User's Manual

32

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

SH7231 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine family/SH7231 Series

Renesas Electronics
www.renesas.com

Rev.2.00 Mar 2012

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

General Precautions in the Handling of MPU/MCU Products

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

1. Treatment of NC Pins

   Do not connect anything to the NC pins.

   — The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.
2. Treatment of Unused Input Pins

   Fix all unused input pins to high or low level.

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

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

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
4. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

   After applying a reset, only release the reset line after the operating clock signal has become stable.

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

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

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

How to Use This Manual

1. Objective and Target Users

   This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

   This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

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

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

   The following documents have been prepared for the SH7231 Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Data Sheet	Overview of hardware and electrical characteristics	—	—
User′s Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	SH7231 Group User′s Manual: Hardware	This user′s manual
User′s Manual: Software	Detailed descriptions of the CPU and instruction set	SH-2A, SH2A-FPU Software Manual	REJ09B0051
Application Note	Examples of applications and sample programs	The latest versions are available from our web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

2. Description of Numbers and Symbols

Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

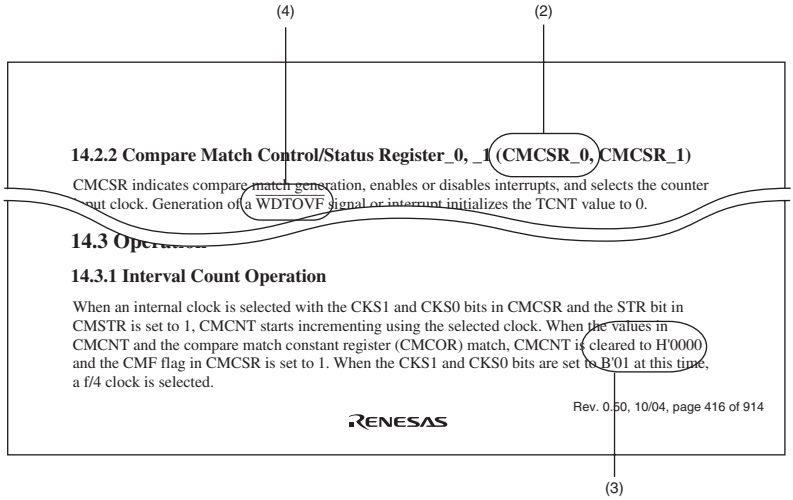
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.
[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$

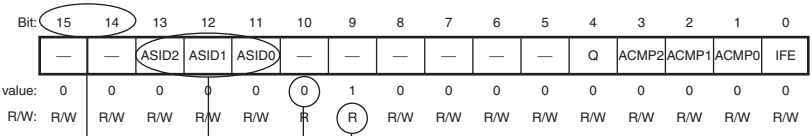


Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

3. Description of Registers

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

[Bit Chart]



[Table of Bits]

(1) (2) (3) (4) (5)

Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved
14	–	0	R	These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10	–	0	R	Reserved This bit is always read as 0.
9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

4. Description of Abbreviations

The abbreviations used in this manual are listed below.

- Abbreviations specific to this product

Abbreviation	Description
BSC	Bus controller
CPG	Clock pulse generator
DTC	Data transfer controller
INTC	Interrupt controller
SCI	Serial communication interface
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communication interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SFR	Special function register
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter
VCO	Voltage-controlled oscillator

Contents

Section 1 Overview

1.1 SH7231 Features

This LSI is a single-chip RISC (reduced instruction set computer) microcontroller that includes a Renesas-original RISC CPU as its core, and the peripheral functions required to configure a system.

The CPU in this LSI is an SH-2A CPU, which provides upward compatibility for SH-1, SH-2, and SH-2E CPUs at object code level. It has a RISC-type instruction set, superscalar architecture, and Harvard architecture, for superior rates of instruction execution. In addition, the 32-bit internal-bus architecture that is independent from the direct memory access controller enhances data processing power. This CPU brings the user the ability to set up high-performance systems with strong performance and functionality at less expense than was achievable with previous microcontrollers, and is even capable of handling realtime control applications that require high-speed characteristics.

