23 are simplified circuits. Operations are not guaranteed in all systems.



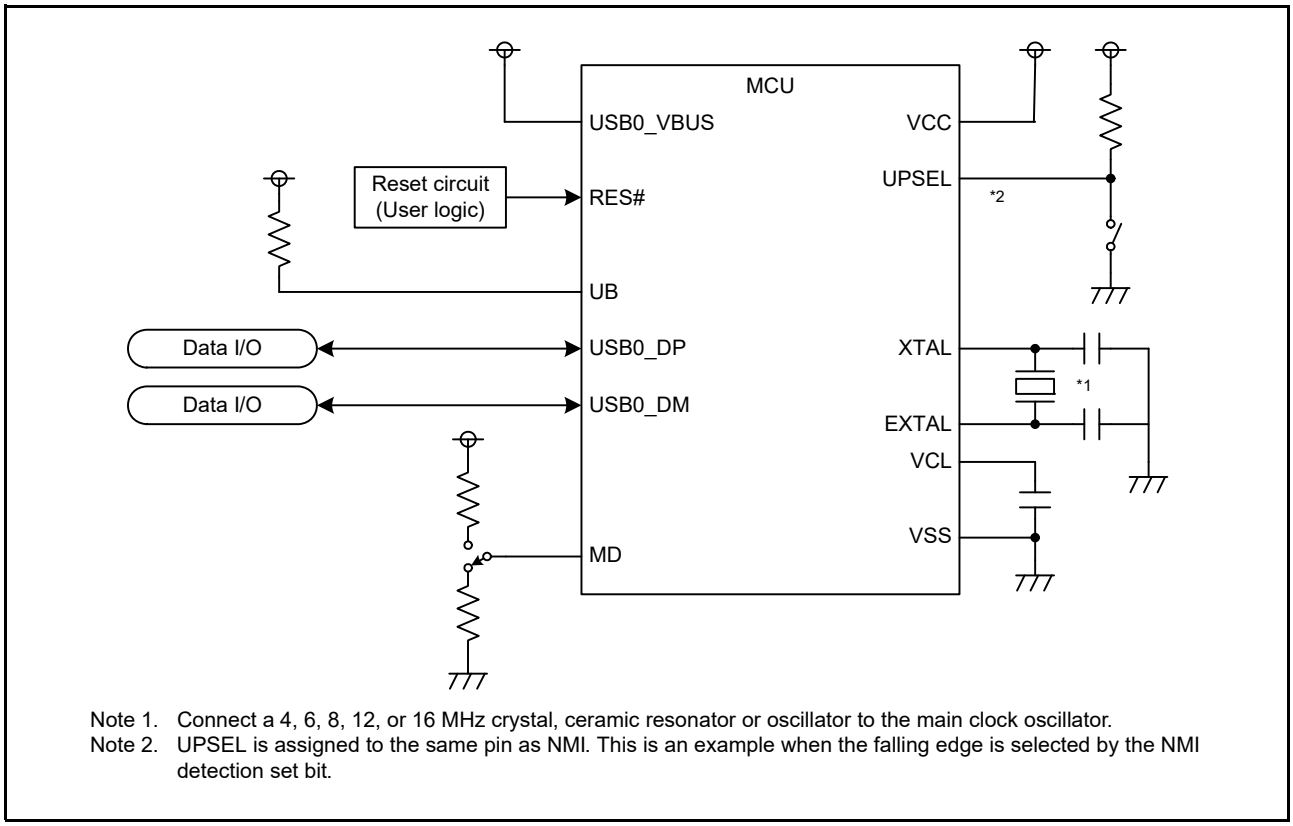
Note 1. Connect a 4, 6, 8, 12, or 16 MHz crystal, ceramic resonator or oscillator to the main clock oscillator.
Note 2. UPSEL is assigned to the same pin as NMI. This is an example when the falling edge is selected by the NMI detection set bit.

Figure 46.22 Example of Pin Connections in Boot Mode (USB Interface) When Self-Powered

Table 46.9 Pin Handling in Boot Mode (USB Interface) When Self-Powered

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input the voltage between 3.0 V and 3.6 V to the VCC pin. Input 0 V to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply	—	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Connect a 4, 6, 8, 12, or 16 MHz crystal or ceramic resonator or oscillator.
MD	Operating mode control	Input	Input low.
PC7/UB	Operating mode control	Input	Input high.*1
P35/UPSEL	USB power mode control	Input	Input low.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Connect to the circuit described in section 30, USB 2.0 FS Host/Function Module (USBe).
USB0_DM	USB on-chip transceiver D– I/O pin	I/O	Connect to the circuit described in section 30, USB 2.0 FS Host/Function Module (USBe).
P16/USB0_VBUS	USB cable connection monitor pin	Input	Connect to the circuit described in section 30, USB 2.0 FS Host/Function Module (USBe).

Note 1. Maintain the input level for 2 ms or longer after a reset is released.



Note 1. Connect a 4, 6, 8, 12, or 16 MHz crystal, ceramic resonator or oscillator to the main clock oscillator.
Note 2. UPSEL is assigned to the same pin as NMI. This is an example when the falling edge is selected by the NMI detection set bit.

Figure 46.23 Example of Pin Connections in Boot Mode (USB Interface) When Bus-Powered

Table 46.10 Pin Handling in Boot Mode (USB Interface) in Bus-Powered

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input the voltage between 3.0 V and 3.6 V to the VCC pin. Input 0 V to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply	—	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Connect a 4, 6, 8, 12, or 16 MHz crystal or ceramic resonator or oscillator.
MD	Operating mode control	Input	Input low.
PC7/UB	Operating mode control	Input	Input high.*1
P35/UPSEL	USB power mode control	Input	Input high.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Connect to the circuit described in section 30, USB 2.0 FS Host/Function Module (USBe).
USB0_DM	USB on-chip transceiver D– I/O pin	I/O	Connect to the circuit described in section 30, USB 2.0 FS Host/Function Module (USBe).
P16/USB0_VBUS	USB cable connection monitor pin	Input	Connect to the USB0_VBUS pin to the VCC pin.

Note 1. Maintain the input level for 2 ms or longer after a reset is released.

46.8.2 Boot Mode (SCI Interface)

The flash memory can be programmed and erased using asynchronous serial communication in boot mode (SCI interface). The user area and data area can be rewritten.
When a reset is released while the MD pin and the UB pin are low, the MCU starts in boot mode (SCI interface).
Contact the manufacturer for details on the serial programmer.

46.8.2.1 Operating Conditions in Boot Mode (SCI Interface)

SCI1 is used to communicate with the serial programmer in boot mode (SCI interface).
Figure 46.24 shows an example of pin connections in boot mode (SCI interface). Table 46.11 lists pin handling in boot mode (SCI interface).
An example of pin connections shown in Figure 46.24 is a simplified circuit. Operations are not guaranteed in all systems.

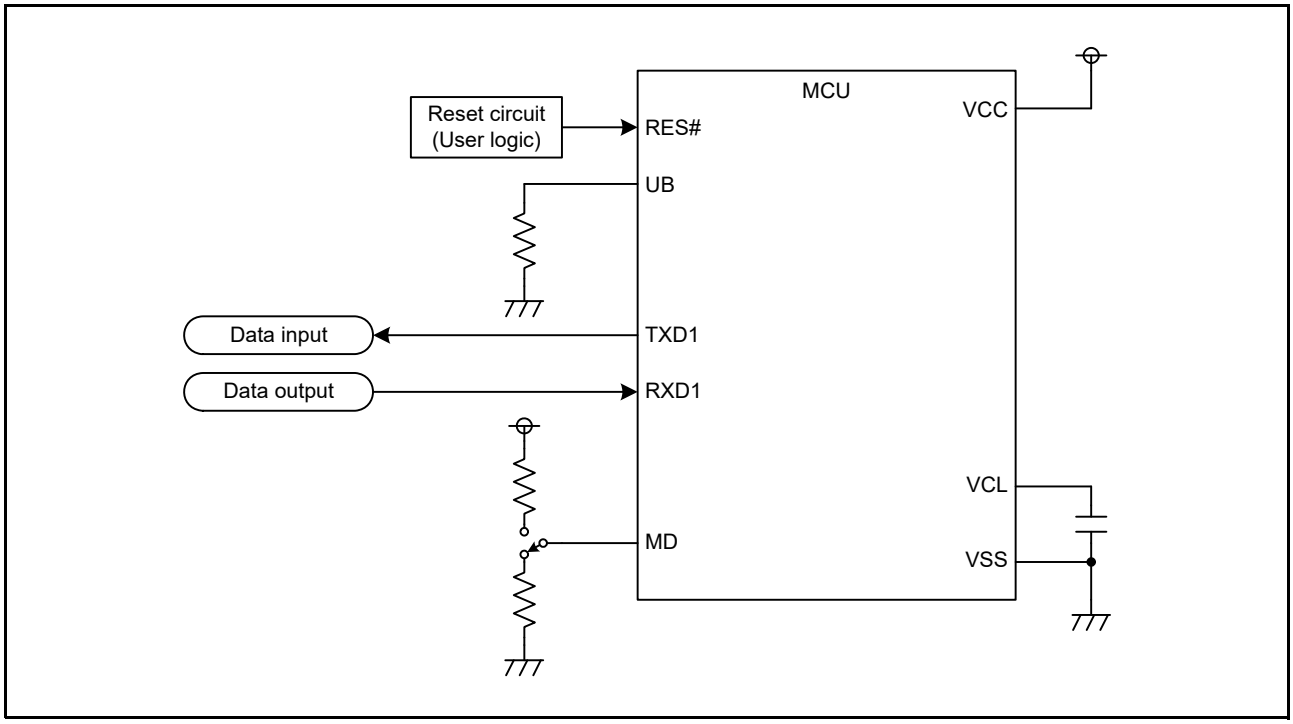


Figure 46.24 Example of Pin Connections in Boot Mode (SCI Interface)

Table 46.11 Pin Handling in Boot Mode (SCI Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input 1.6 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 decoupling capacitor for stabilizing the internal voltage.
MD	Operating mode control	Input	Input low.
PC7/UB	Operating mode control	Input	Input low.*1
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
P30/RXD1	Data input RXD	Input	Input pin for serial data
P26/TXD1	Data output TXD	Output	Output pin for serial data

Note 1. Maintain the input level for 2 ms or longer after a reset is released.

As shown in Figure 46.25, set the format to 8-bit data, 1 stop bit, no parity, and LSB first to communicate with the serial programmer.

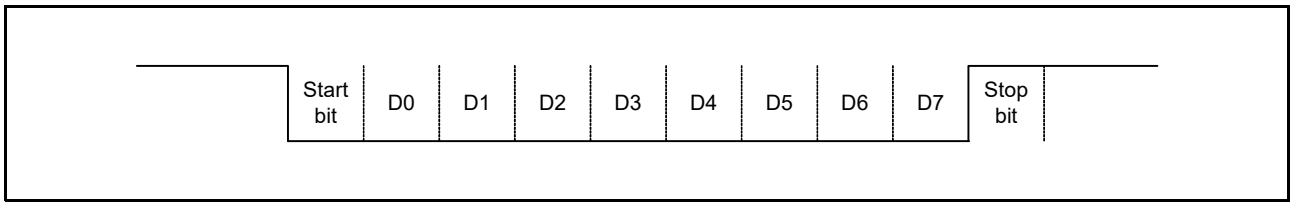


Figure 46.25 Communication Format

Initial communication with the programmer is performed at 9,600 or 19,200 bps. The communication bit rate can be changed after the MCU is connected with the programmer.
The maximum communication bit rate for communication in boot mode (SCI interface) is 2 Mbps.

46.8.2.2 Starting Up in Boot Mode (SCI Interface)

To start the MCU in boot mode (SCI interface), a reset must be released by changing the RES# pin from low to high while the MD pin and UB pin are low. After starting up in boot mode (SCI interface), wait at least 400 ms until communication with the MCU is enabled in boot mode (SCI interface).
As shown in Figure 46.26, keep the signal of each pin unchanged for 400 ms after the reset is released. Use resets according to the range described in section 47.5.2, Reset Timing.

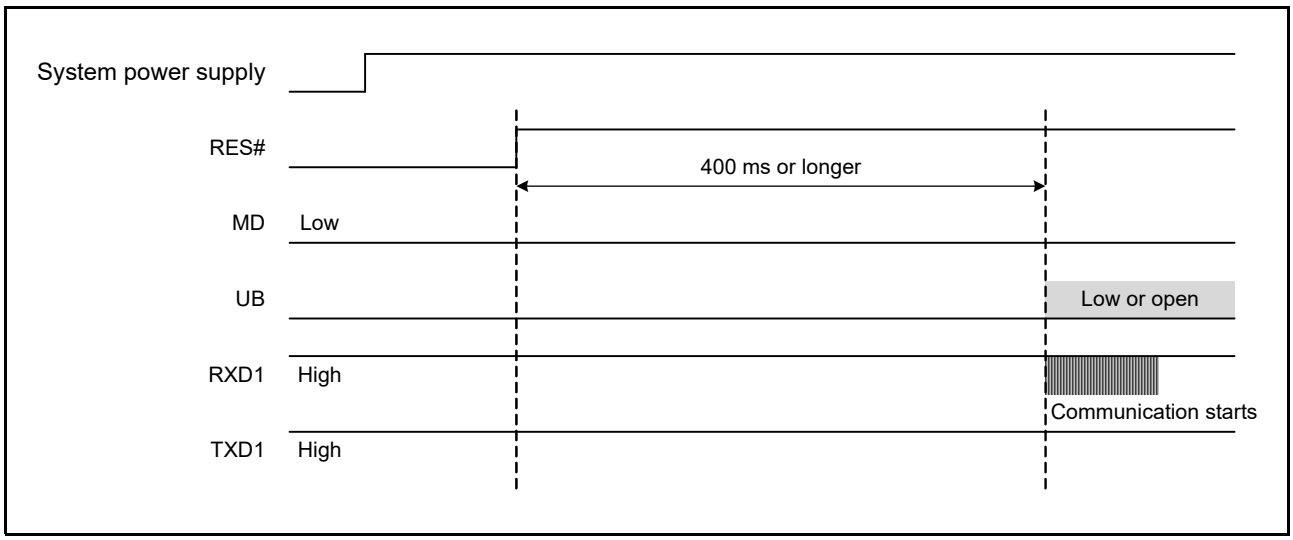


Figure 46.26 Wait Time until Communication Becomes Possible in Boot Mode (SCI Interface)

46.8.3 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The user area and data area can be rewritten.
Contact the manufacturer for details on the serial programmer.

46.8.3.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE interface).
Figure 46.27 shows an example of pin connections in boot mode (FINE interface). Table 46.12 lists pin handling in boot mode (FINE interface).
An example of pin connections shown in Figure 46.27 is a simplified circuit. Operations are not guaranteed in all systems.

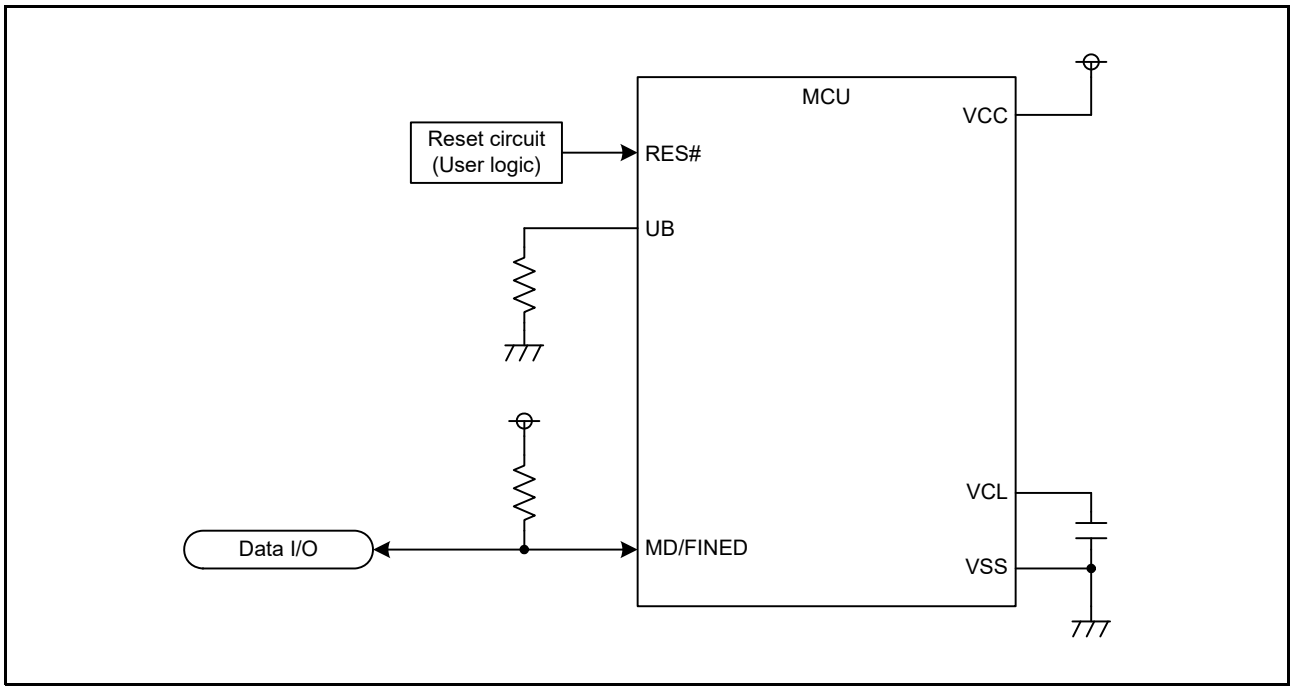


Figure 46.27 Example of Pin Connections in Boot Mode (FINE Interface)

Table 46.12 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input 1.6 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 decoupling capacitor for stabilizing the internal voltage.
MD/FINED	Operating mode control/data I/O	I/O	Connect to the VCC pin via a resistor (pull up).
PC7/UB	Operating mode control	Input	Input low.*1
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

Note 1. Maintain the input level for 2 ms or longer after a reset is released.

46.9 Flash Memory Protection

Flash memory protection prevents the flash memory from being read or rewritten by the third party.
The boot mode ID code protection is for connecting the serial programmer, and the on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator.

46.9.1 ID Code Protection

There are two types of ID code protection: Boot mode ID code protection for connecting the serial programmer and on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator. The same ID codes are used for both functions, but operations differ.
ID codes consist of the control code and ID code 1 to ID code 15. Set ID codes to four 32-bit data in 32-bit units. Figure 46.28 shows the ID code configuration.

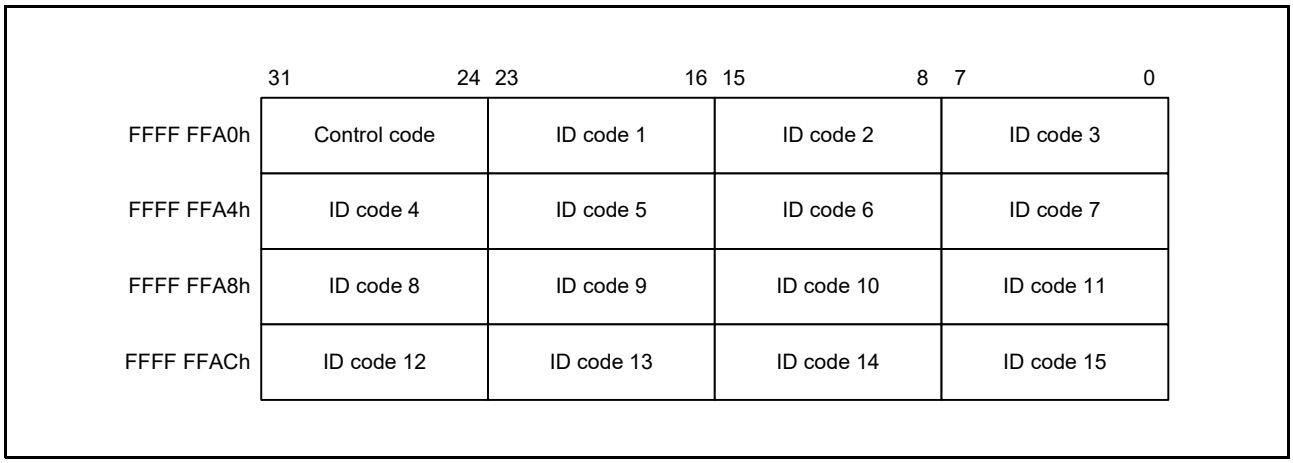


Figure 46.28 ID Code Configuration

The following shows a program example for setting ID codes.
This is an example when setting the control code to 45h and setting ID codes to 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, and 0Fh (from the ID code 1 field to the ID code 15 field).

C language:

```
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE [4] = {0x45010203, 0x04050607,0x08090A0B, 0x0C0D0E0F};
```

Assembly language:

```
.SECTION ID_CODE,CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

46.9.1.1 Boot Mode ID Code Protection

Boot mode ID code protection disables reading and programming of the user area and data area when the serial programmer is connected by the third party.
When the control code indicates 45h or 52h (boot mode ID code protection is enabled), the MCU compares 16-byte ID code sent from the serial programmer with the ID code in the user area. According to the comparison result, reading and programming the user area and data area are enabled.
When the control code indicates a value other than 45h and 52h (boot mode ID code protection is disabled), all blocks in the user area and data area are erased, and reading and programming the user area and data area are enabled.

The control code is used to enable or disable protection. Table 46.13 lists the specifications of boot mode ID code protection, and Figure 46.29 shows the authentication flow of boot mode ID code protection.
ID code 1 to ID code 15 can be set to any desired value.
However, only when disabling connection with the serial programmer, the ID codes must be set to 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, and FFh (from the ID code 1 field to the ID code 15 field).

Table 46.13 Boot Mode ID Code Protection Specifications

ID Code				
Control Code	**ID Code 1 to ID Code 15**	**Protection**	**ID Code Matching Result**	**Operation**
45h	Any desired value	Enabled	Matched	Exit the boot mode ID code authentication state and enter the program/erase host command wait state.
			Not matched	Continue the boot mode ID code authentication state.
			Not matched three times consecutively	Erase all blocks in the user area and data area, and continue boot mode ID code authentication state.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh (8 bytes are all FFh)	Enabled	N/A	Disable reading or rewriting of the flash memory, regardless of the codes sent from the serial programmer.
	Other than above		Matched	Exit the boot mode ID code authentication state and enter the program/erase state.
			Not matched	Continue the boot mode ID code authentication state.
Other than above	Any desired value	Disabled	N/A	Erase all blocks in the user area and data area.

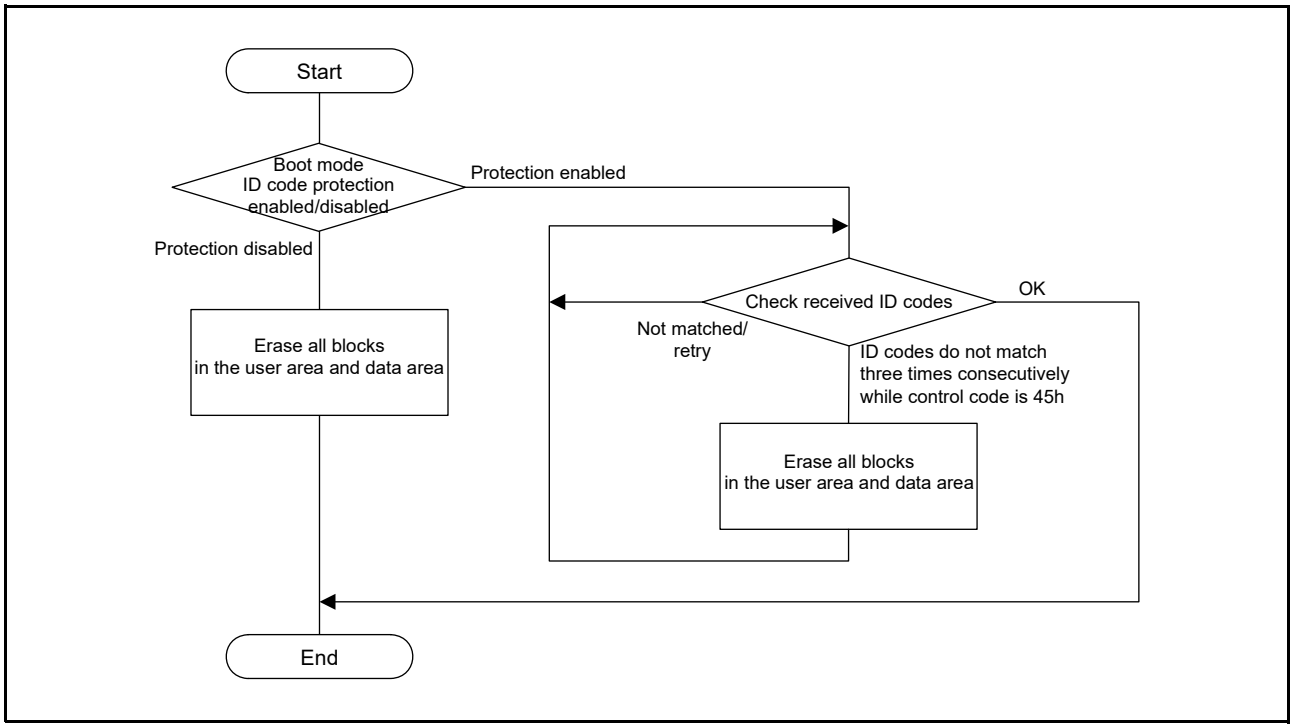


Figure 46.29 Authentication for Boot Mode ID Code Protection

46.9.1.2 On-Chip Debugging Emulator ID Code Protection

On-chip debugging emulator ID code protection enables or disables connection with the on-chip debugging emulator. When the on-chip debugging emulator ID code protection is disabled, connection with the on-chip debugging emulator is enabled. When 16-byte ID codes sent from the on-chip debugging emulator and ID codes in the user area match while on-chip debugging emulator ID code protection is enabled, connection with the on-chip debugging emulator is also enabled.

Table 46.14 lists the specifications of on-chip debugging emulator ID code protection.

Table 46.14 On-Chip Debugging Emulator ID Code Protection Specifications

ID Code				
Control Code	ID Code 1 to ID Code 15	Protection	ID Code Matching Result	Operation
FFh	FFh,…, and FFh (15 bytes are all FFh)	Disabled	N/A	Enable connection with the on-chip debugging emulator.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, and 74h + any 8 bytes	Enabled	N/A	Disable connection with the on-chip debugging emulator, regardless of the codes sent from the on-chip debugging emulator.
Other than above	Other than above	Enabled	Matched	Enable connection with the on-chip debugging emulator.
			Not matched	Continue the ID code wait state.

46.10 Communication Protocol

This section describes the protocol used in boot mode. When developing a serial programmer, control with this communication protocol.

46.10.1 State Transition in Boot Mode (SCI Interface)

Figure 46.30 shows the state transition in boot mode (SCI interface).

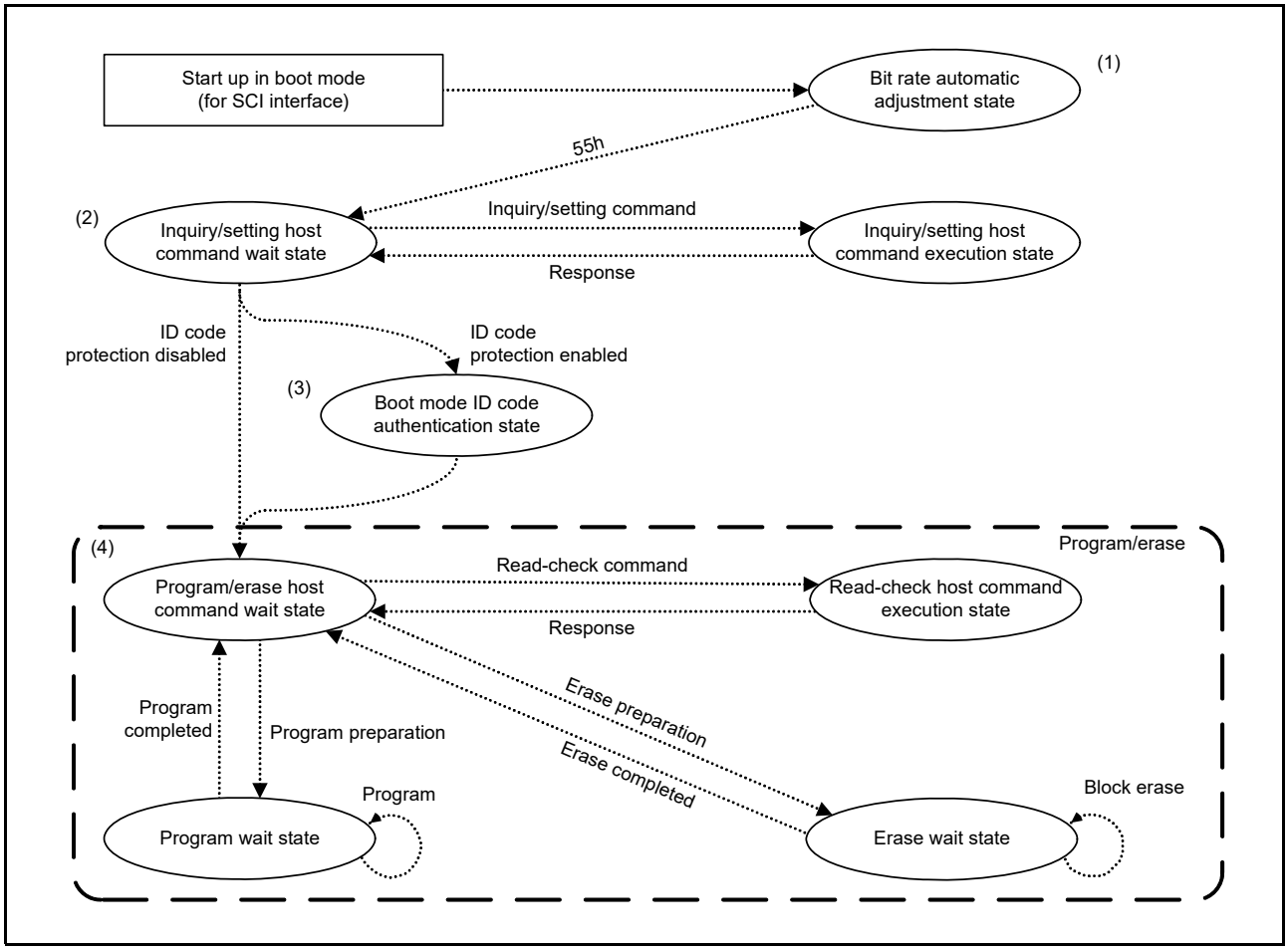


Figure 46.30 Boot Mode (SCI Interface) State Transition

(1) Bit rate automatic adjustment state

In this state, the bit rate is automatically adjusted to 9,600 or 19,200 bps for communication with the host.

When the bit rate adjustment is completed, the MCU sends 00h to the host. After that, when the MCU receives 55h sent from the host, the MCU sends E6h to the host, and enters the inquiry/setting host command wait state.

The host must not send data until 400 ms elapse after a reset of the MCU is released.

(2) Inquiry/setting host command wait state

In this state, the host can make inquiries for the MCU information including block configuration, size, and addresses where the user area and data area are allocated, and select the endian of data and a bit rate.

When the MCU receives the program/erase host state transition command from the host, it determines whether boot mode ID code protection is enabled or disabled. If boot mode ID code protection is disabled, the MCU enters the inquiry/setting host command wait state. If boot mode ID code protection is enabled, the MCU enters the boot mode ID code authentication state.

Refer to section 46.10.5, Inquiry Commands and section 46.10.6, Setting Commands for details on inquiry/setting commands.

(3) Boot mode ID code authentication state
In this state, the MCU accepts the ID code authentication command.
When boot mode ID codes do not match, the MCU remains in the boot mode ID code authentication state.
Refer to section 46.9.1.1, Boot Mode ID Code Protection for details on boot mode ID code protection. Refer to section 46.10.7, ID Code Authentication Command for details on the ID code authentication command.

(4) Program/erase state
In this state, the MCU executes program/erase or read-check commands according to commands sent from the host.
Refer to section 46.10.8, Program/Erase Commands for details on program/erase commands. Refer to section 46.10.9, Read-Check Commands for details on read-check commands.

46.10.2 Command and Response Configuration

The communication protocol is composed of a “Command” sent from the host to the MCU and a “Response” sent from the MCU to the host. Commands include 1-byte commands and multiple-byte commands. Responses include 1-byte responses, multiple-byte responses, and error responses.
A multiple-byte command and multiple-byte response have “Size” for informing the number of transmit/receive data bytes and “SUM” for detecting communication errors.
“Size” indicates the number of transmit/receive data bytes excluding Command code (the first byte), Size, and SUM.
“SUM” indicates byte data that is calculated so the total bytes of Command or Response becomes 00h.
The flash memory addresses for reading are used as the following addresses: the program address specified in the program command, the block start address specified in the block erase command, the AW start and end addresses specified in the access window information program command, and the AW start and end addresses received in the access window read command.

46.10.3 Response to Undefined Commands

When the MCU receives an undefined command, it sends a command error as a response. The contents of the response are shown below. “Command code” in the error response stores the first byte of the command sent from the MCU.

Error response	80h	Command code

46.10.4 Boot Mode Status Inquiry

This command is used to check the current state and which type of an error occurred immediately after a command issued in the boot program.
Table 46.15 and Table 46.16 list a state or error that the MCU responds to.
The boot mode status inquiry command can be used in the inquiry/setting host command wait state and program/erase host command wait state.

Command	4Fh				
Response	5Fh	Size	State	Error	SUM

Size (1 byte): Total bytes of “State” and “Error” (the value is always 02h)
State (1 byte): MCU’s current state (see Table 46.15)
Error (1 byte): Information about the error occurred in response to a command issued immediately before (see Table 46.16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Table 46.15 Information Regarding the States

Code	State*1	Description
11h	Inquiry/setting host command wait state	Device selection wait state
12h/13h		Operating frequency selection wait state
1Fh		Program/erase host command wait state transition command wait state
31h	Boot mode ID code authentication state	The user area and data area are being erased
3Fh	Program/erase host command wait state	Program/erase command wait state
4Fh		Program data reception wait state
5Fh		Block erase specification wait state

Note 1. Refer to Figure 46.30 for details on the states.

Table 46.16 Error Information

Code	Description
00h	No error
11h	SUM error
21h	Device code error
24h	Bit rate selection error
29h	Block start address error
2Ah	Address error
2Bh	Data length error
51h	Erase error
52h	Not blank (blank check error)
53h	Program error
61h	ID code do not match
63h	ID code do not match and erase error
80h	Command error
FFh	Bit rate automatic adjustment error

46.10.5 Inquiry Commands

Inquiry commands are used to obtain necessary information for sending setting commands, program/erase commands, and read-check commands. Table 46.17 lists the inquiry commands. These commands can only be used in the inquiry/setting host command wait state.

Table 46.17 Inquiry Commands

Command	Description
Supported device inquiry	Inquiry for the device code and series name
Data area availability inquiry	Inquiry for the availability of the data area
User area information inquiry	Inquiry for the number of user areas, and the start and end addresses of the user area
Data area information inquiry	Inquiry for the number of data areas, and the start and end addresses of the data area
Block information inquiry	Inquiry for the start address, the block size, and the number of blocks of each of the user and data areas

46.10.5.1 Supported Device Inquiry

This command is used to obtain the device information for identifying the endian of developed software.
When the MCU receives this command, it sends the device information when developed software uses little endian data and the device information when developed software uses big endian data in this order.

Command	20h		
Response	30h	Size	Number of devices
	Number of characters	Device code for little endian	Series name for little endian
	Number of characters	Device code for big endian	Series name for big endian
	SUM		

Size (1 byte): Total bytes of Number of Devices, Characters, Device code, and Series name
Number of devices (1 byte): Number of endian types that the MCU supports (the value is always 02h)
Number of characters (1 byte): Number of characters for the device code and device name
Device code (4 bytes): Identification code indicating the endian of developed software
Series name (n bytes): The series name of the MCU (ASCII code) and the classification of little endian/big endian
SUM (1 byte): Value that is calculated so the sum of response data is 00h

46.10.5.2 Data Area Availability Inquiry

When the MCU receives this command, it sends the result indicating that the data area is available, area protection can be used, the data area program command is available, and the access window protection is available.

Command	2Ah			
Response	3Ah	Size	Availability	SUM

Size (1 byte): Number of characters of Availability (the value is always 01h)
Availability (1 byte): Availability of the data area (the value is always 5Dh)
5Dh represents the data area is available, area protection can be used, data area program command is available, and the access window protection is available.
SUM (1 byte): Value that is calculated so the sum of response data is 00h (the value is always 68h)

46.10.5.3 User Area Information Inquiry

When the MCU receives this command, it sends the number of user areas and addresses.

Command	25h		
Response	35h	Size	Number of areas
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total bytes of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of user areas (the value is always 01h)
Area start address (4 bytes): Start address of the user area
Area end address (4 bytes): End address of the user area
SUM (1 byte): Value that is calculated so the sum of the response data is 00h

46.10.5.4 Data Area Information Inquiry

When the MCU receives this command, it sends the number of data areas and addresses.

Command	2Bh		
Response	3Bh	Size	Number of areas
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total bytes of data of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of areas in the data area (the value is always 01h)
Area start address (4 bytes): Start address of the data area (the value is always 0010 0000h)
Area end address (4 bytes): End address of the data area (the value is always 0010 1FFFh)
SUM (1 byte): Value that is calculated so the sum of the response data is 00h (the value is always 7Dh)

46.10.5.5 Block Information Inquiry

When the MCU receives this command, it sends the start address, the size of one block, and the number of blocks in the user area and data area.

Command	26h		
Response	36h	Size	DDh
	Start address of the user area		
	Block size of one block for the user area		
	Number of blocks of the user area		
	Start address of the data area		
	Block size of one block for the data area		
	Number of blocks of the data area		
	SUM		

Size (2 bytes): Total bytes of data from DDh to Number of blocks of the data area (the value is always 00 19h)
Start address of the user area (4 bytes): Start address of the user area
Block size of one block for the user area (4 bytes): Memory size of one block (the value is always 00 00 08 00h)
Number of blocks of the user area (4 bytes): Number of blocks in the user area
Start address of the data area (4 bytes): Start address of the data area (the value is always 00 10 00 00h)
Block size of one block for the data area (4 bytes): Memory size of one block (the value is always 00 00 01 00h)
Number of blocks of the data area (4 bytes): Number of blocks in the data area (the value is always 00 00 00 20h)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

46.10.6 Setting Commands

Setting commands are used to configure the settings necessary to execute program/erase commands in the MCU. Table 46.18 lists setting commands. These commands can be used only in the inquiry/setting host command wait state.

Table 46.18 Setting Commands

Command	Function
Device select	Select a device code.
Operating frequency select	Change the bit rate for communication.
Program/erase host command wait state transition	Enter the program/erase host command wait state or boot mode ID code authentication state.

46.10.6.1 Device Select

This command is used to specify the endian of developed software. Select a device code from among the device codes obtained in the response to the support device inquiry command.
If the received device code matches the supported device, the MCU sends a response (46h).
If the device is not supported or the SUM of the received command does not match, the MCU sends an error response.

Command	10h	Size	Device code	SUM

Size (1 byte): Number of characters of the device code (the value is always 04h)
Device code (4 bytes): Identification code to identify an endian of the developed software
(code in the response to the support device inquiry command)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	46h

Error response	90h	Error

Error (1 byte): Error code
11h: SUM error
21h: Device code error

46.10.6.2 Operating Frequency Select

This command is used to specify the operating frequency of the MCU and a bit rate for communication with the flash memory programmer. The bit rate selected in this command should be set to a value with error of less than 4% compared to the bit rate obtained by dividing 32 MHz.
If the specified settings can be supported, the MCU sends a response (06h). If the bit rate error is 4% or more or the SUM of the received command does not match, the MCU sends an error response.

After the host receives a response, wait for at least a 1-bit period at the old bit rate, and send communication confirmation data at the new bit rate.
If the MCU successfully receives communication confirmation data, the MCU sends a response (06h). If the MCU fails to receive the communication confirmation data, the MCU sends an error response.

Command	3Fh	Size	Bit rate	Dummy data
	Number of clocks	Multiplier 1	Multiplier 2	
	SUM			

Size (1 byte): Total bytes of data of Bit rate, Dummy data, Number of clocks, and Multiplier (the value is always 07h)
Bit rate (2 bytes): New bit rate
The value is calculated by dividing the bit rate by 100 (Example: Set 00C0h for 19200 bps)
Dummy data (2 bytes): The value should always be set to 0000h
Number of clocks (1 byte): Types of clocks for multiplier setting (the value is always 02h)
Multiplier 1 (1 byte): Multiplier of the system clock (ICLK) (the value is always 01h)
Multiplier 2 (1 byte): Multiplier of the peripheral module clock (PCLK) (the value is always 01h)
SUM (1 byte): Value that is calculated so the sum of command data including dummy data is 00h

Response	06h	
Error response	BFh	Error

Error (1 byte): Error code
11h: SUM error
24h: Bit rate selection error

Communication confirmation	06h
Response	06h
Error response	FFh

- Bit rate selection error

A bit rate selection error occurs when the bit rate specified with the operating frequency select command cannot be set to a value with error of less than 4%. When the new bit rate specified with the operating frequency select command is B, and 32 (MHz) is Pφ, the bit rate error is calculated by the following formula:

$$\text{Error}(\%) = \left(\frac{P\varphi \times 10^6}{B \times 16 \times N} - 1\right) \times 100$$

$$N = \text{INT}\left(\frac{P\varphi \times 10^6}{B \times 16}\right)$$

Pφ: 32 (MHz)
B: New bit rate (bps)
N: Ratio between Pφ and the new bit rate multiplied by 16 (however, 1 ≤ N ≤ 256)

46.10.6.3 Program/Erase Host Command Wait State Transition

This command is used for the transition from the inquiry/setting host command wait state to the program/erase host command wait state.
When the MCU receives this command, it determines whether boot mode ID code protection is enabled or disabled.

When boot mode ID code protection is disabled, all blocks in the user area and data area are erased.
When all blocks are successfully erased, the MCU sends a response (06h) and enters the program/erase host command wait state. If not all blocks are successfully erased, the MCU sends an error response.
When boot mode ID code protection is enabled, the MCU sends a response (16h) and enters boot mode ID code authentication state.

Command	40h

Response	ACK

ACK (1 byte): ACK code
06h: ID code protection is disabled.
16h: ID code protection is enabled.

Error response	C0h	Error

Error (1 byte): Error code
51h: Erase error

46.10.7 ID Code Authentication Command

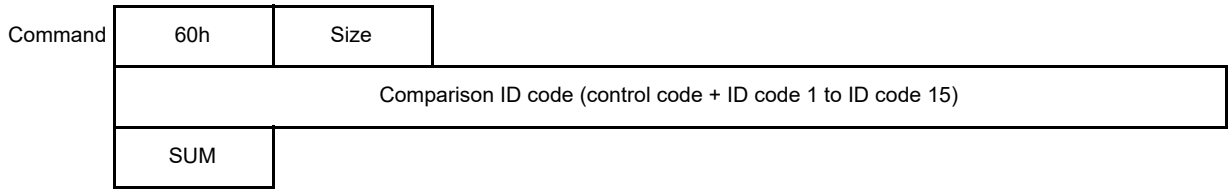
This command is used for ID code authentication when boot mode ID code protection is enabled.
Table 46.19 lists ID code authentication command. This command can be used only in the boot mode ID code authentication state.

Table 46.19 ID Code Authentication Command

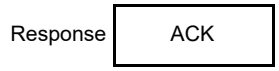
Command	Function
ID code check	Compare the 16-byte code sent from the host and ID code.

46.10.7.1 ID Code Check

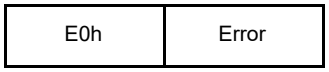
This command is used to unlock boot mode ID code protection.
The comparison ID code specified with the command should be set to the same value as the control code and ID code 1 to ID code 15.
If the comparison ID code sent from the host matches the ID code programmed in the user area, the MCU sends a response (06h) and enters program/erase host command wait state.
If the codes do not match or the SUM of the received command does not match, the MCU sends an error response.
When the ID codes do not match three times consecutively while the control code is 45h, all blocks in the user area and data area are erased. If an error occurs during erasure, the MCU sends an error response.
Also, even if all blocks are successfully erased, the MCU sends an error response and continues the boot mode ID code state. Reset the MCU to enter the program/erase host command wait state.



Size (1 byte): Number of bytes of ID codes (the value is always 10h)
ID code (16 bytes): Control code (1 byte) + ID code 1 to ID code 15 (15 bytes)
SUM (1 byte): Value that is calculated so the sum of the command data is 00h



ACK (1 byte): ACK code
06h: The MCU enters the program/erase host command wait state.



Error (1 byte): Error code
11h: SUM error
61h: ID codes do not match
63h: ID codes do not match and erase error

46.10.8 Program/Erase Commands

Program/erase commands are used to program or erase the user area or data area based on the response to inquiry commands. Table 46.20 lists commands used in each of the program/erase host command wait state, program wait state, and erase wait state. Table 46.21 lists commands that can be accepted in each state.