This LSI has a floating-point unit. It also includes on-chip peripheral functions required for system configuration, such as a large-capacity ROM, a 32-Kbyte high-speed on-chip RAM, 12-Kbyte RAM for data storage, direct memory access controller (DMAC), data transfer controller (DTC), multi-function timer pulse unit 2 (MTU2/MTU2S), port output enable 2 (POE2), compare match timer (CMT), compare match timer 2 (CMT2), serial communications interface with FIFO (SCIF), serial communications interface (SCI), A/D converter (ADC), interrupt controller (INTC), I/O ports, I^2C bus interface 3 (IIC3), LVDS receiving interface (LVDS; only in the SH72135A), key scan controller (KEYC), 32-kHz timer (TIM32C), Renesas serial peripheral interface (RSPI), controller area network (RCAN-ET), and data flash module (FLD).

This LSI also provides an external memory access support function to enable direct connection to various memory devices or peripheral LSIs. These on-chip functions significantly reduce costs of designing and manufacturing application systems.

The features of this LSI are listed in table 1.1.

Table 1.1 SH7231 Features

Items	Specification
Operating modes	• Operating modes Expansion ROM invalid mode Expansion ROM valid mode Single-chip mode • Processing states Program execution state Exception handling state Bus-right released state • Low energy power consumption states Sleep mode Software standby mode Deep software standby mode Module standby mode
Interrupt controller (INTC)	• Twenty five external interrupt pins (NMI, IRQ23 to IRQ0) • On-chip peripheral interrupts: Priority level is set for each module • 16 priority levels available to set up • Register bank enabling fast register saving and restoring in interrupt processing

Items	Specification
Bus state controller (BSC)	• Address space divided into eight areas (CS0 to CS7), each a maximum of 64 Mbytes • The following features settable for each area independently — Bus size (8, 16 or 32 bits): Available sizes depend on the area. — Number of access wait cycles (different wait cycles can be specified for read and write access cycles in some areas) — Idle wait cycle insertion (between the same area access cycles or different area access cycles) — Specifying the memory to be connected to each area enables direct connection to SRAM, SRAM with byte selection, SDRAM, and burst ROM (clocked synchronous or asynchronous). The address/data multiplexed I/O (MPX) interface are also available. — Outputs a chip select signal ($\overline{CS0}$ to $\overline{CS7}$) according to the target area ($\overline{CS}$ assert or negate timing can be selected by software) • SDRAM refresh Auto refresh or self refresh mode selectable • SDRAM burst access
Direct memory access controller (DMAC)	• Four channels; external requests are available for two of them. • Can be activated by on-chip peripheral modules • Burst mode and cycle steal mode • Intermittent mode available (16 and 64 cycles supported) • Transfer information can be automatically reloaded • Transfer addresses can be incremented by four in word transfer.

Items	Specification
CPU	• Renesas original SuperH architecture • Compatible with SH-1 and SH-2 at object code level • 32-bit internal data bus • Support of an abundant register-set — Sixteen 32-bit general registers — Four 32-bit control registers — Four 32-bit system registers — Register bank for high-speed response to interrupts • RISC-type instruction set (upward compatible with SH series) — Instruction length: 16-bit fixed-length basic instructions for improved code efficiency and 32-bit instructions for high performance and usability — Load/store architecture — Delayed branch instructions — Instruction set based on C language • Superscalar architecture to execute two instructions at one time including a floating-point unit • Instruction execution time: Up to two instructions/cycle • Address space: 4 Gbytes • Internal multiplier • Five-stage pipeline • Harvard architecture

Items	Specification
Floating-point unit (FPU)	• Floating-point co-processor included • Supports single-precision (32-bit) and double-precision (64-bit) • Supports data type and exceptions that conforms to IEEE754 standard • Rounding modes: Round to nearest and round to zero • Denormalization mode: Truncate to zero • Floating-point registers — Sixteen 32-bit floating-point registers (single-precision × 16 words or double-precision × 8 words) — Two 32-bit floating-point system registers • Supports FMAC (multiplication and accumulation) instructions • Supports FDIV (division) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution time — Latency (FMAC/FADD/FSUB/FMUL): Three cycles (single-precision), eight cycles (double-precision) — Pitch (FMAC/FADD/FSUB/FMUL): One cycle (single-precision), six cycles (double-precision) Note: FMAC only supports single-precision • Five-stage pipeline
Data transfer controller (DTC)	• Data can be transferred independently of the CPU in response to peripheral I/O interrupts. • Transfer mode can be set per interrupt source (by setting up the transfer mode in memory). • Multiple data can be transferred in response to a single signal from an activating source. • Ample selection of transfer modes: normal mode, repeat mode, and block-transfer mode • Transfer unit can be set as byte, word and long word. • Conveyance to the CPU of interrupt requests which started the data transfer controller. An interrupt can be generated for the CPU after the first data transfer is completed. An interrupt can be generated for the CPU after the whole specified data transfer is completed.

Items	Specification
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EXTAL) or crystal resonator. • Input clock can be multiplied by 8 (max.) by the internal PLL circuit. • Generates five types of clock signal: — CPU clock: Maximum 100 MHz — Bus clock: Maximum 50 MHz — Peripheral clock: Maximum 50 MHz — MTU2S clock: Maximum 100 MHz — AD clock: Maximum 50 MHz; bus clock < AD clock setting is permissible.
Clock pulse generator for KEYC/TIM32C (CPG32)	• Clock mode: Input clock can be selected as external input (EXTAL 32) or 32-kHZ crystal resonator. • Clock for TIM32C/KEYC can be generated. • Continued operation in deep software standby mode is possible. • Output of the 32-kHz clock signal is possible.
Watchdog timer (WDT)	• On-chip one-channel watchdog timer • A counter overflow can reset the LSI
Power-down modes	• Four power-down modes provided to reduce the power consumption in this LSI — Sleep mode — Software standby mode — Deep software standby mode — Module standby mode

Items	Specification
Multi-function timer pulse unit 2 (MTU2)	• Maximum 16 lines of pulse inputs/outputs based on six channels of 16-bit timers and three pulse input • 21 output compare and input capture registers • Input capture function • Pulse output modes Toggle, PWM, complementary PWM, and reset-synchronized PWM modes • Synchronization of multiple counters • Complementary PWM output mode — Non-overlapping waveforms output for 3-phase inverter control — Automatic dead time setting — 0% to 100% PWM duty value specifiable — A/D converter start request delaying function — Interrupt skipping at crest or trough • Reset-synchronized PWM mode Three-phase PWM waveforms in positive and negative phases can be output with a required duty value • Phase counting mode Two-phase encoder pulse counting available
Multi-function timer pulse unit 2S (MTU2S)	Subset of MTU2, including channels 3 to 5
Port output enable2 (POE2)	Control of high-impedance state of high-current pins and CH0 pin for MTU2 on a falling edge or low-level input on the POE pin
Compare match timer (CMT)	• Two-channel 16-bit counters • Four types of clock can be selected (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) • Interrupt request can be issued when a compare match occurs
Compare match timer 2 (CMT2)	• One 32-bit counter • Capable of compare match, and has two input pins to trigger input capture, and two output-compare pins for output • Clock frequency is selectable from among four values (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) • An interrupt request can be issued when a compare match, input capture or output comparison occurs