When a command that is not listed in Table 46.21 is received in each state, the MCU sends a command error response.

Table 46.20 Program/Erase Commands

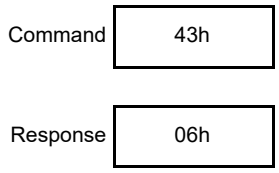
Command	Function
User/data area program preparation	Select the user area or data area to program, and enter the program wait state.
Program	Program the specified data to the selected area in the user area or data area. Or enter the program/erase host command wait state (end of program).
Data area program	Program the specified-size data to the selected area in the data area. Or enter the program/erase host command wait state (end of program).
Erase preparation	Enter the erase wait state.
Block erase	Erase the selected block, or enter the program/erase host command wait state (end of erase).

Table 46.21 Acceptable Commands for Each State

State	Acceptable Command
Program/erase host command wait state	User/data area program preparation command, and erase preparation command
Program wait state	Program command and data area program command
Erase wait state	Block erase command

46.10.8.1 User/Data Area Program Preparation

This command is used to prepare for accepting the program command and the data area program command.
When the MCU receives this command, it recognizes that an instruction to prepare for the program command is issued from the host. Then, the MCU enters the program wait state, where only the program command and the data area program command can be accepted, and sends a response (06h).



46.10.8.2 Program

This command is used to program the specified data to the user area or data area. Set the lower 8 bits to 0 for the program address selected in this command. When the data length is shorter than 256 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 50h FFh FFh FFh FFh B4h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.

Command	50h	Program address
	Program data	
	SUM	

Program address (4 bytes): Address for program destination
Set the lower 8 bits to 0
Set FFFF FFFFh for end of program
Program data (n bytes): Program data (n = 256, 0 for end of program)
When the program data is less than n bytes, set FFh for the missing data.
No program data for the end of program
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	06h

Error response	D0h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
53h: Program error (the data cannot be programmed.)

46.10.8.3 Data Area Program

This command is used to program the specified data to the data area. Set the lower 2 bits to 0 for the program address selected in this command. When the data length is shorter than 4 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 51h FFh FFh FFh FFh 00h B3h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.

Command	51h	Program address	Program data length
	Program data		
	SUM		

Program address (4 bytes): Address for program destination
Set the lower 2 bits of the selected address to 0
Set FFFF FFFFh for end of data area program
Program data length (1 byte): Size of program data
Set 4-byte data
Set 00h for end of data area program
Program data (n bytes): Program data for the data area (n = program data length, 0 for end of program)
Set data of the program data length
When the program is less than n bytes, set FFh for the missing data.
No program data for the end of data area program
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	06h

Error response	D1h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
2Bh: Program data length error
53h: Program error (the data or program data cannot be programmed.)

46.10.8.4 Erase Preparation

This command is used to prepare for accepting the block erase command.
When the MCU receives this command, it recognizes that an instruction to prepare for the erase command is issued from the host. Then, the MCU enters the erase wait state, where only the block erase command can be accepted, and sends a response (06h).

Command	48h
Response	06h

46.10.8.5 Block Erase

This command is used to erase the selected block in the user area or data area.
Specify the block start address selected in the command by calculating the address based on the response to the block information inquiry command.
When the block selected in the block start address is successfully erased, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during an erase operation, the MCU sends an error response.

To enter the program/erase host command wait state after the erase operation ends, send 59h 04h FFh FFh FFh FFh A7h from the host. The MCU enters the program/erase host command wait state and sends a response (06h).

Command	59h	Size
	Block start address	
Response	SUM	

Size (1 byte): Total bytes of Block start address (the value is always 04h)
Block start address (4 bytes): Start address of the block that is erased
Set FFFF FFFFh for end of erase
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h	
Error response	D9h	Error

Error (1 byte): Error code
11h: SUM error
29h: Block start address error
51h: Erase error (the selected block cannot be erased)

46.10.9 Read-Check Commands

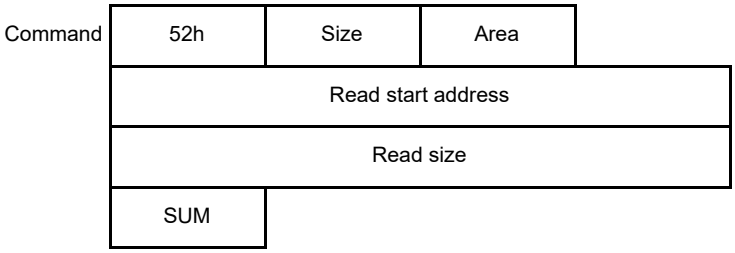
Read-check commands are used to read data or check whether data is programmed in the user area or data area in the MCU based on the response to inquiry commands.
Table 46.22 lists read-check commands used in the program/erase host command wait state.

Table 46.22 Read-Check Commands

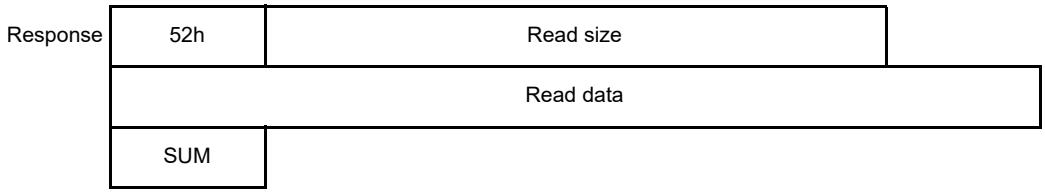
Command	Function
Memory read	Read data from the user area or data area.
User area checksum	Obtain the checksum of the entire user area.
Data area checksum	Obtain the checksum of the entire data area.
User area blank check	Check whether data is programmed in the user area.
Data area blank check	Check whether data is programmed in the data area.
Access window information program	Set the access window.
Access window read	Read the settings of the access window.

46.10.9.1 Memory Read

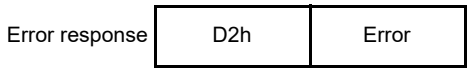
This command is used to read data programmed in the user area or data area.
For a read start address selected in the command, set a value within the range from the area start address to the area end address received in the response to the user area information inquiry command or the data area information inquiry command.
For a read size selected in the command, set a value so the sum of the read start address and the read size is within the range from the area start address to the area end address received in the response to the user area information inquiry command or the data area information inquiry command.
When the MCU performs a read successfully, it sends data of the specified range. If the SUM of the received command does not match or the MCU fails to perform a read successfully, it sends an error response.



Size (1 byte): Total bytes for Read start address and Read size
Area (1 byte): Area that is read
01h: User area or data area
Read start address (4 bytes): Start address of the area that is read
Read size (4 bytes): Size of data that is read (in bytes)
SUM (1 byte): Value that is calculated so the sum of response data is 00h



Read size (4 bytes): Size of Data that is read (in bytes)
Read data (n bytes): Data read from the specified range (n = read size)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

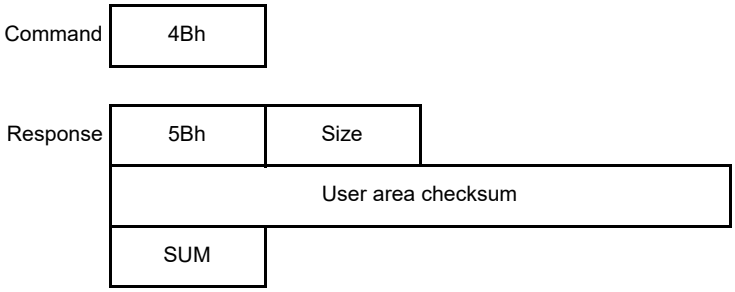


Error (1 byte): Error code
11h: SUM error
2Ah: Address error
- A value other than 01h is set for the “Area” field.
- The read start address is not in the selected area.

2Bh: Size error
- The read size is set to 0000 0000h.
- The read size exceeds the area size.
- The address calculated from the read start address and read size is not in the selected area.

46.10.9.2 User Area Checksum

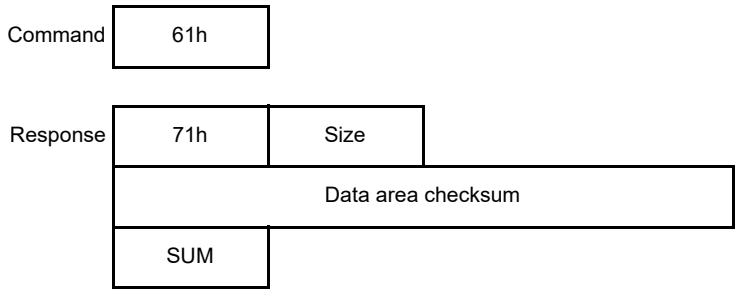
This command used to obtain the checksum of the entire user area.
When the MCU receives this command, it adds data from the start address to the end address in bytes in the user area, and sends the calculated result (checksum) as a response.



Size (1 byte): Number of bytes for checksum of the user area (the value is always 04h)
User area checksum (4 bytes): Calculated result of the data in the user area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

46.10.9.3 Data Area Checksum

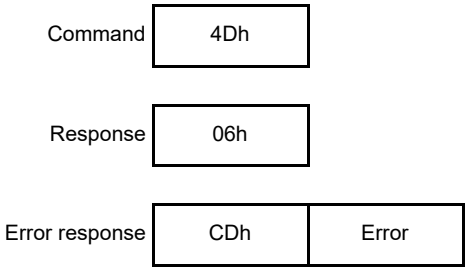
This command used to obtain the checksum of the entire data area.
When the MCU receives this command, it adds data from the start address to the end address in bytes in the data area, and sends the calculated result (checksum) as a response.



Size (1 byte): Number of bytes for checksum of the data area (the value is always 04h)
Data area checksum (4 bytes): Calculated result of the data in the data area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

46.10.9.4 User Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of programmed data, the MCU sends an error response.



Error (1 byte): Error code
52h: Not blank

46.10.9.5 Data Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is programmed data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of programmed data, the MCU sends an error response.

Command	62h	
Response	06h	
Error response	E2h	Error

Error (1 byte): Error code
52h: Not blank

46.10.9.6 Access Window Information Program

This command is used to set the access window used for area protection.
For the access window start address selected in the command, set the start address of the start block. For the access window end address, set the end address of the end block.
When the specified access window settings are successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during the access window settings, the MCU sends an error response.
For details on the access window, see section 46.6, Area Protection.

Command	74h	05h	Access window	
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	SUM			

Access window (1 byte): Set the access window or clear the access window settings
Set 00h to set the access window
Set FFh to clear the access window settings
Access window start address LH (1 byte): Start address of the access window (A15 to A8)
Set A15 to A8 of the start address of the start block.
Set FFh to clear the access window settings
Access window start address HL (1 byte): Start address of the access window (A23 to A16)
Set A23 to A16 of the start address of the start block.
Set FFh to clear the access window settings
Access window end address LH (1 byte): End address of the access window (A15 to A8)
Set A15 to A8 of the end address of the end block.
Set FFh to clear the access window settings
Access window end address HL (1 byte): End address of the access window (A23 to A16)
Set A23 to A16 of the end address of the end block.
Set FFh to clear the access window settings
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h	
Error response	F4h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (specified address is not in the area)
53h: Program error (access window cannot be set)

46.10.9.7 Access Window Read

This command is used to check the set range of the access window.
When the MCU successfully obtains the access window range, the MCU sends the access window start address and end address that it read. If the SUM of the received command does not match, the MCU sends an error response.

Command	73h	01h	FFh	8Dh

Response	73h	05h		
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	FFh			
	SUM			

Access window start address LH (1 byte): Start address of the access window range (A15 to A8)
Access window start address HL (1 byte): Start address of the access window range (A23 to A16)
Access window end address LH (1 byte): End address of the access window range (A15 to A8)
Access window end address HL (1 byte): End address of the access window range (A23 to A16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	F3h	Error

Error (1 byte): Error code
11h: SUM error

46.10.9.8 Access Window Protection

This command is used to protect the settings for the access window used for area protection.
When the protection is successfully set, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during the protect settings, the MCU sends an error response.

Command	7Ch	01h	FFh	84h

Response	06h

Error response	FCh	Error

Error (1 byte): Error code
11h: SUM error
53h: Program error (protection cannot be set)

46.10.9.9 Access Window Protection Flag Read

This command is used to check whether the settings for the access window are protected or not.
When the protection status is successfully obtained, the MCU sends the status. If the SUM of the received command does not match, the MCU sends an error response.

Command	7Bh	01h	FFh	85h

Response	7Bh	01h	Status	SUM

Status (1 byte): Protection status
01h: Protection is enabled
00h: Protection is disabled
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	FBh	Error

Error (1 byte): Error code
11h: SUM error

46.11 Serial Programmer Operation in Boot Mode (SCI Interface)

The following describes the procedure for the serial programmer to program/erase the user area and data area in boot mode (SCI Interface).

1. Automatically adjust the bit rate
2. Receive the MCU information*1
3. Select the device and change the bit rate
4. Enter the program/erase host command wait state
5. Unlock boot mode ID code protection
6. Erase the user area and data area*2, *3
7. Program the user area and data area*2, *3
8. Check data in the user area*2
9. Check data in the data area*2
10. Set the access window in the user area*2
11. Protect the access window settings*2
12. Reset the MCU

Note 1. If the necessary information has been already received, step 2 can be skipped.
Note 2. Processing steps from 6 to 11 can be proceeded as necessary, and their order other than step 12 can be changed.
Note 3. When a timeout occurs or invalid response data is received, stop the operation and perform step 12 (reset the MCU).

Refer to section 46.10.5, Inquiry Commands, section 46.10.6, Setting Commands, section 46.10.7, ID Code Authentication Command, section 46.10.8, Program/Erase Commands, and section 46.10.9, Read-Check Commands for details on the commands used in the above steps 2 to 11.

46.11.1 Bit Rate Automatic Adjustment

The MCU measures the low width of data 00h that is sent from the serial programmer at 9,600 or 19,200 bps to automatically adjust the bit rate.

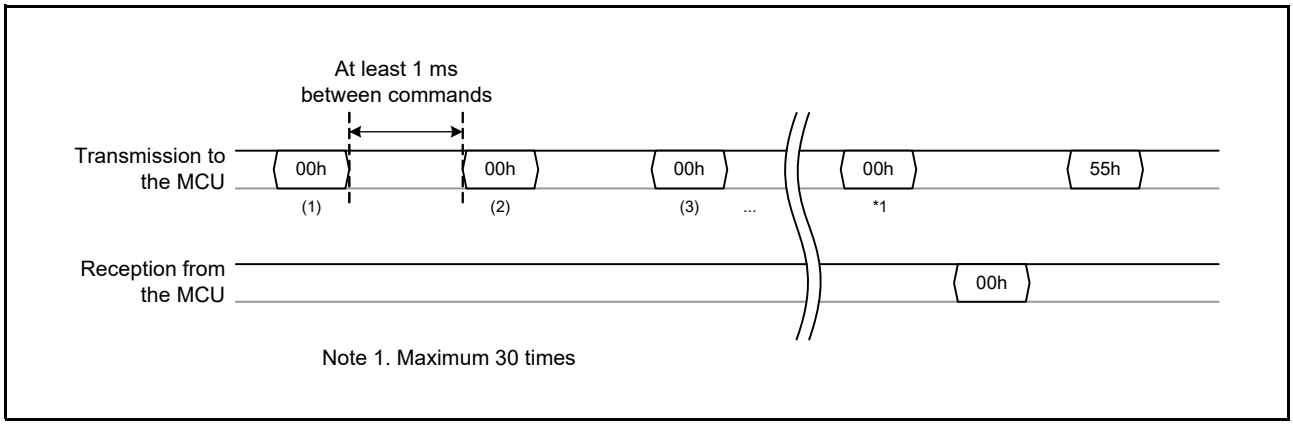


Figure 46.31 Transmit/Receive Data for Bit Rate Automatic Adjustment

After starting up in boot mode, wait for at least 400 ms and then send 00h to the MCU from the serial programmer. When the bit rate adjustment is completed, the MCU sends 00h to the programmer. When the programmer receives 00h, send 55h to the MCU from the programmer. When the programmer can not receive 00h, wait for at least 1 ms and send 00h to the MCU again. When the programmer fails to receive 00h even if it send 00h 30 times, restart the MCU in boot mode and perform the automatic adjustment for the bit rate again.
When the MCU receives 55h, the MCU sends E6h and enters the inquiry/setting command wait state. If the MCU fails to receive 55h, the MCU sends FFh. When the programmer receives FFh, restart the MCU in boot mode, and perform the automatic adjustment for the bit rate again.

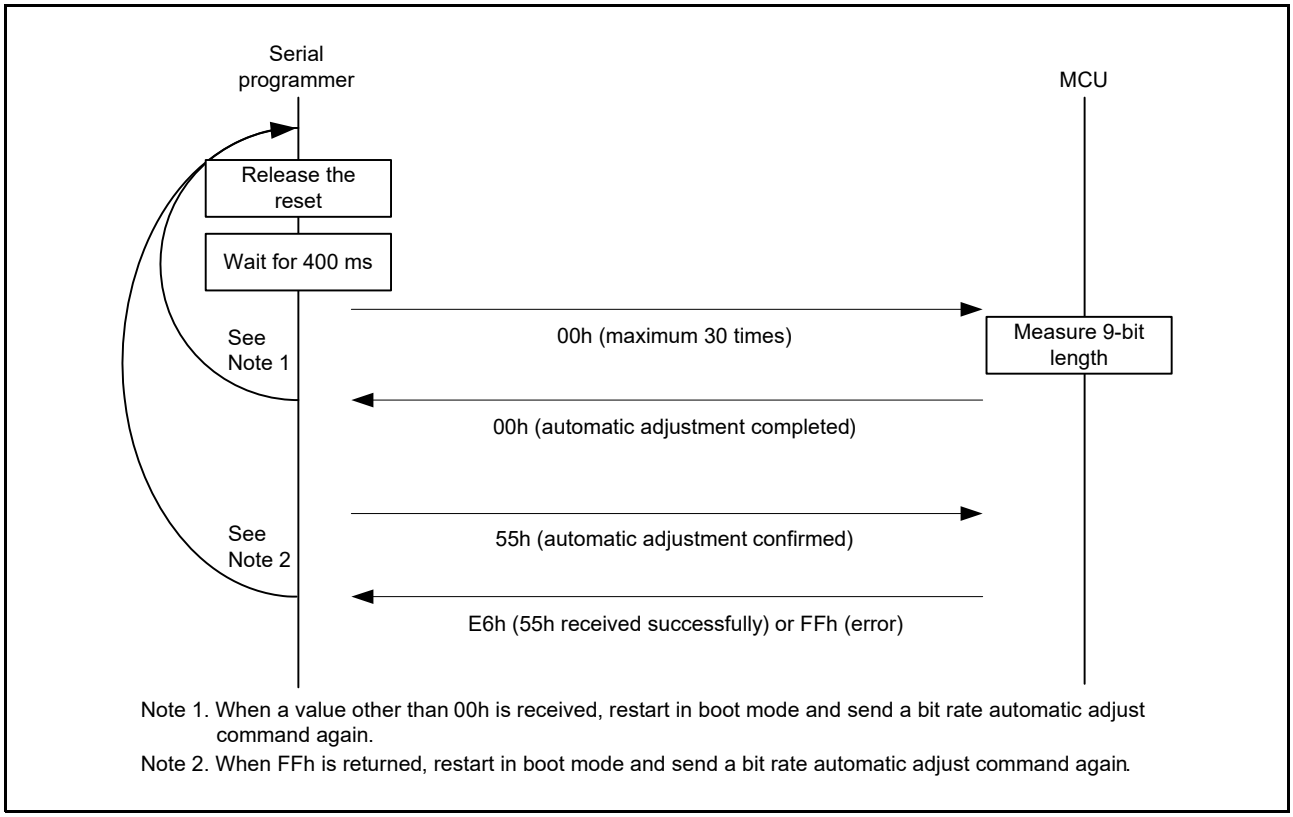


Figure 46.32 Bit Rate Automatic Adjustment Procedure

46.11.2 Receive the MCU Information

Procedure to send inquiry commands, and receive the information necessary to send setting commands, program/erase commands, and read-check commands is as follows.

(1) Send a support device inquiry command (20h) to check what type of endianness the MCU supports. The MCU returns all device codes and series names that it supports.
(2) Send a user area information inquiry command (25h) to check the start and end addresses of the user area. The MCU returns the start and end addresses of the user area.
(3) Send a block information inquiry command (26h) to check the block configuration. The MCU returns the start address, the size of one block, and the number of blocks for the user area and data area.
(4) Send a data area information inquiry command (2Bh) to check the start and end addresses of the data area. The MCU returns the start and end addresses of the data area.

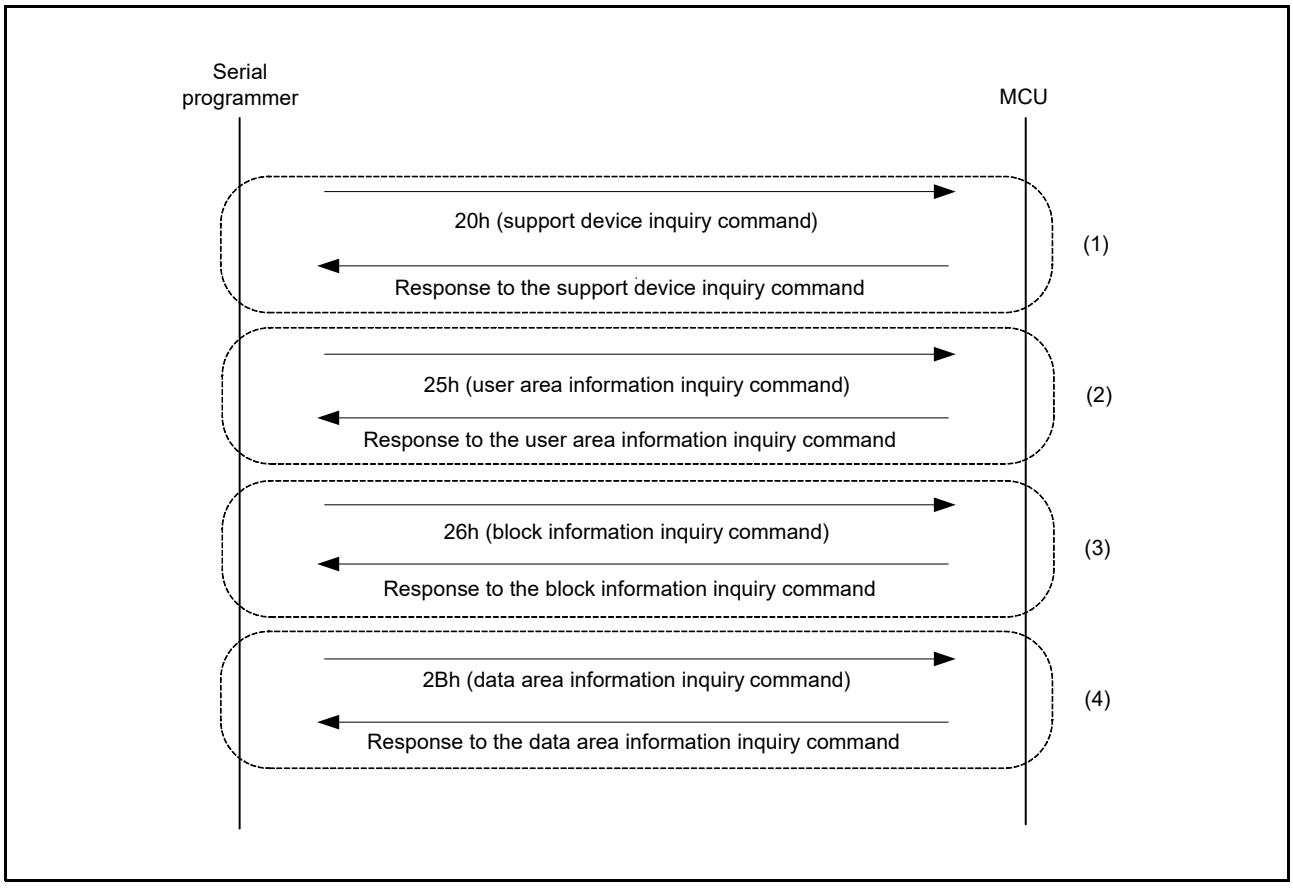


Figure 46.33 Procedure to Receive the MCU Information

46.11.3 Select the Device and Change the Bit Rate

Procedure to select the device to connect with the serial programmer and to change the bit rate for communication is as follows.

(1) Send the device select command (10h). Select the device code according to the endian of developed software.
(2) Send the operating frequency select command (3Fh) to change the communication bit rate from 9,600 or 19,200 bps.

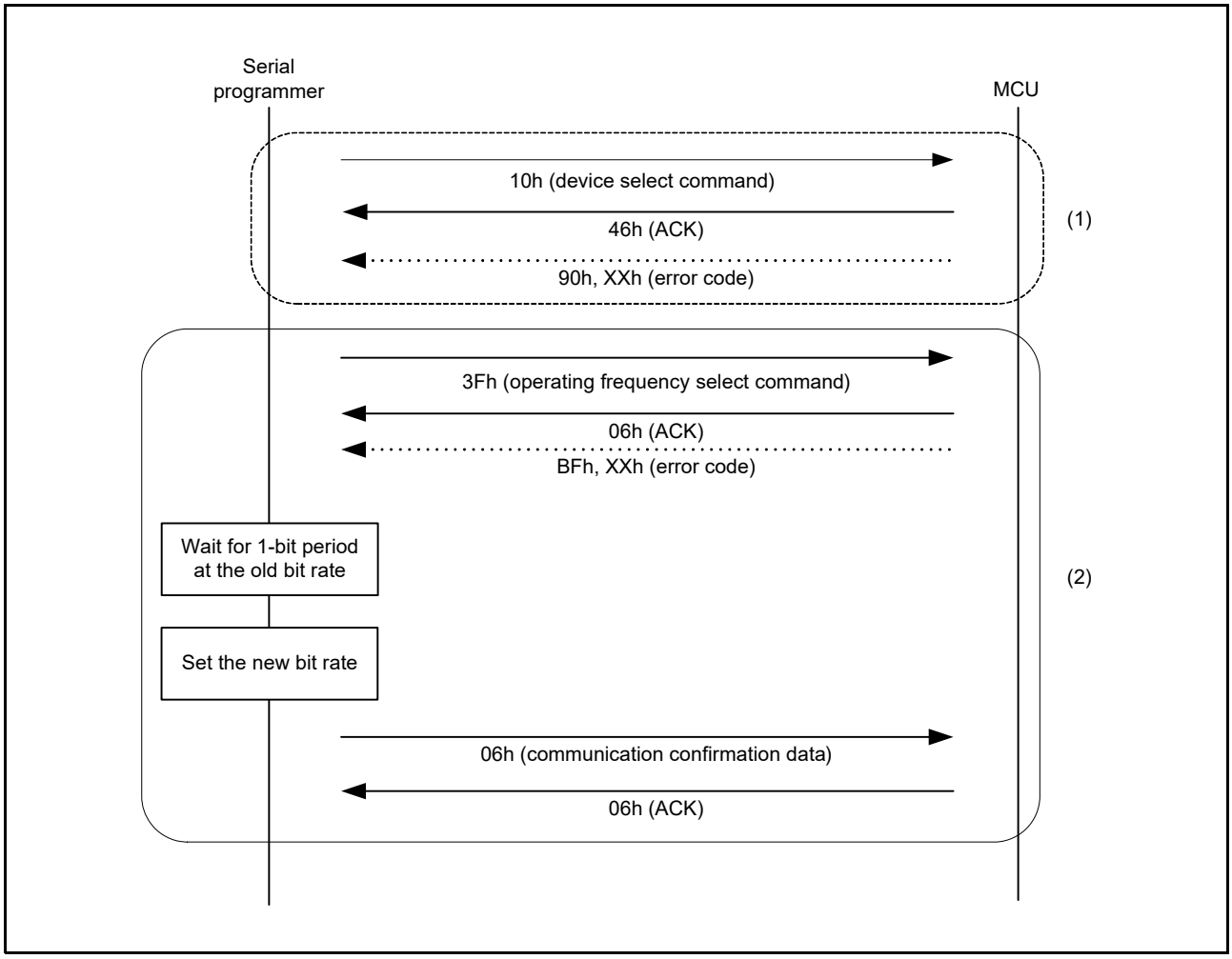


Figure 46.34 Procedure to Select the Device and Change the Bit Rate

46.11.4 Enter the Program/Erase Host Command Wait State

Send the program/erase host command wait state transition command to perform program/erase operations. The MCU sends a response according to whether boot mode ID code protection is enabled or disabled.

(1) When boot mode ID code protection is disabled, the MCU sends a response (06h), and enters the program/erase host command wait state. Use the serial programmer to start from the operation described in section 46.11.6, Erase the User Area and Data Area.

(2) When the boot mode ID code protection is enabled, the MCU sends a response (16h), and enters the ID code authentication wait state. Use the serial programmer to start from the operation described in section 46.11.5, Unlock Boot Mode ID Code Protection.

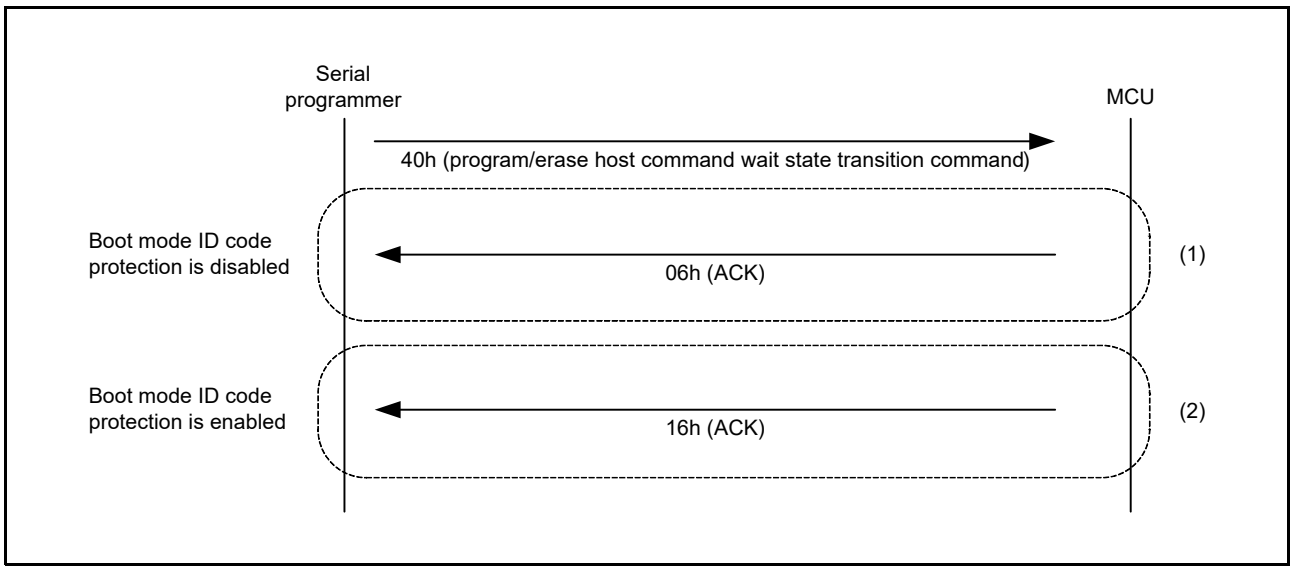


Figure 46.35 Procedure to Transition to the Program/Erase Host Command Wait State

46.11.5 Unlock Boot Mode ID Code Protection

Send the ID code check command to unlock boot mode ID code protection.

(1) When ID codes match, the MCU enters the program/erase host command wait state. Data in the user area and data area are not erased. Use the serial programmer to start from the operation described in section 46.11.6, Erase the User Area and Data Area.

(2) If ID codes do not match consecutively, the MCU remains in the boot mode ID code authentication state. Reset the MCU, and then use the serial programmer to start again from section 46.11.1, Bit Rate Automatic Adjustment.

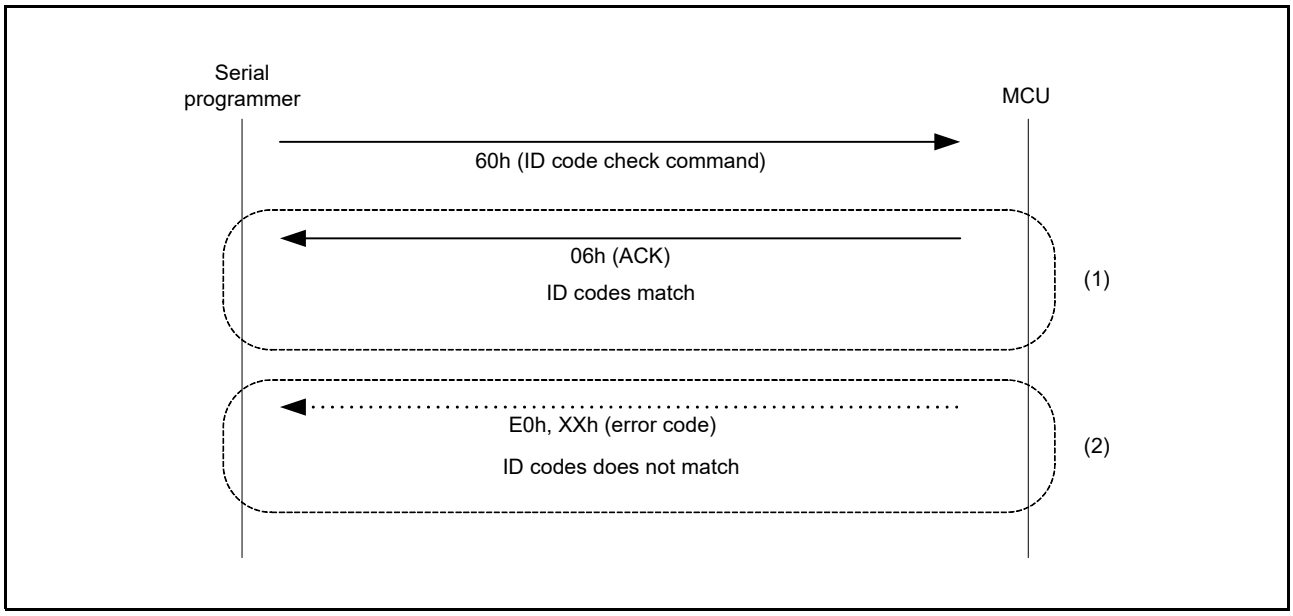


Figure 46.36 Procedure to Unlock ID Code Protection

46.11.6 Erase the User Area and Data Area

Procedure to erase blocks that are programmed in the user area and data area to program a user program and data is as follows.

(1) Send an erase preparation command (48h).
(2) Send a block erase command (59h).
(3) To place the MCU in the program/erase host command wait state, send a block erase command for ending the erasure (59h 04h FFh FFh FFh FFh A7h).

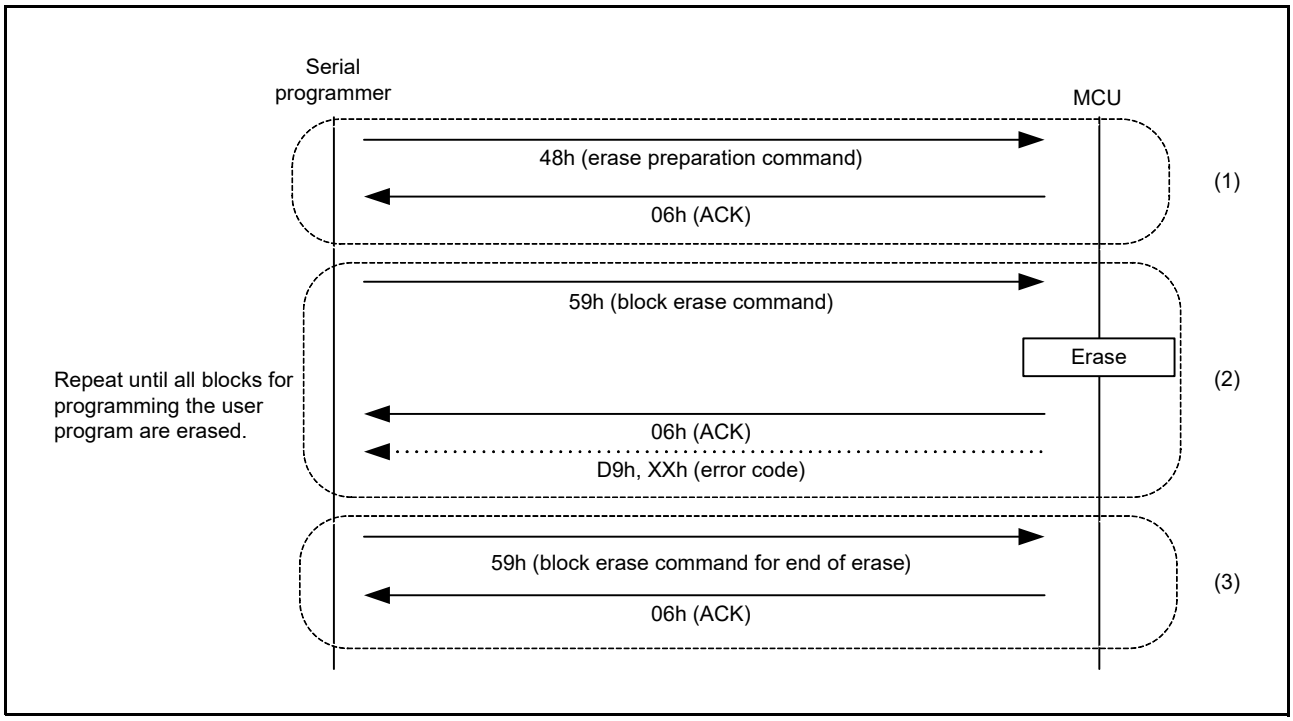


Figure 46.37 Procedure to Erase the User Area and Data Area

46.11.7 Program the User Area and Data Area

Procedure to program a user program and data in the user area and data area is as follows.

(1) Send the user/data area program preparation command (43h).
(2) Send the program command (50h) or the data area program command (51h).
(3) To place the MCU in the program/erase host command wait state, send the program command (50h FFh FFh FFh FFh B4h) or the data area program command (51h FFh FFh FFh FFh 00h B3h) for ending the programming.

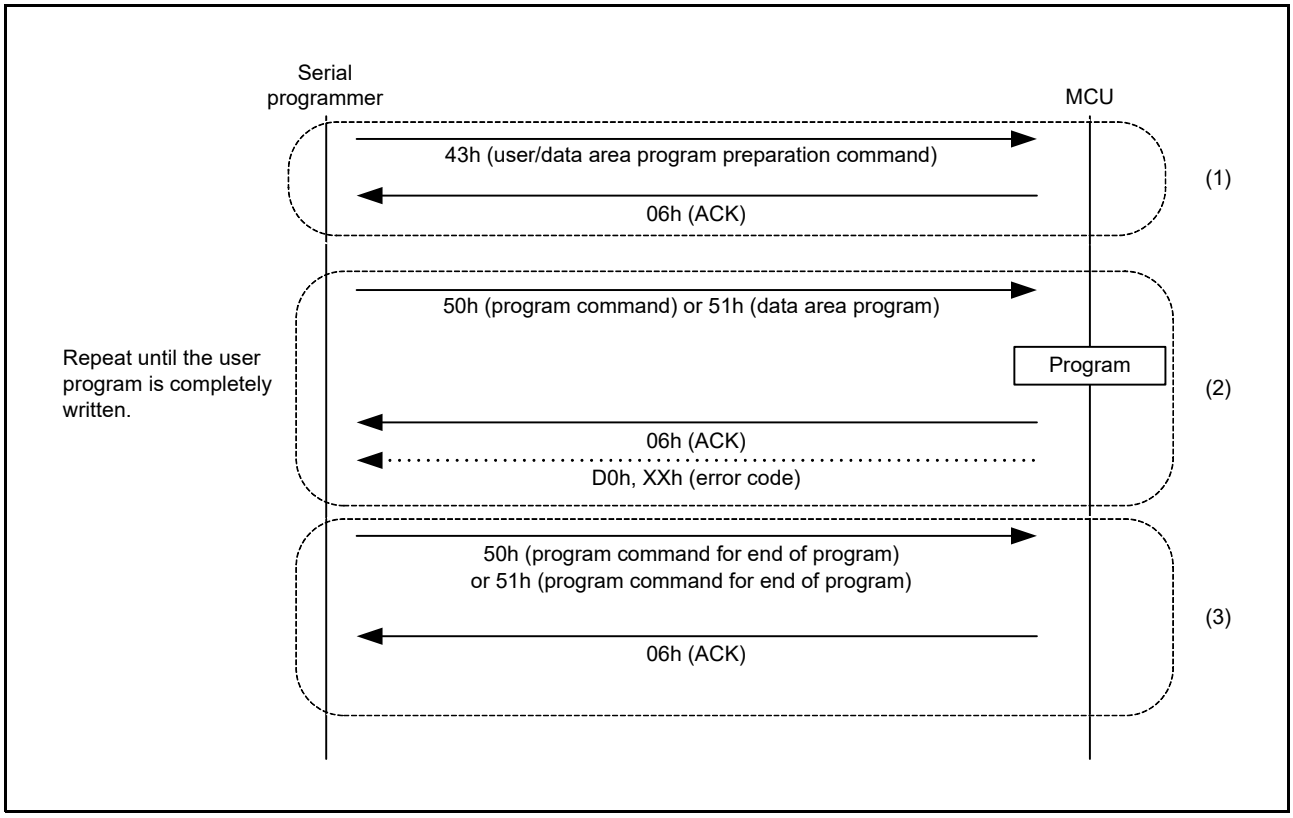


Figure 46.38 Procedure to Program the User Area and Data Area

46.11.8 Check Data in the User Area

Procedure to read and check, checksum, and blank check the user area to check the programmed data in the user area is as follows.

(1) The read and check operation is used to read data in the user area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the user area.
(2) Send the user area checksum command (4Bh) to check program data using the checksum of user area.
(3) Send a user area blank check command (4Dh) to check if the user area has data.

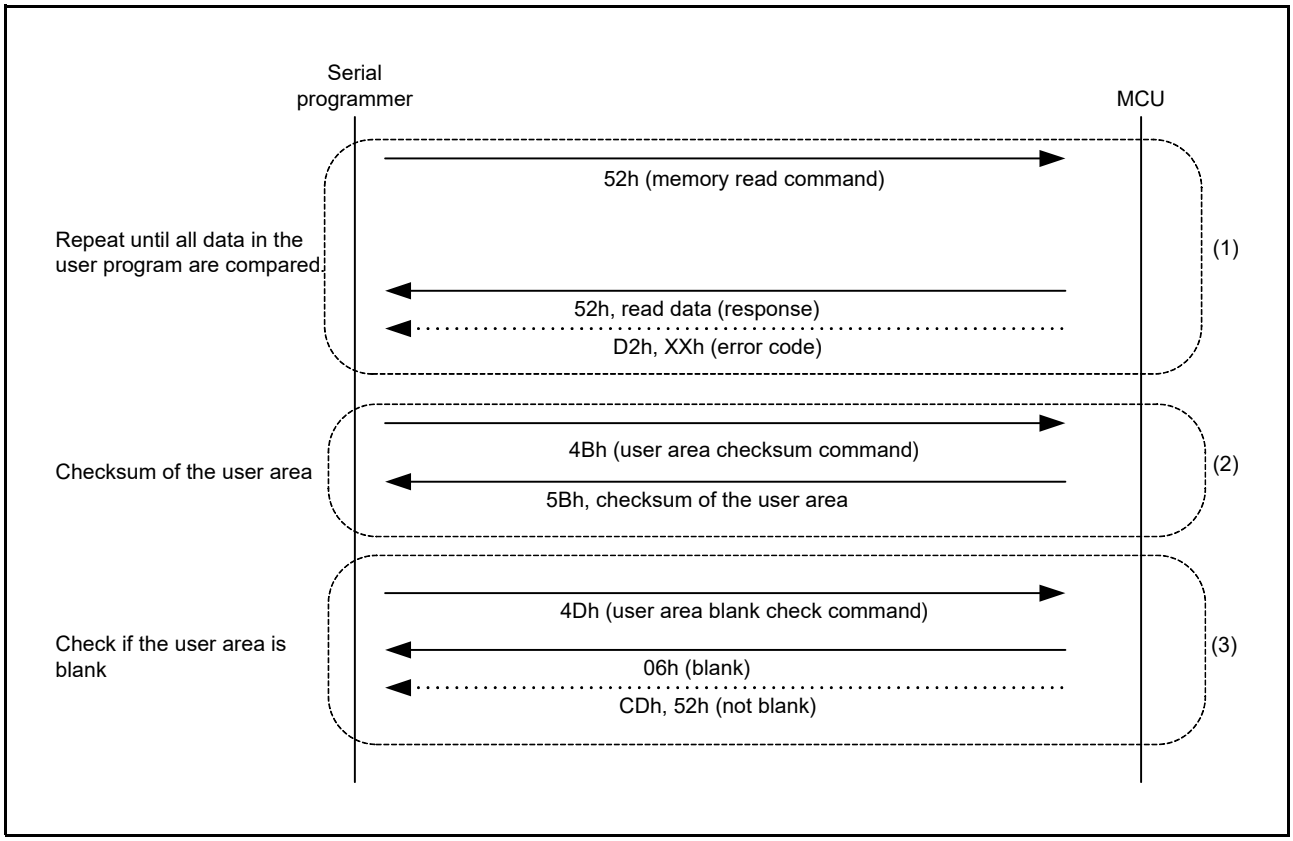


Figure 46.39 Procedure to Check Data in the User Area

46.11.9 Check Data in the Data Area

Procedure to read and check, checksum, and blank check the data area to check the programmed data in the data area is as follows.

(1) The read and check operation is used to read data in the data area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the data area.
(2) Send the data area checksum command (61h) to check program data using the checksum of data area.
(3) Send the data area blank check command (62h) to check if the data area has data.

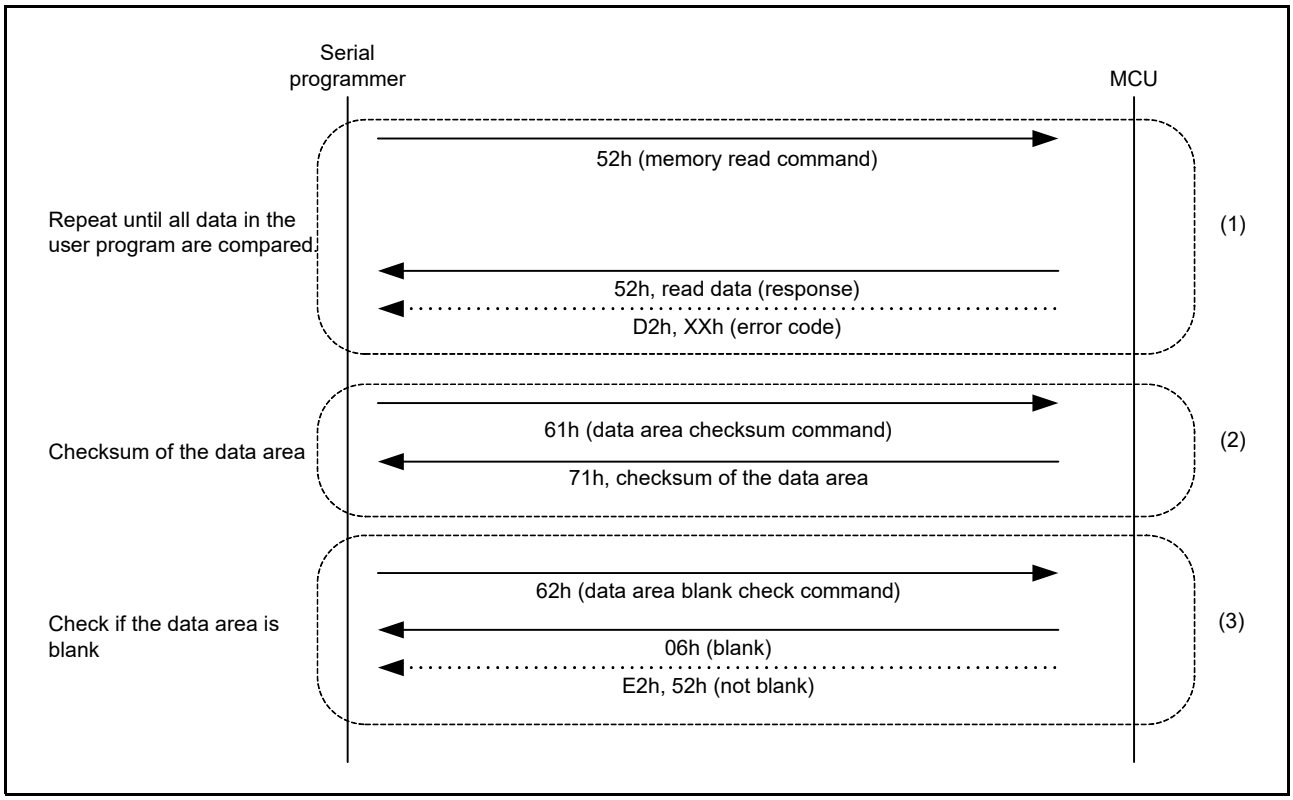


Figure 46.40 Procedure to Check Data in the Data Area

46.11.10 Set the Access Window in the User Area

Procedure to set the access window to avoid unintentionally rewriting the user area during the self-programming is as follows.

(1) Send the access window program command (74h) to set the access window settings.
(2) Send the access window read command (73h) to confirm the access window settings.

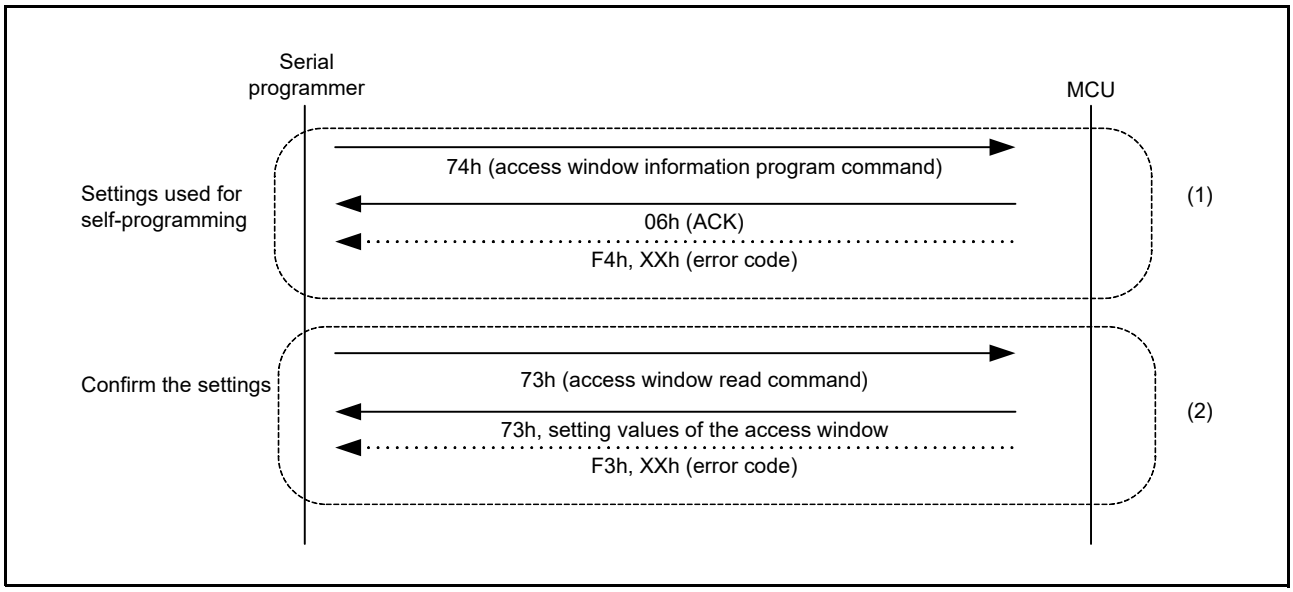


Figure 46.41 Procedure to Set the Access Window in the User Area

46.11.11 Protect the Access Window Settings

Procedure to protect the access window settings to avoid unintentionally rewriting the addresses of the access window during the self-programming is as follows.

(1) Send the access window protection command “7Ch” to protect the access window settings.
(2) Send the access window protection flag read command “7Bh” to confirm the setting of the protection.

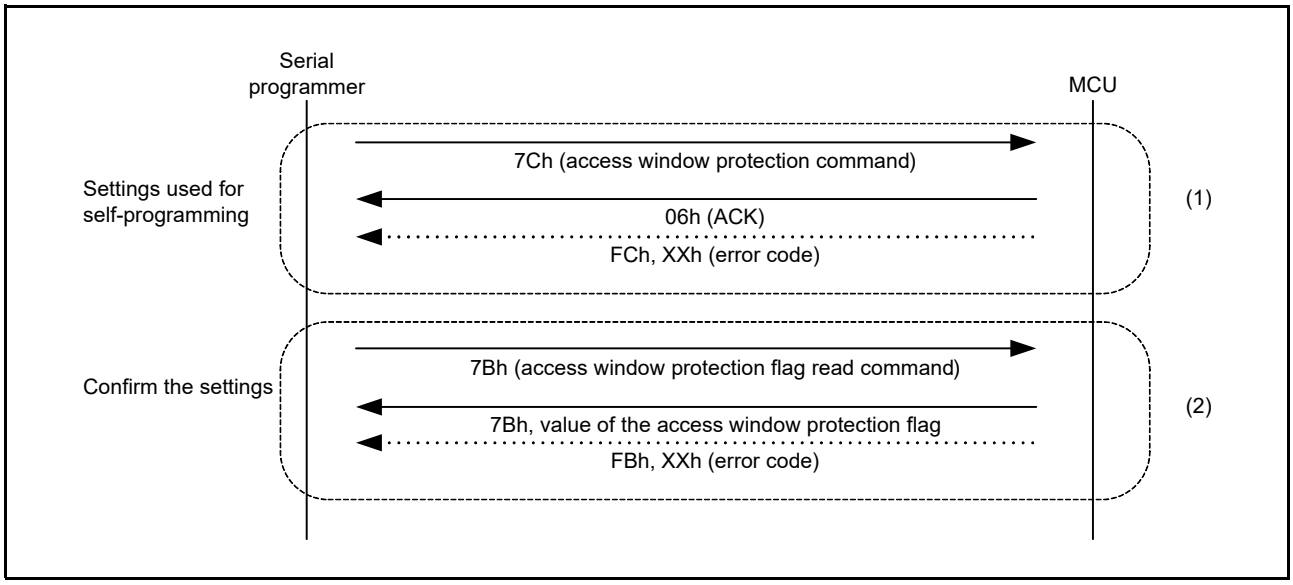


Figure 46.42 Procedure to Protect the Access Window

46.12 Rewriting by Self-Programming

46.12.1 Overview

The MCU supports rewriting of the flash memory by the user program. The ROM and E2 DataFlash can be rewritten by preparing a routine to rewrite the flash memory (flash rewrite routine) in the user program.
When rewriting the E2 DataFlash, the BGO can be used to execute the flash rewrite routine on the ROM. The E2 DataFlash can also be rewritten by executing the flash rewrite routine that is transferred on the RAM in advance.

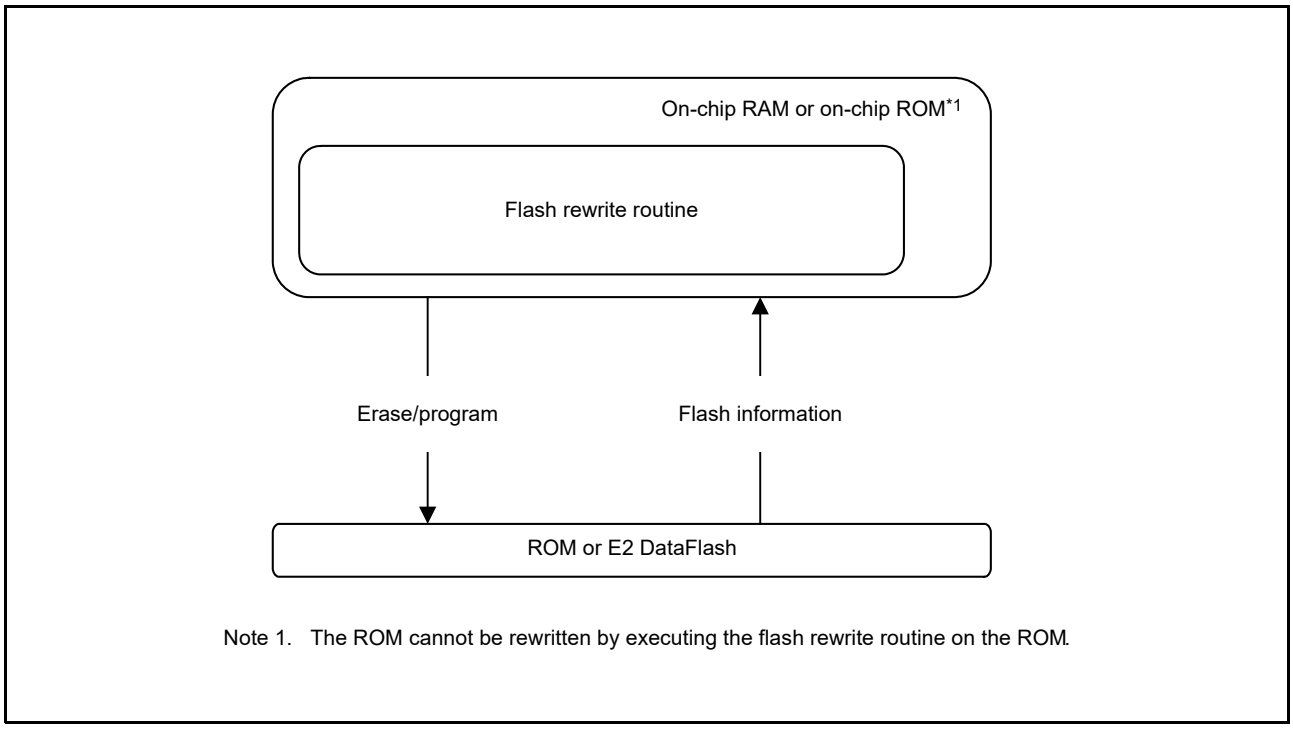


Figure 46.43 Self-Programming Overview

46.13 Usage Notes

(1) Access the Block Where Erase Operation is Forcibly Stopped
When forcibly stopping an erase operation, data in the block where the erase operation is aborted is undefined. To avoid malfunctions caused by reading undefined data, do not execute instructions or read data in the block where an erase operation is forcibly stopped.

(2) Processing After Forced Stop of Erase Operation
When an erase operation is forcibly stopped, issue a block erase command again to the same block.

(3) Additional Programming Disabled
The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

(4) Reset during Program/Erase
If inputting a reset from the RES# pin, release the reset after reset input time of at least tRESW (refer to section 47, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics.
The IWDT reset and software reset can be used regardless of tRESW.

(5) Location of Interrupt Vectors and Exception Vectors during Program/Erase Operation
When an interrupt or an exception occurs during a program/erase operation, the vector may be fetched from the ROM. To avoid fetching the vector from the ROM, allocate the interrupt vector table and exception vector table to the area other than the ROM with the INTB and EXTB registers in the CPU.

(6) Program/Erase in Low-Speed Operating Mode
Do not program or erase the flash memory when low-speed operating mode is selected by the SOPCCR register for low-power consumption functions.

(7) Abnormal Termination during Program/Erase
When the voltage exceeds the range of the operating voltage during a program/erase operation or when a program/erase operation is not completed successfully due to a reset or prohibited actions described in (8), erase the area again.

(8) Actions Prohibited during Program/Erase
To prevent the damage to the flash memory, comply with the following instructions.

- Do not use the MCU power supply that is outside the operating voltage range.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not update the value of the SOPCCR.SOPCM bit.
- Do not change the clock source select bit in the SCKCR3 register.
- Do not enable switching clock sources by setting the RSTCKCR.RSTCKEN bit when exiting sleep mode.
- Do not change the division ratio of the flash interface clock (FCLK).
- Do not place the MCU in deep sleep mode or software standby mode.
- Do not access the E2 DataFlash during a program/erase operation to the ROM.
- Do not change the DFLCTL.DFLEN bit value during a program/erase operation to the E2 DataFlash.

(9) FCLK during Program/Erase
For programming/erasure by self-programming, set the frequency of the FlashIF clock (FCLK), and specify an integer FCLK frequency (MHz) in FISR.PCKA[5:0] bits. Note that when the FCLK is 4 to 32 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (i.e. 12.5 MHz should be set rounded up to 13 MHz). If the FCLK is equal to or less than 4 MHz, only 1, 2, 3, or 4 MHz can be used.

46.14 Usage Notes in Boot Mode

(1) Notes on Communication Errors in Boot Mode
When communication with the MCU cannot be performed properly, reset and start up in boot mode again.

(2) Notes on Option-Setting Memory in Boot Mode
The settings of option function select register 0 (OFS0), option function select register 1 (OFS1), and endian select register (MDE) are disabled in boot mode.

(3) Notes on Clocks in Boot Mode (USB Interface)
When USB interface mode is selected, externally input a clock to the EXTAL or XTAL pin, or connect a crystal or ceramic resonator to supply a clock.
Use a 4, 6, 8, 12, or 16 MHz external clock in boot mode (USB interface). An clock other than a 4, 6, 8, 12, or 16 MHz external clock cannot be used.

(4) Notes on Power Supply Voltage in Boot Mode (USB Interface)
Use a voltage between 3.0 V and 3.6 V in boot mode (USB interface). A voltage that is lower than 3.0 V cannot be used.

(5) Notes on Switching the Start-Up Area
Switch the start-up area by self-programming.

47. Electrical Characteristics

47.1 Absolute Maximum Ratings

Table 47.1 Absolute Maximum Ratings

Conditions: VSS = AVSS0 = VREFL0 = 0 V

Item		Symbol	Value	Unit
Power supply voltage		VCC	–0.3 to +6.5	V
Input voltage	Ports for 5 V tolerant*1	V_{in}	–0.3 to +6.5	V
	P03 to P07, P40 to P47, PJ6, PJ7		–0.3 to AVCC0 + 0.3	V
	Ports other than above		–0.3 to VCC + 0.3	V
Reference power supply voltage		VREFH0	–0.3 to AVCC0 + 0.3	V
Analog power supply voltage		AVCC0	–0.3 to +6.5	V
Analog input voltage	AN000 to AN007 are in use.	V_{AN}	–0.3 to AVCC0 + 0.3	V
	AN016 to AN031 are in use.		–0.3 to VCC + 0.3	
Junction temperature	D-version	T_j	–40 to +105	°C
	G-version		–40 to +112	
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.

To preclude any malfunctions due to noise interference, insert capacitors of high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, and between the VREFH0 and VREFL0 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.

Connect the VCL pin to a VSS pin via a 4.7 µF capacitor. The capacitor must be placed close to the pin, refer to section 47.16.1, Connecting VCL Capacitor and Bypass Capacitors.

Do not input signals or an I/O pull-up power supply to ports other than 5-V tolerant ports while the device is not powered.

The current injection that results from input of such a signal or I/O pull-up may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Even if –0.3 to +6.5 V is input to 5-V tolerant ports, it will not cause problems such as damage to the MCU.

Note 1. P12, P13, P16, and P17 are 5 V tolerant.

47.2 Recommended Operating Conditions

Table 47.2 Recommended Operating Conditions (1)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltages		VCC*1, *2	USB is in use.	3.0	—	3.6	V
			One or more of PLL, PLL2, RSIP, CTSU, internal reference voltage, and temperature sensor are in use.	1.8	—	5.5	
			Other than above	1.6	—	5.5	
		VSS		—	0	—	
Analog power supply voltages		AVCC0*1		1.6	—	5.5	V
		AVSS0		—	0	—	
		VREFH0		1.6	—	AVCC0	
		VREFL0		—	0	—	
Input voltage	Ports for 5 V tolerant: P12, P13, P16, P17	V_{in}		–0.3	—	5.8	V
	P03 to P07, P40 to P47, PJ6, PJ7			–0.3	—	AVCC0 + 0.3	
	Ports other than above			–0.3	—	VCC + 0.3	
Operating temperature*3	D version	T_{opr}		–40	—	85	°C
	G version			–40	—	105	

Note 1. When powering on the VCC and AVCC0 pins, power them on at the same time or the VCC pin first and then the AVCC0 pin.
Note 2. When VCC < 2.4 V, normal operating mode functions of the CTSU are restricted. For details, refer to section 39, Capacitive Touch Sensing Unit (CTSU2SLa).
Note 3. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, refer to section 1.2, List of Products.

Table 47.3 Recommended Operating Conditions (2)

Item	Symbol	Value
Decoupling capacitance to stabilize the internal voltage	C_{VCL}	4.7μF ± 30%*1

Note 1. Use a multilayer ceramic capacitor with a nominal capacitance of 4.7 μF, for which the sum of the capacitance tolerance and change in the capacitance under the usage conditions will be no greater than ±30%.

47.3 DC Characteristics

Table 47.4 DC Characteristics (1)

Conditions: 2.7 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	RIIC input pin (except for SMBus)	V_{IH}	0.7 × VCC	—	—	V	
		V_{IL}	—	—	0.3 × VCC		
		ΔV_T	0.05 × VCC	—	—		
	IRQ input pin, GPTW input pin, POEG input pin, TMR input pin, SCI input pin, RSCI input pin, RSPI input pin, CAC input pin, CANFD input pin, RTC input pin, USB pin, REMC input pin, ADTRG0# input pin*1, RES#, NMI, MD	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
		ΔV_T	0.1 × VCC	—	—		
	ADTRG0# input pin*2	V_{IH}	0.8 × AVCC0	—	—		
		V_{IL}	—	—	0.2 × AVCC0		
		ΔV_T	0.1 × AVCC0	—	—		
Input level voltage (except for schmitt trigger input pins)	EXTAL (external clock input)	V_{IH}	0.8 × VCC	—	—	V	
		V_{IL}	—	—	0.2 × VCC		
	RIIC input pin (SMBus)	V_{IH}	2.2	—	—		VCC = 3.6 to 5.5 V
			2.0	—	—		VCC = 2.7 to 3.6 V
		V_{IL}	—	—	0.8		VCC = 3.6 to 5.5 V
			—	—	0.5		VCC = 2.7 to 3.6 V
	P12 to P17, P20 to P27, P30 to P37, P50 to P55, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PH0 to PH3, PH6, PH7, PJ1, PJ3, PG7	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
	P03 to P07, P40 to P47, PJ6, PJ7	V_{IH}	0.8 × AVCC0	—	—		
		V_{IL}	—	—	0.2 × AVCC0		

Note 1. The ADTRG0# input pin is assigned to P16 and P25.
Note 2. The ADTRG0# input pin is assigned to P07.

Table 47.5 DC Characteristics (2)

Conditions: 1.6 V ≤ VCC < 2.7 V, 1.6 V ≤ AVCC0 < 2.7 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ input pin, GPTW input pin, POEG input pin, TMR input pin, SCI input pin, RSCI input pin, RSPI input pin, CAC input pin, CANFD input pin, RTC input pin, USB pin, REMC input pin, ADTRG0# input pin*1, RES#, NMI, MD	V_{IH}	0.8 × VCC	—	—	V	
		V_{IL}	—	—	0.2 × VCC		
		ΔV_T	0.01 × VCC	—	—		
	ADTRG0# input pin*2	V_{IH}	0.8 × AVCC0	—	—		
		V_{IL}	—	—	0.2 × AVCC0		
		ΔV_T	0.01 × AVCC0	—	—		
Input level voltage (except for schmitt trigger input pins)	EXTAL (external clock input)	V_{IH}	0.8 × VCC	—	—	V	
		V_{IL}	—	—	0.2 × VCC		
	P12 to P17, P20 to P27, P30 to P37, P50 to P55, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PH0 to PH3, PH6, PH7, PJ1, PJ3, PG7	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
	P03 to P07, P40 to P47, PJ6, PJ7	V_{IH}	0.8 × AVCC0	—	—		
		V_{IL}	—	—	0.2 × AVCC0		

Note 1. The ADTRG0# input pin is assigned to P16 and P25.
Note 2. The ADTRG0# input pin is assigned to P07.

Table 47.6 DC Characteristics (3)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	RES#, P35, PH6, PH7	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V, VCC
Three-state leakage current (off-state)	Ports for 5-V tolerant	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0 V, 5.8 V
	PJ6, PJ7, USB0_DM, USB0_DP		—	—	1.0		V_{in} = 0 V, VCC
	Other than ports for 5 V tolerant and PJ6, PJ7		—	—	0.2		V_{in} = 0 V, VCC
Input capacitance	All input pins (except for P35, USB0_DM, USB0_DP)	C_{in}	—	—	15	pF	V_{in} = 0 mV, f = 1 MHz, T_a = 25°C
	P35, USB0_DM, USB0_DP		—	—	30		

Table 47.7 DC Characteristics (4)

Conditions: 1.6 V ≤ VCC < 5.5 V, 1.6 V ≤ AVCC0 < 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input pull-up resistor	All ports (except for P35, PH6, PH7)	R_U	10	20	50	kΩ	V_{in} = 0 V

Table 47.8 DC Characteristics (5) (1/3)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item					Symbol	Typ. *4	Max.	Unit	Test Conditions
Supply current*1	High-speed operating mode	Normal operating mode	No peripheral operation*2	ICLK = 64 MHz	I_{CC}	4.4	—	mA	
				ICLK = 48 MHz		3.4	—		
				ICLK = 32 MHz		2.7	—		
				ICLK = 16 MHz		1.8	—		
				ICLK = 8 MHz		1.4	—		
				ICLK = 4 MHz		1.2	—		
			All peripheral operation: Normal*3	ICLK = 64 MHz		19.5	—		
				ICLK = 48 MHz		14.8	—		
				ICLK = 32 MHz		12.3	—		
				ICLK = 16 MHz		6.8	—		
				ICLK = 8 MHz		4.1	—		
				ICLK = 4 MHz		2.7	—		
			All peripheral operation: Max.*3	ICLK = 64 MHz		—	34.5		
				ICLK = 32 MHz		—	21.7		
		Sleep mode	No peripheral operation*2	ICLK = 64 MHz		2.4	—		
				ICLK = 48 MHz		1.9	—		
				ICLK = 32 MHz		1.6	—		
				ICLK = 16 MHz		1.3	—		
				ICLK = 8 MHz		1.1	—		
				ICLK = 4 MHz		1.0	—		
			All peripheral operation: Normal*3	ICLK = 64 MHz		11.0	—		
				ICLK = 48 MHz		8.4	—		
				ICLK = 32 MHz		7.8	—		
				ICLK = 16 MHz		4.5	—		
				ICLK = 8 MHz		2.8	—		
				ICLK = 4 MHz		2.0	—		
		Deep sleep mode	No peripheral operation*2	ICLK = 64 MHz		1.6	—		
				ICLK = 48 MHz		1.3	—		
				ICLK = 32 MHz		1.2	—		
				ICLK = 16 MHz		1.0	—		
				ICLK = 8 MHz		0.9	—		
				ICLK = 4 MHz		0.9	—		
			All peripheral operation: Normal*3	ICLK = 64 MHz		9.2	—		
				ICLK = 48 MHz		7.0	—		
				ICLK = 32 MHz		6.6	—		
				ICLK = 16 MHz		3.9	—		
				ICLK = 8 MHz		2.5	—		
				ICLK = 4 MHz		1.8	—		
		Increase during flash rewrite*5				2.6	—		

Table 47.8 DC Characteristics (5) (2/3)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

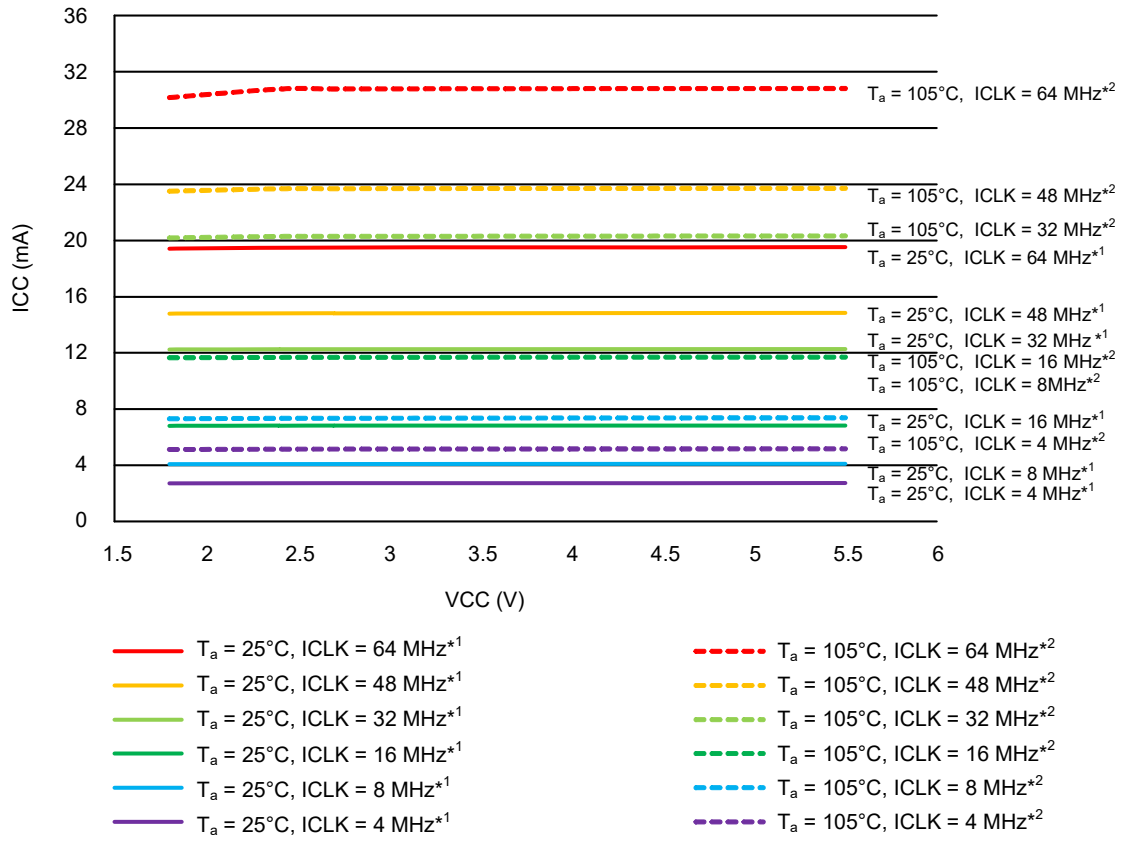
Item					Symbol	Typ. *4	Max.	Unit	Test Conditions
Supply current*1	Middle-speed operating mode	Normal operating mode	No peripheral operation*6	ICLK = 24 MHz	I_{CC}	1.8	—	mA	
				ICLK = 12 MHz		1.2	—		
				ICLK = 8 MHz		1.0	—		
				ICLK = 4 MHz		0.4	—		
				ICLK = 1 MHz		0.2	—		
			All peripheral operation: Normal*7	ICLK = 24 MHz		9.1	—		
				ICLK = 12 MHz		5.0	—		
				ICLK = 8 MHz		3.7	—		
				ICLK = 4 MHz		2.2	—		
				ICLK = 1 MHz		1.2	—		
			All peripheral operation: Max.*7	ICLK = 24 MHz		—	18.1		
		Sleep mode	No peripheral operation*6	ICLK = 24 MHz		1.1	—		
				ICLK = 12 MHz		0.8	—		
				ICLK = 8 MHz		0.7	—		
				ICLK = 4 MHz		0.2	—		
				ICLK = 1 MHz		0.2	—		
			All peripheral operation: Normal*7	ICLK = 24 MHz		5.8	—		
				ICLK = 12 MHz		3.3	—		
				ICLK = 8 MHz		2.7	—		
				ICLK = 4 MHz		1.7	—		
				ICLK = 1 MHz		1.1	—		
		Deep sleep mode	No peripheral operation*6	ICLK = 24 MHz		0.8	—		
				ICLK = 12 MHz		0.6	—		
				ICLK = 8 MHz		0.6	—		
				ICLK = 4 MHz		0.1	—		
				ICLK = 1 MHz		0.1	—		
			All peripheral operation: Normal*7	ICLK = 24 MHz		4.9	—		
				ICLK = 12 MHz		2.8	—		
				ICLK = 8 MHz		2.3	—		
				ICLK = 4 MHz		1.5	—		
				ICLK = 1 MHz		1.0	—		
		Increase during flash rewrite*5				2.1	—		

Table 47.8 DC Characteristics (5) (3/3)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item					Symbol	Typ. *4	Max.	Unit	Test Conditions
Supply current*1	Middle-speed operating mode 2	Normal operating mode	No peripheral operation*8	ICLK = 1 MHz	I_{CC}	160	—	μA	
			All peripheral operation: Normal*9	ICLK = 1 MHz		1170	—		
			All peripheral operation: Max.*9	ICLK = 1 MHz		—	4520		
		Sleep mode	No peripheral operation*8	ICLK = 1 MHz		120	—		
			All peripheral operation: Normal*9	ICLK = 1 MHz		1030	—		
		Deep sleep mode	No peripheral operation*8	ICLK = 1 MHz		110	—		
			All peripheral operation: Normal*9	ICLK = 1 MHz		990	—		
		Increase during flash rewrite*5				1420	—		
	Low-speed operating mode	Normal operating mode	No peripheral operation*10	ICLK = 32.768 kHz		4.3	—		
			All peripheral operation: Normal*11, *12	ICLK = 32.768 kHz		13.9	—		
			All peripheral operation: Max. *11, *12	ICLK = 32.768 kHz		—	1500		
		Sleep mode	No peripheral operation*10	ICLK = 32.768 kHz		2.9	—		
			All peripheral operation: Normal*11	ICLK = 32.768 kHz		9.0	—		
		Deep sleep mode	No peripheral operation*10	ICLK = 32.768 kHz		2.5	—		
			All peripheral operation: Normal*11	ICLK = 32.768 kHz		7.8	—		

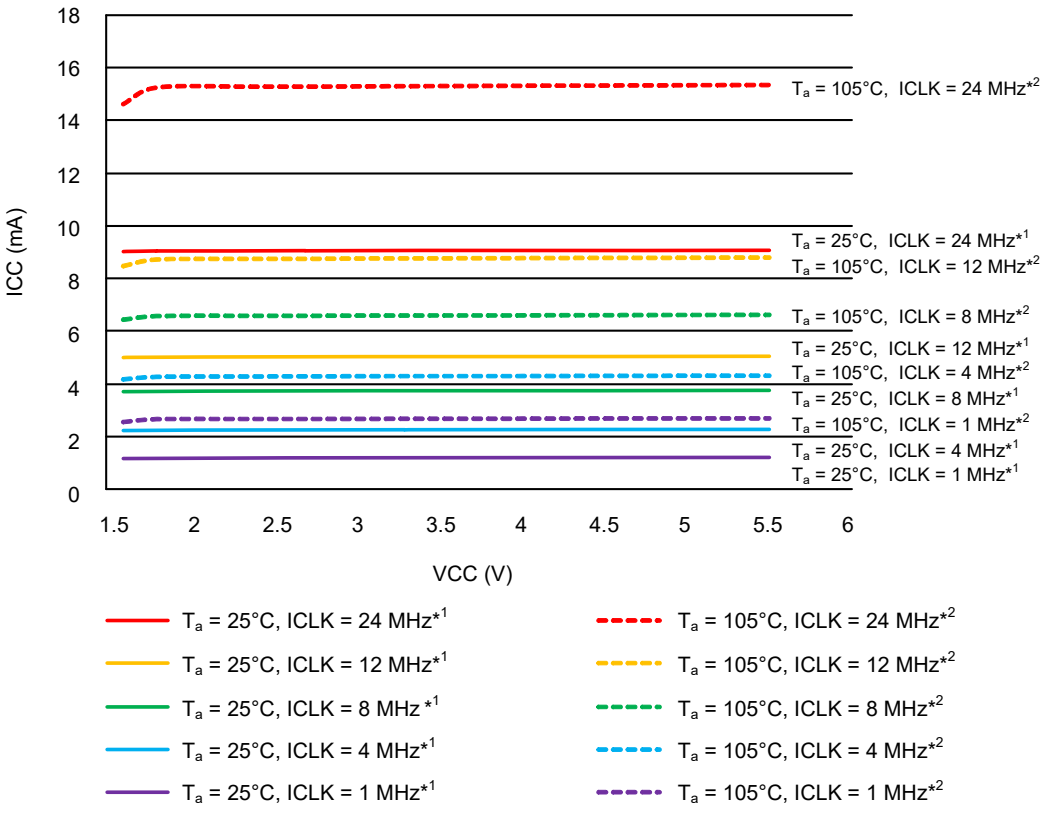
Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is PLL. FCLK and PCLK are set to divided by 64.
Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is PLL. FCLK, PCLKA, and PCLKD are set to the same frequency as ICLK and PCLKB is set to divided by 2 when ICLK is 64 MHz or 48 MHz. FCLK and PCLK are set to the same frequency as ICLK when ICLK is 32 MHz or less.
Note 4. Values when VCC = 3.3 V.
Note 5. This is the increase for programming or erasure of the ROM or E2 DataFlash during program execution.
Note 6. Clock supply to the peripheral function is stopped. The clock source is PLL when ICLK is 24 MHz, HOCO when ICLK is 8 MHz, and LOCO otherwise. FCLK and PCLK are set to divided by 64.
Note 7. Clocks are supplied to the peripheral functions. The clock source is PLL when ICLK is 24 MHz, HOCO when ICLK is (MHz, and LOCO otherwise. FCLK and PCLK are set to the same frequency as ICLK.
Note 8. Clock supply to the peripheral function is stopped. The clock source is LOCO when ICLK is 1 MHz, FCLK and PCLK are set to divided by 64.
Note 9. Clocks are supplied to the peripheral functions. The clock source is LOCO when ICLK is 1 MHz, FCLK and PCLK are set to the same frequency as ICLK.
Note 10. Clock supply to the peripheral functions is stopped. The clock source is the sub-clock oscillator. FCLK and PCLK are set to divided by 64.
Note 11. Clocks are supplied to the peripheral functions. The clock source is the sub-clock oscillator. FCLK and PCLK are set to the same frequency as ICLK.
Note 12. Values when the MSTPCRA.MSTPA17 bit (12-bit A/D converter module stop bit) is set to “transition to the module stop state is made”.



Note 1. All peripheral operation is normal. This does not include BGO operation. Average value of the tested middle samples during product evaluation.

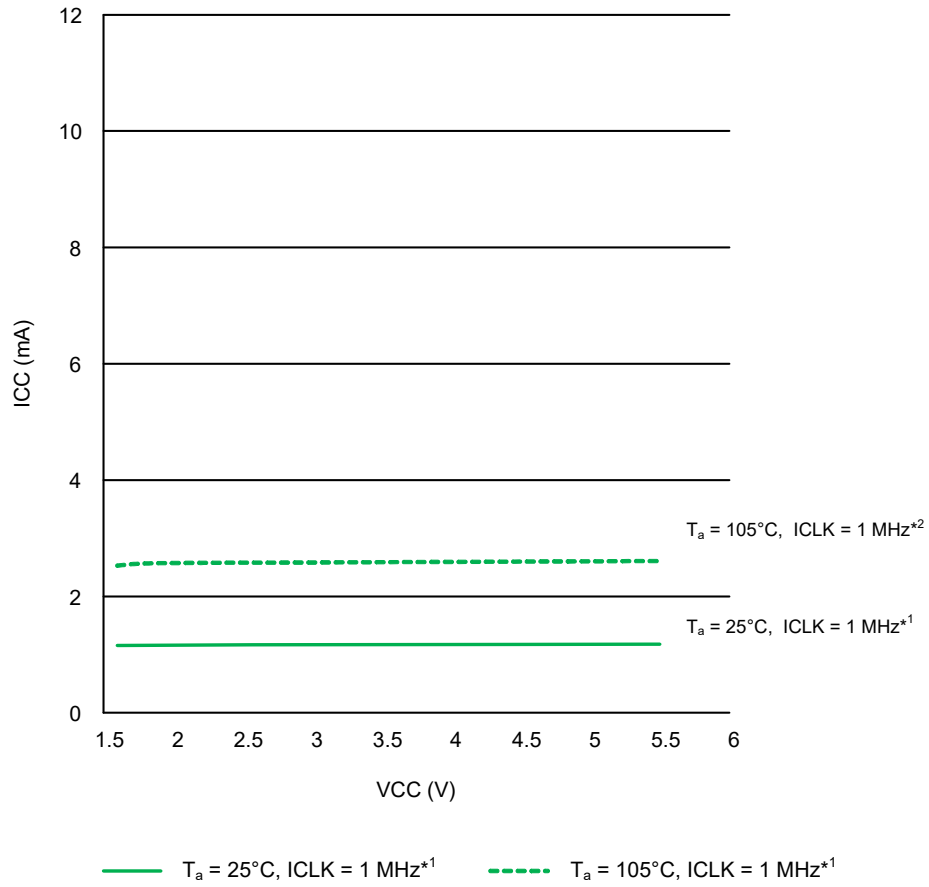
Note 2. All peripheral operation is maximum. This does not include BGO operation. Average value of the tested upper-limit samples during product evaluation.

Figure 47.1 Voltage Dependency in High-Speed Operating Mode



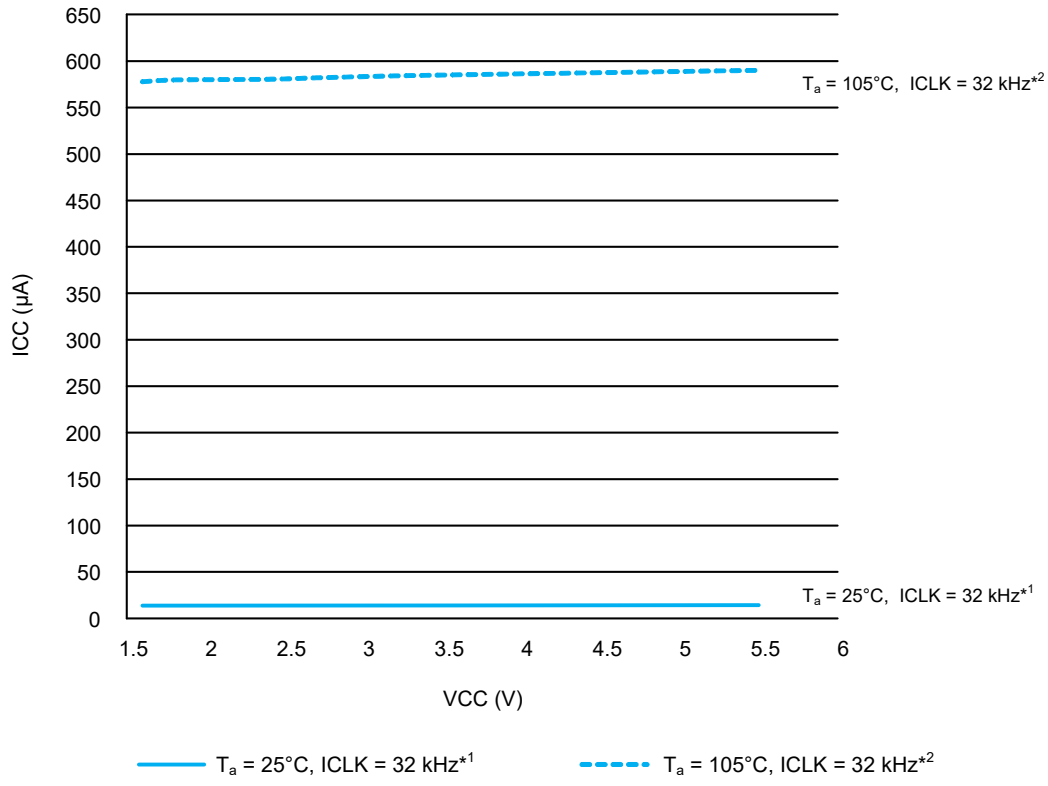
Note 1. All peripheral operation is normal. This does not include BGO operation. Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum. This does not include BGO operation. Average value of the tested upper-limit samples during product evaluation.

Figure 47.2 Voltage Dependency in Middle-Speed Operating Mode



Note 1. All peripheral operation is normal. This does not include BGO operation. Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum. This does not include BGO operation. Average value of the tested upper-limit samples during product evaluation.

Figure 47.3 Voltage Dependency in Middle-Speed Operating Mode 2



Note 1. All peripheral operation is normal. This does not include BGO operation. Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum. This does not include BGO operation. Average value of the tested upper-limit samples during product evaluation.

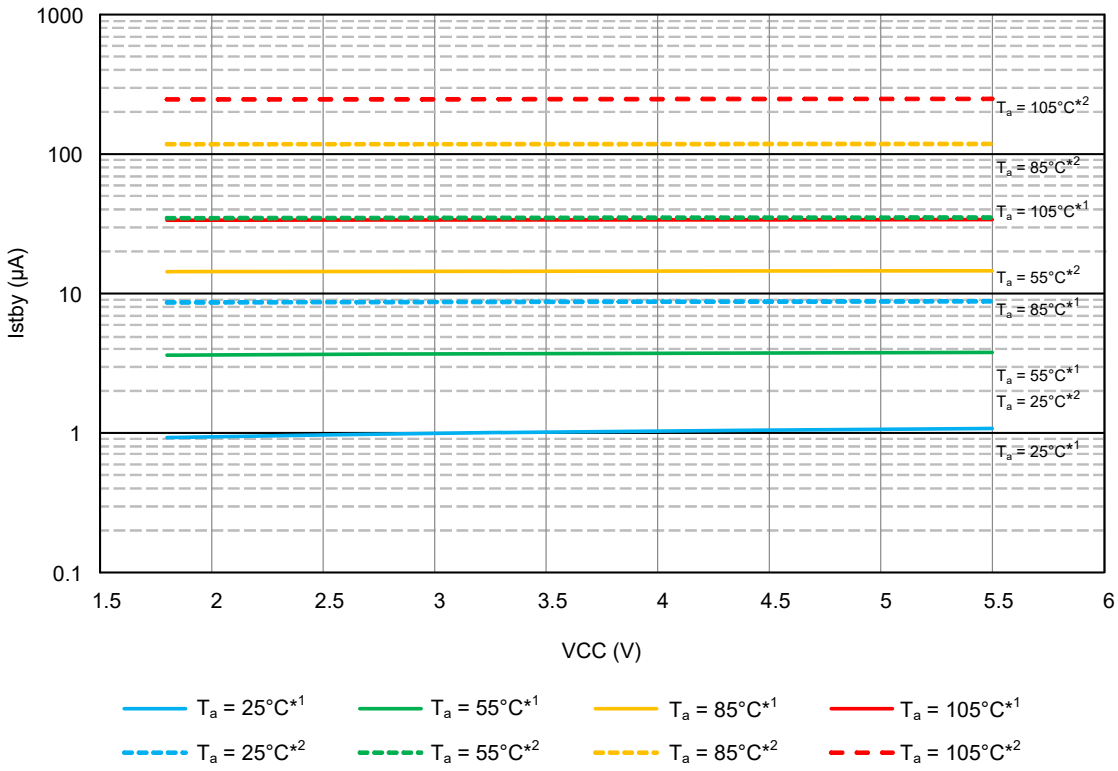
Figure 47.4 Voltage Dependency in Low-Speed Operating Mode

Table 47.9 DC Characteristics (6)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

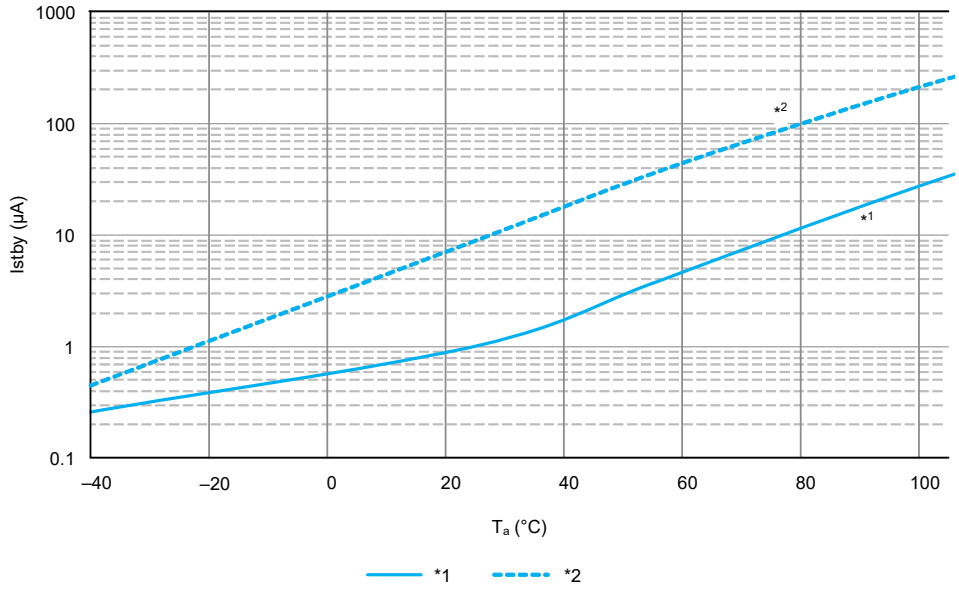
Item				Symbol	Typ.*3	Max.	Unit	Test Conditions
Supply current*1	Software standby mode*2	RAM Power Saving disabled	T_a = 25°C	I_{CC}	1.01	19.54	µA	
			T_a = 55°C		3.71	73.97		
			T_a = 85°C		14.44	229.58		
			T_a = 105°C		33.81	470.46		
	Increment for RTC operation*4				0.99	—		SOMCR.SODRV[1:0] set to drive capacity for standard CL
					0.55	—		SOMCR.SODRV[1:0] set to high drive capacity for low CL
					0.32	—		SOMCR.SODRV[1:0] set to middle drive capacity for low CL
					0.22	—		SOMCR.SODRV[1:0] set to low drive capacity for low CL
	Increment for low-power timer operation				0.33	—		LPTCR1.LPCNTCKSEL set to IWDT-dedicated on-chip oscillator
					16.00	—		LPTCR1.LPCNTCKSEL 2 set to Low-speed on-chip oscillator
	Increment for independent watchdog timer operation				0.32	—		
	Increment for REMC operation*4				0.98	—		REMCON1.CSRC[3:0] set to Sub-clock SOMCR.SODRV[1:0] set to drive capacity for standard CL
					0.60	—		REMCON1.CSRC[3:0] set to Sub-clock SOMCR.SODRV[1:0] set to high drive capacity for low CL
					0.42	—		REMCON1.CSRC[3:0] set to Sub-clock SOMCR.SODRV[1:0] set to middle drive capacity for low CL
					0.31	—		REMCON1.CSRC[3:0] set to Sub-clock SOMCR.SODRV[1:0] set to low drive capacity for low CL
					0.29	—		REMCON1.CSRC[3:0] set to IWDT-dedicated on-chip oscillator

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. The IWDT, LVD, and CMPB are stopped.
Note 3. VCC = 3.3 V.
Note 4. Includes the oscillation circuit.



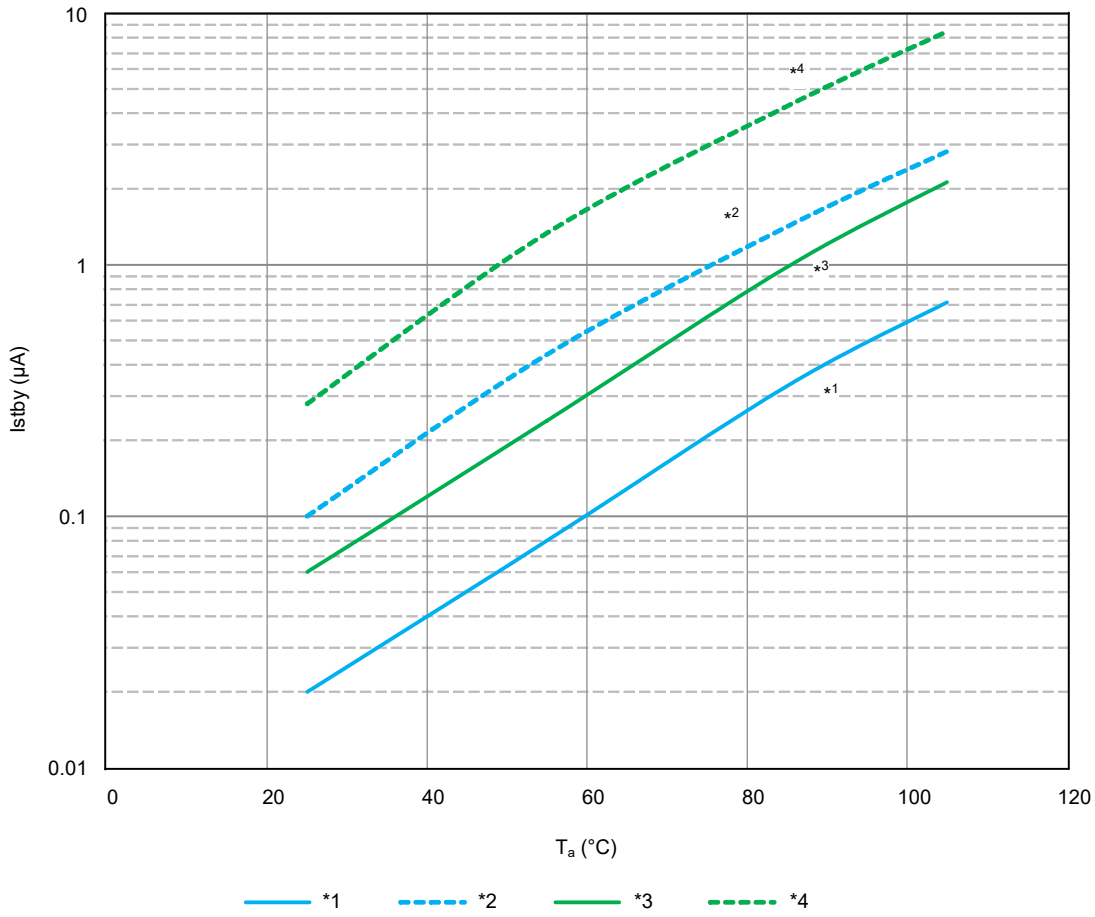
Note 1. Average value of the tested middle samples during product evaluation.
Note 2. Average value of the tested upper-limit samples during product evaluation.

Figure 47.5 Voltage Dependency in Software Standby Mode (Reference Data with RAM Power Saving Disabled)



Note 1. Average value of the tested middle samples during product evaluation.
Note 2. Average value of the tested upper-limit samples during product evaluation.

Figure 47.6 Temperature Dependency in Software Standby Mode (Reference Data with RAM Power Saving Disabled)



Note 1. Average difference of middle samples tested during product evaluation with and without 32 Kbytes of RAM power savings.
Note 2. Average difference of upper-limt samples tested during product evaluation with and without 32 Kbytes of RAM power savings.
Note 3. Average difference of middle samples tested during product evaluation with and without 96 Kbytes of RAM power savings.
Note 4. Average difference of upper-limt samples tested during product evaluation with and without 96 Kbytes of RAM power savings.