Items	Specification
Serial communication interface (SCI)	• Four channels • Clocked synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • Bit rate adjustment function (in asynchronous mode)
Serial communication interface with FIFO (SCIF)	• Four channels • Clocked synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • Separate 16-byte FIFO registers for transmission and reception
I^2C bus interface 3 (IIC3)	• One channel • Master mode and slave mode supported
LVDS receiving interface (LVDS) for SH72315A	• Two channels • Number of data bits: 16 (fixed) • Data throughput: 40 to 320 Mbps • Internal 100-Ω terminating resistance
Renesas serial peripheral interface (RSPI)	• One channel • Synchronous mode serial communications • Master mode and slave mode selectable • Programmable bit length, clock polarity, and clock phase can be selected. • Transfer can be executed in sequential loops. • Maximum transfer rate: 12.5 MHz • Up to four slaves can be controlled in single master mode (depends on the PFC setting) Up to three slaves can be controlled in multi-master mode (depends on the PFC setting)
Controller area network (RCANT-ET)	• One channel • CAN version: Bosch 2.0B active is supported • Buffer size: 15 buffers for transmission/reception and one buffer for reception only

Items	Specification
I/O ports	• Input or output can be selected for each bit • Input pull-up function
A/D converter (ADC)	• Two modules • 10-bit resolution • Sixteen input channels (eight channels × 2 modules) • A/D conversion requests by external trigger or timer trigger • Simultaneous initiation of A/D conversion by both modules in response to an external trigger or timer trigger • Adjustable sampling time • The results of conversion by A/D_0 and A/D_1 can be read out in a single action by a longword-read instruction.
User break controller (UBC)	• Break channels: eight (of these, six are only for use with instruction fetching by the CPU) • Addresses, data values, type of access, and data size can all be set as break conditions.
User debugging interface (H-UDI)	• E10A emulator support • JTAG-standard pin assignment • For boundary scans
On-chip ROM	• 1 Mbyte (SH72315A/SH72315L) and 768 Kbytes (SH72314L)
On-chip RAM	• 32 Kbytes of memory for high-speed operation • 12 Kbytes of memory for data retention (with data retained in deep standby mode)
Key scan controller (KEYC)	• 32 channels • Setting up for a key matrix is possible. Key input: 32 pins Key input: 24 pins, matrix output: 4 pins, matrix input: 4 pins Key input: 16 pins, matrix output: 8 pins, matrix input: 8 pins • Setting of pull-up time is possible. • Setting of pull-up cycle is possible. • Storage of matrix values in a register • Operation in deep standby mood is possible. • Interrupts can initiate release from deep software standby mode.

Items	Specification
32-kHz timer (TIM32C)	• Three channels, two of which are dedicated timers for phase counting. • Can continue to operate in deep software standby mode. • Interrupts can initiate release from deep software standby mode.
Power supply voltage	• Vcc : 3.0 to 3.6 V, LVDSVcc : 3.0 to 3.6 V PLLVcc : 3.0 to 3.6 V PVcc1 : 1.65 to 1.95 V or 3.0 to 3.6 V PVcc2 : 1.65 to 1.95 V or 3.0 to 3.6 V AVcc : 3.0 to 3.6 V
Packages	• P-LFBGA1111-256 (0.5 pitch) • P-FBGA1717-272 (0.8 pitch)

1.2 Block Diagram

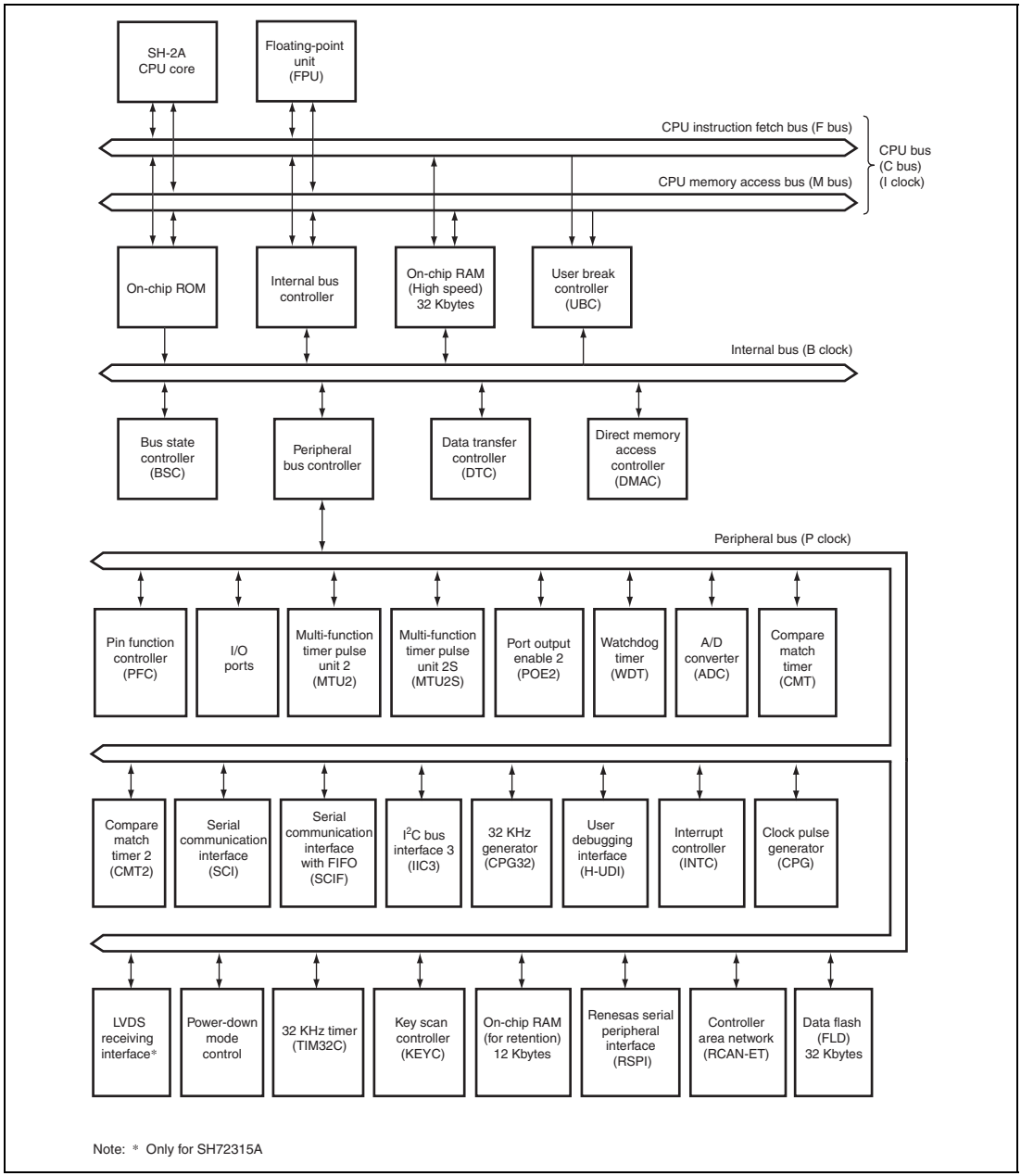


Figure 1.1 Block Diagram

1.3 Pin Assignment

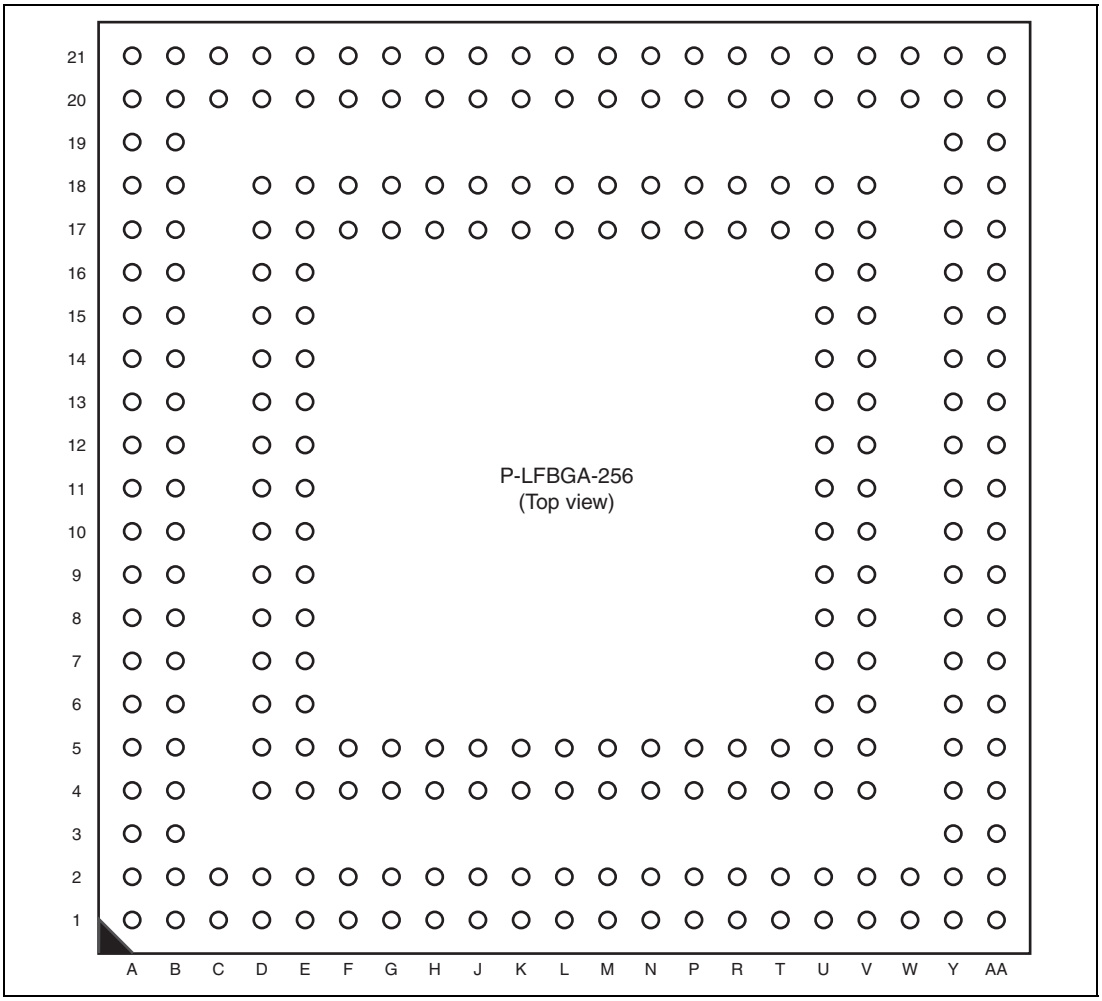


Figure 1.2 Pin Assignment Diagram (P-LFBGA1111-256)

Table 1.2 Pin assignment (P-LFBGA1111-256)

Pin number	Pin name	I/O buffer power supply
A1	PE23/TEND1/TIOC4DS/RXD7	Vcc
A2	PE20/TEND0/TIOC4AS/RXD6	Vcc
A3	PE19/DACK0/TIOC3DS/TXD6	Vcc
A4	PE21/DREQ1/TIOC4BS/SCK7	Vcc
A5	Vss	—
A6	PE17/$\overline{\text{MRES}}$/TIOC3BS/RXD0	Vcc
A7	Vss	—
A8	PE7/IRQ7/TIOC2B/RXD2	Vcc
A9	PE4/IRQ4/TIOC1A/RXD3	Vcc
A10	PE0/IRQ0/TIOC0A/SCK4	Vcc
A11	PF10/AN10	AVcc
A12	AVcc	—
A13	AVss	—
A14	PF2/AN2	AVcc
A15	PH15/TOC1	Vcc
A16	Vcc	—
A17	AUDATA1	Vcc
A18	AUDATA3	Vcc
A19	PLLVss	—
A20	PLLVcc	—
A21	PH10	Vcc
B1	PJ0/IRQ10	PVcc1
B2	PE18/DREQ0/TIOC3CS/SCK6	Vcc
B3	PE22/DACK1/TIOC4CS/TXD7	Vcc
B4	Vcc	—
B5	PE12/IRQ12/TIOC4A/SCK5	Vcc
B6	PE11/IRQ11/TIOC3D/RXD1	Vcc
B7	Vcc	—
B8	PE5/IRQ5/TIOC1B/TXD3	Vcc
B9	PE2/IRQ2/TIOC0C/RXD4	Vcc
B10	$\overline{\text{WDTOVF}}$	Vcc