Figure 47.7 Temperature Dependency in Software Standby Mode (Reference Data with RAM Power Saving Enabled)

Table 47.10 DC Characteristics (7)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.*4	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion (at high-speed conversion)	I_{AVCC}	—	0.7	1.4	mA	
	During A/D conversion (at low-speed conversion)		—	0.3	0.7		
	During D/A conversion (per channel)*1		—	—	0.5		
	Waiting for A/D and D/A conversion		—	—	2.0	µA	
Reference power supply current	During A/D conversion (at high-speed conversion)	I_{REFH0}	—	53	122	µA	
	Waiting for A/D conversion		—	—	0.3	µA	
LVD0	—	I_{LVD}	—	0.04	—	µA	
LVD1, LVD2	Per channel		—	0.12	—	µA	
Temperature sensor*3	—	I_{TEMP}	—	120	—	µA	
Comparator B operating current*3	Window function enabled	I_{CMP}*2	—	7.5	12.5	µA	
	Comparator high-speed mode (per channel)		—	5.0	10.0	µA	
	Comparator low-speed mode (per channel)		—	1.5	3.0	µA	
USB operating current*3	During USB communication operation under the following settings and conditions • Host controller operation is set to full-speed mode Bulk OUT transfer (64 bytes) × 1, bulk IN transfer (64 bytes) × 1 • Connect peripheral devices via a 1-meter USB cable from the USB port.	I_{USBH}*5	—	3.5	—	µA	
	During USB communication operation under the following settings and conditions • Function controller operation is set to full-speed mode Bulk OUT transfer (64 bytes) × 1, bulk IN transfer (64 bytes) × 1 • Connect the host device via a 1-meter USB cable from the USB port.	I_{USBF}*5	—	4.0	—	mA	
	During suspended state under the following setting and conditions • Function controller operation is set to full-speed mode (pull up the USB0_DP pin) • Software standby mode • Connect the host device via a 1-meter USB cable from the USB port.	I_{SUSP}*6	—	160	—	µA	
RSIP operating current*3	During a self-test	I_{RSIP}*7	—	—	11.3	mA	PCLKB = 32 MHz
	After release from the module-stop state		—	3.6	—	mA	

Note 1. The value for the D/A converter is the value of the power supply current, including the reference current.
Note 2. The values are only for the current drawn by the comparator B module.
Note 3. The values are for the current drawn by the power supply (VCC).
Note 4. The values apply when VCC = AVCC0 = 3.3 V.
Note 5. The value is only for the current drawn by the USB module.
Note 6. The value includes the current supplied from the pull-up resistor of the USB0_DP pin to the pull-down resistor of the host device, in addition to the current drawn by this MCU in the suspended state.
Note 7. The values are only for the current drawn by the RSIP module.

Table 47.11 DC Characteristics (8)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM standby voltage	V_{RAM}	1.6	—	—	V	

Table 47.12 DC Characteristics (9)

Conditions: 0 V ≤ VCC ≤ 5.5 V, 0 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power-on VCC rising gradient	At normal startup*1	SrVCC	0.02	—	20	ms/V	
	During fast startup time*2		0.02	—	2		
	Voltage monitoring 0 reset enabled at startup*3, *4		0.02	—	—		

Note 1. When OFS1.(FASTSTUP, LVDAS) = 11b.
Note 2. When OFS1.(FASTSTUP, LVDAS) = 01b.
Note 3. When OFS1.LVDAS = 0.
Note 4. Turn on the power supply voltage according to the normal startup rising gradient because the register settings set by OFS1 are not read in boot mode.

Table 47.13 DC Characteristics (10)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

The ripple voltage must meet the allowable ripple frequency $f_{r\,(VCC)}$ within the range between the VCC upper limit and lower limit. When VCC change exceeds VCC ±10%, the allowable voltage change rising/falling gradient dt/dVCC must be met.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Allowable ripple frequency	$f_{r\,(VCC)}$	—	—	10	kHz	Figure 47.8 $V_{r\,(VCC)}$ ≤ 0.2 × VCC
		—	—	1	MHz	Figure 47.8 $V_{r\,(VCC)}$ ≤ 0.08 × VCC
		—	—	10	MHz	Figure 47.8 $V_{r\,(VCC)}$ ≤ 0.06 × VCC
Allowable voltage change rising/falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

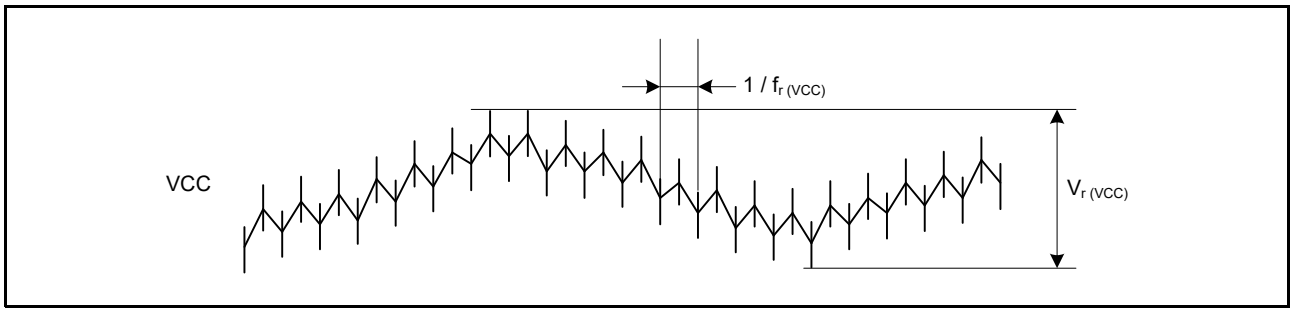


Figure 47.8 Ripple Waveform

Table 47.14 Permissible Output Currents (1)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +85°C

Item		Symbol	Max.	Unit
Permissible output low current (average value per pin)	P03 to P07, P40 to P47, PJ6, PJ7	I_{OL}	8.0	mA
	Ports other than above		8.0	
Permissible output low current (maximum value per pin)	P03 to P07, P40 to P47, PJ6, PJ7		8.0	
	Ports other than above		8.0	
Permissible output low current	Total of P03 to P07, P40 to P47, PJ6, PJ7	ΣI_{OL}	40	
	Total of P12 to P17, P20 to P27, P30 to P37, PG7, PH2, PH3, PH6, PH7, PJ1, PJ3		40	
	Total of P50 to P55, PB0 to PB7, PC0 to PC7, PH0, PH1		40	
	Total of PA0 to PA7, PD0 to PD7, PE0 to PE7		40	
	Total of all output pins		80	
Permissible output high current (average value per pin)	P03 to P07, P40 to P47, PJ6, PJ7	I_{OH}	–4.0	
	Ports other than above		–4.0	
Permissible output high current (maximum value per pin)	P03 to P07, P40 to P47, PJ6, PJ7		–4.0	
	Ports other than above		–4.0	
Permissible output high current	Total of P03 to P07, P40 to P47, PJ6, PJ7	ΣI_{OH}	–40	
	Total of P12 to P17, P20 to P27, P30 to P37, PG7, PH2, PH3, PH6, PH7, PJ1, PJ3		–40	
	Total of P50 to P55, PB0 to PB7, PC0 to PC7, PH0, PH1		–40	
	Total of PA0 to PA7, PD0 to PD7, PE0 to PE7		–40	
	Total of all output pins		–80	

Note: Do not exceed the permissible total supply current.

Table 47.15 Permissible Output Currents (2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Max.	Unit
Permissible output low current (average value per pin)	P03 to P07, P40 to P47, PJ6, PJ7	I_{OL}	8.0	mA
	Ports other than above		8.0	
Permissible output low current (maximum value per pin)	P03 to P07, P40 to P47, PJ6, PJ7		8.0	
	Ports other than above		8.0	
Permissible output low current	Total of P03 to P07, P40 to P47, PJ6, PJ7	ΣI_{OL}	30	
	Total of P12 to P17, P20 to P27, P30 to P37, PG7, PH2, PH3, PH6, PH7, PJ1, PJ3		30	
	Total of P50 to P55, PB0 to PB7, PC0 to PC7, PH0, PH1		30	
	Total of PA0 to PA7, PD0 to PD7, PE0 to PE7		30	
	Total of all output pins		60	
Permissible output high current (average value per pin)	P03 to P07, P40 to P47, PJ6, PJ7	I_{OH}	–4.0	
	Ports other than above		–4.0	
Permissible output high current (maximum value per pin)	P03 to P07, P40 to P47, PJ6, PJ7		–4.0	
	Ports other than above		–4.0	
Permissible output high current	Total of P03 to P07, P40 to P47, PJ6, PJ7	ΣI_{OH}	–30	
	Total of P12 to P17, P20 to P27, P30 to P37, PG7, PH2, PH3, PH6, PH7, PJ1, PJ3		–30	
	Total of P50 to P55, PB0 to PB7, PC0 to PC7, PH0, PH1		–30	
	Total of PA0 to PA7, PD0 to PD7, PE0 to PE7		–30	
	Total of all output pins		–60	

Note: Do not exceed the permissible total supply current.

Table 47.16 Output Values of Voltage (1)

Conditions: 1.6 V ≤ VCC < 1.8 V, 1.6 V ≤ AVCC0 < 1.8 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports (except for RIIC)		V_{OL}	—	0.3	V	I_{OL} = 0.3 mA
Output high	All output ports	P03 to P07, P40 to P47, PJ6, PJ7	V_{OH}	AVCC0 – 0.3	—	V	I_{OH} = –0.5 mA
		Ports other than above		VCC – 0.3	—		

Table 47.17 Output Values of Voltage (2)

Conditions: 1.8 V ≤ VCC < 2.7 V, 1.8 V ≤ AVCC0 < 2.7 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports (except for RIIC)		V_{OL}	—	0.3	V	I_{OL} = 1.0 mA
Output high	All output ports	P03 to P07, P40 to P47, PJ6, PJ7	V_{OH}	AVCC0 – 0.3	—	V	I_{OH} = –0.5 mA
		Ports other than above		VCC – 0.3	—		

Table 47.18 Output Values of Voltage (3)

Conditions: 2.7 V ≤ VCC < 4.0 V, 2.7 V ≤ AVCC0 < 4.0 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports (except for RIIC)		V_{OL}	—	0.5	V	I_{OL} = 2.0 mA
	RIIC pins			—	0.6		I_{OL} = 6.0 mA
Output high	All output ports	P03 to P07, P40 to P47, PJ6, PJ7	V_{OH}	AVCC0 – 0.5	—	V	I_{OH} = –1.0 mA
		Ports other than above		VCC – 0.5	—		

Table 47.19 Output Values of Voltage (4)

Conditions: 4.0 V ≤ VCC ≤ 5.5 V, 4.0 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports (except for RIIC)		V_{OL}	—	0.8	V	I_{OL} = 4.0 mA
	RIIC pins			—	0.6		I_{OL} = 6.0 mA
Output high	All output ports	P03 to P07, P40 to P47, PJ6, PJ7	V_{OH}	AVCC0 – 0.8	—	V	I_{OH} = –2.0 mA
		Ports other than above		VCC – 0.8	—		

Table 47.20 Thermal Resistance Value (Reference Values)

Item	Package	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Thermal resistance	100-pin LFQFP(PLQP0100KB-B)	θja	—	—	47.9	°C/W	JESD51-2 and JESD51-7 compliant
	80-pin LFQFP (PLQP0080KB-B)		—	—	46.0		
	64-pin LFQFP (PLQP0064KB-C)		—	—	44.8		
	48-pin LFQFP (PLQP0048KB-B)		—	—	53.6		
	48-pin HWQFN (PWQN0048KC-A)		—	—	20.0*1		
	100-pin LFQFP(PLQP0100KB-B)	Ψjt	—	—	0.81		
	80-pin LFQFP (PLQP0080KB-B)		—	—	0.81		
	64-pin LFQFP (PLQP0064KB-C)		—	—	0.81		
	48-pin LFQFP (PLQP0048KB-B)		—	—	1.30		
	48-pin HWQFN (PWQN0048KC-A)		—	—	0.11*1		

Note: The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, refer to the JEDEC standards.

Note 1. This value applies when the exposed die pad for this purpose is connected to VSS.

47.4 Normal I/O Pin Output Characteristics

Table 47.21 Normal I/O Pin VOH Voltage Characteristics (Reference Values)

Conditions: VCC = AVCC0 = 2.0 V, VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high level voltage	All output pins	V_{OH}	—	VCC – 0.05	—	V	I_{OH} = –0.5 mA
			—	VCC – 0.09	—		I_{OH} = –1.0 mA
			—	VCC – 0.20	—		I_{OH} = –2.0 mA
			—	VCC – 0.49	—		I_{OH} = –4.0 mA

Table 47.22 Normal I/O Pin VOH Voltage Characteristics (Reference Values)

Conditions: VCC = AVCC0 = 3.3 V, VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high level voltage	All output pins	V_{OH}	—	VCC – 0.02	—	V	I_{OH} = –0.5 mA
			—	VCC – 0.05	—		I_{OH} = –1.0 mA
			—	VCC – 0.10	—		I_{OH} = –2.0 mA
			—	VCC – 0.22	—		I_{OH} = –4.0 mA

Table 47.23 Normal I/O Pin VOH Voltage Characteristics (Reference Values)

Conditions: VCC = AVCC0 = 5.0 V, VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high level voltage	All output pins	V_{OH}	—	VCC – 0.02	—	V	I_{OH} = –0.5 mA
			—	VCC – 0.04	—		I_{OH} = –1.0 mA
			—	VCC – 0.08	—		I_{OH} = –2.0 mA
			—	VCC – 0.15	—		I_{OH} = –4.0 mA

Table 47.24 Normal I/O Pin VOL Voltage Characteristics (Reference Values)

Conditions: VCC = AVCC0 = 2.0 V, VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	0.02	—	V	I_{OL} = 0.5 mA
			—	0.04	—		I_{OL} = 1.0 mA
			—	0.08	—		I_{OL} = 2.0 mA
			—	0.17	—		I_{OL} = 4.0 mA
			—	0.43	—		I_{OL} = 8.0 mA

Table 47.25 Normal I/O Pin VOL Voltage Characteristics (Reference Values)

Conditions: VCC = AVCC0 = 3.3 V, VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	0.01	—	V	I_{OL} = 0.5 mA
			—	0.02	—		I_{OL} = 1.0 mA
			—	0.04	—		I_{OL} = 2.0 mA
			—	0.08	—		I_{OL} = 4.0 mA
			—	0.17	—		I_{OL} = 8.0 mA

Table 47.26 Normal I/O Pin VOL Voltage Characteristics (Reference Values)

Conditions: VCC = AVCC0 = 5.0 V, VSS = AVSS0 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	0.01	—	V	I_{OL} = 0.5 mA
			—	0.01	—		I_{OL} = 1.0 mA
			—	0.03	—		I_{OL} = 2.0 mA
			—	0.06	—		I_{OL} = 4.0 mA
			—	0.12	—		I_{OL} = 8.0 mA

47.5 AC Characteristics

47.5.1 Clock Timing

Table 47.27 Operating Frequency Value (High-Speed Operating Mode)

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.8 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC				Unit
			1.8 V ≤ VCC < 2.4 V*5	1.8 V ≤ VCC < 2.4 V*6	2.4 V ≤ VCC ≤ 5.5 V	When USB is in Use, 3.0 V ≤ VCC ≤ 3.6 V	
Maximum operating frequency *4	System clock (ICLK)	f_{max}	16	48	64	64	MHz
	FlashIF clock (FCLK) *1, *2		16	48	64	64	
	Peripheral module clock (PCLKA)		16	48	64	64	
	Peripheral module clock (PCLKB)		16	32	32	32	
	Peripheral module clock (PCLKD)*3		16	48	64	64	
	USB clock (UCLK)	f_{usb}	—	—	—	48	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the A/D converter is in use.
Note 4. The maximum operating frequency does not include HOCO error or PLL jitter. See Table 47.34, HOCO Clock Timing, Table 47.35, PLL Clock Timing and Table 47.36, PLL2 Clock Timing.
Note 5. This is applicable when the RSIP module is to be used.
Note 6. The RSIP module cannot be used. Do not make the settings for release from the module-stop state.

Table 47.28 Operating Frequency Value (Middle-Speed Operating Mode)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC					Unit
			1.6 V ≤ VCC < 1.8 V*6	1.8 V ≤ VCC < 2.4 V*5	1.8 V ≤ VCC < 2.4 V*6	2.4V ≤ VCC ≤ 5.5 V	When USB is in Use, 3.0 V ≤ VCC ≤ 3.6 V	
Maximum operating frequency *4	System clock (ICLK)	f_{max}	4	16	24	24	24	MHz
	FlashIF clock (FCLK) *1, *2		4	16	24	24	24	
	Peripheral module clock (PCLKA)		4	16	24	24	24	
	Peripheral module clock (PCLKB)		4	16	24	24	24	
	Peripheral module clock (PCLKD)*3		4	16	24	24	24	
	USB clock (UCLK)	f_{usb}	—	—	—	—	48	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the A/D converter is in use.
Note 4. The maximum operating frequency does not include HOCO error or PLL jitter. See Table 47.34, HOCO Clock Timing, Table 47.35, PLL Clock Timing and Table 47.36, PLL2 Clock Timing.
Note 5. This is applicable when the RSIP module is to be used.
Note 6. The RSIP module cannot be used. Do not make the settings for release from the module-stop state.

Table 47.29 Operating Frequency Value (Middle-Speed Operating Mode 2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC		Unit
			1.6 V ≤ VCC < 1.8 V*4	1.8 V ≤ VCC ≤ 5.5 V	
Maximum operating frequency*4	System clock (ICLK)	f_{max}	1	1	MHz
	FlashIF clock (FCLK)*1, *2		1	1	
	Peripheral module clock (PCLKA)		1	1	
	Peripheral module clock (PCLKB)		1	1	
	Peripheral module clock (PCLKD)*3		1	1	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory.
Note 2. The frequency accuracy of FCLK should be ±3.5%.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the A/D converter is in use.
Note 4. The RSIP module cannot be used. Do not make the settings for release from the module-stop state.

Table 47.30 Operating Frequency Value (Low-Speed Operating Mode)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC	Unit
			1.6 V ≤ VCC ≤ 5.5 V	
Maximum operating frequency	System clock (ICLK)	f_{max}	32.768	kHz
	FlashIF clock (FCLK)*1		32.768	
	Peripheral module clock (PCLKA)		32.768	
	Peripheral module clock (PCLKB)		32.768	
	Peripheral module clock (PCLKD)*2		32.768	

Note 1. Programming and erasing the flash memory is impossible.
Note 2. The A/D converter cannot be used.

Table 47.31 EXTAL Clock Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	1.8 V ≤ VCC ≤ 5.5 V	t_{Xcyc}	50	—	—	ns	Figure 47.9
	1.6 V ≤ VCC < 1.8 V		250	—	—	ns	
EXTAL external clock input high pulse width	1.8 V ≤ VCC ≤ 5.5 V	t_{XH}	20	—	—	ns	
	1.6 V ≤ VCC < 1.8 V		120	—	—	ns	
EXTAL external clock input low pulse width	1.8 V ≤ VCC ≤ 5.5 V	t_{XL}	20	—	—	ns	
	1.6 V ≤ VCC < 1.8 V		120	—	—	ns	
EXTAL external clock rise time		t_{Xr}	—	—	5	ns	
EXTAL external clock fall time		t_{Xf}	—	—	5	ns	
EXTAL external clock input wait time*1		t_{XWT}	0.5	—	—	μs	

Note 1. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating).

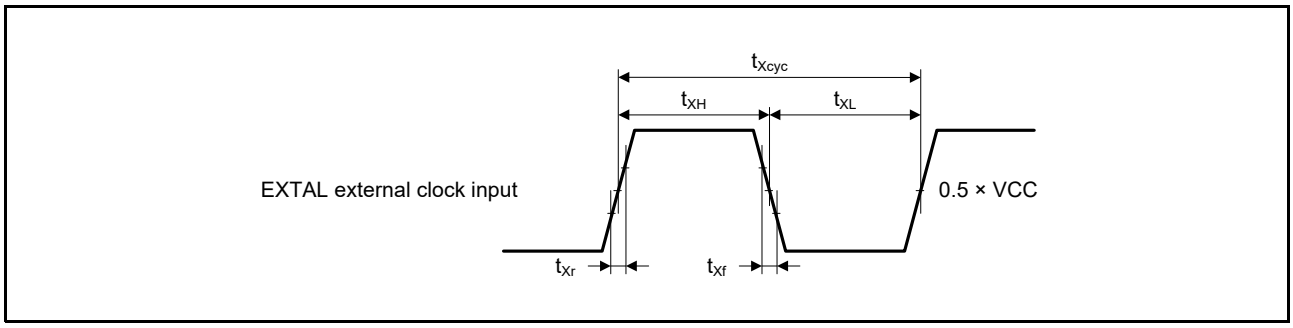


Figure 47.9 EXTAL External Clock Input Timing

Table 47.32 Main Clock Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Main clock oscillator oscillation frequency	f_{MAIN}	1	—	20	MHz	
Main clock oscillation stabilization time (crystal)*1	$t_{MAINOSC}$	—	3	—	ms	Figure 47.10
Main clock oscillation stabilization time (ceramic resonator)*1	$t_{MAINOSC}$	—	50	—	μs	

Note 1. Reference values when an 8-MHz resonator is used.
When specifying the main clock oscillator stabilization time, set the MOSCWTCR register with a stabilization time value that is equal to or greater than the resonator-manufacturer-recommended value.
After changing the setting of the MOSCCR.MOSTP bit so that the main clock oscillator operates, read the OSCOVFSR.MOOVF flag to confirm that is has become 1, and then start using the main clock.

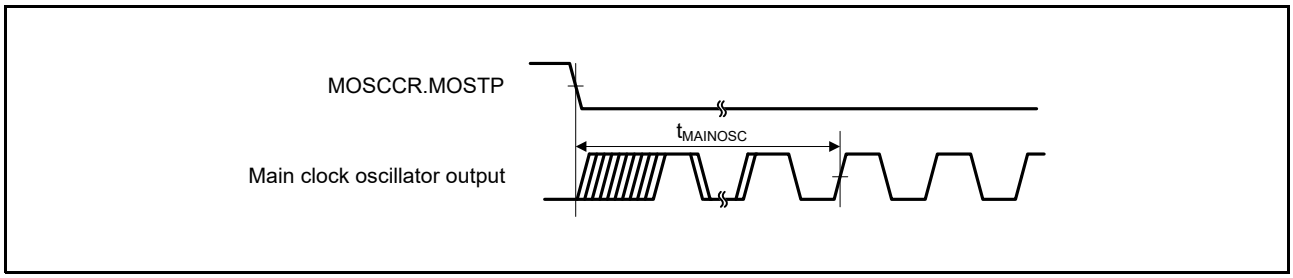


Figure 47.10 Main Clock Oscillation Start Timing

Table 47.33 LOCO and IWDT-Dedicated Low-Speed Clock Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LOCO clock oscillation frequency	f_{LOCO}	3.44	4.0	4.56	MHz	
LOCO clock oscillation frequency error	Δf_{LOCO}	—	—	±14	%	
LOCO clock oscillation stabilization time	t_{LOCO}	—	—	0.5	µs	Figure 47.11
IWDT-dedicated clock oscillation frequency	f_{ILOCO}	12.75	15	17.25	kHz	
IWDT-dedicated clock oscillation frequency error	Δf_{ILOCO}	—	—	±15	%	
IWDT-dedicated clock oscillation stabilization time	t_{ILOCO}	—	—	80	µs	Figure 47.12

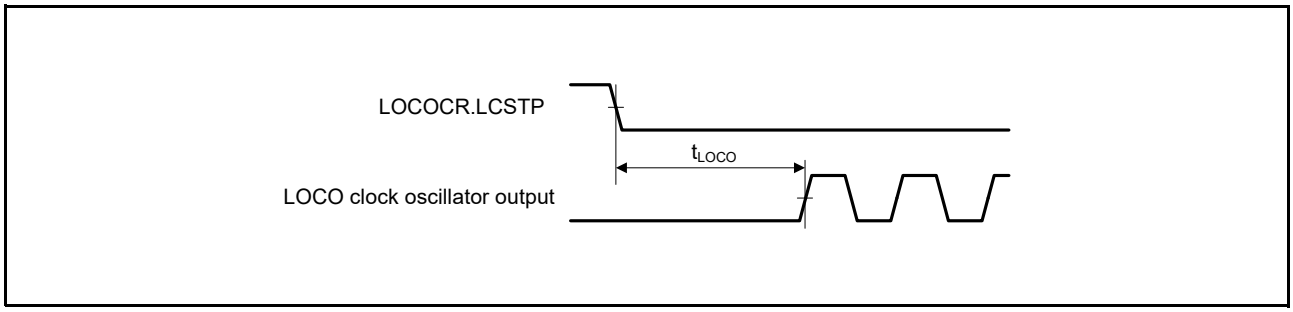


Figure 47.11 LOCO Clock Oscillation Start Timing

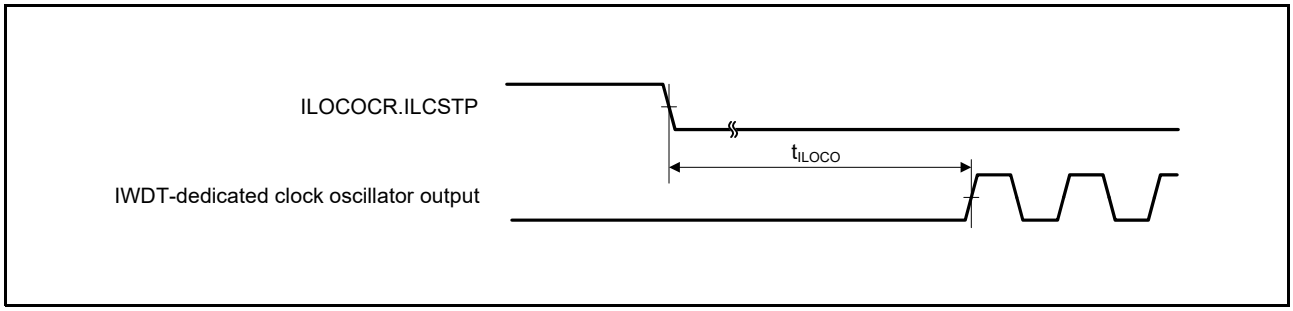


Figure 47.12 IWDT-Dedicated Clock Oscillation Start Timing

Table 47.34 HOCO Clock Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HOCO clock oscillation frequency	f_{HOCO}	23.76 (–1.0%)	24	24.24 (+1.0%)	MHz	T_a = –40 to +105°C
		31.68 (–1.0%)	32	32.32 (+1.0%)		
		47.52 (–1.0%)	48	48.48 (+1.0%)		
		63.36 (–1.0%)	64	64.64 (+1.0%)		
HOCO oscillation frequency error	Δf_{HOCO}	—	—	±1.0	%	T_a = –40 to +105°C
HOCO clock oscillation stabilization time	t_{HOCO}	—	—	4.95	μs	Figure 47.14

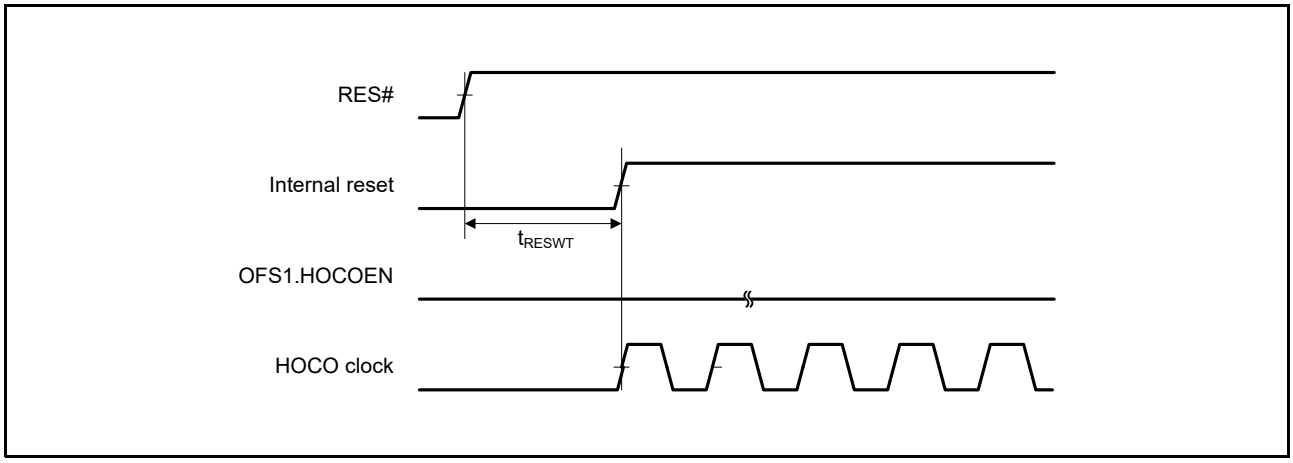


Figure 47.13 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0)

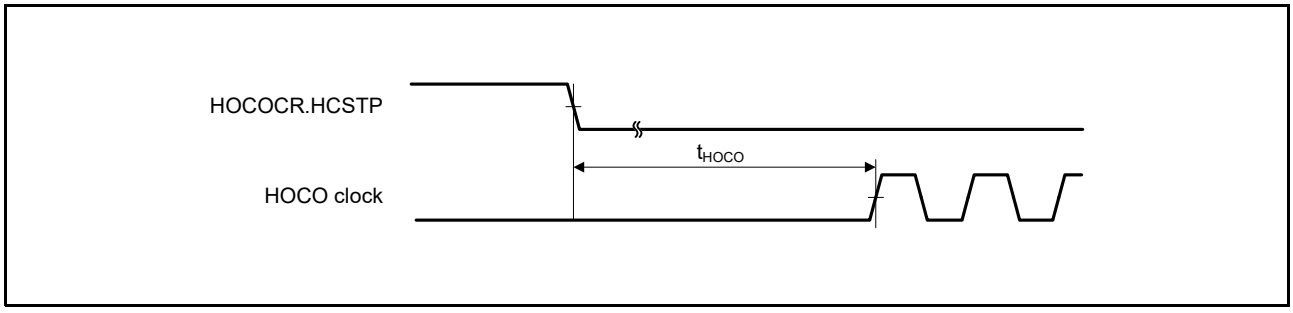


Figure 47.14 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit)

Table 47.35 PLL Clock Timing

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL input frequency	f_{PLLIN}	4	—	12.5	MHz	
PLL circuit oscillation frequency	f_{PLL}	24	—	64	MHz	
PLL clock oscillation stabilization time	t_{PLL}	—	—	81.4	µs	Figure 47.15
PLL free-running oscillation frequency	f_{PLLFR}	—	9	—	MHz	

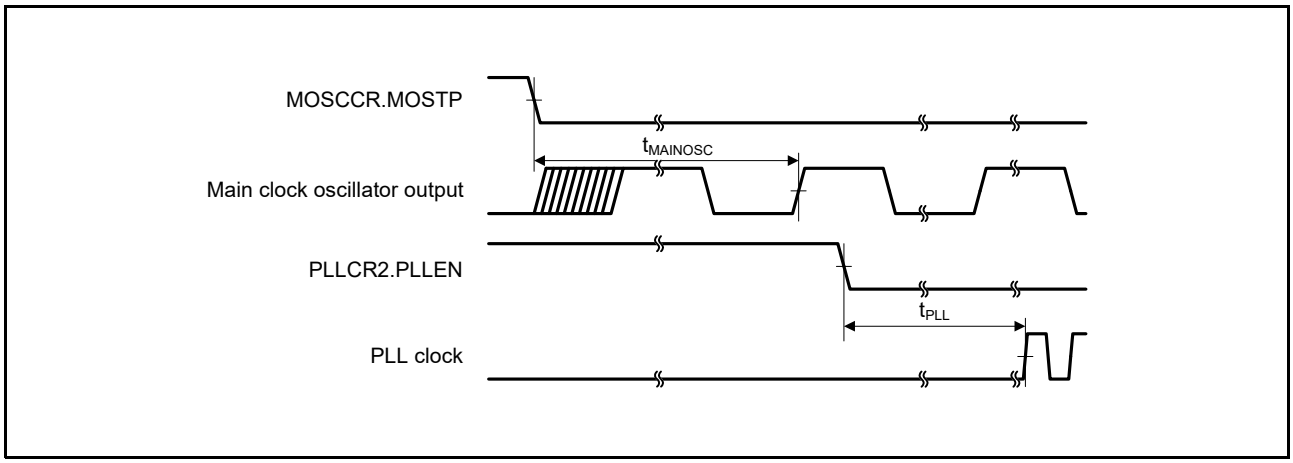


Figure 47.15 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)

Table 47.36 PLL2 Clock Timing

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL2 input frequency	f_{PLLIN}	4	—	12.5	MHz	
PLL2 circuit oscillation frequency	f_{PLL}	24	—	64	MHz	
PLL2 clock oscillation stabilization time	t_{PLL}	—	—	81.4	µs	Figure 47.16

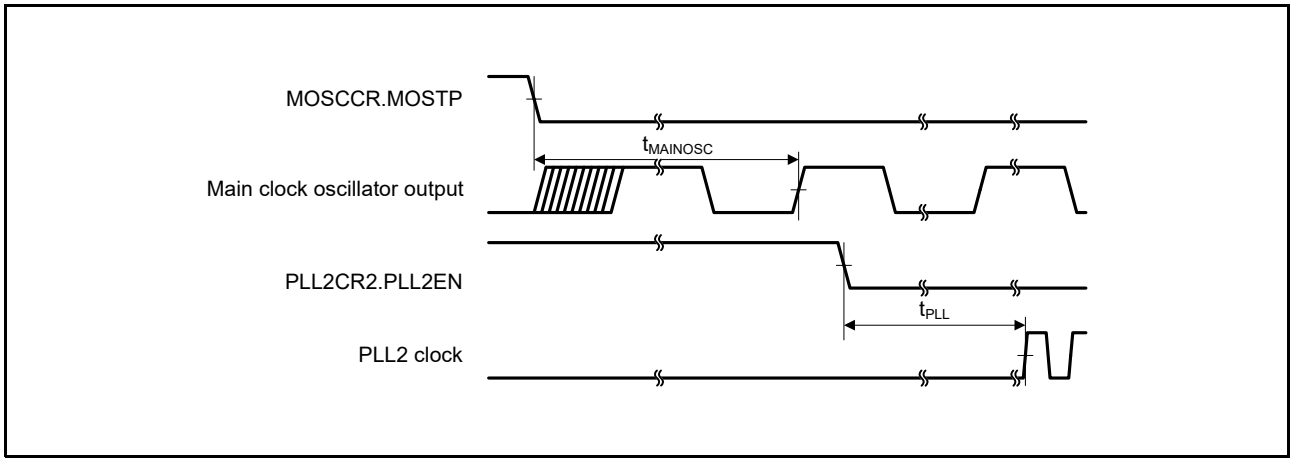


Figure 47.16 PLL2 Clock Oscillation Start Timing (PLL2 is Operated after Main Clock Oscillation Has Settled)

Table 47.37 EXCIN Clock Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXCIN external clock input cycle time	t_{Xcyc}	31.25	—	—	µs	Figure 47.17
EXCIN external clock input high pulse width	t_{XH}	15.62	—	—	µs	
EXCIN external clock input low pulse width	t_{XL}	15.62	—	—	µs	
EXCIN external clock rise time	t_{Xr}	—	—	5.0	ns	
EXCIN external clock fall time	t_{Xf}	—	—	5.0	ns	
EXCIN external clock input wait time*1	t_{XWT}	0.2	—	—	ms	

Note 1. Time until the clock can be used after the sub-clock oscillator stop bit (SOSCCR.SOSTP) is set to 0 (operating).

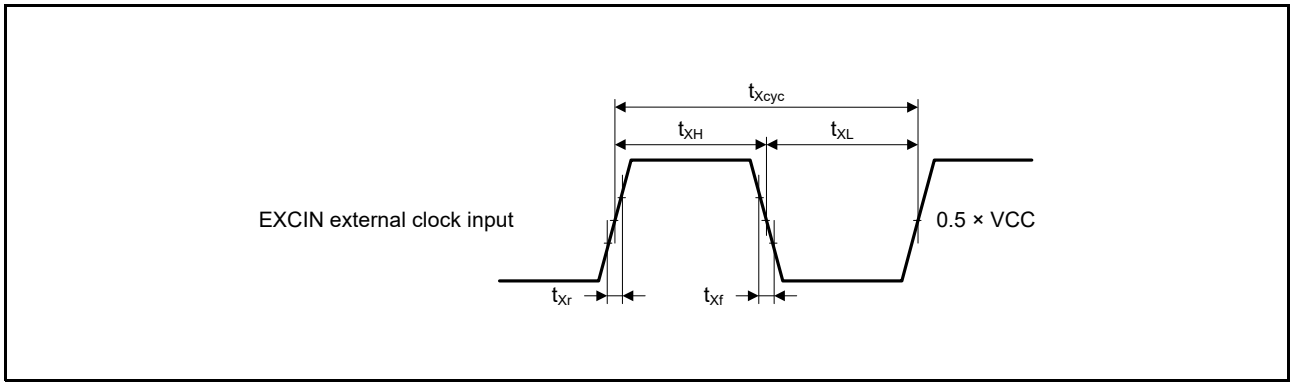


Figure 47.17 EXCIN External Clock Input Timing

Table 47.38 Sub-Clock Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Sub-clock oscillator oscillation frequency*2	f_{SUB}	—	32.768	—	kHz	
Sub-clock oscillation stabilization time*1	t_{SUBOSC}	—	0.5	—	s	Figure 47.18

Note 1. Reference value when a 32.768-kHz resonator is used.
After changing the setting of the SOSCCR.SOSTP bit so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization wait time that is equal to or greater than the oscillator-manufacturer-recommended value has elapsed.

Note 2. Only 32.768-kHz can be used.

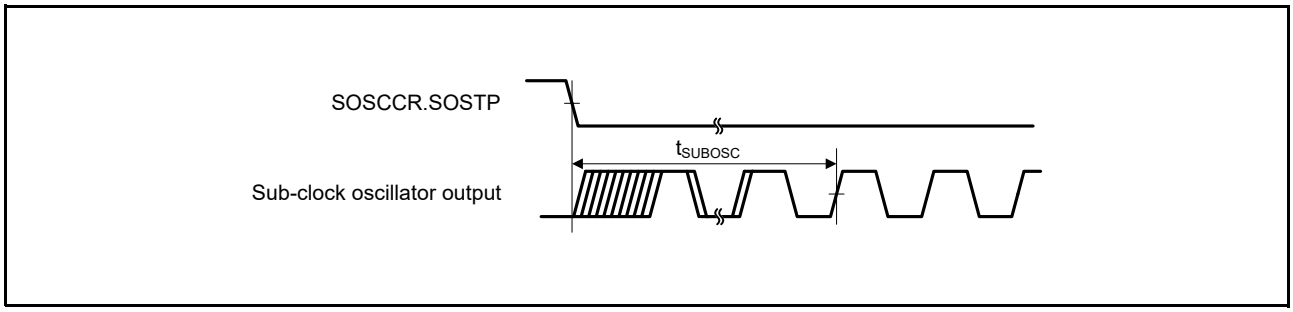


Figure 47.18 Sub-Clock Oscillation Start Timing

47.5.2 Reset Timing

Table 47.39 Reset Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	At power-on	t_{RESWP}	10.5	—	—	ms	Figure 47.19
	Other than above	t_{RESW}	30	—	—	μs	Figure 47.20
Wait time after RES# cancellation (at power-on)	At normal startup*1	t_{RESWT}	—	8.5	—	ms	Figure 47.19
	During fast startup time*2	t_{RESWT}	—	850	—	μs	
Wait time after RES# cancellation (during powered-on state)	LVD0 disabled*3	t_{RESWT}	—	140	—	μs	Figure 47.20
	LVD0 enabled*4		—	850	—	μs	
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)	LVD0 disabled*3	t_{RESWT2}	—	210	—	μs	
	LVD0 enabled*4		—	910	—	μs	

Note 1. When OFS1.(LVDAS, FASTSTUP) = 11b.
Note 2. When OFS1.(LVDAS, FASTSTUP) = a value other than 11b.
Note 3. When OFS1.LVDAS = 1b.
Note 4. When OFS1.LVDAS = 0b.

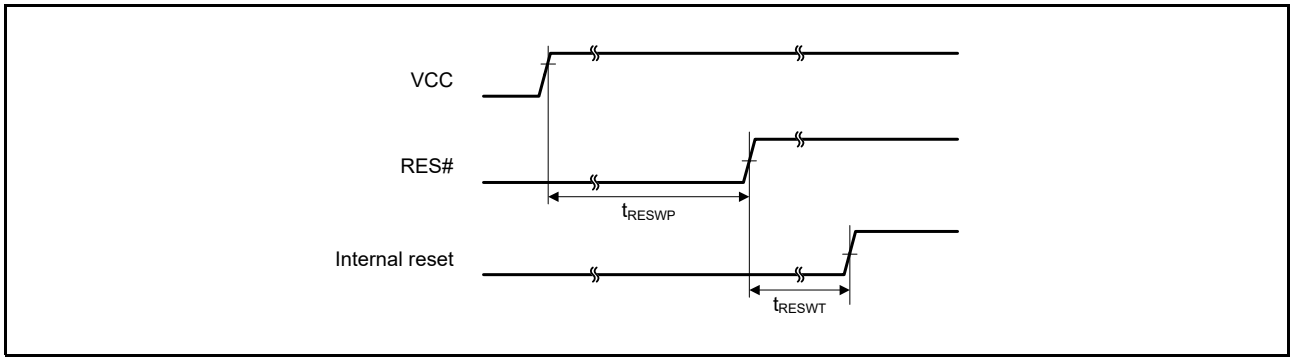


Figure 47.19 Reset Input Timing at Power-On

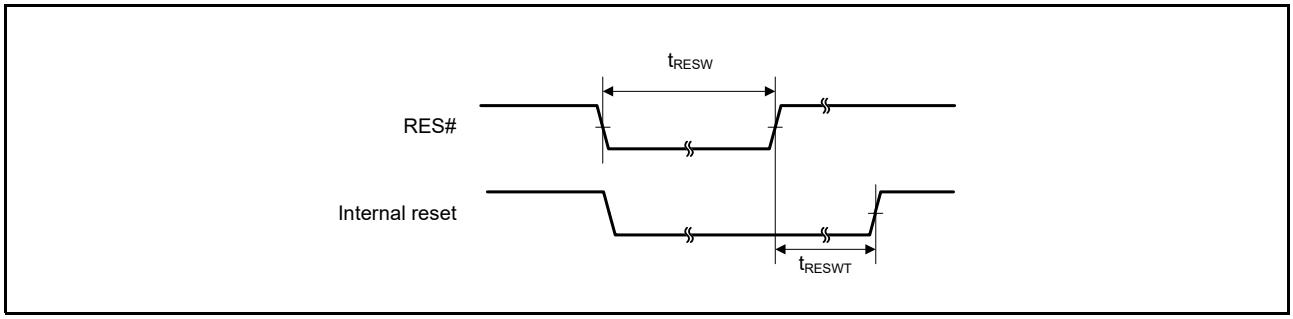


Figure 47.20 Reset Input Timing (1)

47.5.3 Timing of Recovery from Low Power Consumption Modes

Table 47.40 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
The oscillation stabilization wait time*1	High-speed operating mode/ Middle-speed operating mode	Main clock oscillator operating	Main clock oscillator operating	$t_{SBYOSCWTMC}$	—	—	0.65 + t_{LOCO} + (16 + Number of cycles specified in MOSCWTCR) / f_{LOCO} + 2 / f_{MOSC} + 1 / f_{ICLK}	μs	
			Main clock oscillator and PLL circuit operating	$t_{SBYOSCWTPC}$	—	—	0.65 + t_{LOCO} + (288 + Number of cycles specified in MOSCWTCR) / f_{LOCO} + 2 / f_{PLL} + 1 / f_{ICLK}		
		Sub-clock oscillator operating		$t_{SBYOSCWTSC}$	—	—	0.65 + 3 / f_{SOSC} + 1 / f_{ICLK}		
		High-speed on-chip oscillator		$t_{SBYOSCWTHO}$	—	—	0.65 + t_{LOCO} + 16 / f_{LOCO} + 2 / f_{HOCO} + 1 / f_{ICLK}		
		Low-speed on-chip oscillator		$t_{SBYOSCWTLO}$	—	—	0.65 + t_{LOCO} + 1 / f_{ICLK}		