Pin number	Pin name	I/O buffer power supply
B11	PF13/AN13	AVcc
B12	PF9/AN9	AVcc
B13	PF7/AN7	AVcc
B14	PF4/AN4	AVcc
B15	PF0/AN0	AVcc
B16	PH13/TIC1	Vcc
B17	AUDCK	Vcc
B18	Vss	—
B19	$\overline{\text{AUDSYNC}}$	Vcc
B20	PH11	Vcc
B21	Vss	—
C1	PJ4/TXD4	PVcc1
C2	PJ1/IRQ11	PVcc1
C20	Vcc	—
C21	EXTAL	Vcc
D1	PVcc1	—
D2	PJ3/SCK4	PVcc1
D4	PE14/IRQ14/TIOC4C/RXD5	Vcc
D5	PE13/IRQ13/TIOC4B/TXD5	Vcc
D6	PE15/IRQ15/TIOC4D/SCK0	Vcc
D7	PE9/IRQ9/TIOC3B/SCK1	Vcc
D8	PE6/IRQ6/TIOC2A/TXD2	Vcc
D9	PE1/IRQ1/TIOC0B/TXD4	Vcc
D10	PF15/AN15	AVcc
D11	PF14/AN14	AVcc
D12	PF11/AN11	AVcc
D13	PF8/AN8	AVcc
D14	PF5/AN5	AVcc
D15	PF1/AN1	AVcc
D16	PH14/TOC0	Vcc
D17	AUDATA0	Vcc
D18	PH9	Vcc

Pin number	Pin name	I/O buffer power supply
D20	PH7/TXD7	Vcc
D21	XTAL	Vcc
E1	PVss1	—
E2	PJ8/RXD5	PVcc1
E4	PJ6/SCK5	PVcc1
E5	PJ2/IRQ12	PVcc1
E6	PE16/$\overline{\text{UBCTRG}}$/TIOC3AS/TXD0	Vcc
E7	PE10/IRQ10/TIOC3C/TXD1	Vcc
E8	PE8/IRQ8/TIOC3A/SCK2	Vcc
E9	PE3/IRQ3/TIOC0D/SCK3	Vcc
E10	MD1	Vcc
E11	PF12/AN12	AVcc
E12	AVref	—
E13	PF6/AN6	AVcc
E14	PF3/AN3	AVcc
E15	MD0	Vcc
E16	PH12/TIC0	Vcc
E17	AUDATA2	Vcc
E18	PH5/TIC5WS	Vcc
E20	PH6/SCK7	Vcc
E21	PH8/RXD7	Vcc
F1	PJ14/SSL2	PVcc1
F2	PJ11/MISO0	PVcc1
F4	PJ9/RSPCK0	PVcc1
F5	PJ5/RXD4	PVcc1
F17	PH4/TIC5VS	Vcc
F18	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Vcc
F20	PH3/TIC5US	Vcc
F21	Vcc	—
G1	PVss2	—
G2	PJ15/SSL3	PVcc1
G4	PJ12/SSL0	PVcc1