The time required for operations by the software standby release sequencer*2				t_{SBYSEQ}	—	—	4 / f_{LOCO} + 11 / f_{ICLK} + 3 / f_{PCLKB} + 3n / $f_{source\ clock}$		
Recovery time from software standby mode*3	High-speed operating mode/ Middle-speed operating mode	Main clock oscillator operating	Main clock oscillator operating	t_{SBYMC}	—	—	$t_{SBYOSCWTMC}$ + t_{SBYSEQ}		Figure 47.21
			Main clock oscillator and PLL circuit operating	t_{SBYPC}	—	—	$t_{SBYOSCWTPC}$ + t_{SBYSEQ}		
		Sub-clock oscillator operating		t_{SBYSC}	—	—	$t_{SBYOSCWTSC}$ + t_{SBYSEQ}		
		High-speed on-chip oscillator		t_{SBYHO}	—	—	$t_{SBYOSCWTHO}$ + t_{SBYSEQ}		
		Low-speed on-chip oscillator		t_{SBYLO}	—	—	$t_{SBYOSCWTLO}$ + t_{SBYSEQ}		

Note 1. When multiple oscillators are operating before entering software standby mode, the oscillation stabilization wait time will be selected from the largest value among the operating oscillators.

Note 2. For n, the greatest value is selected from among the internal clock division settings.

Note 3. The time for recovery from software standby mode is determined by the value obtained by adding the oscillation stabilization waiting time and the time required for operations by the software standby release sequencer.

Table 47.41 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
The oscillation stabilization wait time*1	Middle-speed operating mode 2	Main clock oscillator operating	Main clock oscillator operating	$t_{SBYOSCWTMC}$	—	—	0.65 + t_{LOCO} + (16 + Number of cycles specified in MOSCWTCR) / f_{LOCO} + 2 / f_{MOSC} + 1 / f_{ICLK}	µs	
			Main clock oscillator and PLL circuit operating	$t_{SBYOSCWTPC}$	—	—	0.65 + t_{LOCO} + (288 + Number of cycles specified in MOSCWTCR) / f_{LOCO} + 2 / f_{PLL} + 1 / f_{ICLK}		
		Sub-clock oscillator operating		$t_{SBYOSCWTSC}$	—	—	0.65 + 3 / f_{SOSC} + 1 / f_{ICLK}		
		High-speed on-chip oscillator		$t_{SBYOSCWTHO}$	—	—	0.65 + t_{LOCO} + 16 / f_{LOCO} + 2 / f_{HOCO} + 1 / f_{ICLK}		
		Low-speed on-chip oscillator		$t_{SBYOSCWTLO}$	—	—	0.65 + t_{LOCO} + 1 / f_{ICLK}		
The time required for operations by the software standby release sequencer*2				t_{SBYSEQ}	—	—	9 / f_{ICLK} + 3 / f_{PCLKB} + 3n / $f_{source\ clock}$		
Recovery time from software standby mode*3	Middle-speed operating mode 2	Main clock oscillator operating	Main clock oscillator operating	t_{SBYMC}	—	—	$t_{SBYOSCWTMC}$ + t_{SBYSEQ}		Figure 47.21
			Main clock oscillator and PLL circuit operating	t_{SBYPC}	—	—	$t_{SBYOSCWTPC}$ + t_{SBYSEQ}		
		Sub-clock oscillator operating		t_{SBYSC}	—	—	$t_{SBYOSCWTSC}$ + t_{SBYSEQ}		
		High-speed on-chip oscillator		t_{SBYHO}	—	—	$t_{SBYOSCWTHO}$ + t_{SBYSEQ}		
		Low-speed on-chip oscillator		t_{SBYLO}	—	—	$t_{SBYOSCWTLO}$ + t_{SBYSEQ}		

Note 1. When multiple oscillators are operating before entering software standby mode, the oscillation stabilization wait time will be selected from the largest value among the operating oscillators.

Note 2. For n, the greatest value is selected from among the internal clock division settings.

Note 3. The time for recovery from software standby mode is determined by the value obtained by adding the oscillation stabilization waiting time and the time required for operations by the software standby release sequencer.

Table 47.42 Timing of Recovery from Low Power Consumption Modes (3)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
The oscillation stabilization wait time*1	Low-speed operating mode	Sub-clock oscillator operating	$t_{SBYOSCWTSC}$	—	—	0.65 + 3 / f_{SOSC} + 1 / f_{ICLK}	µs	
The time required for operations by the software standby release sequencer*1			t_{SBYSEQ}	—	—	9 / f_{ICLK} + 3 / f_{PCLKB} + 3n / $f_{source\ clock}$		
Recovery time from software standby mode*2	Low-speed operating mode	Sub-clock oscillator operating	t_{SBYSC}	—	—	$t_{SBYOSCWTSC}$ + t_{SBYSEQ}		Figure 47.21

Note 1. For n, the greatest value is selected from among the internal clock division settings.

Note 2. The time for recovery from software standby mode is determined by the value obtained by adding the oscillation stabilization waiting time and the time required for operations by the software standby release sequencer.

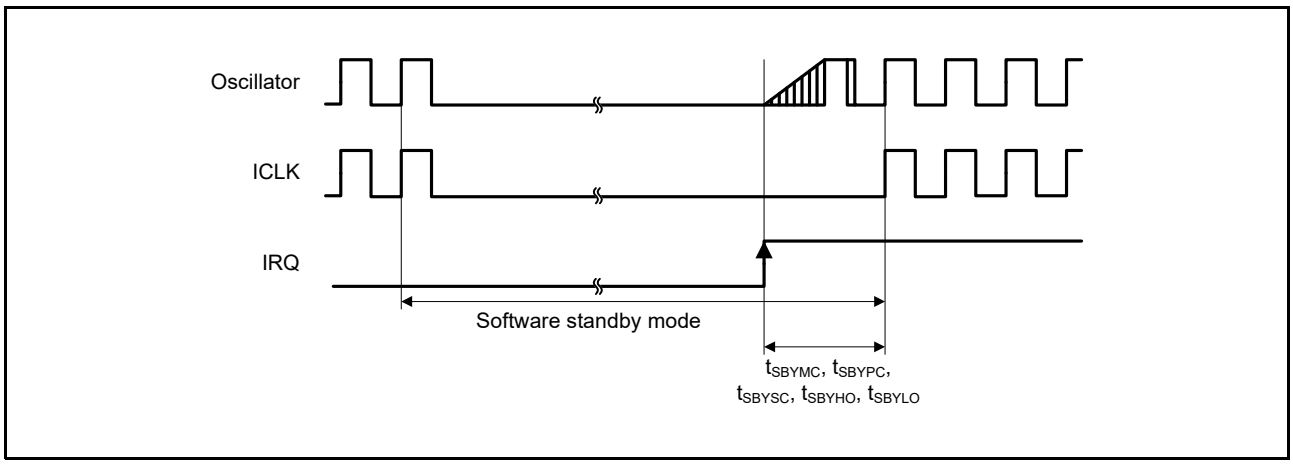


Figure 47.21 Software Standby Mode Recovery Timing

Table 47.43 Timing of Recovery from Low Power Consumption Modes (4)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
The oscillation stabilization wait time*1	Main clock oscillator operating	Main clock oscillator operating	$t_{SBYOSCWTMC}$	—	—	0.65 + t_{LOCO} + (16 + Number of cycles specified in MOSCWTCR) / f_{LOCO} + 2 / f_{MOSC} + 1 / f_{ICLK}	µs	
		Main clock oscillator and PLL circuit operating	$t_{SBYOSCWTPC}$	—	—	0.65 + t_{LOCO} + (288 + Number of cycles specified in MOSCWTCR) / f_{LOCO} + 2 / f_{PLL} + 1 / f_{ICLK}		
	Sub-clock oscillator operating		$t_{SBYOSCWTSC}$	—	—	0.65 + 3 / f_{SOSC} + 1 / f_{ICLK}		
	High-speed on-chip oscillator		$t_{SBYOSCWTHO}$	—	—	0.65 + t_{LOCO} + 16 / f_{LOCO} + 2 / f_{HOCO} + 1 / f_{ICLK}		
	Low-speed on-chip oscillator		$t_{SBYOSCWTLO}$	—	—	0.65 + t_{LOCO} + 1 / f_{ICLK}		
The time required for operations by the software standby release sequencer*2			t_{SBYSEQ}	—	—	3 / f_{ICLK} + 2n / $f_{source\ clock}$		
Time to shift to the snooze mode from the software standby mode*3	Main clock oscillator operating	Main clock oscillator operating	t_{SNZMC}	—	—	$t_{SBYOSCWTMC}$ + t_{SBYSEQ}		Figure 47.22
		Main clock oscillator and PLL circuit operating	t_{SNZPC}	—	—	$t_{SBYOSCWTPC}$ + t_{SBYSEQ}		
	Sub-clock oscillator operating		t_{SNZSC}	—	—	$t_{SBYOSCWTSC}$ + t_{SBYSEQ}		
	High-speed on-chip oscillator		t_{SNZHO}	—	—	$t_{SBYOSCWTHO}$ + t_{SBYSEQ}		
	Low-speed on-chip oscillator		t_{SNZLO}	—	—	$t_{SBYOSCWTLO}$ + t_{SBYSEQ}		

Note 1. When multiple oscillators are operating before entering software standby mode, the oscillation stabilization wait time will be selected from the largest value among the operating oscillators.

Note 2. For n, the greatest value is selected from among the internal clock division settings.

Note 3. Time to shift to the snooze mode from the software standby mode is determined by the value obtained by adding the oscillation stabilization waiting time and the time required for operations by the software standby release sequencer.

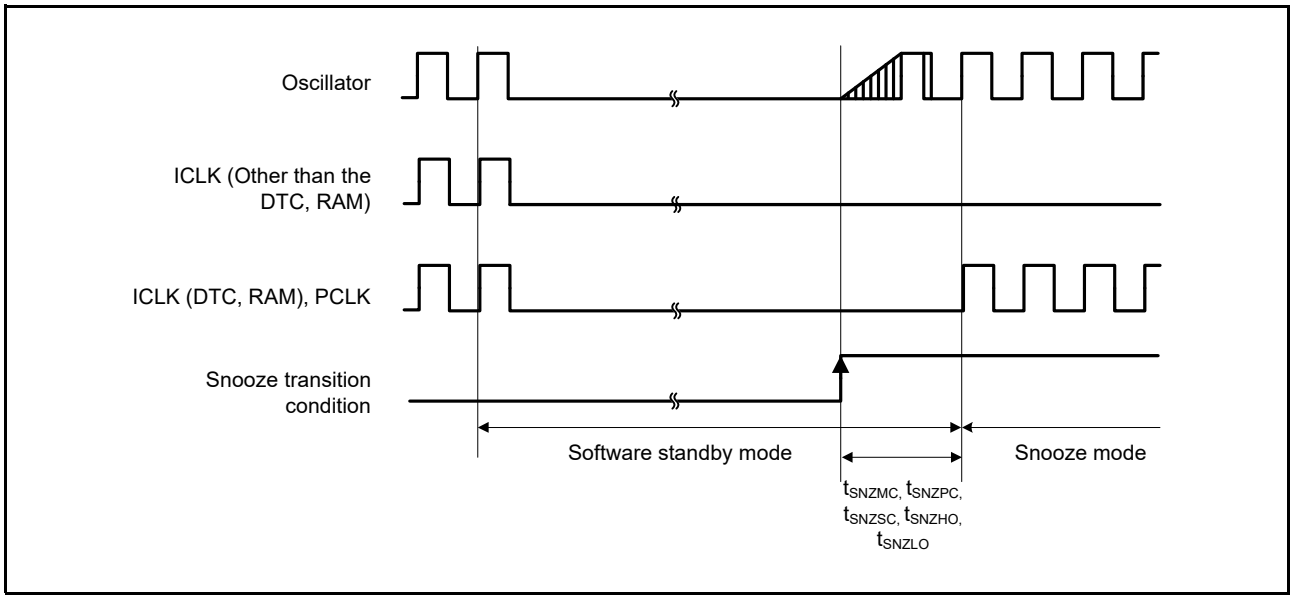


Figure 47.22 Timing to shift to the Snooze Mode from the Software Standby Mode

Table 47.44 Timing of Recovery from Low Power Consumption Modes (5)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.*2	Unit	Test Conditions
Recovery time from deep sleep mode*1	High-speed operating mode	t_{DSLP}	—	—	4 / f_{LOCO} + 8 / f_{ICLK} + 2 / f_{PCLKB} + 3n / $f_{source\ clock}$	µs	Figure 47.23
	Middle-speed operating mode				4 / f_{LOCO} + 8 / f_{ICLK} + 2 / f_{PCLKB} + 3n / $f_{source\ clock}$		
	Middle-speed operating mode 2				6 / f_{ICLK} + 2 / f_{PCLKB} + 3n / $f_{source\ clock}$		
	Low-speed operating mode				6 / f_{ICLK} + 2 / f_{PCLKB} + 3n / $f_{source\ clock}$		

Note 1. Oscillators continue oscillating in deep sleep mode.
Note 2. n represents the largest frequency divisor among those for the internal clock signals.

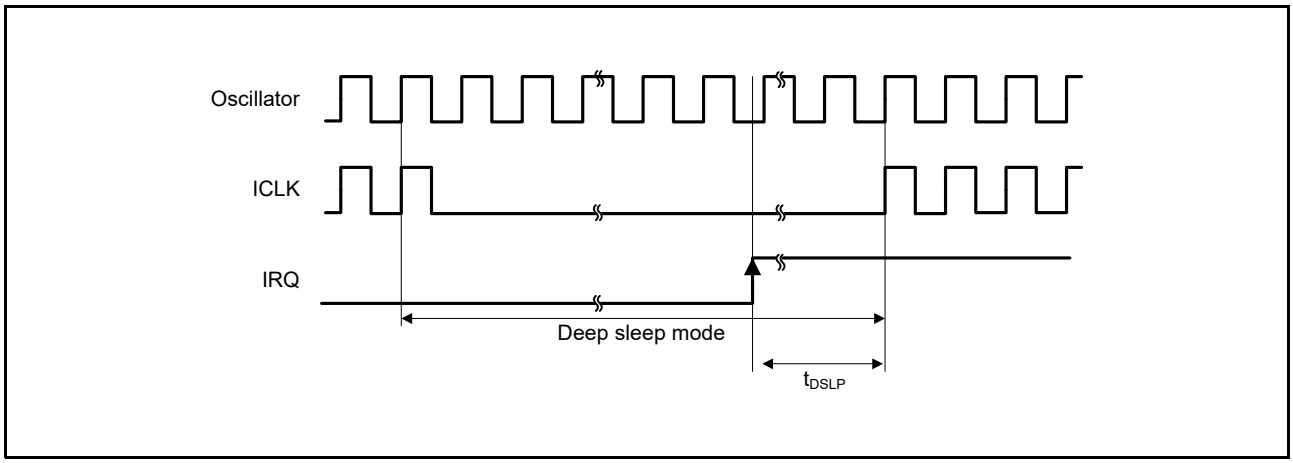


Figure 47.23 Deep Sleep Mode Recovery Timing

47.5.4 Operating Mode Transition Time

Table 47.45 Operating Mode Transition Time

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Mode before Transition	Mode after Transition	Transition Time				Unit
		Min.	Typ.		Max.	
			$f_{ICLK} \geq f_{FCLK}$	$f_{ICLK} < f_{FCLK}$		
High-speed operating mode	Middle-speed operating mode	—	5 / f_{ICLK} + 3 / f_{FCLK}	8 / f_{ICLK}	—	µs
	Middle-speed operating mode 2	—	5 / f_{ICLK} + 3 / f_{FCLK}	8 / f_{ICLK}	—	
	Low-speed operating mode	—	3 / f_{ICLK} + 2 / f_{FCLK}	5 / f_{ICLK}	—	
Middle-speed operating mode	High-speed operating mode	—	5 / f_{ICLK} + 3 / f_{FCLK}	8 / f_{ICLK}	—	
	Middle-speed operating mode 2	—	5 / f_{ICLK} + 3 / f_{FCLK}	8 / f_{ICLK}	—	
	Low-speed operating mode	—	3 / f_{ICLK} + 2 / f_{FCLK}	5 / f_{ICLK}	—	
Middle-speed operating mode 2	High-speed operating mode	—	5 / f_{ICLK} + 3 / f_{FCLK}	8 / f_{ICLK}	—	
	Middle-speed operating mode	—	5 / f_{ICLK} + 3 / f_{FCLK}	8 / f_{ICLK}	—	
	Low-speed operating mode	—	3 / f_{ICLK} + 2 / f_{FCLK}	5 / f_{ICLK}	—	
Low-speed operating mode	High-speed operating mode	—	3 / f_{ICLK} + 3 / f_{FCLK}	6 / f_{ICLK}	—	
	Middle-speed operating mode	—	3 / f_{ICLK} + 3 / f_{FCLK}	6 / f_{ICLK}	—	
	Middle-speed operating mode 2	—	3 / f_{ICLK} + 3 / f_{FCLK}	6 / f_{ICLK}	—	

47.5.5 Control Signal Timing

Table 47.46 Control Signal Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter disabled (NMIFLTE.NFLTEN = 0)	t_{Pcyc} × 2 ≤ 200 ns
		t_{PBcyc} × 2*1	—	—			t_{Pcyc} × 2 > 200 ns
		200	—	—		NMI digital filter enabled (NMIFLTE.NFLTEN = 1)	t_{NMICK} × 3 ≤ 200 ns
		t_{NMICK} × 3.5*2	—	—			t_{NMICK} × 3 > 200 ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled (IRQFLTE0.FLTENi = 0)	t_{Pcyc} × 2 ≤ 200 ns
		t_{PBcyc} × 2*1	—	—			t_{Pcyc} × 2 > 200 ns
		200	—	—		IRQ digital filter enabled (IRQFLTE0.FLTENi = 1)	t_{IRQCK} × 3 ≤ 200 ns
		t_{IRQCK} × 3.5*3	—	—			t_{IRQCK} × 3 > 200 ns

Note: 200 ns minimum in software standby mode.
Note 1. t_{PBcyc} indicates the cycle of PCLKB.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 7).

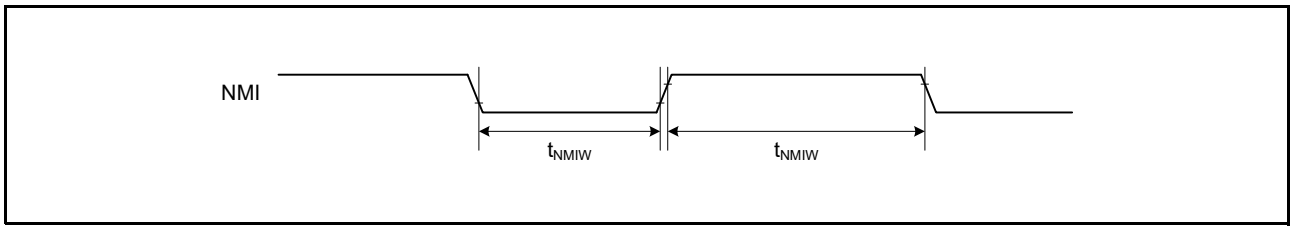


Figure 47.24 NMI Interrupt Input Timing

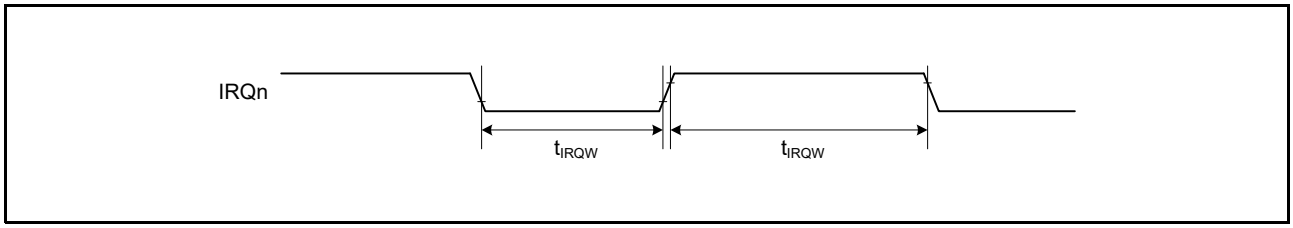


Figure 47.25 IRQ Interrupt Input Timing

47.5.6 Timing of On-Chip Peripheral Modules

47.5.6.1 I/O Port Input Timing

Table 47.47 I/O Port Input Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Max.	Unit *1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	t_{PBcyc}	Figure 47.26

Note 1. t_{PBcyc}: PCLKB cycle

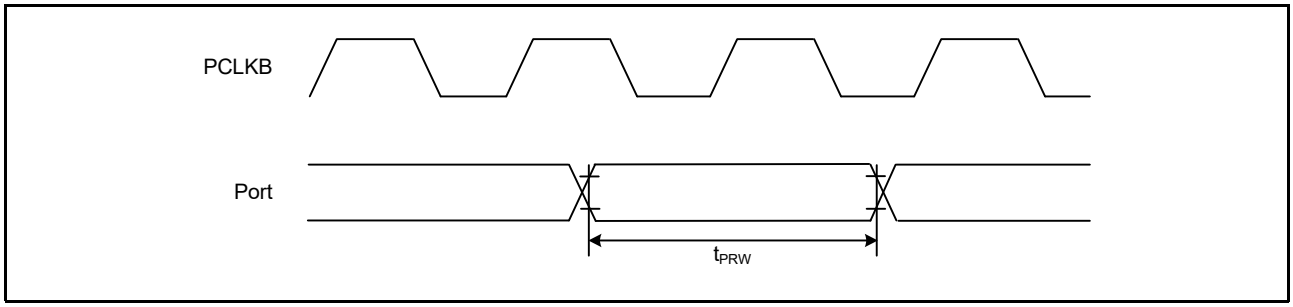


Figure 47.26 I/O Port Input Timing

47.5.6.2 GPTW

Table 47.48 GPTW Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
GPTW	Input capture input pulse width	Single-edge setting	t_{GTICW}	1.5	—	t_{PAcyc}	Figure 47.27
		Both-edge setting		2.5	—		
	Input capture rise/fall time		t_{GTICr}/ t_{GTICf}	—	0.1	µs/V	Figure 47.27
	External trigger input pulse width	Single-edge setting	t_{GTEW}	1.5	—	t_{PAcyc}	Figure 47.28
		Both-edge setting		2.5	—		
	Timer clock pulse width		t_{GTCKWH} / t_{GTCKWL}	1.5	—	t_{PAcyc}	Figure 47.29
	Timer clock rise/fall time		t_{GTCKr}/ t_{GTCKf}	—	0.1	µs/V	Figure 47.29

Note 1. t_{PAcyc}: PCLKA cycle

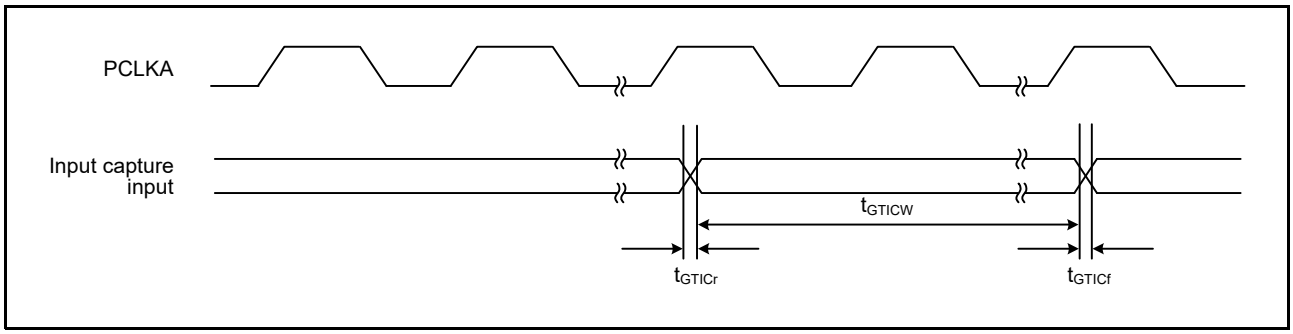


Figure 47.27 GPTW Input Capture Input Timing

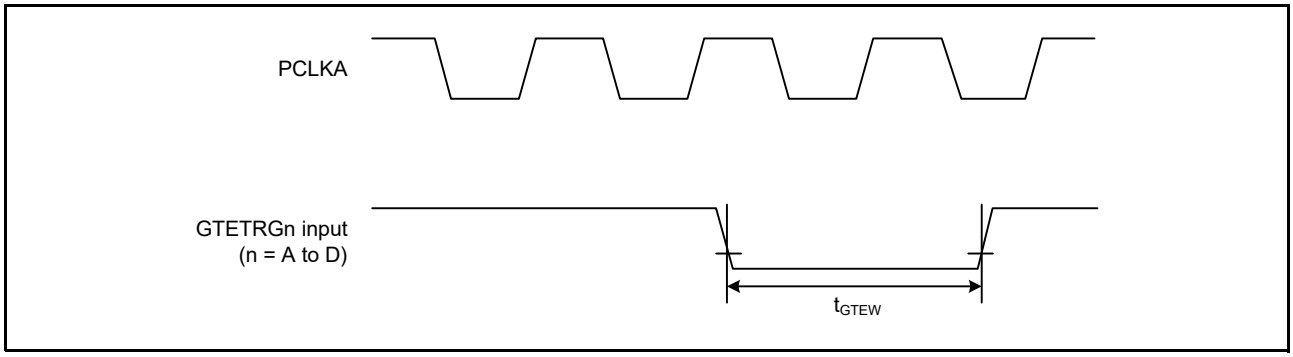


Figure 47.28 GPTW External Trigger Input Timing

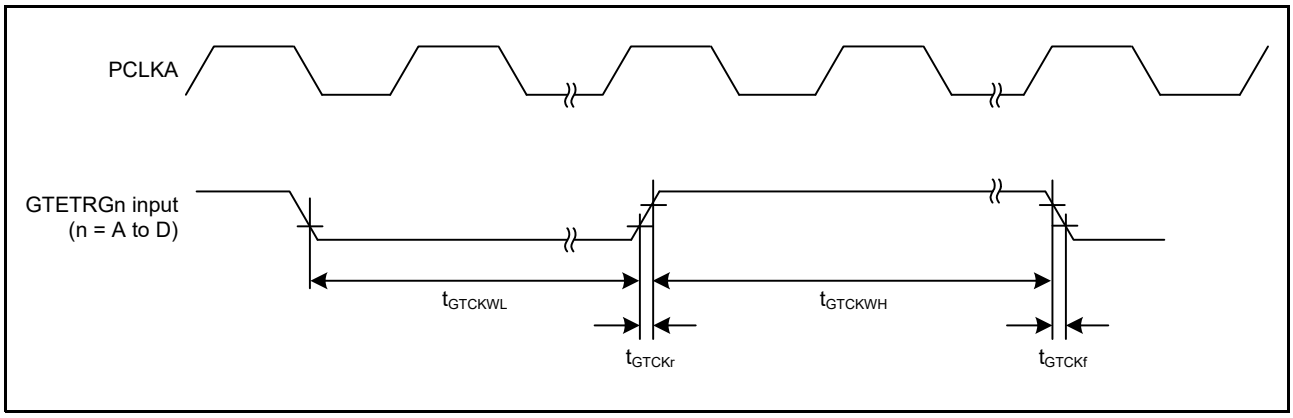


Figure 47.29 GPTW Clock Input Timing

47.5.6.3 POEG

Table 47.49 POEG Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item			Symbol	Min.	Typ.	Max.	Unit*1	Test Conditions
POEG	GTETRGn input pulse width (n = A to D)		t_{POEGW}	1.5	—	—	t_{PBcyc}	Figure 47.30
	GTERGn input rise/fall time		t_{POEGr}/ t_{POEGf}	—	—	0.1	µs/V	Figure 47.30
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	—	3 PCLKB + 0.34	µs	Figure 47.31 When the digital noise filter is not in use (POEGGn.NFEN = 0 (n = A to D))
		Detection of the output stopping signal from GPTW (simultaneous high output or simultaneous low output)	t_{POEGDE}	—	—	0.5	µs	Figure 47.32
		Edge detection signal from a comparator	t_{POEGDC}	—	—	4 PCLKB + 0.5	µs	Figure 47.33 The time is that when the noise filter for comparator B is not in use (CPBF.CPB0FEN = 0 and CPBF.CPB1FEN = 0) and excludes the time for detection by comparator B.
		Register setting	t_{POEGDS}	—	—	1 PCLKB + 0.3	µs	Figure 47.34 Time for access to the register is not included.
		Oscillation stop detection	$t_{POEGDOS}$	—	—	21	µs	Figure 47.35

Note 1. t_{PBcyc}: PCLKB cycle

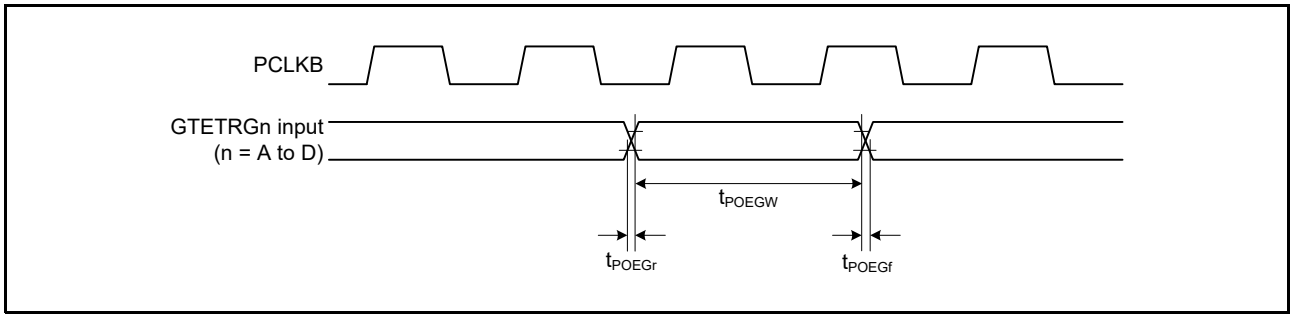


Figure 47.30 POEG Input Timing

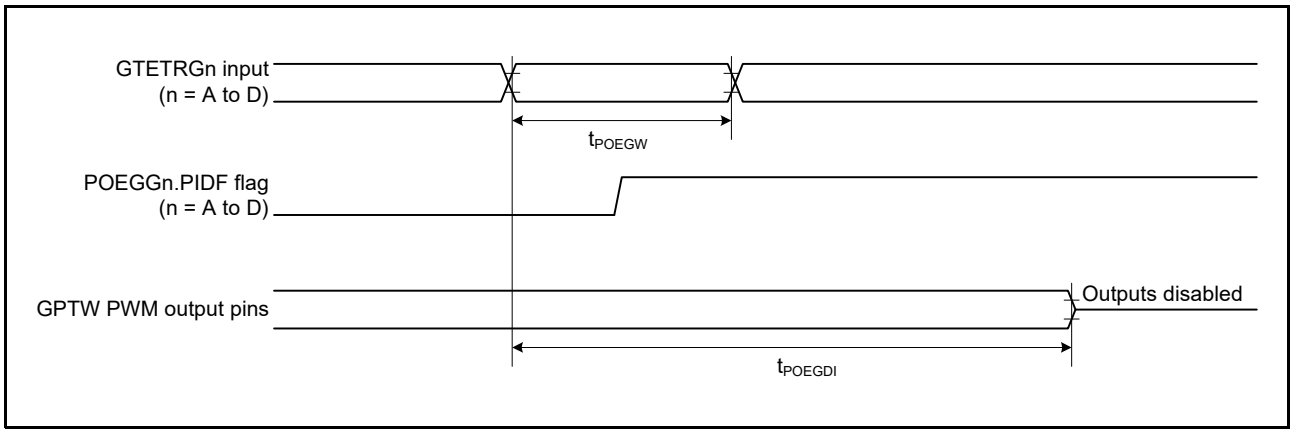
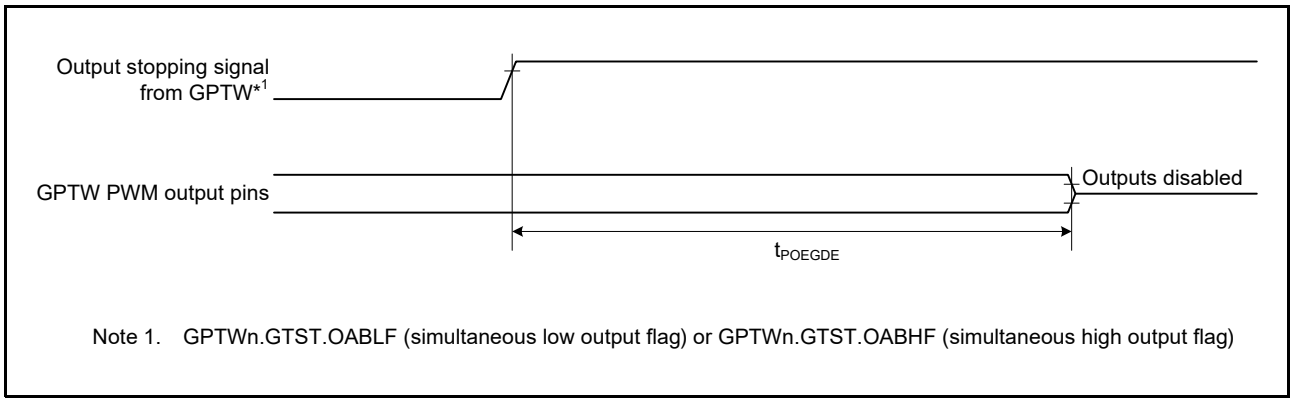


Figure 47.31 Output Disable Time for POEG via Detection Flag in Response to the Input Level Detection of the GTETRGn pin



Note 1. GPTWn.GTST.OABLF (simultaneous low output flag) or GPTWn.GTST.OABHF (simultaneous high output flag)

Figure 47.32 Output Disable Time for POEG in Response to Detection of the Output Stopping Signal from GPTW

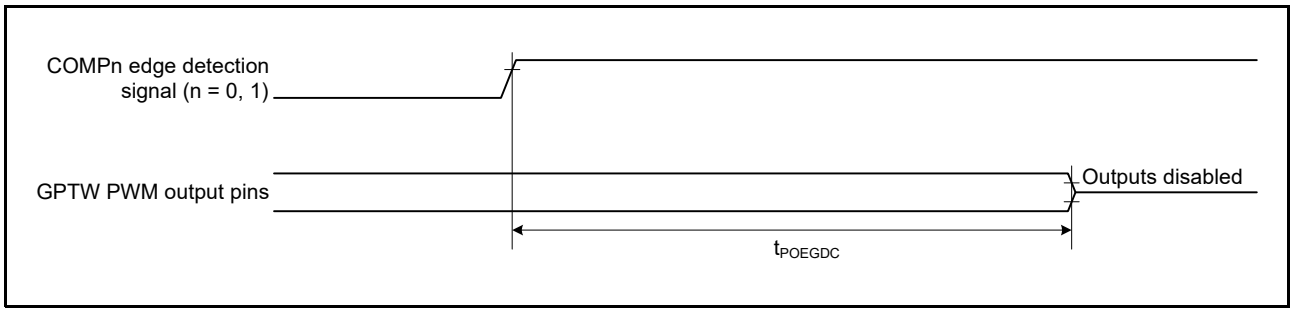


Figure 47.33 Output Disable Time for POEG in Response to Edge Detection Signal from a Comparator

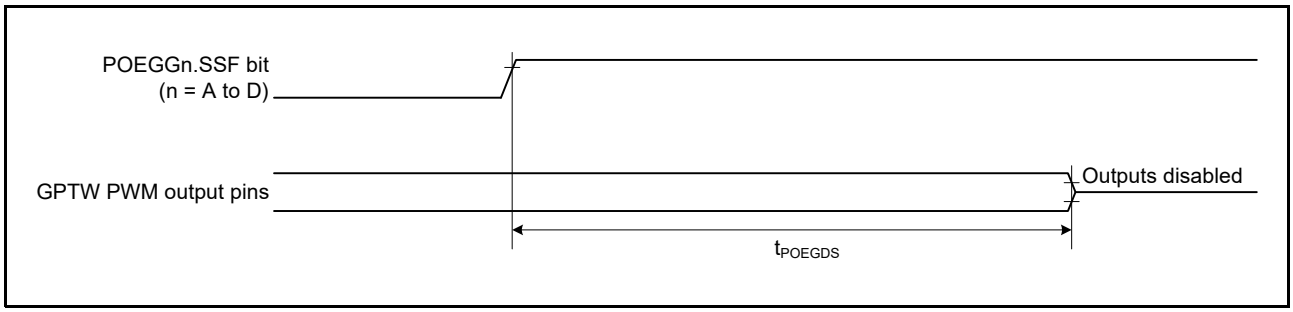


Figure 47.34 Output Disable Time for POEG in Response to the Register Setting

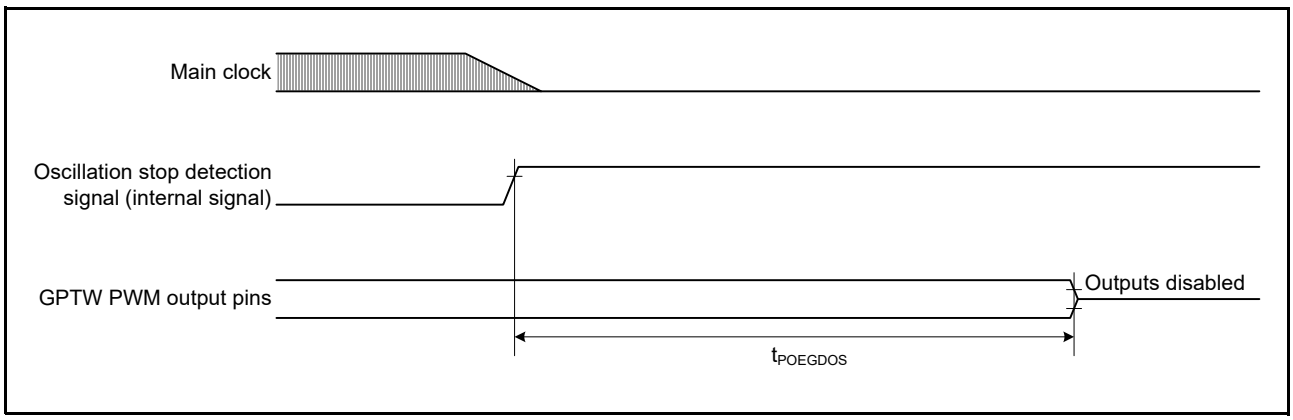


Figure 47.35 Output Disable Time of POEG in Response to the Oscillation Stop Detection

47.5.6.4 TMR

Table 47.50 TMR Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

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 47.36
		Both-edge setting		2.5	—		
	Timer clock rise/fall time		t_{TMCr}, t_{TMCf}	—	0.1	µs/V	

Note 1. t_{PBcyc}: PCLKB cycle

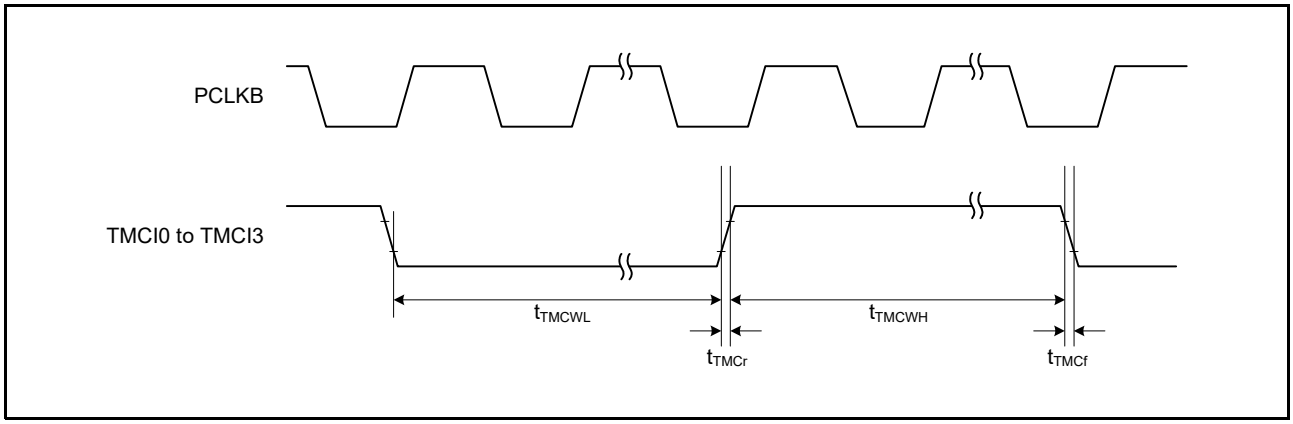


Figure 47.36 TMR Clock Input Timing

47.5.6.5 SCI

Table 47.51 SCI Timing (1/2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF

Item				Symbol	Min.	Max.	Unit *1	Test Conditions
SCI (channel 1, 5, 6)	Input clock cycle time	Asynchronous		t_{Scyc}	4	—	t_{PBcyc}	Figure 47.37
		Clock synchronous	2.4 V ≤ VCC ≤ 5.5 V		6	—		
			1.8 V ≤ VCC < 2.4 V		8	—		
			1.6 V ≤ VCC < 1.8 V		6	—		
	Input clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time			t_{SCKr}	—	20	ns	
	Input clock fall time			t_{SCKf}	—	20	ns	
	Output clock cycle time	Asynchronous		t_{Scyc}	6	—	t_{PBcyc}	Figure 47.38
		Clock synchronous	2.4 V ≤ VCC ≤ 5.5 V		4	—		
			1.8 V ≤ VCC < 2.4 V		8	—		
			1.6 V ≤ VCC < 1.8 V		4	—		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKr}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Output clock fall time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKf}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Transmit data delay time (master)	Clock synchronous	1.8 V ≤ VCC ≤ 5.5 V	t_{TXD}	—	40	ns	
			1.6 V ≤ VCC < 1.8 V		—	45	ns	
	Transmit data delay time (slave)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		—	55	ns	
			2.4 V ≤ VCC < 2.7 V		—	60	ns	
			1.8 V ≤ VCC < 2.4 V		—	100	ns	
			1.6 V ≤ VCC < 1.8 V		—	125	ns	
	Receive data setup time (master)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{RXS}	45	—	ns	
			2.4 V ≤ VCC < 2.7 V		55	—	ns	
			1.8 V ≤ VCC < 2.4 V		90	—	ns	
			1.6 V ≤ VCC < 1.8 V		110	—	ns	
	Receive data setup time (slave)	Clock synchronous	1.8 V ≤ VCC ≤ 5.5 V		40	—	ns	
			1.6 V ≤ VCC < 1.8 V		45	—	ns	
	Receive data hold time	Clock synchronous		t_{RXH}	40	—	ns	

Table 47.51 SCI Timing (2/2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF

Item				Symbol	Min.	Max.	Unit *1	Test Conditions
SCI (channel 12)	Input clock cycle time	Asynchronous		t_{Scyc}	4	—	t_{PBcyc}	Figure 47.37
		Clock synchronous	2.4 V ≤ VCC ≤ 5.5 V		6	—		
			1.8 V ≤ VCC < 2.4 V		8	—		
			1.6 V ≤ VCC < 1.8 V		6	—		
	Input clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time			t_{SCKr}	—	20	ns	
	Input clock fall time			t_{SCKf}	—	20	ns	
	Output clock cycle time	Asynchronous*2		t_{Scyc}	8	—	t_{PBcyc}	Figure 47.38
		Clock synchronous	2.4 V ≤ VCC ≤ 5.5 V		4	—		
			1.8 V ≤ VCC < 2.4 V		8	—		
			1.6 V ≤ VCC < 1.8 V		4	—		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKr}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Output clock fall time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKf}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Transmit data delay time (master)	Clock synchronous	1.8 V ≤ VCC ≤ 5.5 V	t_{TXD}	—	40	ns	
			1.6 V ≤ VCC < 1.8 V		—	45	ns	
	Transmit data delay time (slave)	Clock synchronous	2.4 V ≤ VCC ≤ 5.5 V		—	65	ns	
			1.8 V ≤ VCC < 2.4 V		—	100	ns	
			1.6 V ≤ VCC < 1.8 V		—	125	ns	
	Receive data setup time (master)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{RXS}	45	—	ns	
			2.4 V ≤ VCC < 2.7 V		55	—	ns	
			1.8 V ≤ VCC < 2.4 V		90			
			1.6 V ≤ VCC < 1.8 V		110	—	ns	
	Receive data setup time (slave)	Clock synchronous	1.8 V ≤ VCC ≤ 5.5 V		40	—	ns	
			1.6 V ≤ VCC < 1.8 V		45	—	ns	
	Receive data hold time	Clock synchronous		t_{RXH}	40	—	ns	

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. When SEMR.ABCS = 1 and SEMR.BGDM = 1

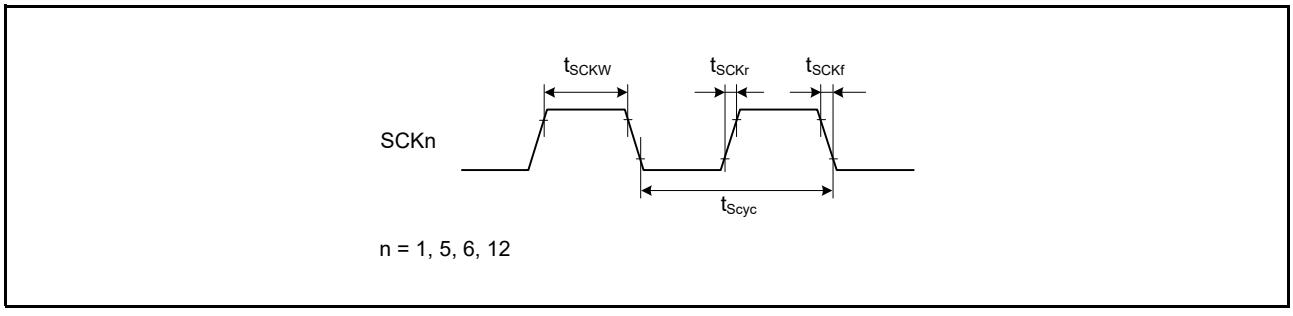


Figure 47.37 SCK Clock Input Timing

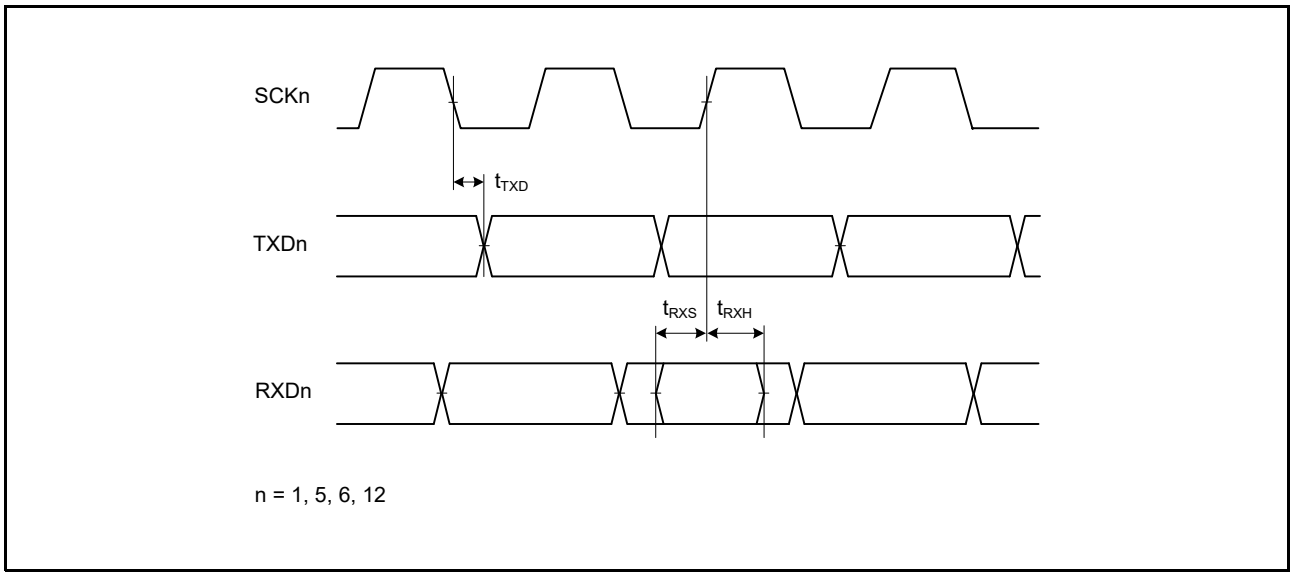


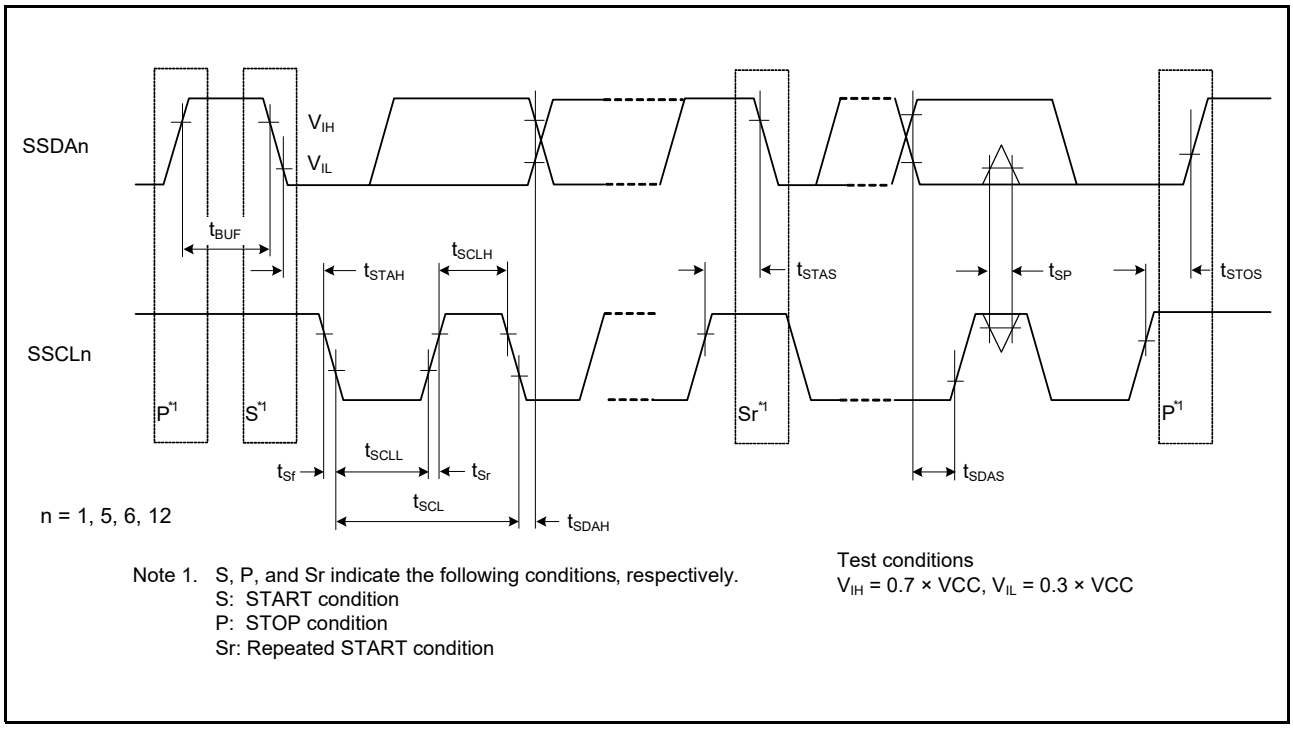
Figure 47.38 SCI Input/Output Timing: Clock Synchronous Mode

Table 47.52 Simple I²C Timing

Conditions: 2.7 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple I²C (standard mode)	SDA rise time	t_{Sr}	—	1000	ns	Figure 47.39
	SDA fall time	t_{Sf}	—	300	ns	
	SDA spike pulse removal time	t_{SP}	0	4 × t_{PBcyc}	ns	
	Data setup time	t_{SDAS}	250	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	
Simple I²C (fast mode)	SDA rise time	t_{Sr}	—	300	ns	Figure 47.39
	SDA fall time	t_{Sf}	—	300	ns	
	SDA spike pulse removal time	t_{SP}	0	4 × t_{PBcyc}	ns	
	Data setup time	t_{SDAS}	100	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	

Note: t_{PBcyc}: PCLKB cycle
Note 1. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: START condition
P: STOP condition
Sr: Repeated START condition

Figure 47.39 Output Timing and Simple I²C Bus Interface Input/Output Timing

Table 47.53 Simple SPI Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)	2.4 V ≤ VCC ≤ 5.5 V	t_{SPBcyc}	4	65536	t_{PBcyc}	Figure 47.40
		1.8 V ≤ VCC < 2.4 V		8	65536		
		1.6 V ≤ VCC < 1.8 V		4	65536		
	SCK clock cycle input (slave)	2.4 V ≤ VCC ≤ 5.5 V	t_{SPBcyc}	6	—	t_{PBcyc}	
		1.8 V ≤ VCC < 2.4 V		8	—		
		1.6 V ≤ VCC < 1.8 V		6	—		
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPBcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPBcyc}	
	SCK clock rise/fall time	1.8 V ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	—	20	ns	
		1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Data input setup time (master)	2.7 V ≤ VCC ≤ 5.5 V	t_{SU}	45	—	ns	Figure 47.41, Figure 47.42
		2.4 V ≤ VCC < 2.7 V		55	—		
		1.8 V ≤ VCC < 2.4 V		80	—		
		1.6 V ≤ VCC < 1.8 V		110	—		
	Data input setup time (slave)	1.8 V ≤ VCC ≤ 5.5 V		40	—		
		1.6 V ≤ VCC < 1.8 V		45	—		
	Data input hold time		t_H	40	—	ns	
	SSL input setup time		t_{LEAD}	1	—	t_{SPBcyc}	
	SSL input hold time		t_{LAG}	1	—	t_{SPBcyc}	
	Data output delay time (master)	1.8 V ≤ VCC ≤ 5.5 V	t_{OD}	—	40	ns	
		1.6 V ≤ VCC < 1.8 V		—	50		
	Data output delay time (slave)	2.4 V ≤ VCC ≤ 5.5 V		—	65		
		1.8 V ≤ VCC < 2.4 V		—	100		
		1.6 V ≤ VCC < 1.8 V		—	125		
	Data output hold time (master)	2.7 V ≤ VCC ≤ 5.5 V	t_{OH}	–10	—	ns	
		1.8 V ≤ VCC < 2.7 V		–20	—		
		1.6 V ≤ VCC < 1.8 V		–40	—		
	Data output hold time (slave)			–10	—		
	Data rise/fall time	1.8 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	—	20	ns	
		1.6 V ≤ VCC < 1.8 V		—	30		
	SSL input rise/fall time		t_{SSLr}, t_{SSLf}	—	20	ns	
	Slave access time	2.4 V ≤ VCC ≤ 5.5 V	t_{SA}	—	6	t_{PBcyc}	Figure 47.43, Figure 47.44
		1.8 V ≤ VCC < 2.4 V, 24 MHz < PCLKB ≤ 32 MHz		—	7		
		1.8 V ≤ VCC < 2.4 V, PCLKB ≤ 24 MHz		—	6		
		1.6 V ≤ VCC < 1.8 V		—	6		
	Slave output release time	2.4 V ≤ VCC ≤ 5.5 V	t_{REL}	—	6	t_{PBcyc}	
		1.8 V ≤ VCC < 2.4 V, 24 MHz < PCLKB ≤ 32 MHz		—	7		
		1.8 V ≤ VCC < 2.4 V, PCLKB ≤ 24 MHz		—	6		
		1.6 V ≤ VCC < 1.8 V		—	6		

Note 1. t_{PBcyc}: PCLKB cycle

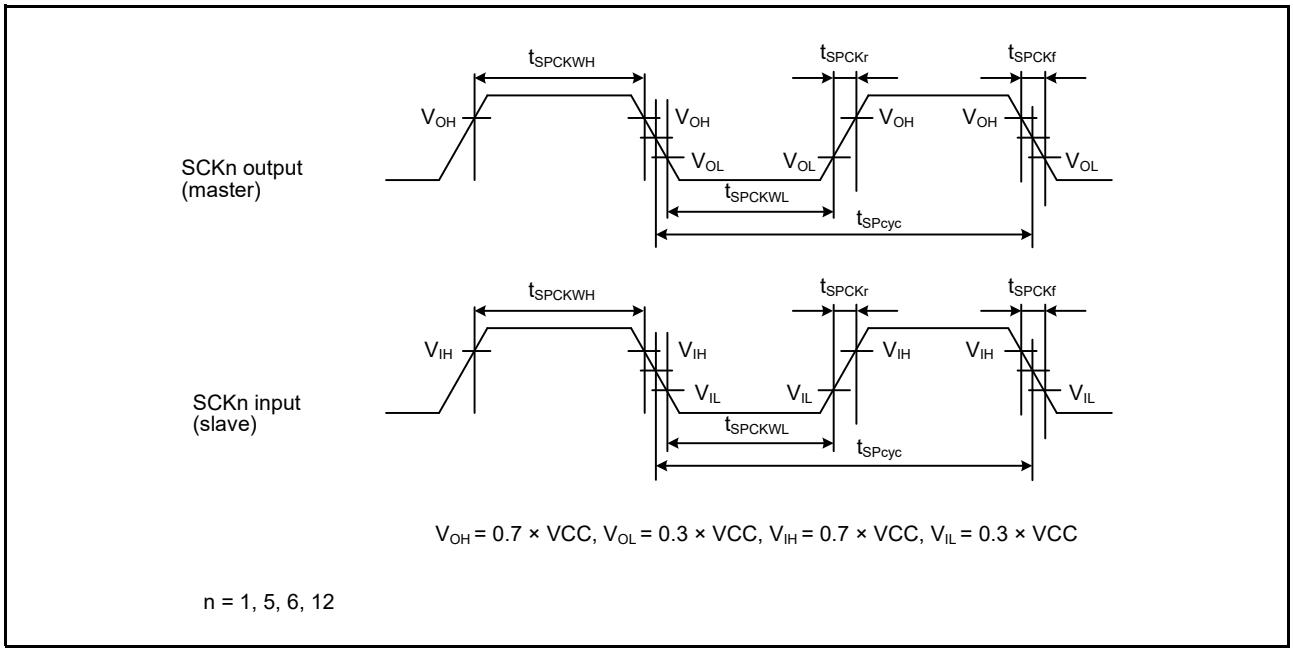


Figure 47.40 Simple SPI Clock Timing

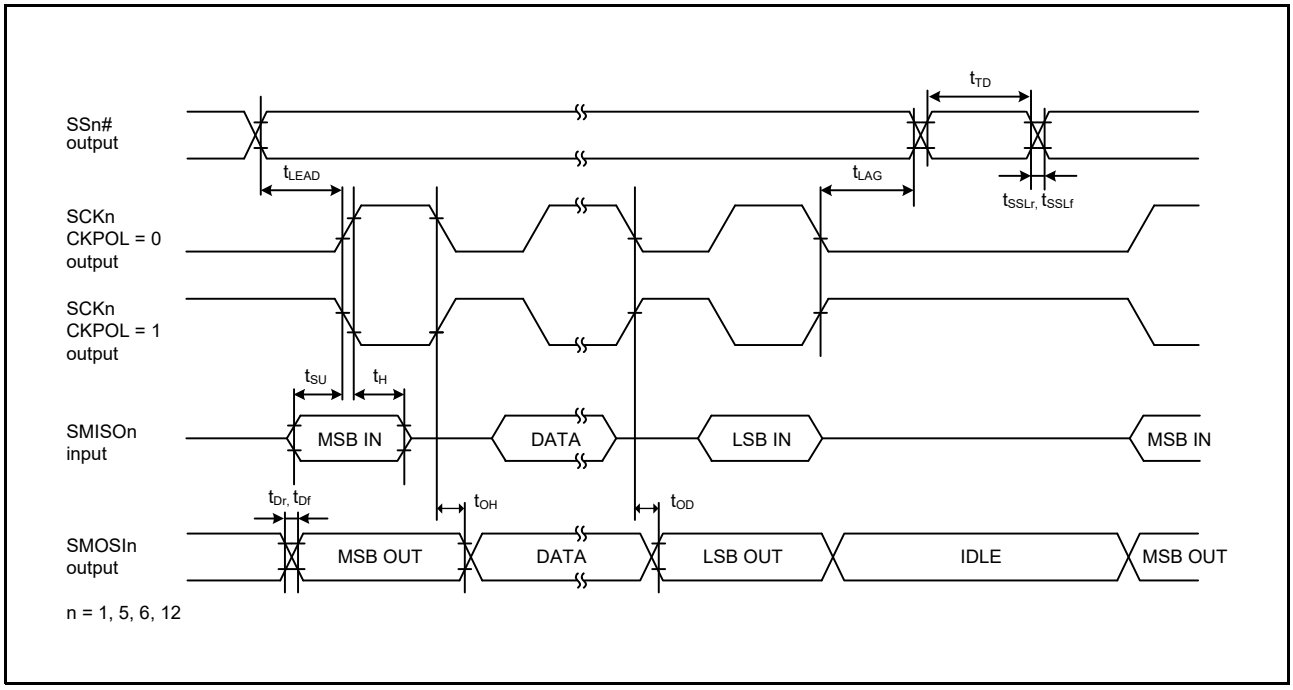


Figure 47.41 Simple SPI Clock Timing (Master, CKPH = 1)

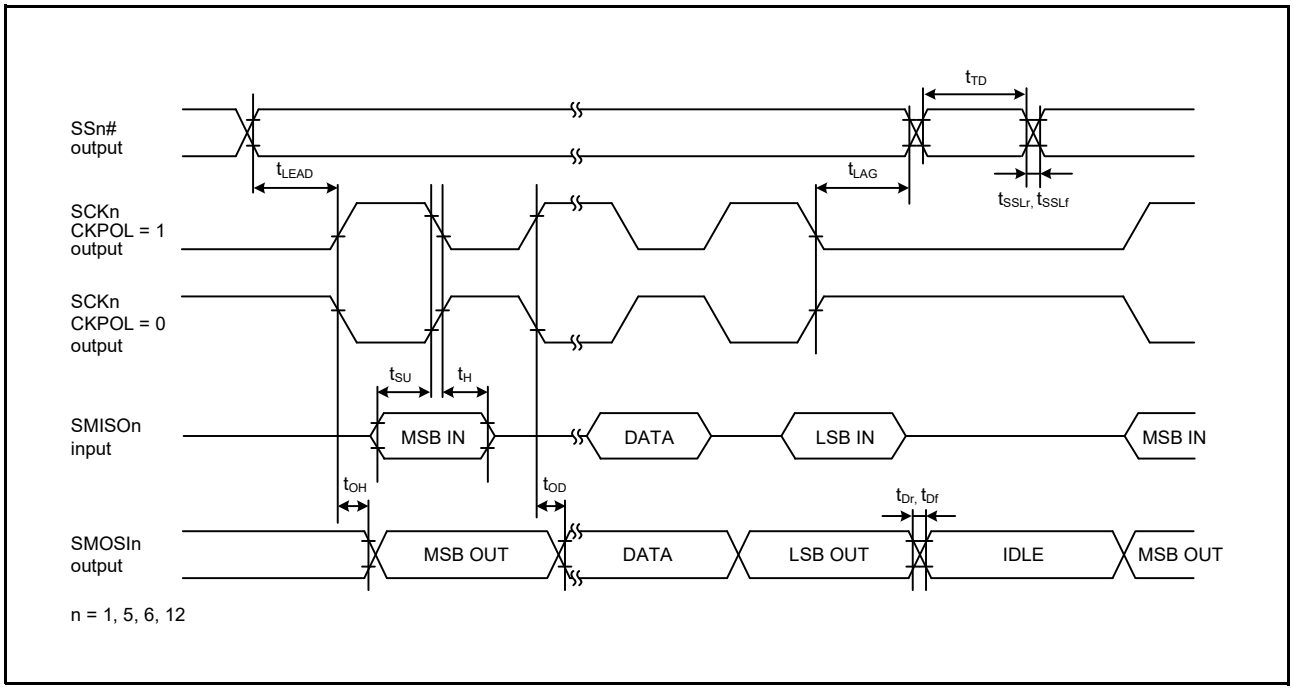


Figure 47.42 Simple SPI Clock Timing (Master, CKPH = 0)

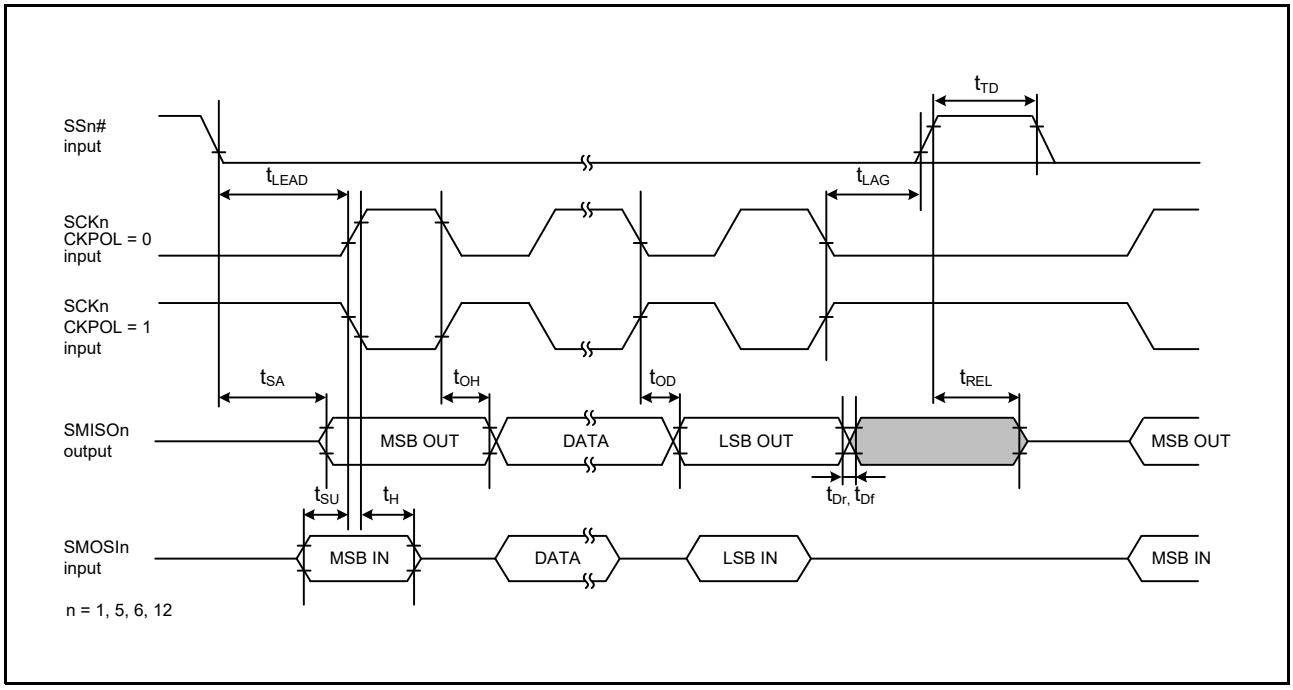


Figure 47.43 Simple SPI Clock Timing (Slave, CKPH = 1)

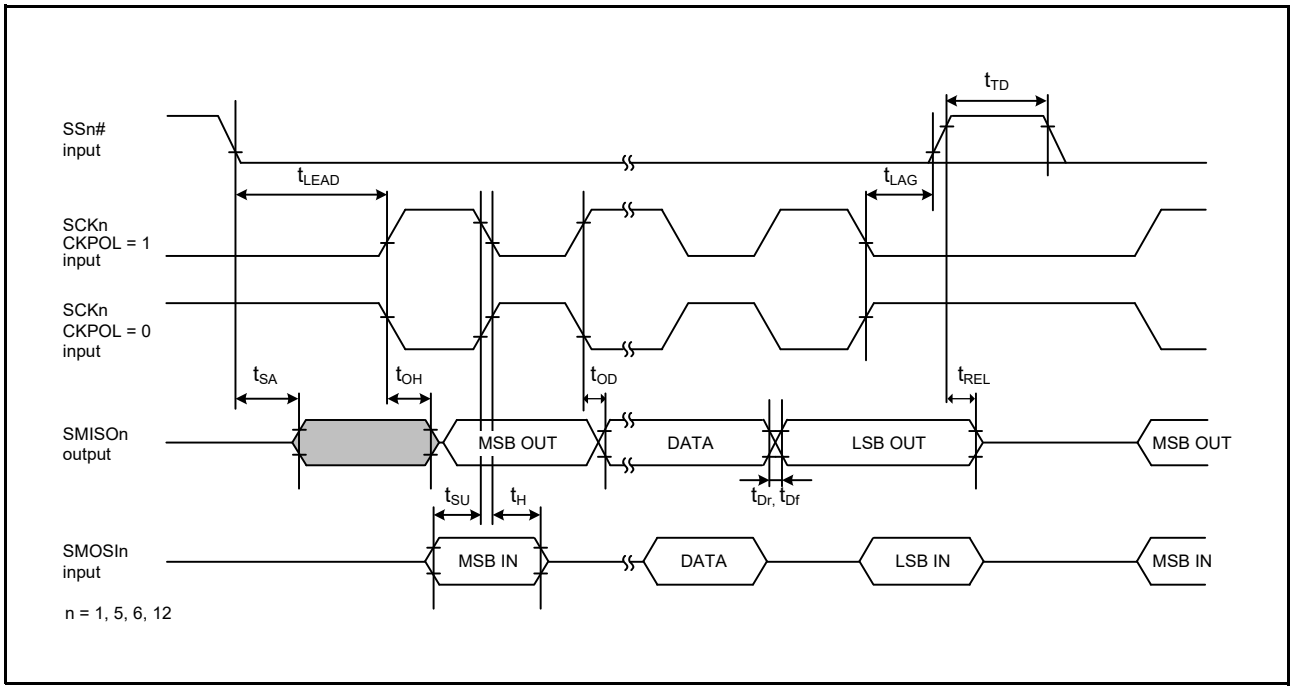


Figure 47.44 Simple SPI Clock Timing (Slave, CKPH = 0)

47.5.6.6 RSCI

Table 47.54 RSCI Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF

Item					Symbol	Min.	Max.	Unit *1	Test Conditions
RSCI (channel 0, 8, 9)	Input clock cycle time	Asynchronous			t_{Scyc}	4	—	t_{PBcyc}	Figure 47.45
		Clock synchronous	4.5 V ≤ VCC ≤ 5.5 V	24 MHz < PCLKB ≤ 32 MHz		4	—		
				PCLKB ≤ 24 MHz		2	—		
			2.4 V ≤ VCC < 4.5 V			6	—		
			1.8 V ≤ VCC < 2.4 V			8	—		
			1.6 V ≤ VCC <1.8 V			2	—		
	Input clock pulse width				t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time				t_{SCKr}	—	20	ns	
	Input clock fall time				t_{SCKf}	—	20	ns	
	Output clock cycle time	Asynchronous			t_{Scyc}	6	—	t_{PBcyc}	Figure 47.46
		Clock synchronous	4.5 V ≤ VCC ≤ 5.5 V	24 MHz < PCLKB ≤ 32 MHz		4	—		
				PCLKB ≤ 24 MHz		2	—		
			2.7 V ≤ VCC < 4.5 V			4	—		
			2.4 V ≤ VCC < 2.7 V			6	—		
			1.8 V ≤ VCC < 2.4 V	24 MHz < PCLKB ≤ 32 MHz		8	—		
				PCLKB ≤ 24 MHz		6	—		
			1.6 V ≤ VCC <1.8 V			2	—		
	Output clock pulse width				t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time			4.5 V ≤ VCC ≤ 5.5 V	t_{SCKr}	—	5	ns	
				1.8 V ≤ VCC < 4.5 V		—	20	ns	
				1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Output clock fall time			4.5 V ≤ VCC ≤ 5.5 V	t_{SCKf}	—	5	ns	
				1.8 V ≤ VCC < 4.5 V		—	20	ns	
				1.6 V ≤ VCC < 1.8 V		—	30	ns	
	Transmit data delay time (master)	Clock synchronous		4.5 V ≤ VCC ≤ 5.5 V	t_{TXD}	—	10	ns	
				1.8 V ≤ VCC < 4.5 V		—	40	ns	
				1.6 V ≤ VCC < 1.8 V		—	45	ns	
	Transmit data delay time (slave)	Clock synchronous		4.5 V ≤ VCC ≤ 5.5 V		—	30	ns	
				2.7 V ≤ VCC < 4.5 V		—	55	ns	
				2.4 V ≤ VCC < 2.7 V		—	60	ns	
				1.8 V ≤ VCC < 2.4 V		—	100	ns	
				1.6 V ≤ VCC < 1.8 V		—	125	ns	
	Receive data setup time (master)	Clock synchronous		4.5 V ≤ VCC ≤ 5.5 V	t_{RXS}	25	—	ns	
				2.7 V ≤ VCC < 4.5 V		45	—	ns	
				2.4 V ≤ VCC < 2.7 V		55	—	ns	
				1.8 V ≤ VCC < 2.4 V		90	—	ns	
				1.6 V ≤ VCC < 1.8 V		110	—	ns	
	Receive data setup time (slave)	Clock synchronous		4.5 V ≤ VCC ≤ 5.5 V		10	—	ns	
				1.8 V ≤ VCC < 4.5 V		40	—	ns	
				1.6 V ≤ VCC < 1.8 V		45	—	ns	
	Receive data hold time	Clock synchronous			t_{RXH}	10	—	ns	

Note 1. t_{PBcyc}: PCLKB cycle

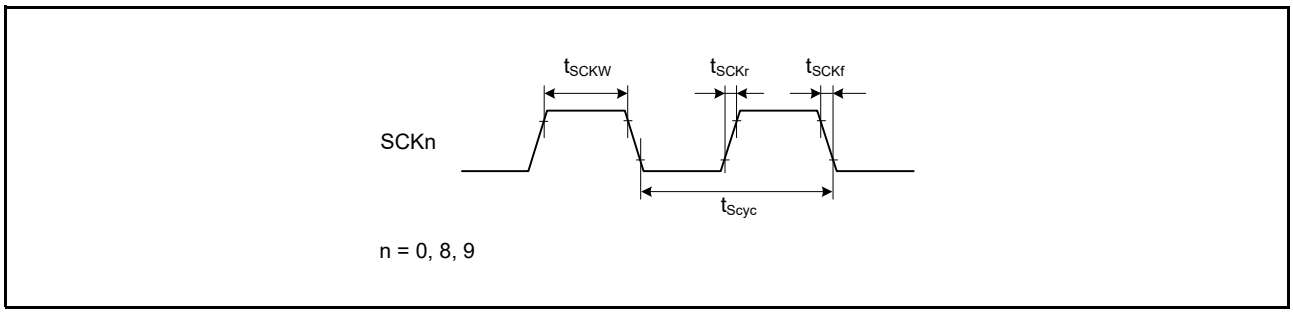


Figure 47.45 SCK Clock Input Timing

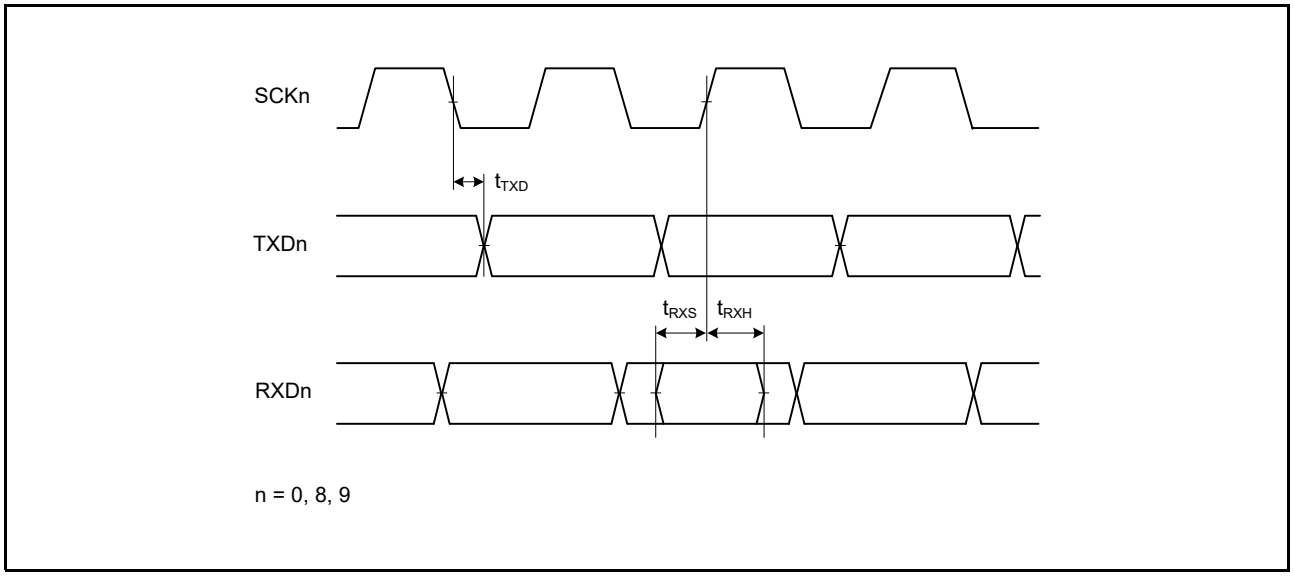


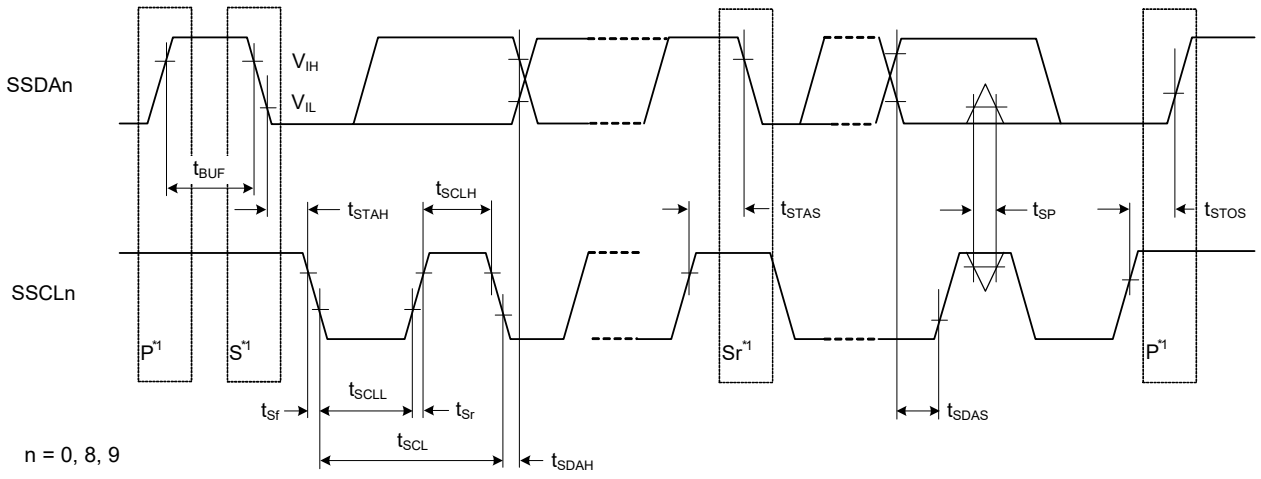
Figure 47.46 SCI Input/Output Timing: Clock Synchronous Mode

Table 47.55 Simple I²C Timing

Conditions: 2.7 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple I²C (standard mode)	SDA rise time	t_{Sr}	—	1000	ns	Figure 47.47
	SDA fall time	t_{Sf}	—	300	ns	
	SDA spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data setup time	t_{SDAS}	250	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	
Simple I²C (fast mode)	SDA rise time	t_{Sr}	—	300	ns	Figure 47.47
	SDA fall time	t_{Sf}	—	300	ns	
	SDA spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data setup time	t_{SDAS}	100	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	

Note: t_{Pcyc}: PCLKB cycle
Note 1. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: START condition
P: STOP condition
Sr: Repeated START condition

Figure 47.47 Output Timing and Simple I²C Bus Interface Input/Output Timing

Table 47.56 Simple SPI Timing (1/2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item				Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)	4.5 V ≤ VCC ≤ 5.5 V	24 MHz < PCLKB ≤ 32 MHz	t_{SPcyc}	4	65536	t_{PBcyc}	Figure 47.48
			PCLKB ≤ 24 MHz		2	65536		
		2.7 V ≤ VCC < 4.5 V			4	65536		
		2.4 V ≤ VCC < 2.7 V			6	65536		
		1.8 V ≤ VCC < 2.4 V	24 MHz < PCLKB ≤ 32 MHz		8	65536		
			PCLKB ≤ 24 MHz		6	65536		
		1.6 V ≤ VCC < 1.8 V			2	65536		
	SCK clock cycle input (slave)	4.5 V ≤ VCC ≤ 5.5 V	24 MHz < PCLKB ≤ 32 MHz	t_{SPcyc}	4	—	t_{PBcyc}	
			PCLKB ≤ 24 MHz		2	—		
		2.7 V ≤ VCC < 4.5 V			6	—		
		1.8 V ≤ VCC < 2.4 V			8	—		
		1.6 V ≤ VCC < 1.8 V			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	4.5 V ≤ VCC ≤ 5.5 V		t_{SPCKr}, t_{SPCKf}	—	5	ns	
		1.8 V ≤ VCC < 4.5 V			—	20	ns	
		1.6 V ≤ VCC < 1.8 V			—	30	ns	
	Data input setup time (master)	4.5 V ≤ VCC ≤ 5.5 V		t_{SU}	25	—	ns	Figure 47.49, Figure 47.50
		2.7 V ≤ VCC < 4.5 V			45	—		
		2.4 V ≤ VCC < 2.7 V			55	—		
		1.8 V ≤ VCC < 2.4 V			80	—		
		1.6 V ≤ VCC < 1.8 V			110	—		
	Data input setup time (slave)	4.5 V ≤ VCC ≤ 5.5 V			10	—		
		1.8 V ≤ VCC < 4.5 V			40	—		
		1.6 V ≤ VCC < 1.8 V			45	—		
	Data input hold time			t_H	10	—	ns	
	SSL input setup time	t_{SPcyc} < 6 t_{PBcyc}		t_{LEAD}	2	—	t_{SPcyc}	
		t_{SPcyc} ≥ 6 t_{PBcyc}			1	—		
	SSL input hold time			t_{LAG}	1	—	t_{SPcyc}	
	Data output delay time (master)	4.5 V ≤ VCC ≤ 5.5 V		t_{OD}	—	10	ns	
		1.8 V ≤ VCC < 4.5 V			—	40		
		1.6 V ≤ VCC < 1.8 V			—	50		
	Data output delay time (slave)	4.5 V ≤ VCC ≤ 5.5 V			—	30		
		2.4 V ≤ VCC < 4.5 V			—	65		
		1.8 V ≤ VCC < 2.4 V			—	100		
		1.6 V ≤ VCC < 1.8 V			—	125		
	Data output hold time (master)	2.7 V ≤ VCC ≤ 5.5 V		t_{OH}	–10	—	ns	
		1.8 V ≤ VCC < 2.7 V			–20	—		
		1.6 V ≤ VCC < 1.8 V			–40	—		
	Data output hold time (slave)				–10	—		
	Data rise/fall time	4.5 V ≤ VCC ≤ 5.5 V		t_{Dr}, t_{Df}	—	5	ns	
		1.8 V ≤ VCC < 4.5 V			—	20		
		1.6 V ≤ VCC < 1.8 V			—	30		

Table 47.56 Simple SPI Timing (2/2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item				Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SSL input rise/fall time			t_{SSLr}, t_{SSLf}	—	20	ns	Figure 47.49, Figure 47.50
	Slave access time	2.4 V ≤ VCC ≤ 5.5 V		t_{SA}	—	6	t_{PBcyc}	Figure 47.51, Figure 47.52
		1.8 V ≤ VCC < 2.4 V	24 MHz < PCLKB ≤ 32 MHz		—	7		
			PCLKB ≤ 24 MHz		—	6		
		1.6 V ≤ VCC < 1.8 V			—	6		
	Slave output release time	2.4 V ≤ VCC ≤ 5.5 V		t_{REL}	—	6	t_{PBcyc}	
		1.8 V ≤ VCC < 2.4 V	24 MHz < PCLKB ≤ 32 MHz		—	7		
			PCLKB ≤ 24 MHz		—	6		
		1.6 V ≤ VCC < 1.8 V			—	6		

Note 1. t_{PBcyc}: PCLKB cycle

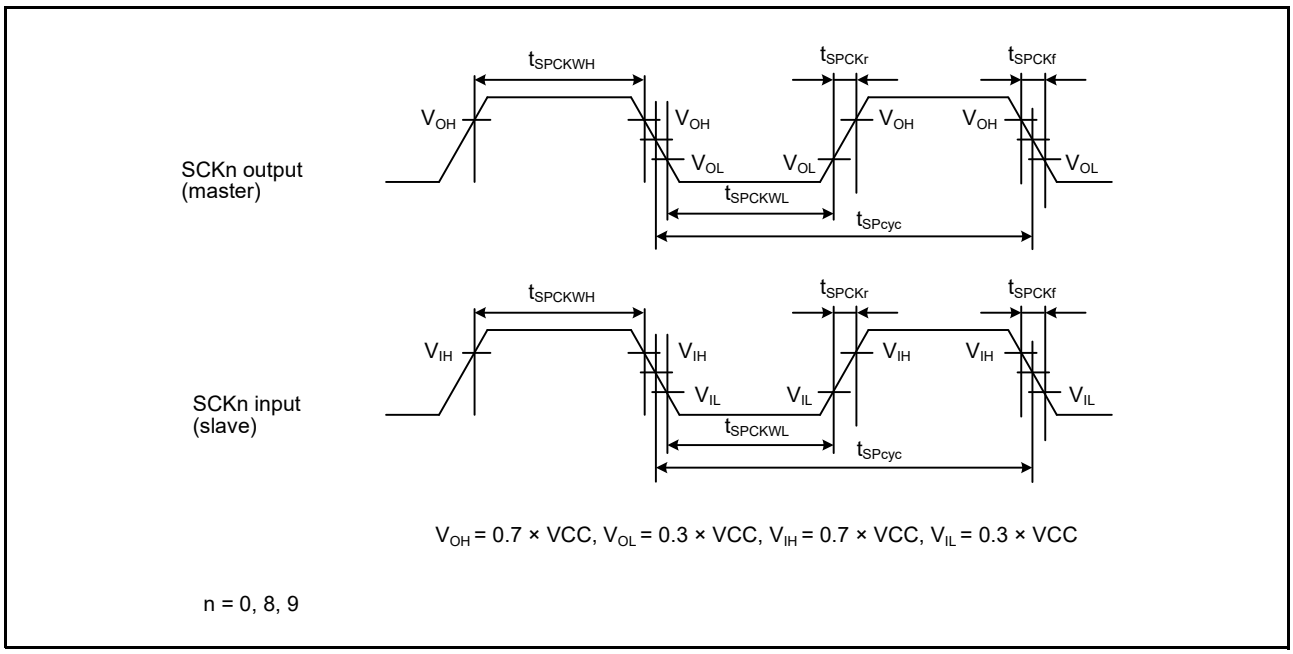


Figure 47.48 Simple SPI Clock Timing

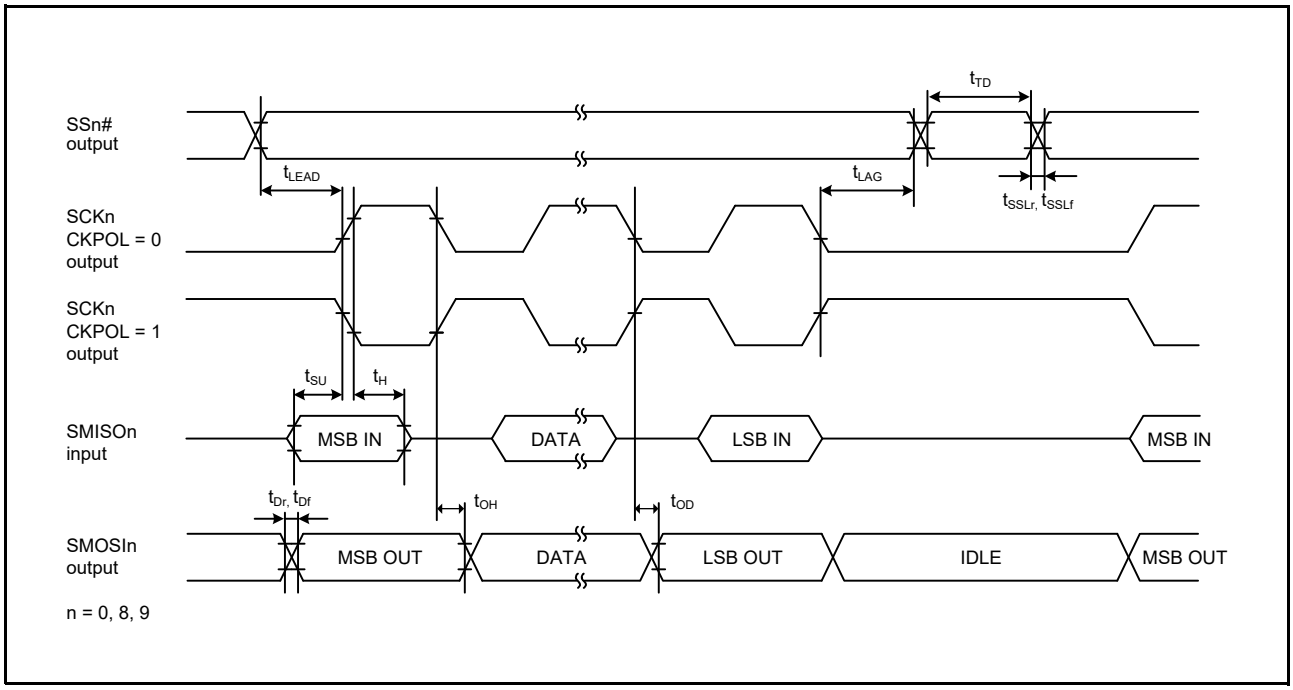


Figure 47.49 Simple SPI Clock Timing (Master, CPHA = 1)

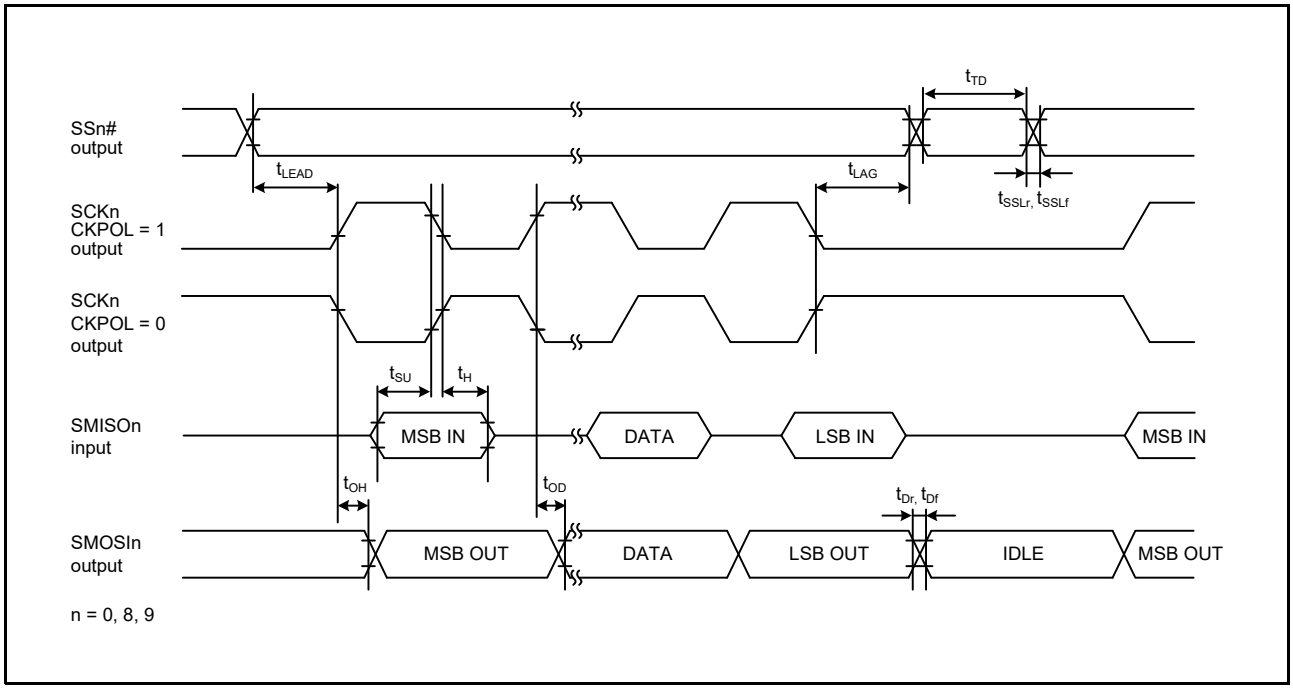


Figure 47.50 Simple SPI Clock Timing (Master, CPHA = 0)

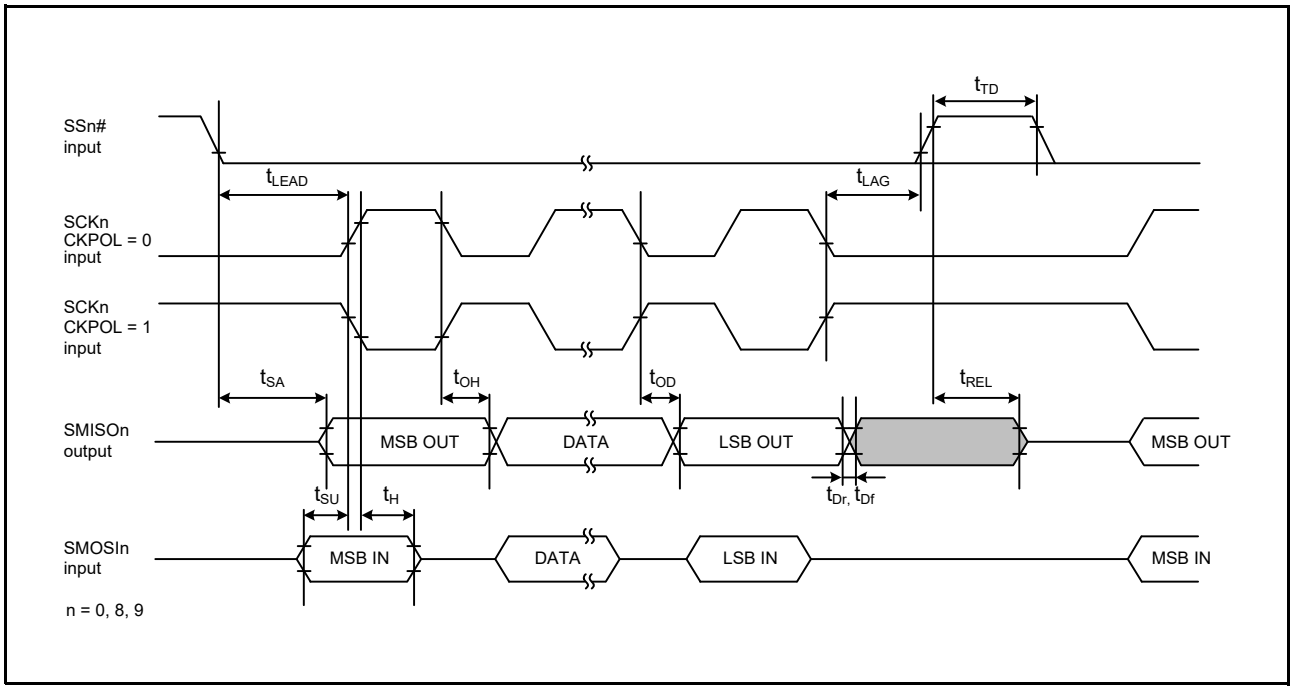


Figure 47.51 Simple SPI Clock Timing (Slave, CPHA = 1)

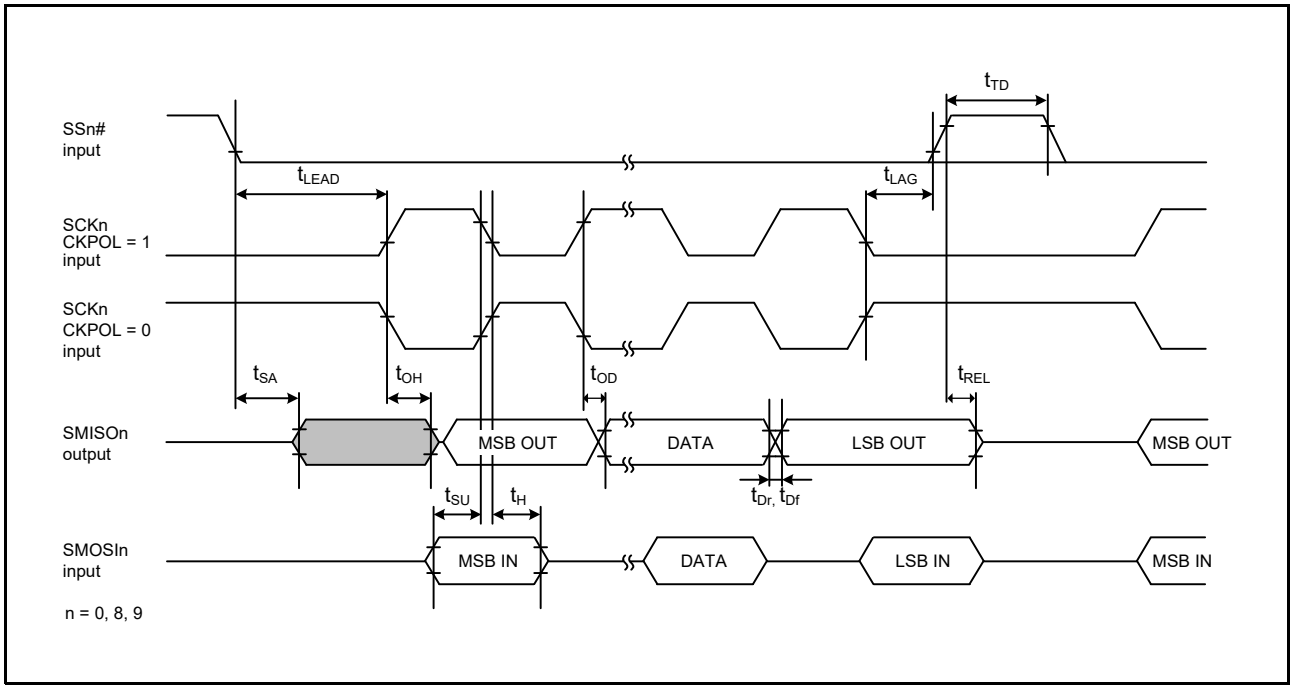


Figure 47.52 Simple SPI Clock Timing (Slave, CPHA = 0)

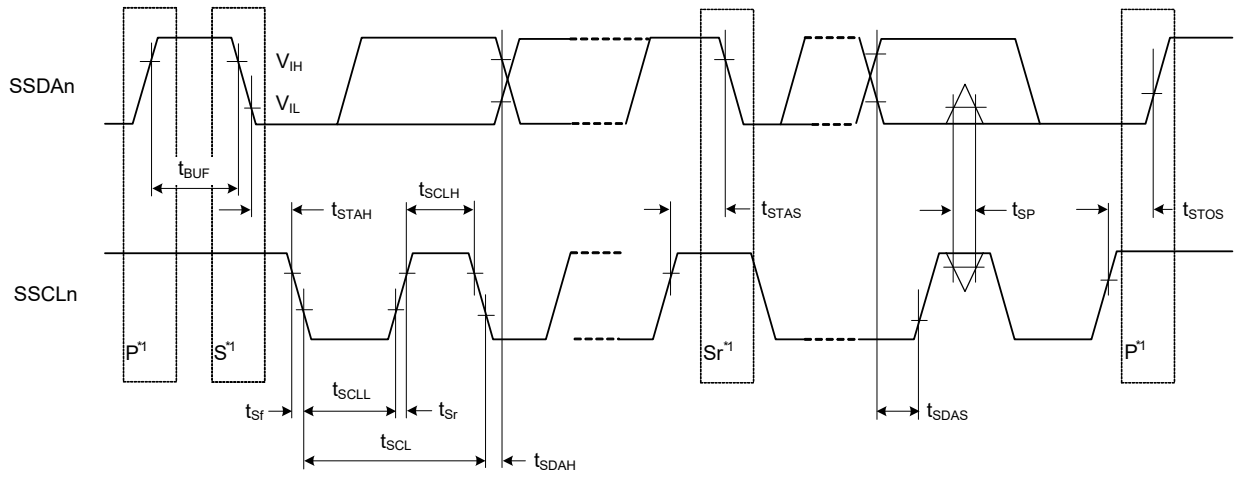
47.5.6.7 RIIC

Table 47.57 RIIC Timing

Conditions: 2.7 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
RIIC (standard mode, SMBus)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 47.53
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA rise time	t_{Sr}	—	1000	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition setup time	t_{STAS}	1000	—	ns	
	STOP condition setup time	t_{STOS}	1000	—	ns	
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*2	—	400	pF	
RIIC (fast mode)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	—	ns	Figure 47.53
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA rise time	t_{Sr}	—	300	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition setup time	t_{STAS}	300	—	ns	
	STOP condition setup time	t_{STOS}	300	—	ns	
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*2	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference count clock (IICφ) cycle
Note 1. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bit = 1.
Note 2. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: START condition
P: STOP condition
Sr: Repeated START condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Figure 47.53 RIIC Bus Interface Input/Output Timing

47.5.6.8 RSPI

Table 47.58 RSPI Timing (1/2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C, C = 30 pF,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item				Symbol	Min.	Max.	Unit	Test Conditions
RSPI	RSPCK clock cycle	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{SPcyc}	2	4096	t_{PBcyc} *1	Figure 47.54
			1.8 V ≤ VC C< 2.7 V		4	4096		
			1.6 V ≤ VCC < 1.8 V		2	4096		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		4	—		
			2.4 V ≤ VCC < 2.7 V		6	—		
			1.8 V ≤ VCC < 2.4 V		8	—		
			1.6 V ≤ VCC < 1.8 V		4	—		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	(t_{SPcyc} – t_{SPCKr} – t_{SPCKf}) / 2 – 3	—	ns	
		Slave			(t_{SPcyc} – t_{SPCKr} – t_{SPCKf}) / 2	—		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	(t_{SPcyc} – t_{SPCKr} – t_{SPCKf}) / 2 – 3	—	ns	
		Slave			(t_{SPcyc} – t_{SPCKr} – t_{SPCKf}) / 2	—		
	RSPCK clock rise/ fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	—	10	ns	
			2.4 V ≤ VCC < 2.7 V		—	15		
			1.8 V ≤ VCC < 2.4 V		—	20		
			1.6 V ≤ VCC < 1.8 V		—	30		
		Input			—	0.1	µs/V	
	Data input setup time	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{SU}	10	—	ns	Figure 47.55 to Figure 47.60
			1.8 V ≤ VCC < 2.7 V		30	—		
			1.6 V ≤ VCC < 1.8 V		10	—		
		Slave	2.4 V ≤ VCC ≤ 5.5 V		10	—		
			1.8 V ≤ VCC < 2.4 V		15	—		
			1.6 V ≤ VCC < 1.8 V		20	—		
	Data input hold time	Master	RSPCK set to a division ratio other than PCLKB divided by 2	t_H	t_{PBcyc}	—	ns	
			RSPCK set to PCLKB divided by 2	t_{HF}	0	—		
		Slave		t_H	20	—		
	SSL setup time	Master	1.8 V ≤ VCC ≤ 5.5 V	t_{LEAD}	–30 + N*2 × t_{SPcyc}	—	ns	
			1.6 V ≤ VCC < 1.8 V		–50 + N*2 × t_{SPcyc}	—		
		Slave			6	—	t_{PBcyc}	
	SSL hold time	Master		t_{LAG}	–30 + N*3 × t_{SPcyc}	—	ns	
		Slave			6	—	t_{PBcyc}	
	Data output delay time	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{OD}	—	14	ns	
			2.4 V ≤ VCC < 2.7 V		—	20		
			1.8 V ≤ VCC < 2.4 V		—	25		
			1.6 V ≤ VCC < 1.8 V		—	30		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		—	50		
			2.4 V ≤ VCC < 2.7 V		—	60		
			1.8 V ≤ VCC < 2.4 V		—	85		
			1.6 V ≤ VCC < 1.8 V		—	110		

Table 47.58 RSPI Timing (2/2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C, C = 30 pF,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item				Symbol	Min.	Max.	Unit	Test Conditions
RSPI	Data output hold time	Master		t_{OH}	0	—	ns	Figure 47.55 to Figure 47.60
		Slave			0	—		
	Successive transmission delay time	Master		t_{TD}	t_{SPcyc} + 2 × t_{PBcyc}	8 × t_{SPcyc} + 2 × t_{PBcyc}	ns	
		Slave			6 × t_{PBcyc}	—		
	MOSI and MISO rise/fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	—	10	ns	
			2.4 V ≤ VCC < 2.7 V		—	15		
			1.8 V ≤ VCC < 2.4 V		—	20		
			1.6 V ≤ VCC < 1.8 V		—	30		
		Input			—	1	µs	
	SSL rise/fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{SSLr}, t_{SSLf}	—	10	ns	
			2.4 V ≤ VCC < 2.7 V		—	15	ns	
			1.8 V ≤ VCC < 2.4 V		—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30	ns	
		Input			—	1	µs	
	Slave access time		2.4 V ≤ VCC ≤ 5.5 V	t_{SA}	—	2 × t_{PBcyc} + 100	ns	Figure 47.59, Figure 47.60
			1.8 V ≤ VCC < 2.4 V		—	2 × t_{PBcyc} + 140	ns	
			1.6 V ≤ VCC < 1.8 V		—	2 × t_{PBcyc} + 180	ns	
	Slave output release time		2.4 V ≤ VCC ≤ 5.5 V	t_{REL}	—	2 × t_{PBcyc} + 100	ns	
			1.8 V ≤ VCC < 2.4 V		—	2 × t_{PBcyc} + 140	ns	
			1.6 V ≤ VCC < 1.8 V		—	2 × t_{PBcyc} + 180	ns	

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. N: An integer from 1 to 8 that can be set by the RSPI clock delay register (SPCKD)
Note 3. N: An integer from 1 to 8 that can be set by the RSPI slave select negation delay register (SSLND)

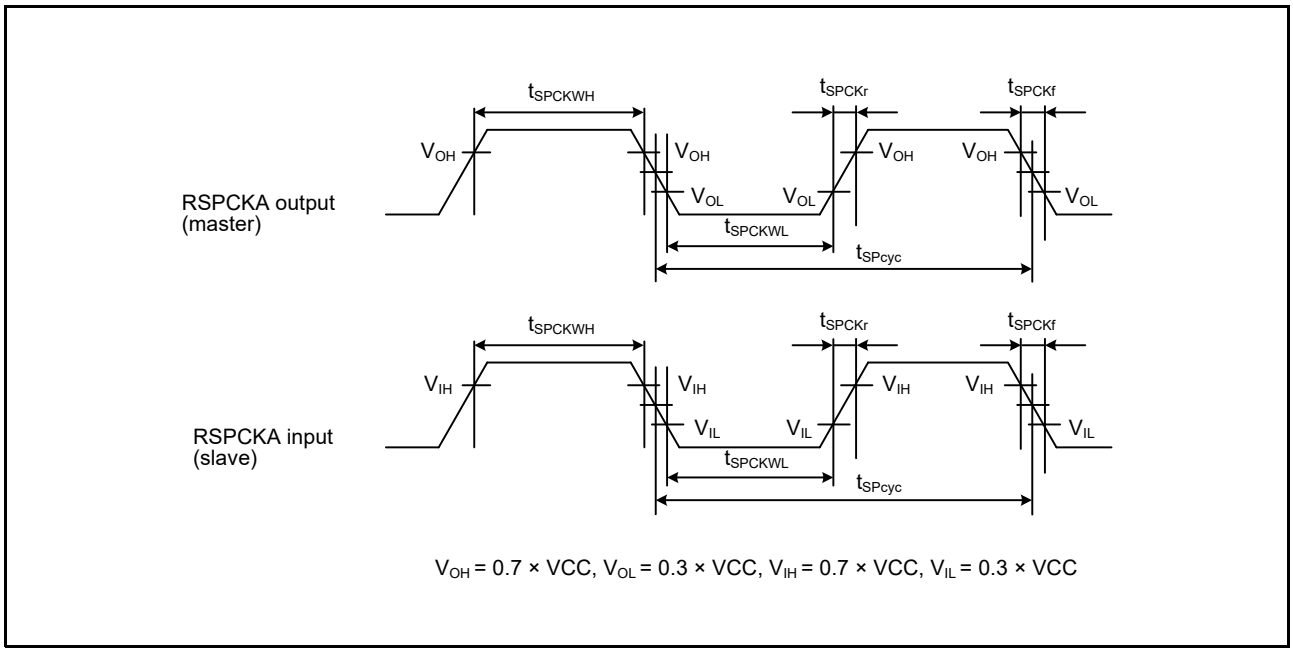


Figure 47.54 RSPI Clock Timing

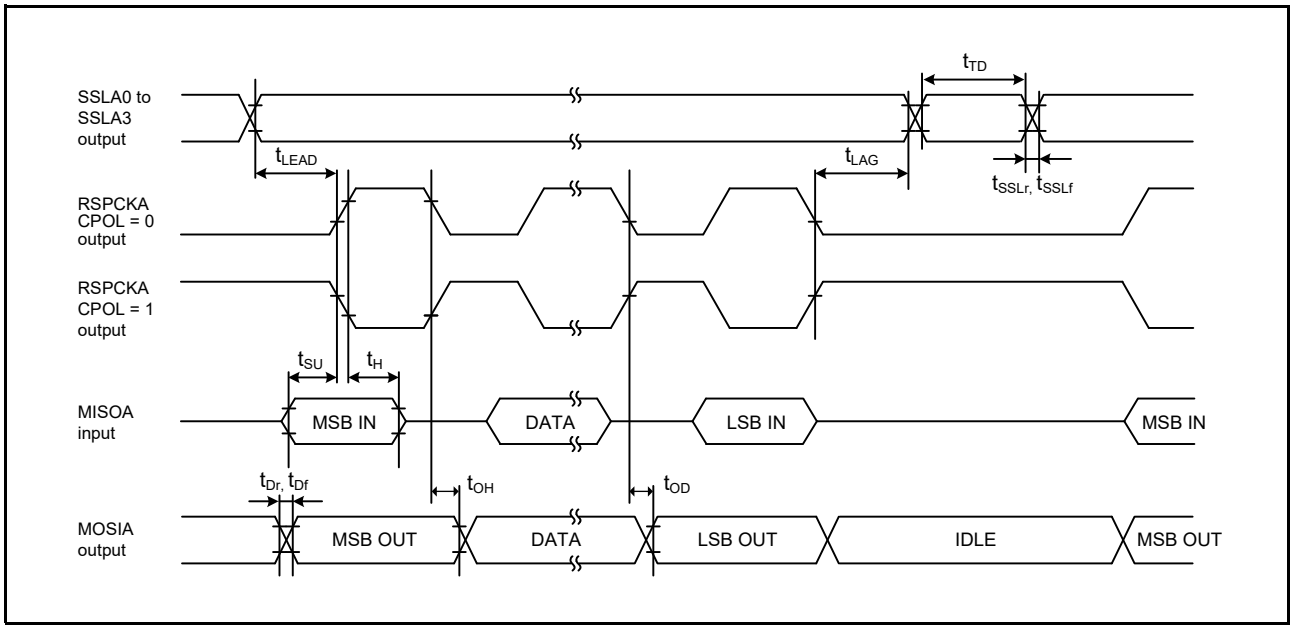


Figure 47.55 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

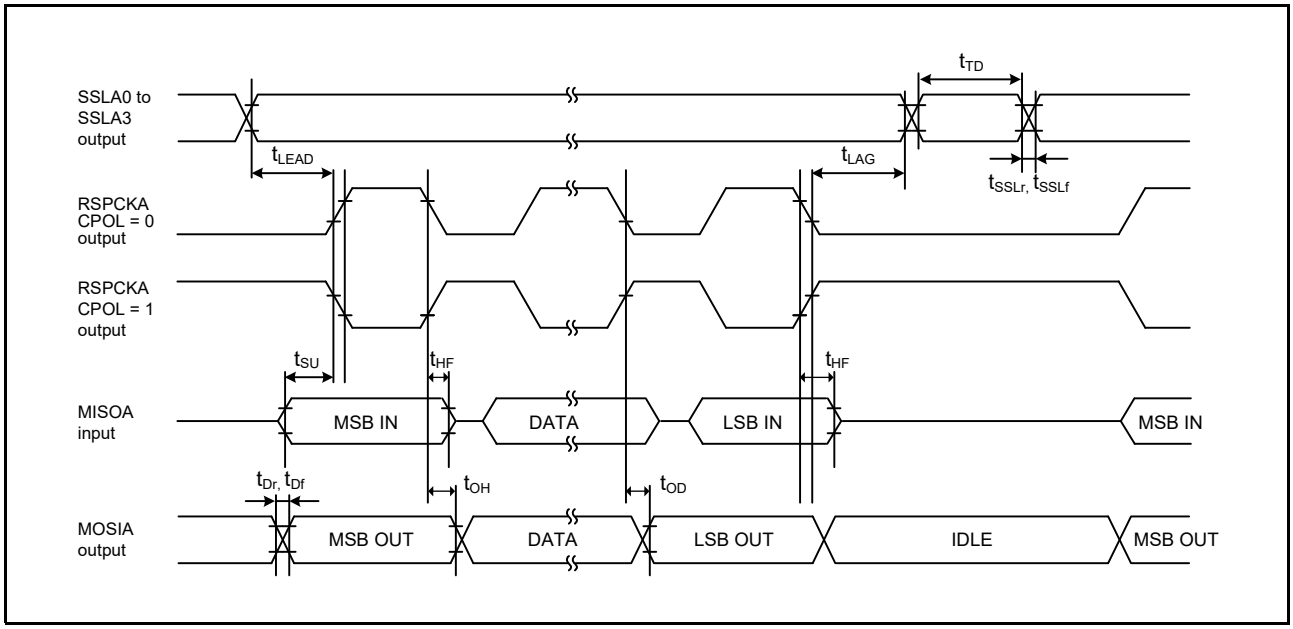


Figure 47.56 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to 1/2)

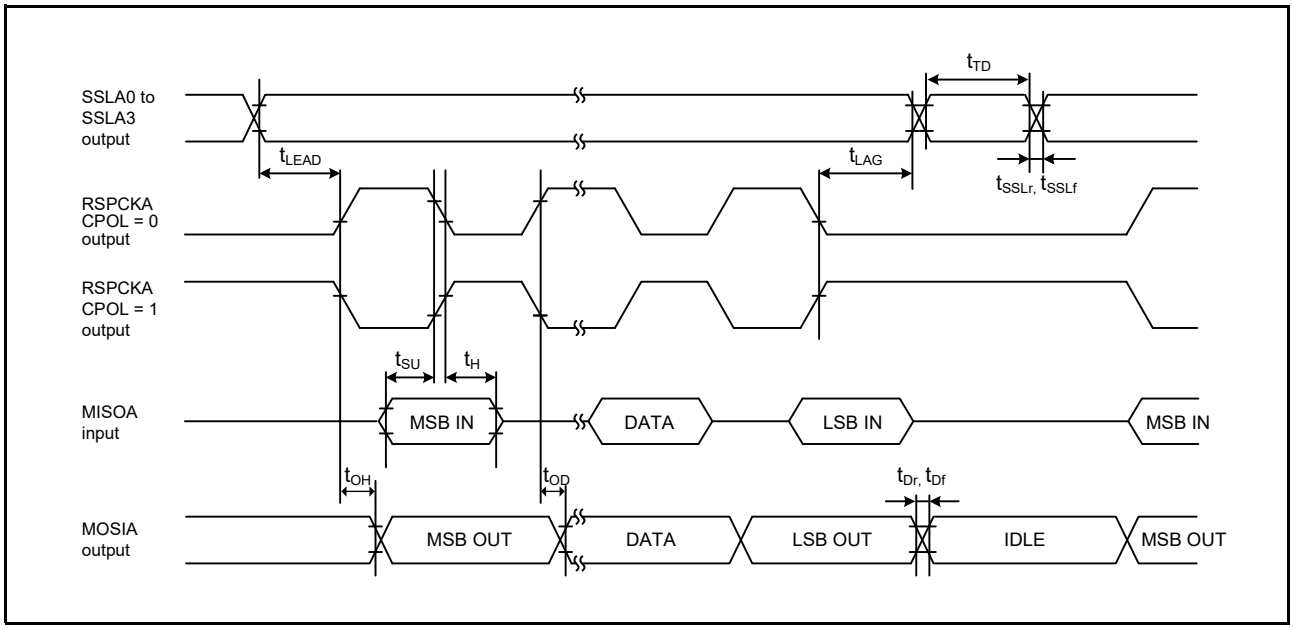


Figure 47.57 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

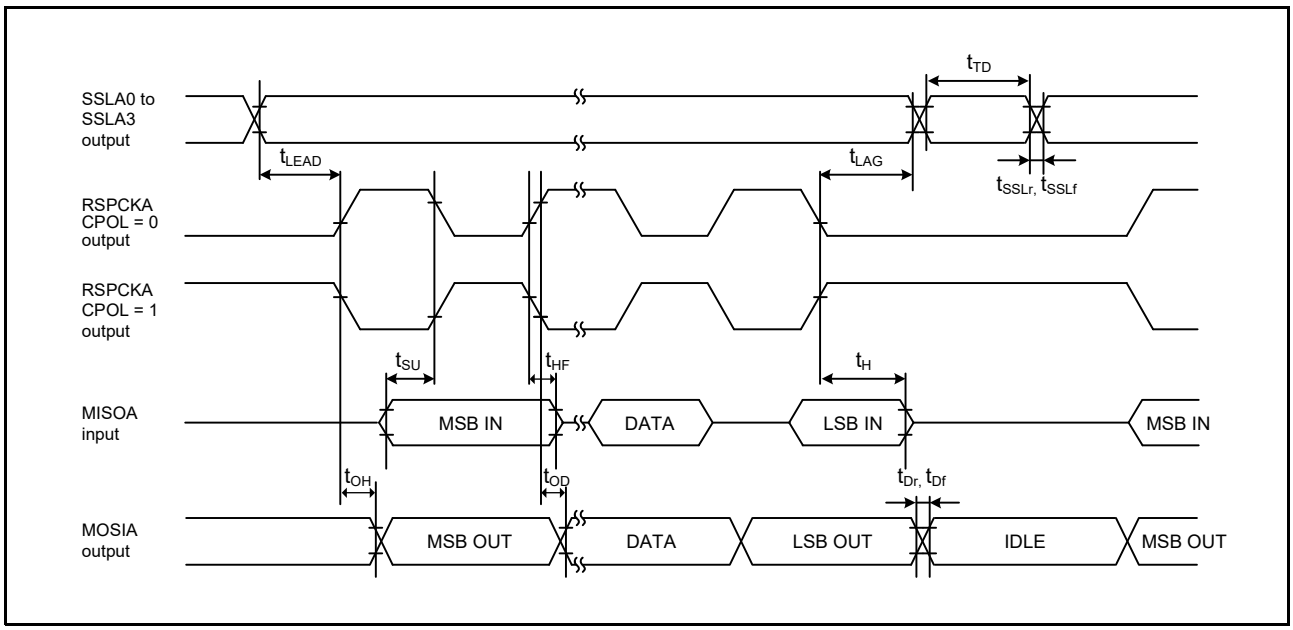


Figure 47.58 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to 1/2)

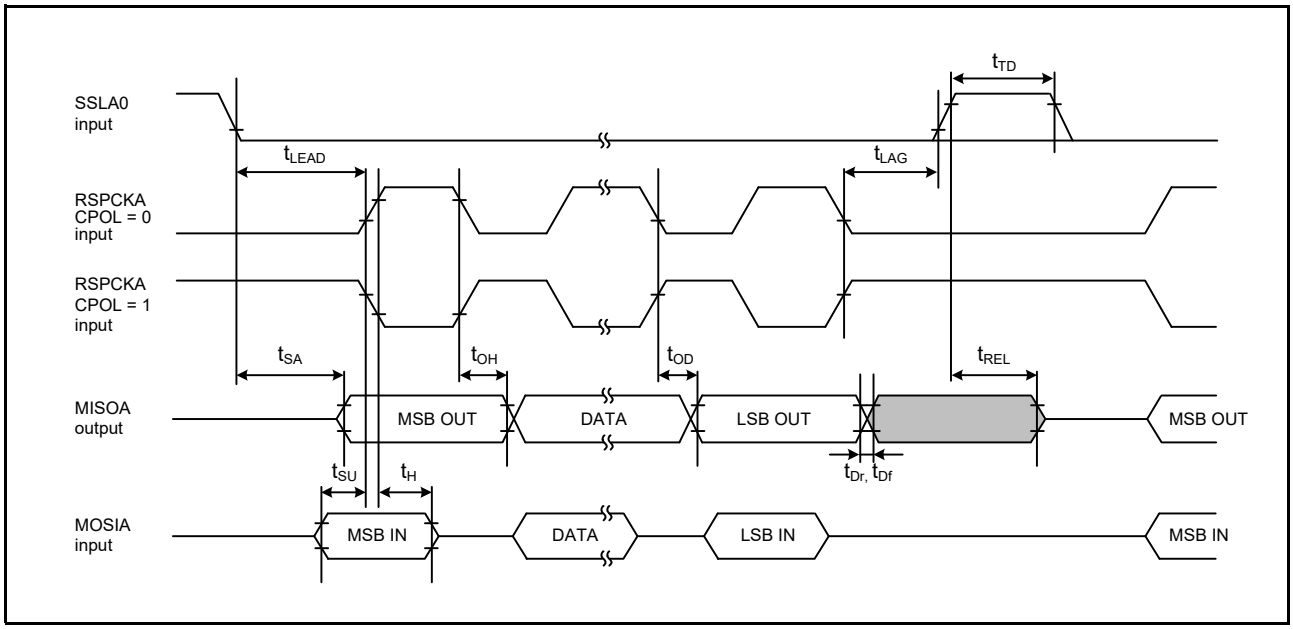


Figure 47.59 RSPI Timing (Slave, CPHA = 0)

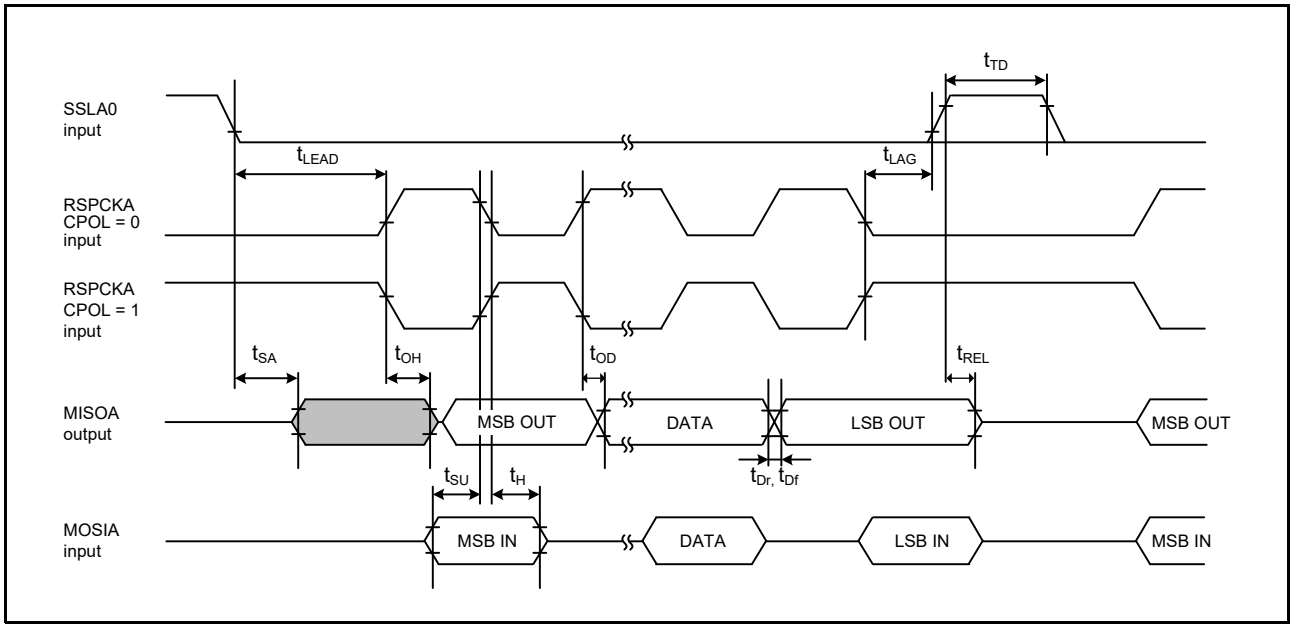


Figure 47.60 RSPI Timing (Slave, CPHA = 1)

47.5.6.9 CANFD

Table 47.59 CANFD Timing

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.8 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item		Symbol	Min.	Max.	Unit	Test Conditions
Internal delay time	2.4 V ≤ VCC ≤ 5.5 V	t_{NODE}	—	50	ns	Figure 47.61
	1.8 V ≤ VCC < 2.4 V		—	75		

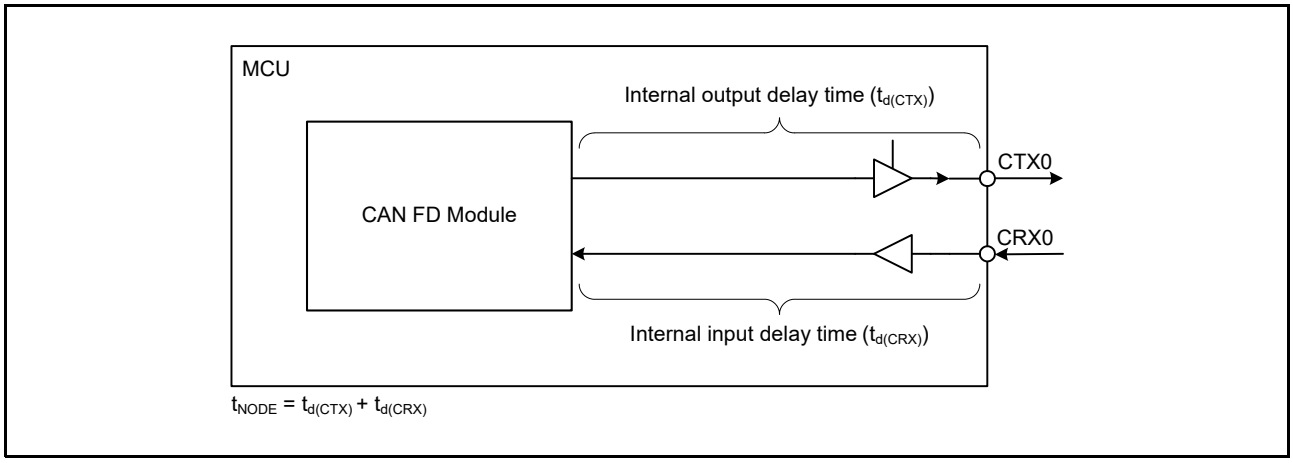


Figure 47.61 Definition of Internal Delay Time

47.5.6.10 A/D Converter Trigger

Table 47.60 A/D Converter Trigger Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Max.	Unit *1	Test Conditions
A/D converter	Trigger input pulse width	t_{TRGW}	1.5	—	t_{PBcyc}	Figure 47.62

Note 1. t_{PBcyc}: PCLKB cycle

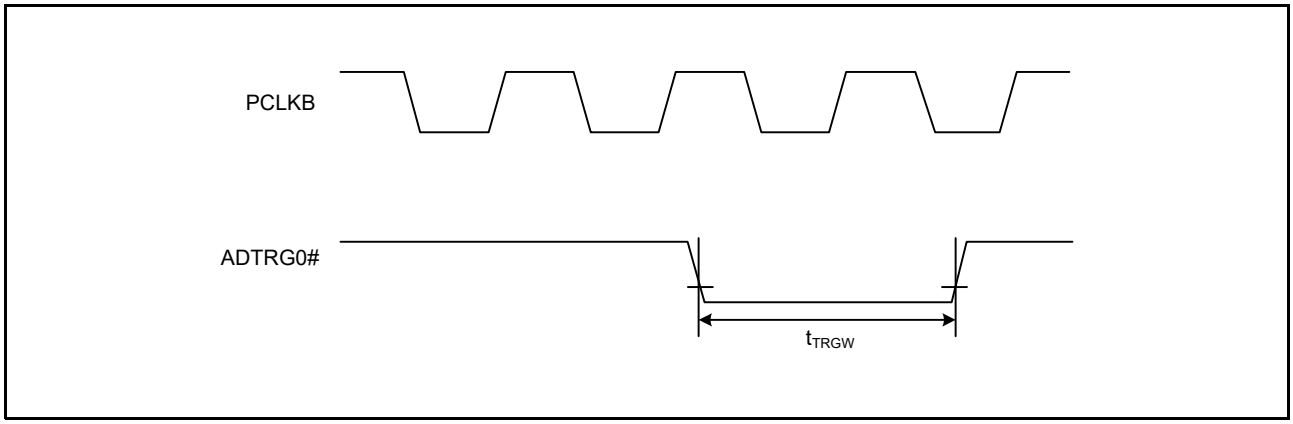


Figure 47.62 A/D Converter External Trigger Input Timing

47.5.6.11 CAC

Table 47.61 CAC Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit *1	Test Conditions
CAC	CACREF input pulse width	$t_{PBcyc} \leq t_{cac}$ *2	t_{CACREF}	4.5 t_{cac} + 3 t_{PBcyc}	—	ns	
		$t_{PBcyc} > t_{cac}$ *2		5 t_{cac} + 6.5 t_{PBcyc}			
	CACREF input rise/fall time		$t_{CACREFr}$, $t_{CACREFf}$	—	0.1	µs/V	

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. t_{cac}: CAC count clock source cycle

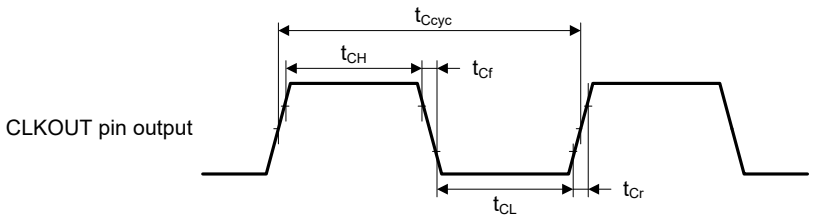
47.5.6.12 CLKOUT

Table 47.62 CLKOUT Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C,
Output load conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Item			Symbol	Min.	Max.	Unit	Test Conditions
CLKOUT	CLKOUT pin output cycle*2	2.7 V ≤ VCC ≤ 5.5 V	t_{Ccyc}	62.5	—	ns	Figure 47.63
		1.8 V ≤ VCC < 2.7 V		125			
		1.6 V ≤ VCC < 1.8 V		250			
	CLKOUT pin high pulse width*1	2.7 V ≤ VCC ≤ 5.5 V	t_{CH}	15	—	ns	
		1.8 V ≤ VCC < 2.7 V		30			
		1.6 V ≤ VCC < 1.8 V		80			
	CLKOUT pin low pulse width*1	2.7 V ≤ VCC ≤ 5.5 V	t_{CL}	15	—	ns	
		1.8 V ≤ VCC < 2.7 V		30			
		1.6 V ≤ VCC < 1.8 V		80			
	CLKOUT pin output rise time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cr}	—	12	ns	
		1.8 V ≤ VCC < 2.7 V			25		
		1.6 V ≤ VCC < 1.8 V			30		
	CLKOUT pin output fall time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cf}	—	12	ns	
		1.8 V ≤ VCC < 2.7 V			25		
		1.6 V ≤ VCC < 1.8 V			30		

Note 1. When the LOCO is selected as the clock output source (CKOCR.CKOSEL[3:0] bits = 0000b), set the clock output division ratio selection to divided by 2 (CKOCR.CKODIV[2:0] bits = 001b).
Note 2. When the XTAL external clock input or an oscillator is used with divided by 1 (CKOCR.CKOSEL[3:0] bits = 010b and CKOCR.CKODIV[2:0] bits = 000b) to output from CLKOUT, the above should be satisfied with an input duty cycle of 45 to 55%.



Test conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, I_{OH} = -1.0 mA, I_{OL} = 1.0 mA, C = 30 pF

Figure 47.63 CLKOUT Output Timing

47.6 USB Characteristics

Table 47.63 USB Characteristics (USB0_DP and USB0_DM Pin Characteristics)

Conditions: 3.0 V ≤ VCC ≤ 3.6 V, 3.0 V ≤ AVCC0 ≤ 3.6 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions	
Input characteristics	Input high level voltage		V_{IH}	2.0	—	V		
	Input low level voltage		V_{IL}	—	0.8	V		
	Differential input sensitivity		V_{DI}	0.2	—	V	| USB0_DP – USB0_DM |	
	Differential common mode range		V_{CM}	0.8	2.5	V		
Output characteristics	Output high level voltage		V_{OH}	2.8	VCC	V	I_{OH} = –200 µA	
	Output low level voltage		V_{OL}	0.0	0.3	V	I_{OL} = 2 mA	
	Cross-over voltage		V_{CRS}	1.3	2.0	V		Figure 47.64, Figure 47.65
	Rise time	FS	t_r	4	20	ns		
		LS		75	300			
	Fall time	FS	t_f	4	20	ns		
		LS		75	300			
	Rise/fall time ratio	FS	t_r/t_f	90	111.11	%	t_r/t_f	
		LS		80	125			
	Output resistance		Z_{DRV}	28	44	Ω	(Adjusting the resistance by external elements is not necessary.)	
Pull-up, pull-down	Pull-down resistor		R_{PD}	14.25	24.80	kΩ		
	Pull-up resistor		R_{PUI}	0.9	1.575	kΩ	During idle state	
			R_{PUA}	1.425	3.09	kΩ	During transmission and reception	

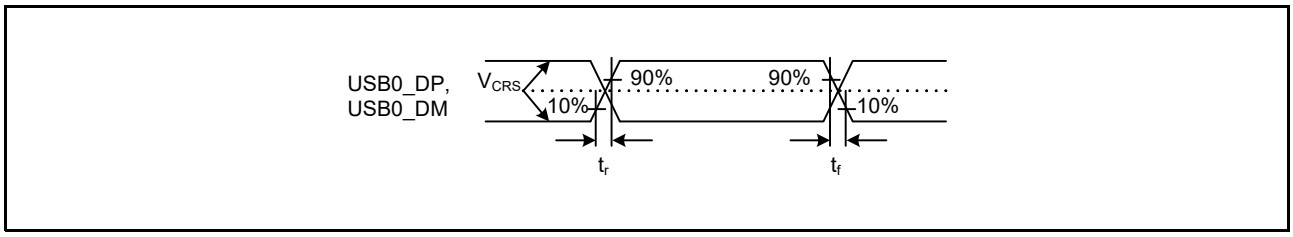


Figure 47.64 USB0_DP and USB0_DM Output Timing

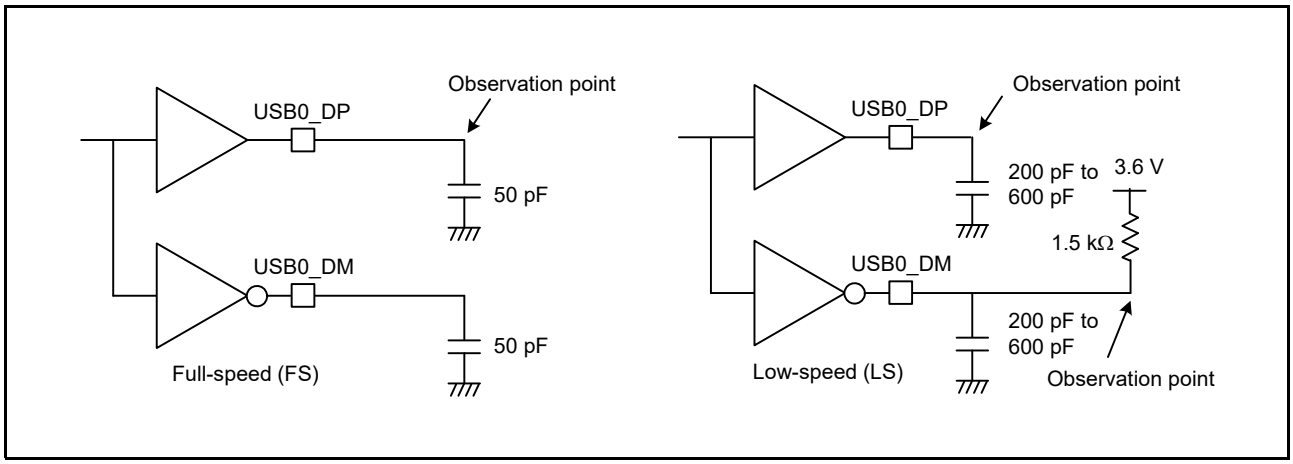


Figure 47.65 Test Circuit

47.7 A/D Conversion Characteristics

Table 47.64 A/D Conversion Characteristics (1)

Conditions: 4.5 V ≤ VCC ≤ 5.5 V, 4.5 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C, signal source impedance = 0.5 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	64	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (operation at PCLKD = 64 MHz)		0.50 (0.156)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 0 ADSSTRn = 0Ah ADCCR.CCS = 1
		0.97 (0.625)*2	—	—	µs	Normal-precision channel ADCSR.ADHSC bit = 0 ADSSTRn = 28h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	1.3*3	kΩ	High-precision channel
		—	—	5.0*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		—	± 0.5	—	LSB	
Absolute accuracy		—	±2.5	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential nonlinearity error		—	±1.0	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.65 A/D Conversion Characteristics (2)

Conditions: 2.7 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C,
signal source impedance = 0.3 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	48	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (operation at PCLKD = 48 MHz)		0.67 (0.208)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 0 ADSSTRn = 0Ah ADCCR.CCS = 1
		1.29 (0.833)*2	—	—	µs	Normal-precision channel ADCSR.ADHSC bit = 0 ADSSTRn = 28h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	1.9*3	kΩ	High-precision channel
		—	—	6.0*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		—	± 0.5	—	LSB	
Absolute accuracy		—	±2.5	±5.5	LSB	High-precision channel
				±8.5	LSB	Other than above
DNL differential nonlinearity error		—	±1.0	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.66 A/D Conversion Characteristics (3)

Conditions: 2.4 V ≤ VCC ≤ 5.5 V, 2.4 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C,
signal source impedance = 1.3 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	32	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 32 MHz)		1.00 (0.313)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 0 ADSSTRn = 0Ah ADCCR.CCS = 1
		1.94 (1.250)*2	—	—	µs	Normal-precision channel ADCSR.ADHSC bit = 0 ADSSTRn = 28h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	2.2*3	kΩ	High-precision channel
		—	—	7.0*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±2.5	±5.5	LSB	High-precision channel
				±8.5	LSB	Other than above
DNL differential nonlinearity error		—	±1.0	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.67 A/D Conversion Characteristics (4)

Conditions: 2.7 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C,
signal source impedance = 1.1 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	24	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (operation at PCLKD = 24 MHz)		1.58 (0.417)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 0Ah ADCCR.CCS = 1
		2.00 (0.833)*2	—	—		Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 14h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	1.9*3	kΩ	High-precision channel
		—	—	6*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.25	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±2.5	±5.5	LSB	High-precision channel
				±8.5	LSB	Other than above
DNL differential nonlinearity error		—	±1.0	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.68 A/D Conversion Characteristics (5)

Conditions: 2.4 V ≤ VCC ≤ 5.5 V, 2.4 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C, signal source impedance = 2.2 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	16	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (operation at PCLKD = 16 MHz)		2.38 (0.625)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 0Ah ADCCR.CCS = 1
		3.00 (1.250)*2	—	—		Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 14h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	2.2*3	kΩ	High-precision channel
		—	—	7*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.25	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±2.5	±5.5	LSB	High-precision channel
				±8.5	LSB	Other than above
DNL differential nonlinearity error		—	±1.0	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.69 A/D Conversion Characteristics (6)

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.8 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C,
signal source impedance = 5 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	8	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (operation at PCLKD = 8 MHz)		4.75 (1.250)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 0Ah ADCCR.CCS = 1
		6.00 (2.500)*2	—	—		Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 14h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	6*3	kΩ	High-precision channel
		—	—	14*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.25	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Full-scale error		—	±1.5	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±3.0	±8.0	LSB	High-precision channel
				±11.0	LSB	Other than above
DNL differential nonlinearity error		—	±1.25	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.5	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.70 A/D Conversion Characteristics (7)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C,
signal source impedance = 9.9 kΩ
Reference voltage = VREFH0

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	4	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (operation at PCLKD = 4 MHz)		9.50 (2.500)*2	—	—	µs	High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 0Ah ADCCR.CCS = 1
		12.00 (5.000)*2	—	—		Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn = 14h ADCCR.CCS = 1
Analog input capacitance	Cs	—	—	9*3	pF	High-precision channel
		—	—	10*3		Normal-precision channel
Analog input resistance	Rs	—	—	12*3	kΩ	High-precision channel
		—	—	28*3		Normal-precision channel
Analog input effective range		0	—	VREFH0	V	
Offset error		—	±1.25	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Full-scale error		—	±1.5	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±3.0	±8.0	LSB	High-precision channel
				±11.0	LSB	Other than above
DNL differential nonlinearity error		—	±1.25	—	LSB	
INL integral nonlinearity error		—	±1.5	±3.5	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The values in ( ) show the sampling times.

Note 3. The values are reference values.

Table 47.71 A/D Converter Channel Classification

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN007	AVCC0 = 1.6 to 5.5 V	Pins AN000 to AN007 cannot be used as digital outputs when the A/D converter is in use.
Normal-precision channel	AN016 to AN031		
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 1.6 to 5.5 V	
Temperature sensor input channel	Temperature sensor output	AVCC0 = 1.6 to 5.5 V	
CTSU input channels	AN008	AVCC0 = 1.6 to 5.5V	

Table 47.72 A/D Internal Reference Voltage Characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ VREFH0 = AVCC0 ≤ 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Internal reference voltage input channel*1	1.42	1.48	1.54	V	

Note 1. The A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the A/D converter.

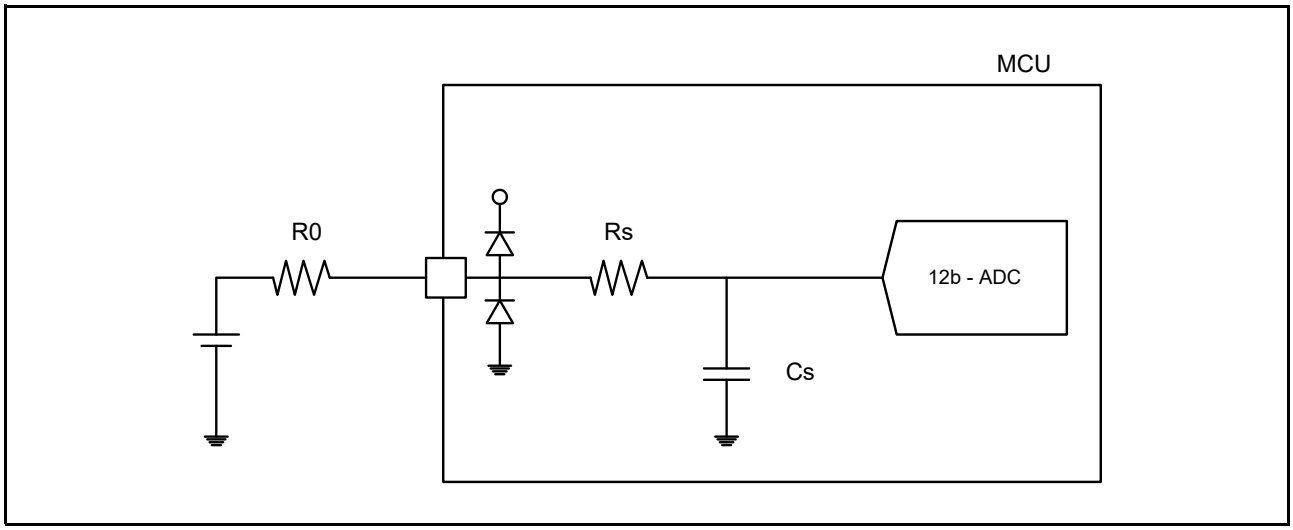


Figure 47.66 Equivalent Circuit

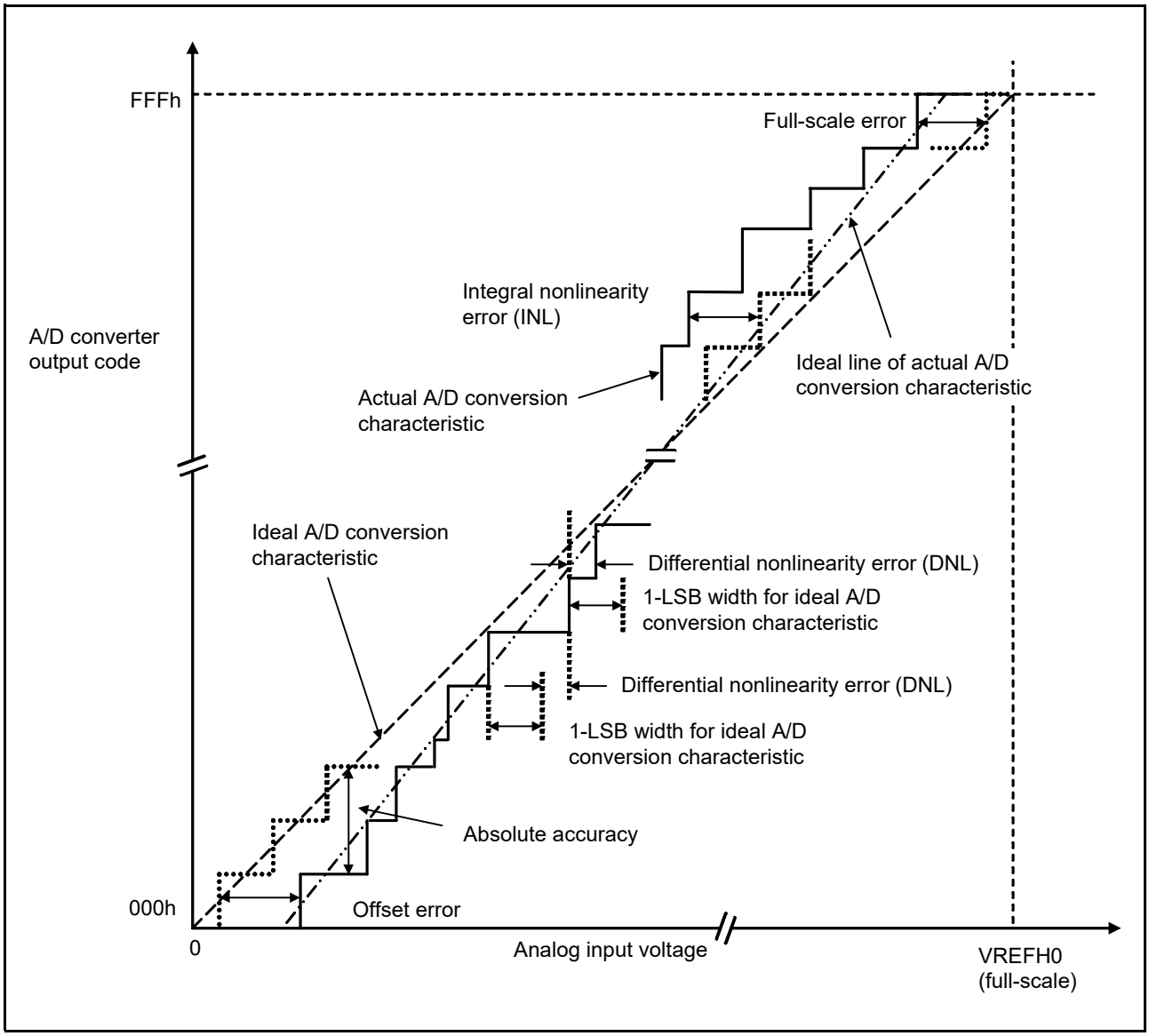


Figure 47.67 Illustration of A/D Converter Characteristic Terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and if reference voltage (VREFH0 = 3.072 V), then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, 1.5 mV, … are used as analog input voltages.

If analog input voltage is 6 mV, absolute accuracy = ±5 LSB means that the actual A/D conversion result is in the range of 003h to 00Dh though an output code, 008h, can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

Offset error is the difference between a transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

47.8 D/A Conversion Characteristics

Table 47.73 D/A Conversion Characteristics

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Resolution		—	—	—	8	Bit	
Conversion time	AVCC0 = 1.6 to 5.5 V	t_{DCONV}	—	—	3.0	µs	35-pF capacitive load
Absolute accuracy	AVCC0 = 2.4 to 5.5 V	—	—	—	±3.0	LSB	2-MΩ resistive load
	AVCC0 = 1.8 to 2.4 V	—	—	—	±3.5		
	AVCC0 = 1.6 to 1.8 V	—	—	—	±4.0		
	AVCC0 = 2.4 to 5.5 V	—	—	—	±2.0	LSB	4-MΩ resistive load
	AVCC0 = 1.8 to 2.4 V	—	—	—	±2.5		
	AVCC0 = 1.6 to 1.8 V	—	—	—	±3.0		
RO output resistance		—	—	9.0	—	kΩ	

47.9 Temperature Sensor Characteristics

Table 47.74 Temperature Sensor Characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	—	±1.5	—	°C	2.4 V or above
		—	±2.0	—		Below 2.4 V
Temperature slope	—	—	–3.3	—	mV/°C	
Output voltage (25°C)	—	—	1.05	—	V	VCC = 3.3 V
Temperature sensor start time	t_{START}	—	—	5	µs	
Sampling time	—	5	—	—	µs	

47.10 Comparator Characteristics

Table 47.75 Comparator Characteristics

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
CVREFB0 to CVREFB1 input reference voltage		VREF	0	—	VCC – 1.4	V	
CMPB0 to CMPB1 input voltage		VI	0	—	VCC	V	
Internal reference voltage*1		—	1.34	1.44	1.54	V	
Offset	Comparator high-speed mode	—	—	—	50	mV	
	Comparator high-speed mode Window function enabled	—	—	—	60	mV	
	Comparator low-speed mode	—	—	—	40	mV	
Comparator output delay time	Comparator high-speed mode	Td	—	—	1.2	µs	VCC = 3 V, input slew rate ≥ 50 mV/µs
	Comparator high-speed mode Window function enabled	Tdw	—	—	2.0	µs	
	Comparator low-speed mode	Td	—	—	9.0	µs	
High-side reference voltage (comparator high-speed mode, window function enabled)		VRFH	—	0.76 × VCC	—	V	
Low-side reference voltage (comparator high-speed mode, window function enabled)		VRFL	—	0.24 × VCC	—	V	
Operation stabilization wait time (high-speed mode)	VCC = 1.6 V to 5.5 V	Tcmp	100	—	—	µs	
Operation stabilization wait time (low-speed mode)	VCC = 1.8 V to 5.5 V		100	—	—		
	VCC = 1.6 V to 1.8 V		1000	—	—		

Note 1. The internal reference voltage cannot be used when VCC < 1.8 V.

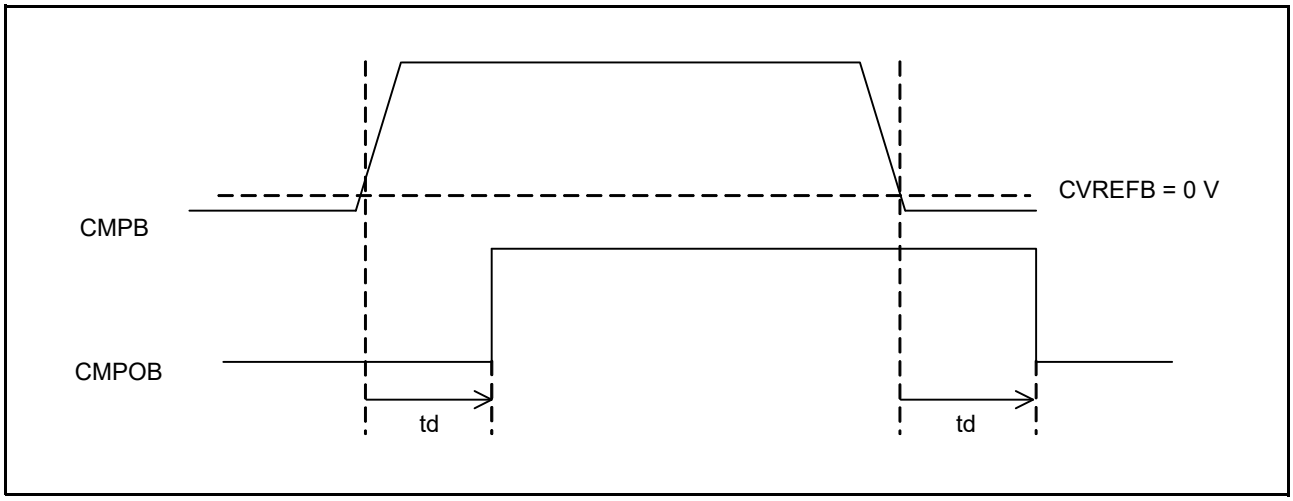


Figure 47.68 Comparator Output Delay Time in Comparator High-Speed Mode and Low-Speed Mode

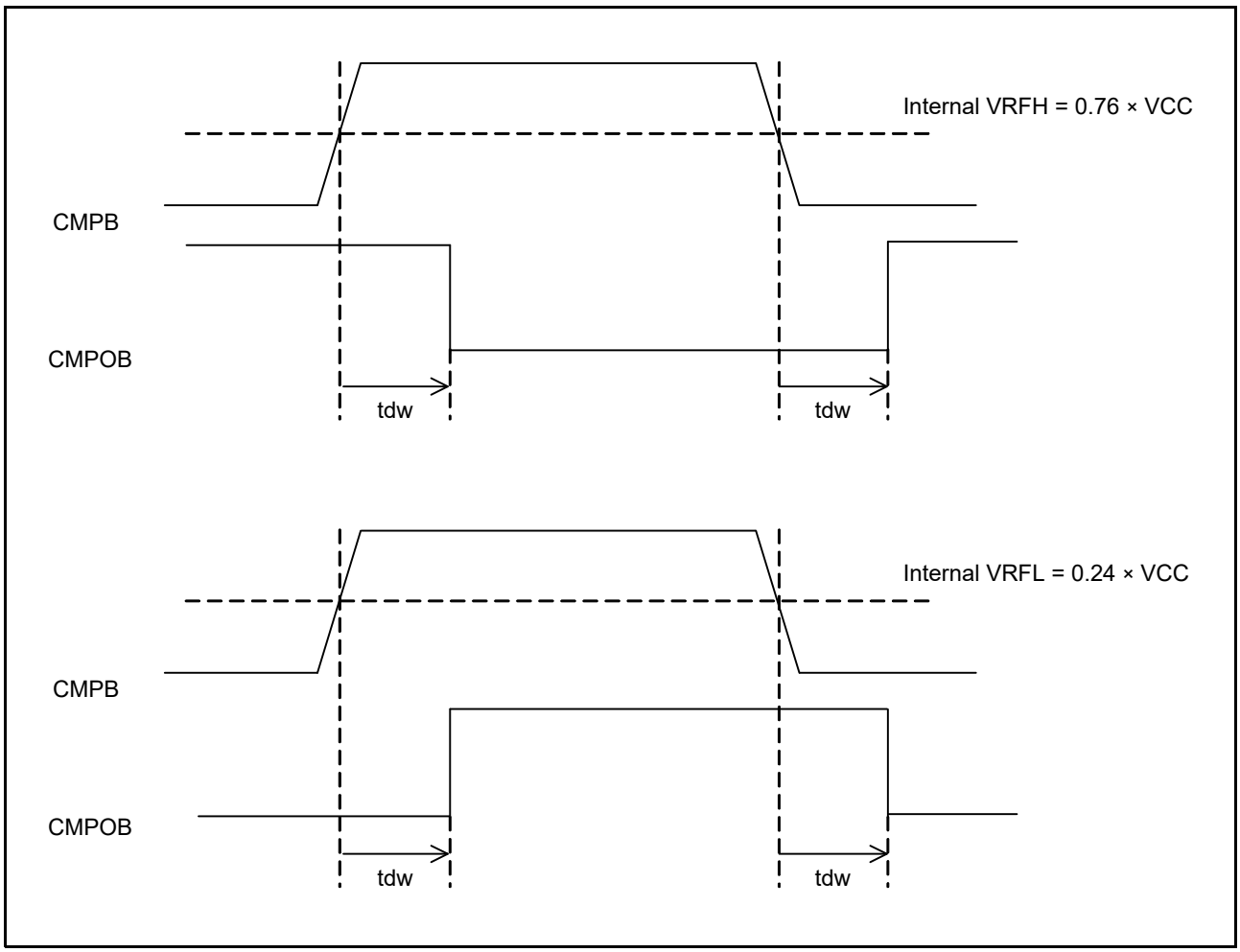


Figure 47.69 Comparator Output Delay Time in High-Speed Mode with Window Function Enabled

47.11 CTSU Characteristics

Table 47.76 CTSU Characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
External capacitance connected to TSCAP pin	C_{tscap}	9	10	11	nF	

47.12 Power-On Reset Circuit and Voltage Detection Circuit Characteristics

Table 47.77 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (1)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	1.46	1.50	1.54	V	Figure 47.70, Figure 47.71
	Voltage detection circuit (LVD0)*1	V_{det0_0}	3.67	3.85	3.97	V	Figure 47.72 At falling edge VCC
		V_{det0_1}	2.70	2.85	3.00		
		V_{det0_2}	2.37	2.53	2.67		
		V_{det0_3}	1.80	1.90	1.99		
		V_{det0_4}	1.60	1.69	1.80		
	Voltage detection circuit (LVD1)*2	V_{det1_0}	4.12	4.29	4.42	V	Figure 47.73 At falling edge VCC
		V_{det1_1}	3.98	4.16	4.28		
		V_{det1_2}	3.86	4.03	4.16		
		V_{det1_3}	3.68	3.86	3.98		
		V_{det1_4}	2.99	3.10	3.29		
		V_{det1_5}	2.89	3.00	3.19		
		V_{det1_6}	2.79	2.90	3.09		
		V_{det1_7}	2.68	2.80	2.98		
		V_{det1_8}	2.57	2.68	2.87		
		V_{det1_9}	2.47	2.59	2.67		
		V_{det1_A}	2.37	2.48	2.57		
		V_{det1_B}	2.10	2.20	2.30		
		V_{det1_C}	1.86	1.96	2.06		
		V_{det1_D}	1.80	1.86	1.96		
		V_{det1_E}	1.69	1.75	1.81		
		V_{det1_F}	1.60	1.65	1.70		
Voltage detection level	Voltage detection circuit (LVD2)*3	V_{det2_0}	4.08	4.32	4.48	V	Figure 47.74 At falling edge VCC
		V_{det2_1}	3.95	4.17	4.35		
		V_{det2_2}	3.82	4.03	4.22		
		V_{det2_3}	3.62	3.84	4.02		

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD2), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. n in the symbol Vdet0_n denotes the value of the OFS1.VDSEL2, VDSEL[1:0] bits.

Note 2. n in the symbol Vdet1_n denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

Note 3. n in the symbol Vdet2_n denotes the value of the LVDLVLR.LVD2LVL[1:0] bits.

Table 47.78 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (2)

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Wait time after power-on reset cancellation	At normal startup*1	t_{POR}	—	12.5	—	ms	Figure 47.71
	During fast startup time*2	t_{POR}	—	5.0	—		
Wait time after voltage monitoring 0 reset cancellation		t_{LVD0}	—	880	—	µs	Figure 47.72
Wait time after voltage monitoring 1 reset cancellation	LVD0 disabled*4	t_{LVD1}	—	180	—	µs	Figure 47.73
	LVD0 enabled*5		—	880	—	µs	
Wait time after voltage monitoring 2 reset cancellation	LVD0 disabled*4	t_{LVD2}	—	180	—	µs	Figure 47.74
	LVD0 enabled*5		—	880	—	µs	
PDR response delay time		t_{det}	—	—	500	µs	Figure 47.70
LVD0 response delay time			—	—	500	µs	Figure 47.70
LVD1 response delay time			—	—	360	µs	Figure 47.70
LVD2 response delay time			—	—	600	µs	Figure 47.70
POR/LVD0 minimum VCC down time*3		t_{VOFF}	500	—	—	µs	Figure 47.70, VCC = 1.0 V or above
LVD1 minimum VCC down time*3			300	—	—	µs	Figure 47.70, VCC = 1.0 V or above
LVD2 minimum VCC down time*3			600	—	—	µs	Figure 47.70, VCC = 1.0 V or above
Power-on reset enable time		$t_{W(POR)}$	1	—	—	ms	Figure 47.71, VCC = below 1.0 V
LVD1 operation stabilization time (after LVD is enabled)		$t_{d(E-A)}$	—	—	300	µs	Figure 47.73
LVD2 operation stabilization time (after LVD is enabled)		$t_{d(E-A)}$	—	—	1200	µs	Figure 47.74
Hysteresis width (power-on rest (POR))		V_{PORH}	—	10	—	mV	
Hysteresis width (LVD0, LVD1, and LVD2)		V_{LVH}	—	60	—	mV	Vdet0_0 to Vdet0_4 selected
			—	110	—		Vdet1_0 to Vdet1_2 selected
			—	70	—		Vdet1_3 to Vdet1_9 selected
			—	60	—		Vdet1_A to Vdet1_B selected
			—	50	—		Vdet1_C to Vdet1_F selected
			—	90	—		LVD2 selected

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD1), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. When OFS1.(LVDAS, FASTSTUP) = 11b.

Note 2. When OFS1.(LVDAS, FASTSTUP) ≠ 11b.

Note 3. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, V_{det1}, and V_{det2} for the POR/LVD.

Note 4. When OFS1.LVDAS = 1b.

Note 5. When OFS1.LVDAS = 0b.

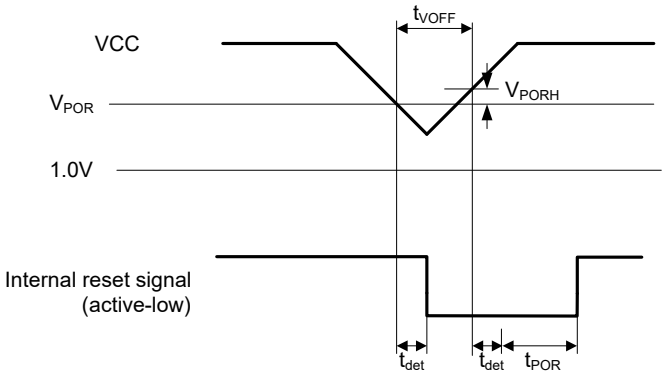
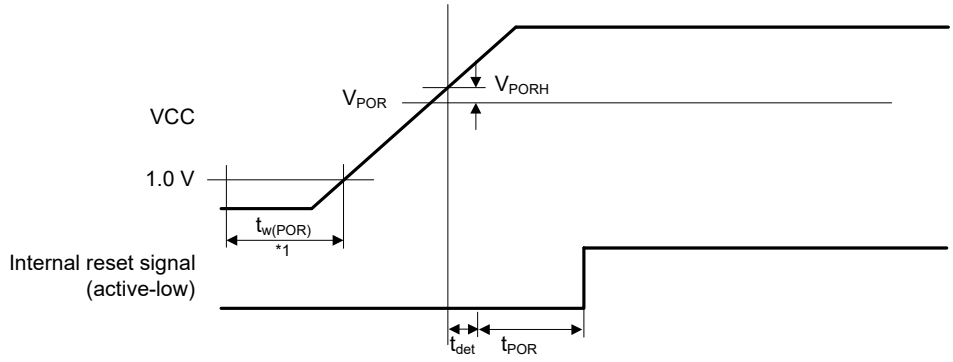


Figure 47.70 Voltage Detection Reset Timing



Note 1. $t_{w(POR)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When VCC turns on, maintain $t_{w(POR)}$ for 1.0 ms or more.

Figure 47.71 Power-On Reset Timing

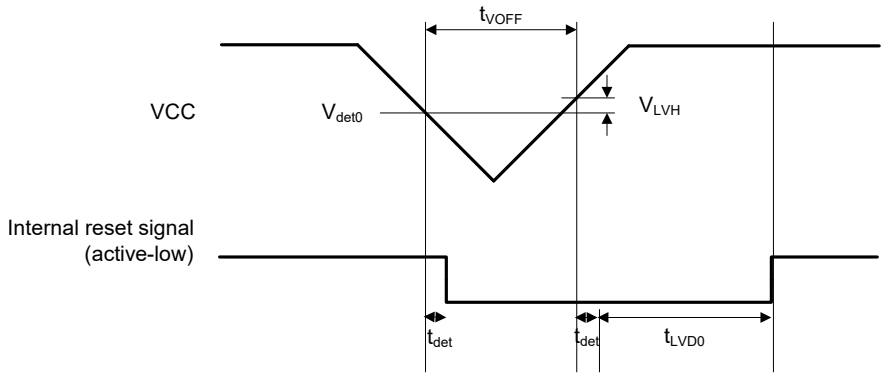


Figure 47.72 Voltage Detection Circuit Timing (V_{det0})

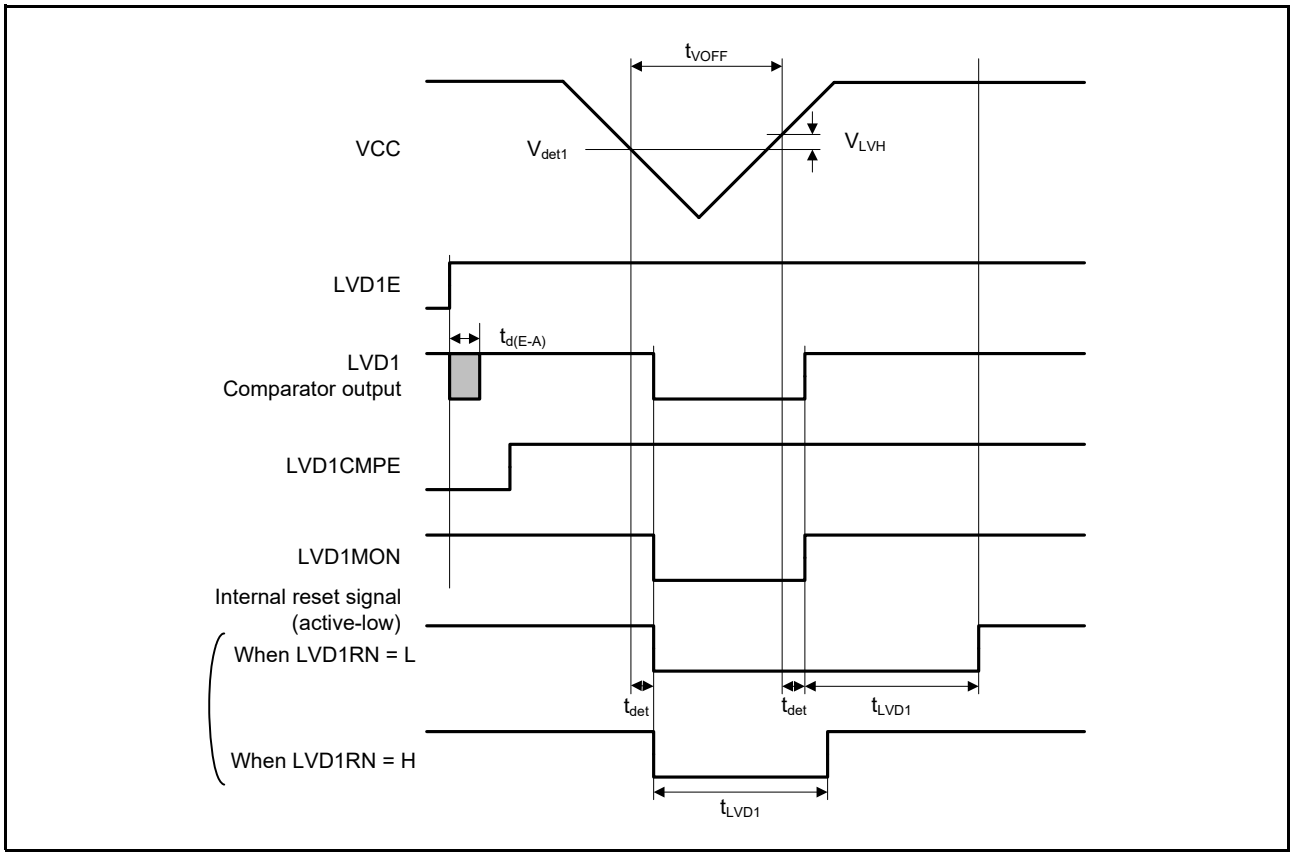


Figure 47.73 Voltage Detection Circuit Timing (V_{det1})

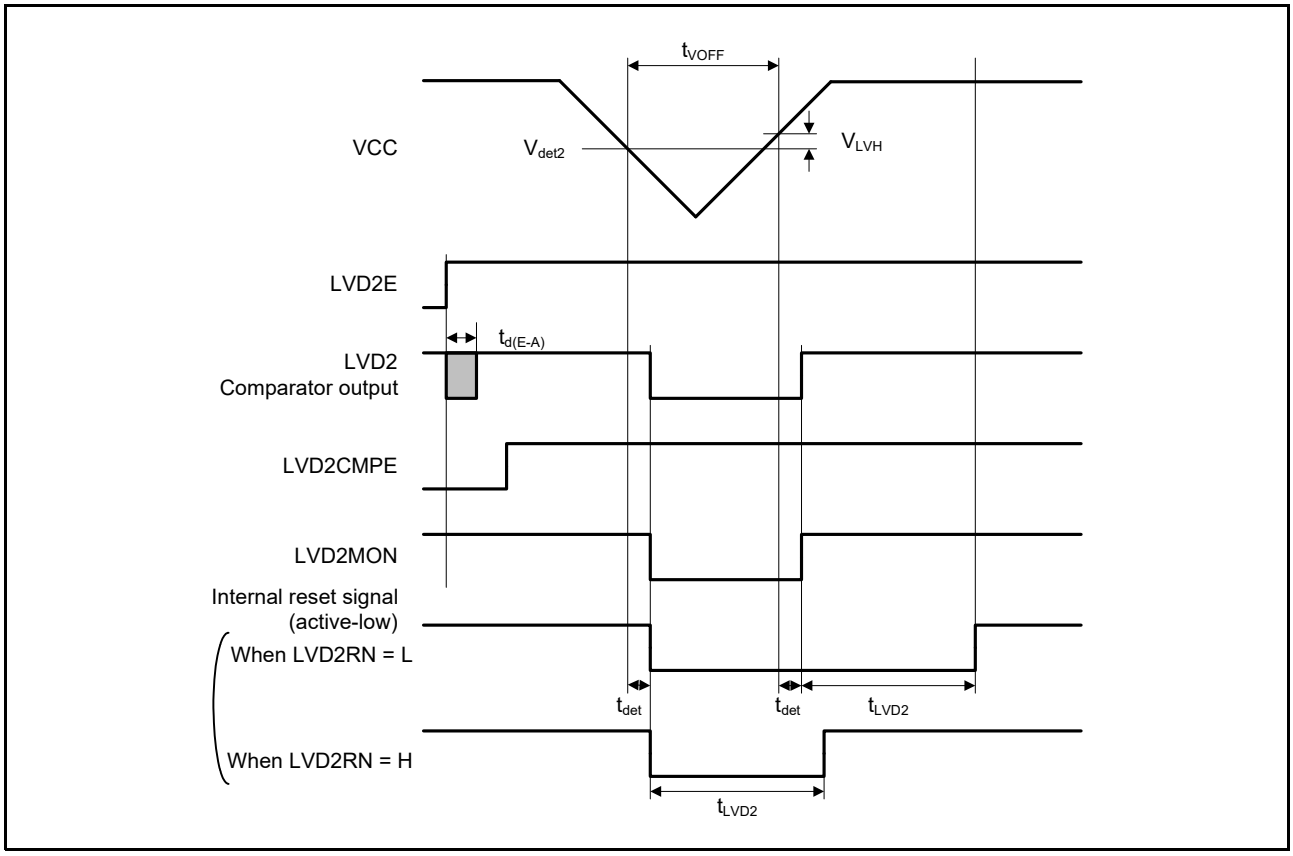


Figure 47.74 Voltage Detection Circuit Timing (V_{det2})

47.13 Oscillation Stop Detection Timing

Table 47.79 Oscillation Stop Detection Timing

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 47.75

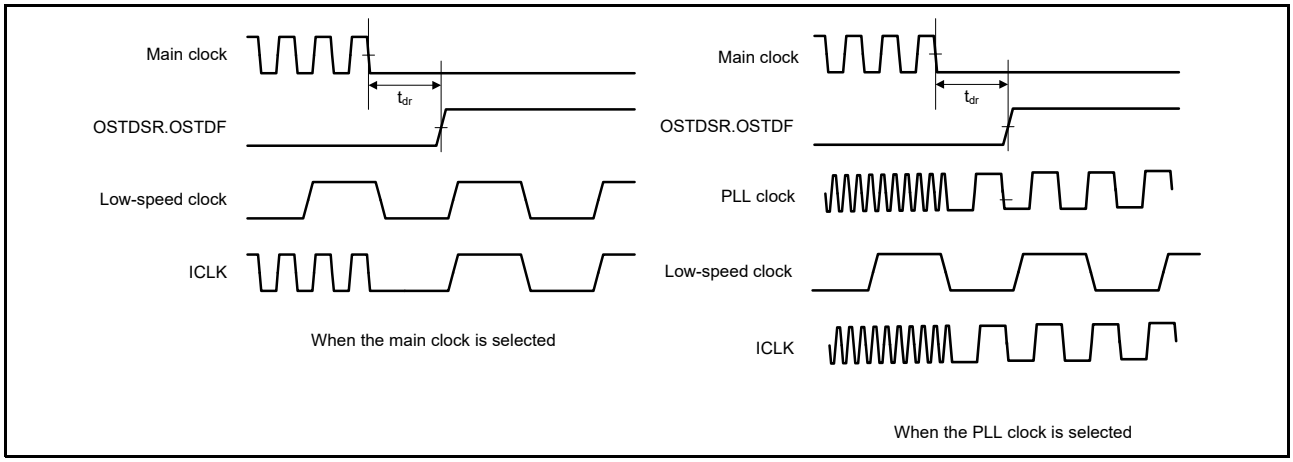


Figure 47.75 Oscillation Stop Detection Timing

47.14 ROM (Flash Memory for Code Storage) Characteristics

Table 47.80 ROM (Flash Memory for Code Storage) Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Reprogramming/erasure cycle*1		N_{PEC}	1K	—	—	Times	
Data retention*2, *3	After 1K times of N_{PEC}	t_{DRP}	20	—	—	Year	T_a = +105°C

Note 1. Definition of reprogram/erase cycle: The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1K), erasing can be performed n times for each block. For instance, when 8-byte programming is performed 256 times for different addresses in 2-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. Characteristic when using the flash memory programmer and the self-programming library provided from Renesas Electronics.
Note 3. This result is obtained from reliability testing.

Table 47.81 ROM (Flash Memory for Code Storage) Characteristics (2) High-Speed Operating Mode

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.8 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V

Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			FCLK = 48 MHz*1			FCLK = 64 MHz*1			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	94.0	843.5	—	45.1	446.0	—	45.0	445.0	µs
Erasure time	2-Kbyte	t_{E2K}	—	8.3	282.0	—	5.4	220.1	—	5.4	220.4	ms
	512-Kbyte (block erase command)	t_{E512K}	—	807.1	17356.0	—	67.9	1651.9	—	70.6	1709.3	ms
	512-Kbyte (all-block erase command)	t_{EA512K}	—	801.9	17140.5	—	62.7	1436.9	—	65.4	1494.3	ms
Blank check time	8-byte	t_{BC8}	—	—	45.0	—	—	8.7	—	—	8.6	µs
	2-Kbyte	t_{BC2K}	—	—	1573	—	—	115	—	—	120	µs
Erase operation forcible stop time		t_{SED}	—	—	22.8	—	—	11.0	—	—	10.9	µs
Start-up area switching setting time		t_{SAS}	—	8.2	503.3	—	5.6	437.7	—	5.6	437.9	ms
Access window setting time		t_{AWS}	—	8.2	503.3	—	5.6	437.7	—	5.6	437.9	ms
ROM mode transition wait time		t_{MS}	15	—	—	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%.
Note 1. 2.4 V ≤ VCC ≤ 5.5 V

Table 47.82 ROM (Flash Memory for Code Storage) Characteristics (3) Middle-Speed Operating Mode

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V

Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			FCLK = 24 MHz*1			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	94.0	843.5	—	45.7	450.7	µs
Erasure time	2-Kbyte	t_{E2K}	—	8.3	282.0	—	5.4	220.2	ms
	512-Kbyte (block erase command)	t_{E512K}	—	807.1	17356.0	—	67.9	1653.0	ms
	512-Kbyte (all-block erase command)	t_{EA512K}	—	801.9	17140.5	—	62.7	1438.1	ms
Blank check time	8-byte	t_{BC8}	—	—	45	—	—	9	µs
	2-Kbyte	t_{BC2K}	—	—	1573	—	—	115	µs
Erase operation forcible stop time		t_{SED}	—	—	22.8	—	—	11.2	µs
Start-up area switching setting time		t_{SAS}	—	8.2	503.3	—	5.6	437.7	ms
Access window setting time		t_{AWS}	—	8.2	503.3	—	5.6	437.7	ms
ROM mode transition wait time		t_{MS}	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%.
Note 1. 2.4 V ≤ VCC ≤ 5.5 V

Table 47.83 ROM (Flash Memory for Code Storage) Characteristics (4) Middle-Speed Operating Mode 2

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V

Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			Unit
			Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	94.0	843.5	µs
Erasure time	2-Kbyte	t_{E2K}	—	8.3	282.0	ms
	512-Kbyte (block erase command)	t_{E512K}	—	807.1	17356.0	ms
	512-Kbyte (all-block erase command)	t_{EA512K}	—	801.9	17140.5	ms
Blank check time	8-byte	t_{BC8}	—	—	45	µs
	2-Kbyte	t_{BC2K}	—	—	1573	µs
Erase operation forcible stop time		t_{SED}	—	—	22.8	µs
Start-up area switching setting time		t_{SAS}	—	8.2	503.3	ms
Access window setting time		t_{AWS}	—	8.2	503.3	ms
ROM mode transition wait time		t_{MS}	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory.
Note: The frequency accuracy of FCLK should be ±3.5%.

47.15 E2 DataFlash Characteristics (Flash Memory for Data Storage)

Table 47.84 E2 DataFlash Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Reprogramming/erasure cycle*1		N_{DPEC}	100K	1000K	—	Times	
Data retention	After 10K times of N_{DPEC}	t_{DDRP}	20*2, *3	—	—	Year	T_a = +105°C
	After 100K times of N_{DPEC}		5*2, *3	—	—	Year	
	After 1000K times of N_{DPEC}		—	1*2, *3	—	Year	T_a = +25°C

Note 1. The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 100K), erasing can be performed n times for each block. For instance, when 1-byte programming is performed 256 times for different addresses in 256-byte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. Characteristic when using the flash memory programmer and the self-programming library provided from Renesas Electronics.
Note 3. These results are obtained from reliability testing.

Table 47.85 E2 DataFlash Characteristics (2) High-speed operating mode

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, 1.8 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V

Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			FCLK = 48 MHz			FCLK = 64 MHz*1			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1-byte	t_{DP1}	—	83.0	729.5	—	34.8	338.8	—	34.6	337.7	µs
Erasure time	256-byte	t_{DE256}	—	8.3	282.0	—	5.4	220.1	—	5.4	220.4	ms
	8-Kbyte	t_{DE8K}	—	104.8	2331.4	—	12.4	368.0	—	12.7	375.2	ms
Blank check time	1-byte	t_{DBC1}	—	—	44.6	—	—	8.7	—	—	8.6	µs
	256-byte	t_{DBC256}	—	—	1573	—	—	115	—	—	120	µs
Erase operation forcible stop time		t_{DSED}	—	—	22.8	—	—	11.0	—	—	10.9	µs
DataFlash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	250	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%.
Note 1. 2.4 V ≤ VCC ≤ 5.5 V

Table 47.86 E2 DataFlash Characteristics (3) Middle-speed operating mode

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V

Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			FCLK = 24 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1-byte	t_{DP1}	—	83.0	729.5	—	35.3	343.2	µs
Erasure time	256-byte	t_{DE256}	—	8.3	282.0	—	5.4	220.2	ms
	8-Kbyte	t_{DE8K}	—	104.8	2331.4	—	12.4	368.2	ms
Blank check time	1-byte	t_{DBC1}	—	—	44.6	—	—	9.0	µs
	256-byte	t_{DBC256}	—	—	1573	—	—	0.1	ms
Erase operation forcible stop time		t_{DSED}	—	—	22.8	—	—	11.2	µs
DataFlash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%.

Table 47.87 E2 DataFlash Characteristics (4) Middle-speed operating mode 2

Conditions: 1.6 V ≤ VCC ≤ 5.5 V, 1.6 V ≤ AVCC0 ≤ 5.5 V, VSS = AVSS0 = 0 V

Temperature range for the programming/erasure operation: T_a = –40 to +105°C

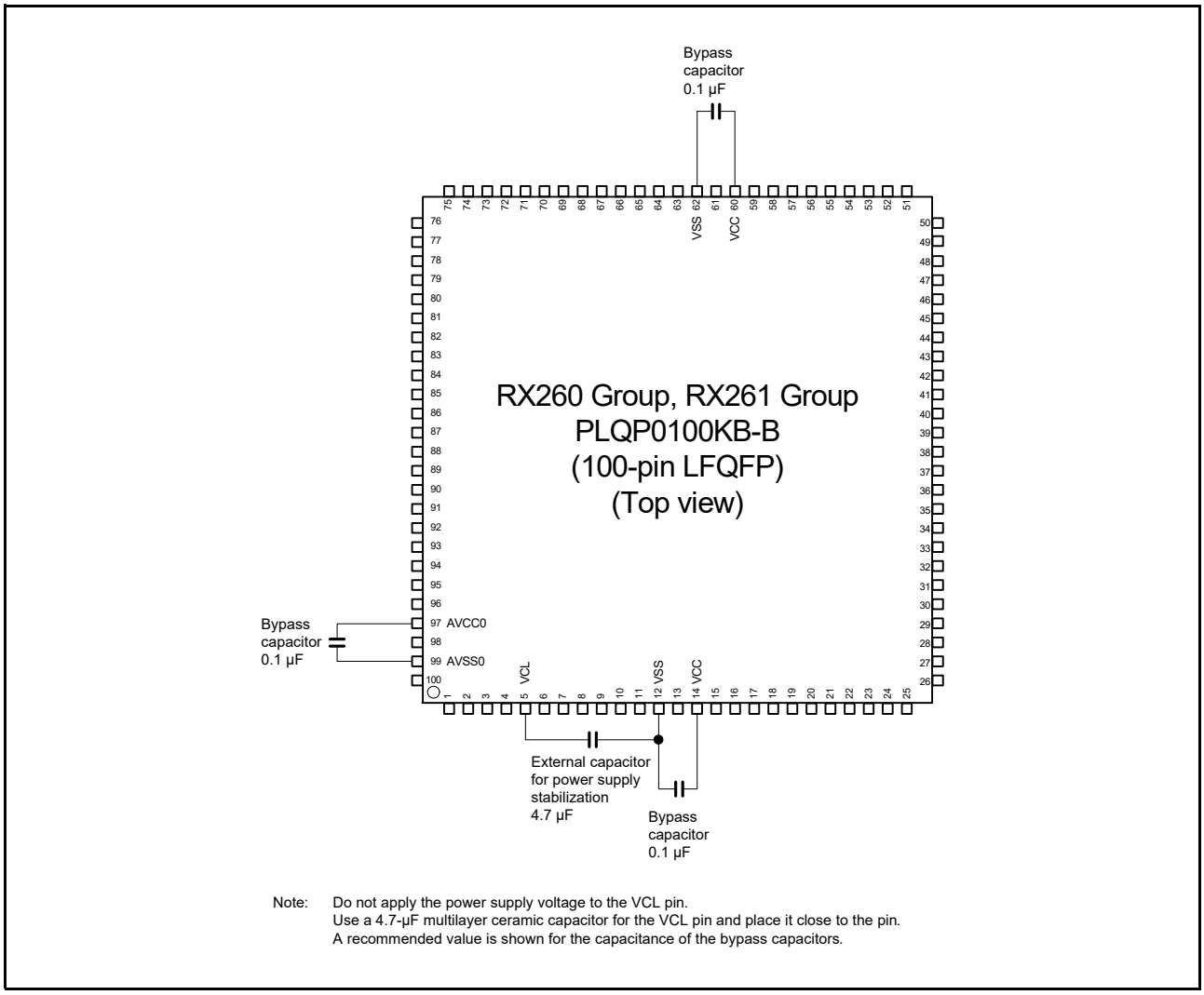
Item		Symbol	FCLK = 1 MHz			Unit
			Min.	Typ.	Max.	
Programming time	1-byte	t_{DP1}	—	83.0	729.5	µs
Erasure time	256-byte	t_{DE256}	—	8.3	282.0	ms
	8-Kbyte	t_{DE8K}	—	104.8	2331.4	ms
Blank check time	1-byte	t_{DBC1}	—	—	44.6	µs
	256-byte	t_{DBC256}	—	—	1573	ms
Erase operation forcible stop time		t_{DSED}	—	—	22.8	µs
DataFlash STOP recovery time		t_{DSTOP}	250	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory.
Note: The frequency accuracy of FCLK should be ±3.5%.

47.16 Usage Notes

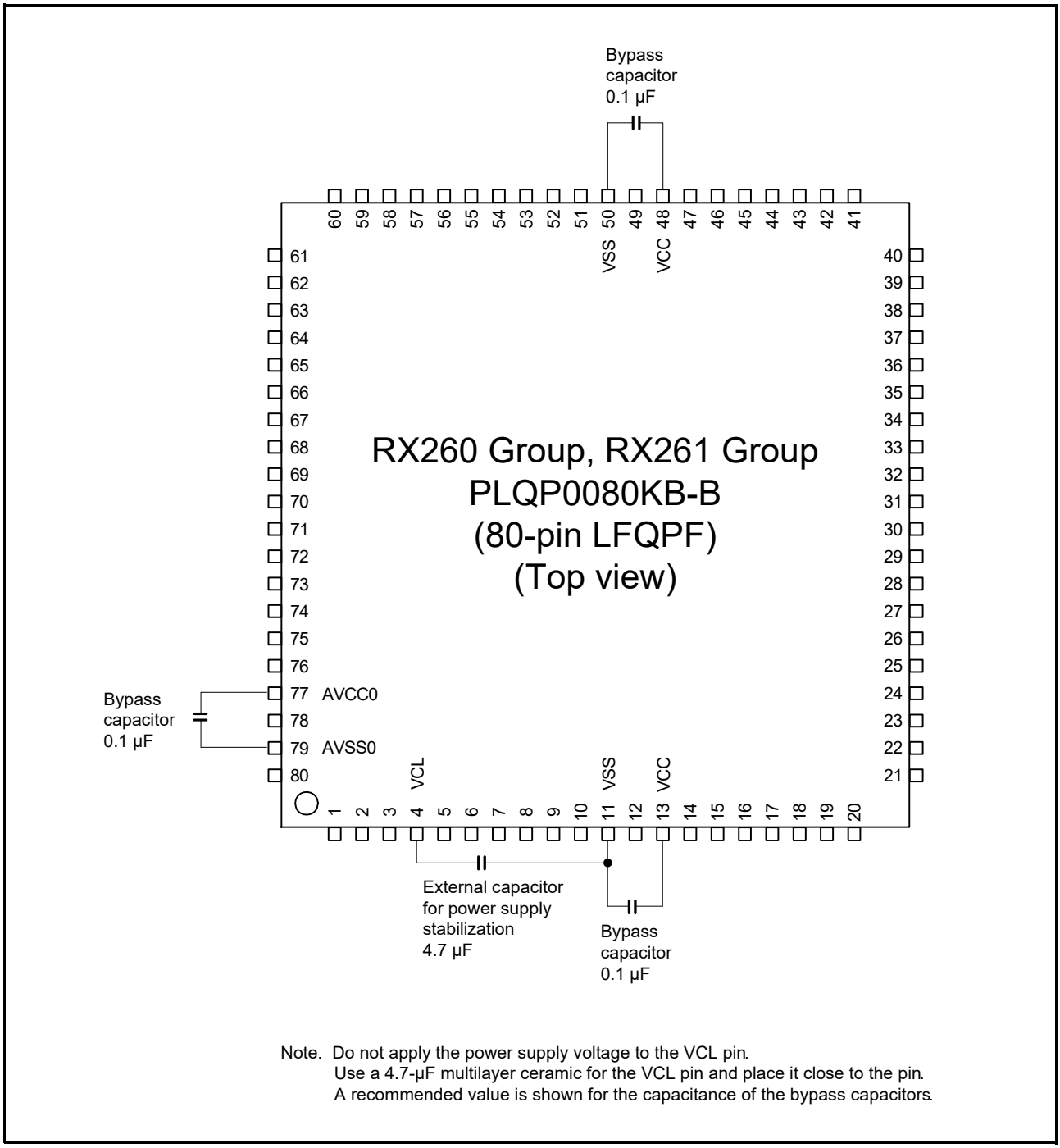
47.16.1 Connecting VCL Capacitor and Bypass Capacitors

This MCU integrates an internal voltage-down circuit, which is used for lowering the power supply voltage in the internal MCU to adjust automatically to the optimum level. A 4.7-µF capacitor needs to be connected between this internal voltage-down power supply (VCL pin) and VSS pin. Figure 47.77 to Figure 47.79 shows how to connect external capacitors. Place an external capacitor close to the pins. Do not apply the power supply voltage to the VCL pin. Insert a multilayer ceramic capacitor as a bypass capacitor between each pair of the power supply pins. Implement a bypass capacitor to the MCU power supply pins as close as possible. Use a recommended value of 0.1 µF as the capacitance of the capacitors. For the capacitors related to crystal oscillation, see section 9, Clock Generation Circuit. For the capacitors related to analog modules, also see section 40, 12-Bit A/D Converter (S12ADE).
For notes on designing the printed circuit board, see the descriptions of the application note “Hardware Design Guide” (R01AN1411EJ). The latest version can be downloaded from Renesas Electronics Website.



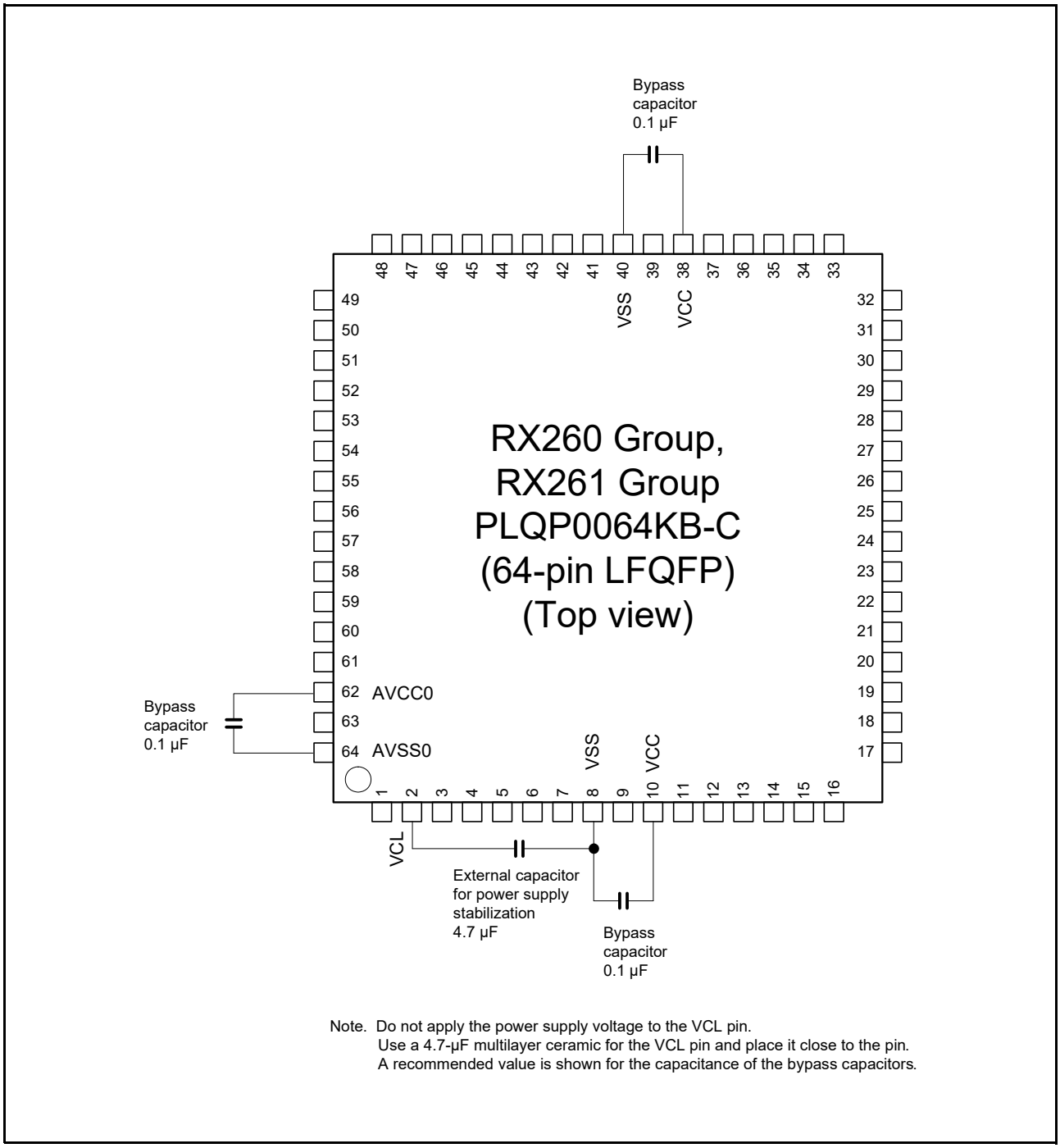
Note: Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic capacitor for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 47.76 Connecting Capacitors (100 Pins)



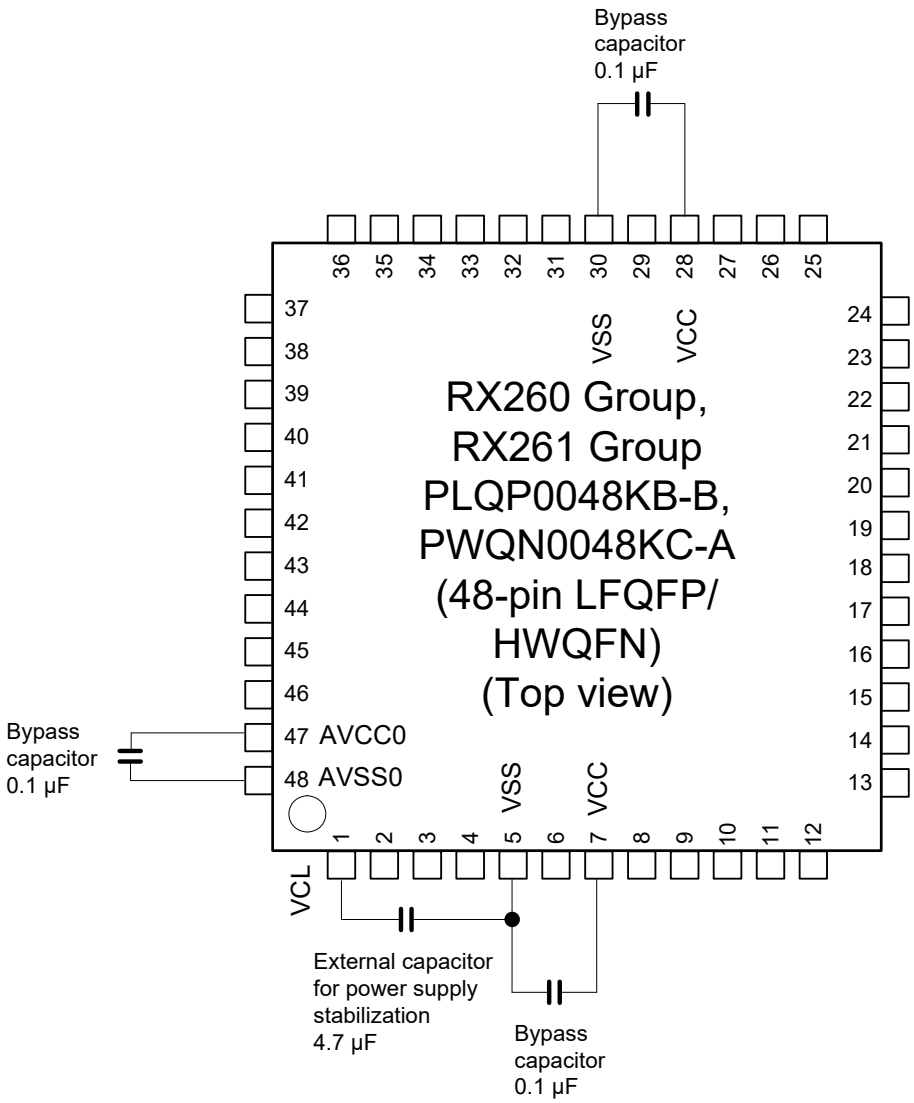
Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 47.77 Connecting Capacitors (80 Pins)



Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 47.78 Connecting Capacitors (64 Pins)



Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 47.79 Connecting Capacitors (48 Pins)

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing Mode (1/2)

Port Name (Pin Name)	Reset	Software Standby Mode	
P03 (DA0)	Hi-Z	DA0 output (DAOE0 = 1)	DA output retained
		Other than the above (DAOE0 = 0)	Keep-O
P05 (DA1)	Hi-Z	DA1 output (DAOE1 = 1)	DA output retained
		Other than the above (DAOE1 = 0)	Keep-O
P04, P06, P07	Hi-Z	Keep-O	
P12 (IRQ2)	Hi-Z	Keep-O*1	
P13 (IRQ3)	Hi-Z	Keep-O*1	
P14 (IRQ4/USB0_OVRCURA)	Hi-Z	Keep-O*1, *2	
P15 (IRQ5)	Hi-Z	Keep-O*1	
P16 (IRQ6/RTCOUT/USB0_VBUS/ USB0_OVRCURB)	Hi-Z	RTCOUT output	RTCOUT output
		Other than the above	Keep-O*1, *2
P17 (IRQ7)	Hi-Z	Keep-O*1	
P20, P21, P23, P24	Hi-Z	Keep-O	
P22 (USB0_OVRCURB)	Hi-Z	Keep-O*2	
P26 (CLKOUT)	Hi-Z	CLKOUT output	CLKOUT output
		Other than the above	Keep-O
P27	Hi-Z	Keep-O	
P30 (IRQ0)	Hi-Z	Keep-O*1	
P31 (IRQ1)	Hi-Z	Keep-O*1	
P32 (IRQ2/RTCOUT)	Hi-Z	RTCOUT output	RTCOUT output
		Other than the above	Keep-O*1
P33 (IRQ3)	Hi-Z	Keep-O*1	
P34 (IRQ4)	Hi-Z	Keep-O*1	
P35 (NMI)	Hi-Z	Keep*1	
P36 (IRQ2)	Hi-Z	Keep-O*1	
P37 (IRQ4)	Hi-Z	Keep-O*1	
P40 to P47	Hi-Z	Keep-O	
P50, P52, P54, P55	Hi-Z	Keep-O	
P51, P53 (PMC0)	Hi-Z	Keep-O*3	
PA0, PA1	Hi-Z	Keep-O	
PA2 (RXD5)	Hi-Z	Keep-O*4	
PA3 (IRQ6/RXD5)	Hi-Z	Keep-O*1, *4	
PA4 (IRQ5)	Hi-Z	Keep-O*1	
PA5 to PA7	Hi-Z	Keep-O	
PB0	Hi-Z	Keep-O	
PB1 (IRQ4/CMPOB1)	Hi-Z	CMPOB1 output	CMPOB1 output
		Other than the above	Keep-O*1
PB2, PB4, PB6, PB7	Hi-Z	Keep-O	
PB3 (PMC0)	Hi-Z	Keep-O*3	
PB5 (USB0_VBUS)	Hi-Z	Keep-O*2	

Table 1.1 Port States in Each Processing Mode (2/2)

Port Name (Pin Name)	Reset	Software Standby Mode	
PC0, PC1, PC6, PC7	Hi-Z	Keep-O	
PC2 (RXD5)	Hi-Z	Keep-O*4	
PC3 to PC5 (PMC0)	Hi-Z	Keep-O*3	
PD0 (IRQ0)	Hi-Z	Keep-O*1	
PD1 (IRQ1)	Hi-Z	Keep-O*1	
PD2 (IRQ2)	Hi-Z	Keep-O*1	
PD3 (IRQ3)	Hi-Z	Keep-O*1	
PD4 (IRQ4)	Hi-Z	Keep-O*1	
PD5 (IRQ5)	Hi-Z	Keep-O*1	
PD6 (IRQ6)	Hi-Z	Keep-O*1	
PD7 (IRQ7)	Hi-Z	Keep-O*1	
PE0, PE1	Hi-Z	Keep-O	
PE2 (IRQ7)	Hi-Z	Keep-O*1	
PE3, PE4 (CLKOUT)	Hi-Z	CLKOUT output	CLKOUT output
		Other than the above	Keep-O
PE5 (IRQ5/CMPOB0)	Hi-Z	CMPOB0 output	CMPOB0 output
		Other than the above	Keep-O*1
PE6 (IRQ6)	Hi-Z	Keep-O*1	
PE7 (IRQ7)	Hi-Z	Keep-O*1	
PH0	Hi-Z	Keep-O	
PG7/MD	Pullup	Keep-O	
PH1 (IRQ0)	Hi-Z	Keep-O*1	
PH2 (IRQ1)	Hi-Z	Keep-O*1	
PH3	Hi-Z	Keep-O	
PH6/XCOUT	Hi-Z	Sub-Clock output	XCOU output
		Other than the above	Hi-Z
PH7/XCIN	Hi-Z	Sub-Clock Input	XCIN Input
		Other than the above	Hi-Z
PJ1, PJ3, PJ6, PJ7	Hi-Z	Keep-O	

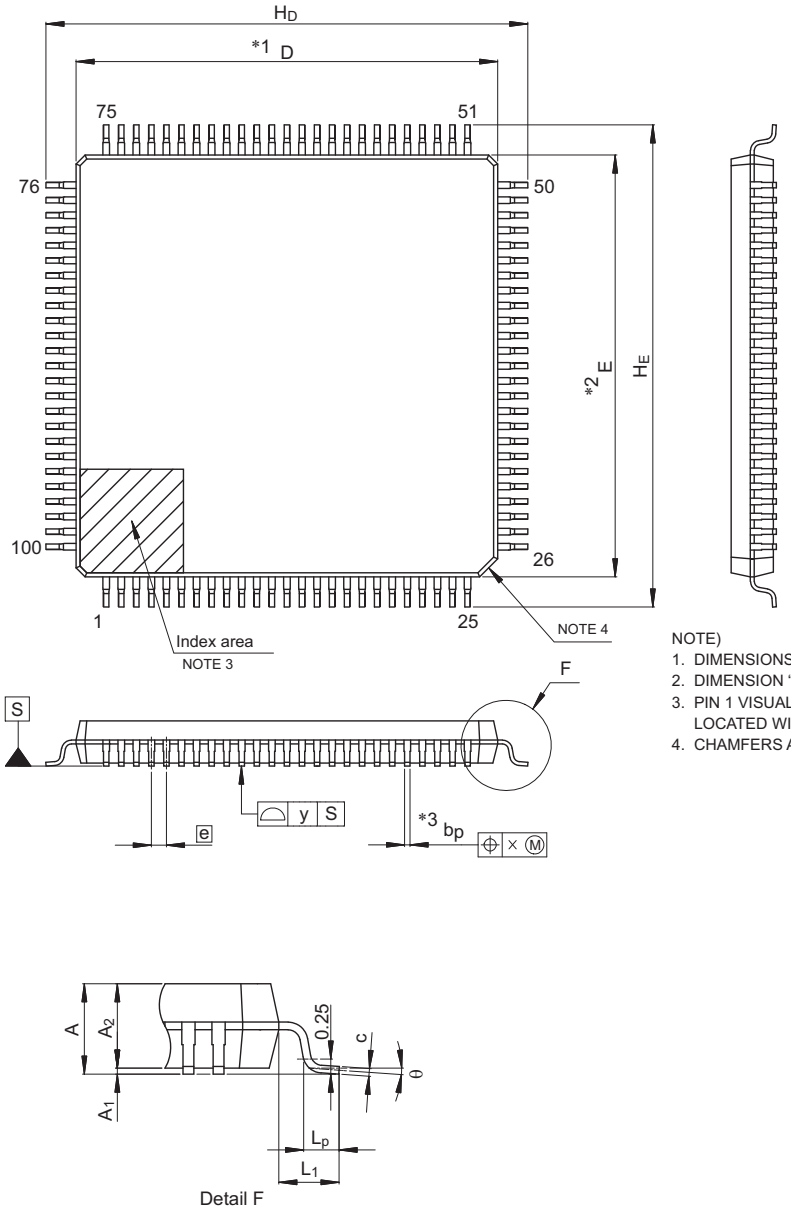
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Keep: Retained the pin state in software standby mode (Pull-up and open-drain settings are also retained).
Hi-Z: High-impedance
Note 1. Input is possible, but only when the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.
Note 2. Input is enabled if the pin is used as a USB pin (USB0_VBUS/USB0_OVRCURA/USB0_OVRCURB).
Note 3. Input is enabled even in software standby mode when it is used as an REMC external pulse signal input pin.
Note 4. Input is possible even in the software standby mode, but only when the pin is in use as the RXD pin of SCI5.

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-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm



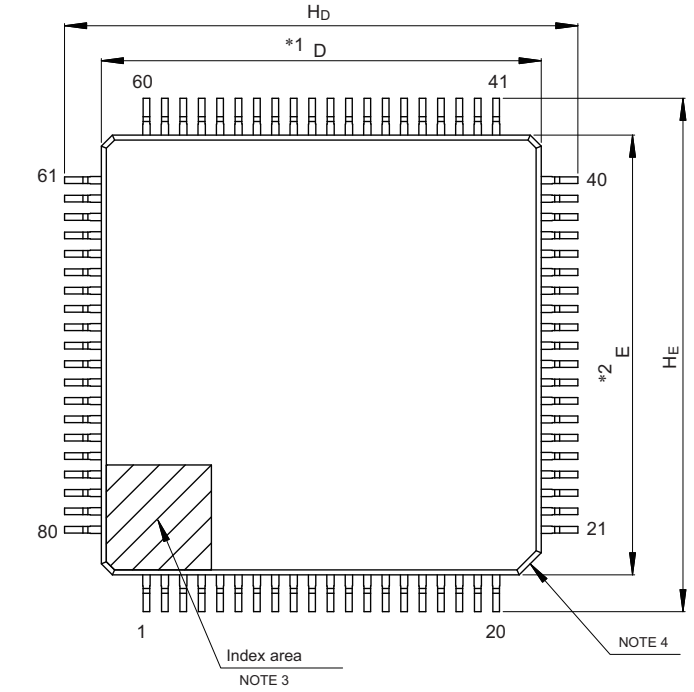
NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

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	—

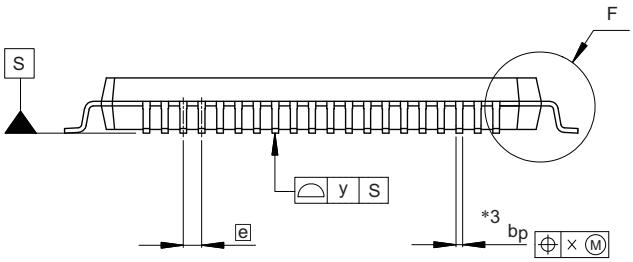
Figure A 100-Pin LFQFP (PLQP0100KB-B)

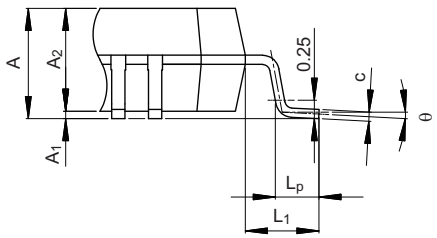
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5

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.





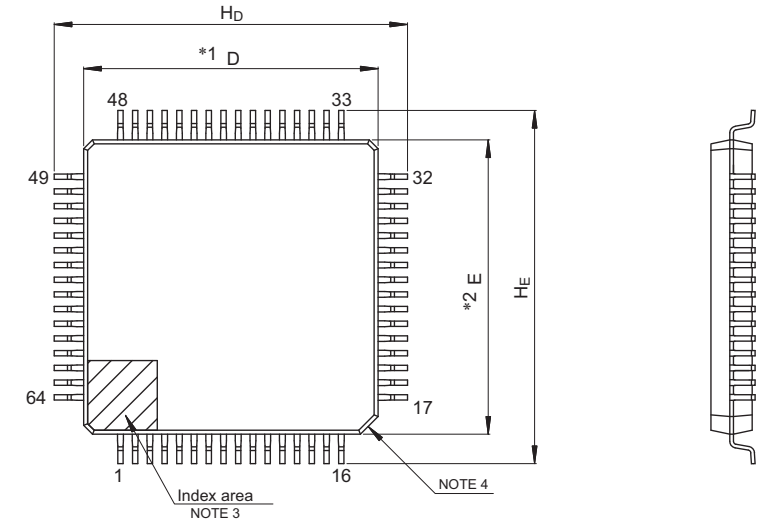
Detail F

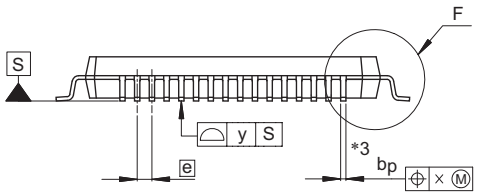
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

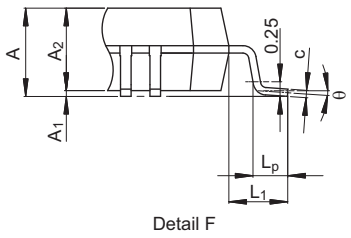
Figure B 80-Pin LFQFP (PLQP0080KB-B)

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm







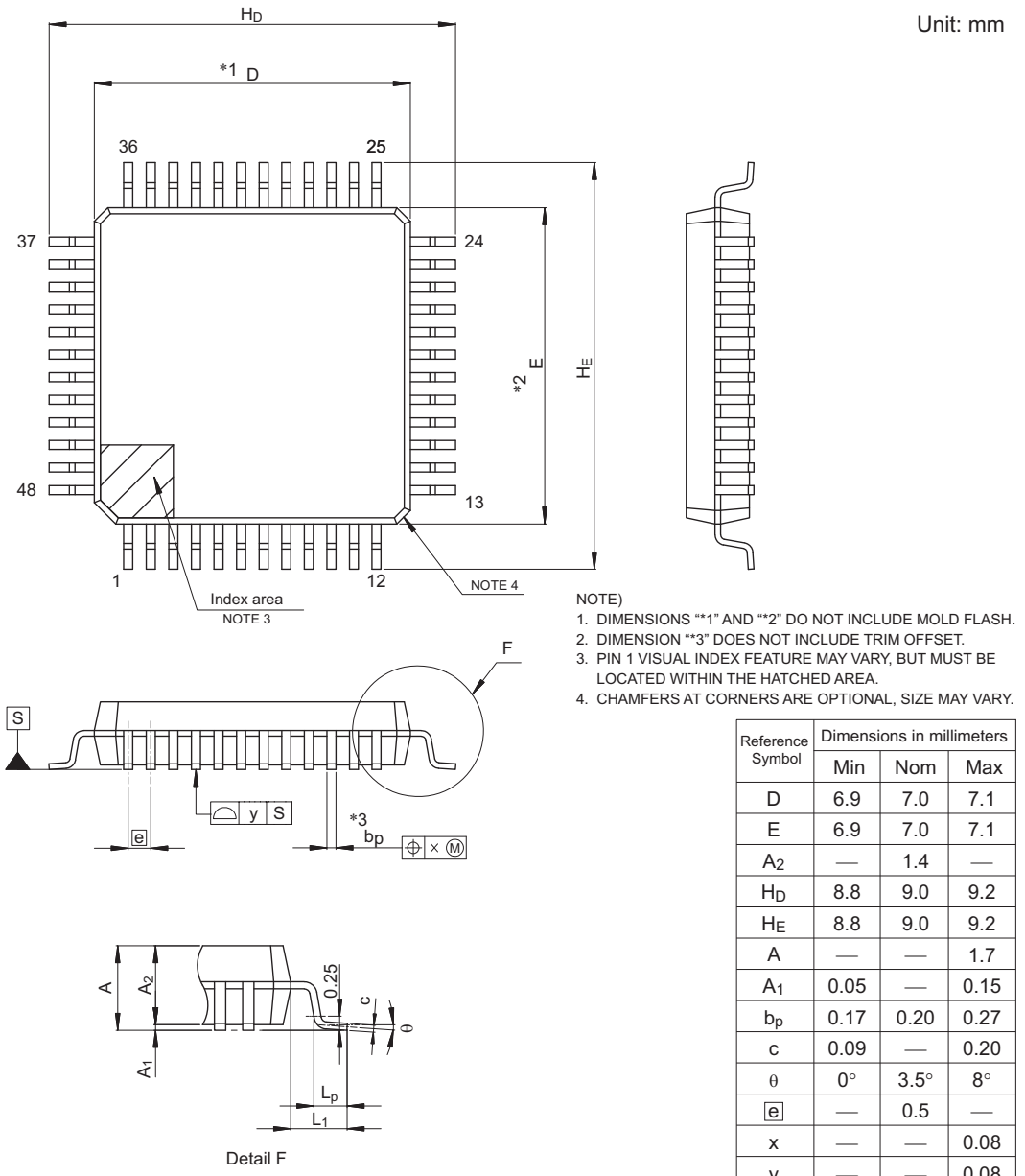
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

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 C 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

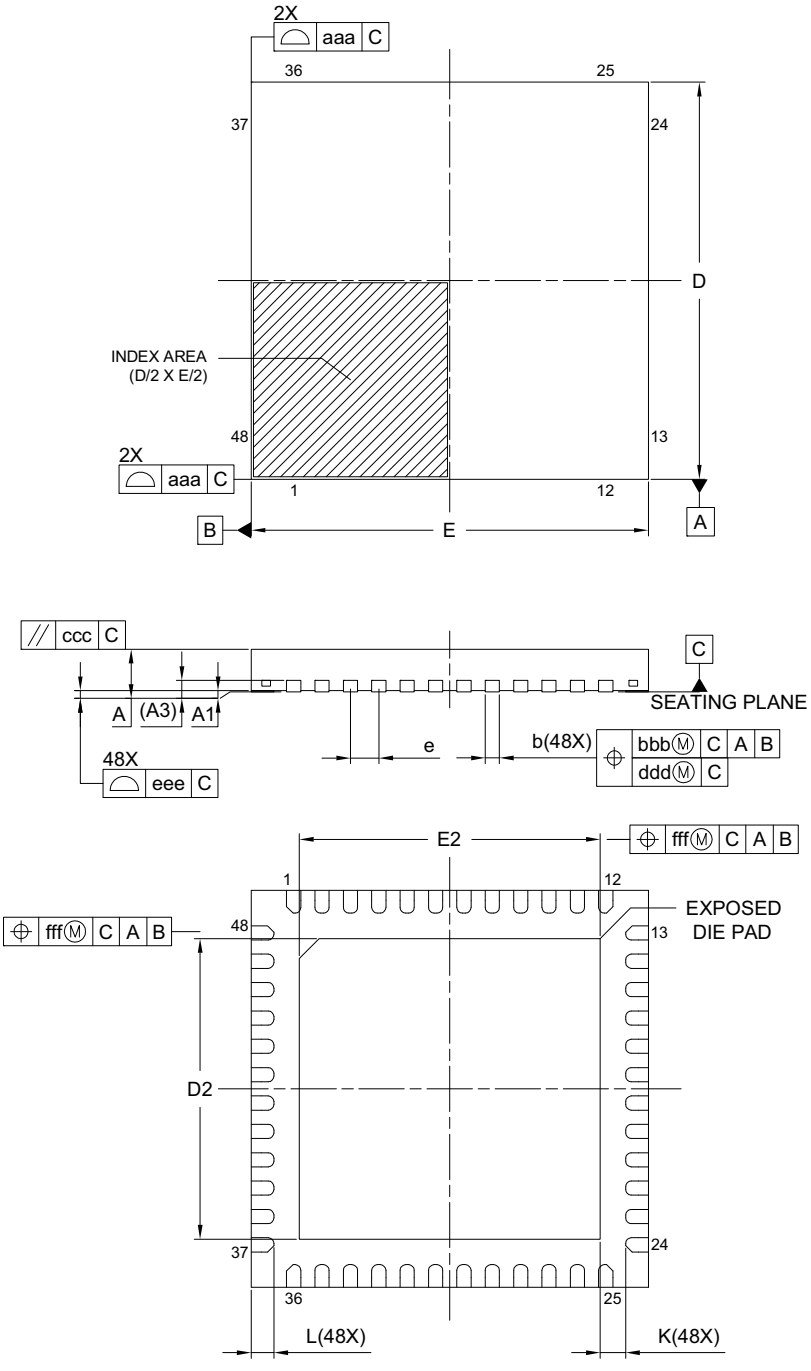


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

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 D 48-Pin LFQFP (PLQP0048KB-B)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure E 48-Pin HWQFN (PWQN0048KC-A)

REVISION HISTORY	RX260 Group, RX261 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	Jul 31, 2024	—	First edition, issued	

RX260 Group, RX261 Group User’s Manual: Hardware

Publication Date: Rev.1.00 Jul 31, 2024

Published by: Renesas Electronics Corporation

RX260 Group, RX261 Group

Renesas Electronics Corporation

R01UH1045EJ0100