Pin number	Pin name	I/O buffer power supply
G5	PJ7/TXD5	PVcc1
G17	TMS	Vcc
G18	TDO	Vcc
G20	TCK	Vcc
G21	$\overline{\text{RES}}$	Vcc
H1	PVcc2	—
H2	PK2/IRQ15	PVcc2
H4	PK0/IRQ13	PVcc2
H5	PJ10/MOSI0	PVcc1
H17	Vss	—
H18	TDI	Vcc
H20	$\overline{\text{ASEMD0}}$	Vcc
H21	$\overline{\text{TRST}}$	Vcc
J1	PL2/RXIN0P (for SH72315A)	LVDSVcc
	PL2 (for SH72315L/SH72314L)	Vcc
J2	LVDSVss (for SH72315A)	—
	Vss (for SH72315L/SH72314L)	—
J4	PK1/IRQ14	PVcc2
J5	PJ13/SSL1	PVcc1
J17	Vcc	—
J18	PH0/TIC5U	Vcc
J20	PH2/TIC5W	Vcc
J21	NMI	Vcc
K1	PL3/RXIN0M (for SH72315A)	LVDSVcc
	PL3 (for SH72315L/SH72314L)	Vcc
K2	PL0/RXCLKINP (for SH72315A)	LVDSVcc
	PL0 (for SH72315L/SH72314L)	Vcc
K4	PK6	PVcc2
K5	PK3/SCK6	PVcc2
K17	PC13/A13/IRQ21/SCK7	Vcc
K18	PC14/A14/IRQ22/TXD7	Vcc
K20	PC15/A15/IRQ23/RXD7	Vcc

Pin number	Pin name	I/O buffer power supply
K21	PH1/TIC5V	Vcc
L1	PL4/RXIN1P (for SH72315A)	LVDSVcc
	PL4 (for SH72315L/SH72314L)	Vcc
L2	PL1/RXCLKINM (for SH72315A)	LVDSVcc
	PL1 (for SH72315L/SH72314L)	Vcc
L4	PK7	PVcc2
L5	PK4/TXD6	PVcc2
L17	PC12/A12/IRQ20/RXD4	Vcc
L18	PC11/A11/IRQ19/TXD4	Vcc
L20	Vss	—
L21	PA15/CK/IRQ11/SCK5	Vcc
M1	PL5/RXIN1M (for SH72315A)	LVDSVcc
	PL5 (for SH72315L/SH72314L)	Vcc
M2	LVDSVss (for SH72315A)	—
	Vss (for SH72315L/SH72314L)	—
M4	LVDSVcc (for SH72315A)	—
	Vcc (for SH72315L/SH72314L)	—
M5	PK5/RXD6	PVcc2
M17	V_{CL}	—
M18	PC10/A10/IRQ18/SCK4	Vcc
M20	Vcc	—
M21	PC6/A6/$\overline{\text{UBCTRG}}$/RXD3	Vcc
N1	PG2/IRQ2	Vcc
N2	PG0/IRQ0	Vcc
N4	PG1/IRQ1	Vcc
N5	PG3/IRQ3	Vcc
N17	PC9/A9/IRQ17/RXD6	Vcc
N18	PC5/A5/TIC5US/TXD3	Vcc
N20	PC2/A2/TIC5W/$\overline{\text{POE2}}$	Vcc
N21	PC1/A1/TIC5V/$\overline{\text{POE1}}$	Vcc
P1	PG6/IRQ6	Vcc
P2	PG4/IRQ4	Vcc

Pin number	Pin name	I/O buffer power supply
P4	PG5/IRQ5	Vcc
P5	PG7/IRQ7	Vcc
P17	PC8/A8/IRQ16/TXD6	Vcc
P18	PC4/A4/TIC5VS/SCK3	Vcc
P20	PC0/A0/TIC5U/$\overline{\text{POE0}}$	Vcc
P21	Vcc	—
R1	Vss	—
R2	Vcc	—
R4	PG8/IRQ8	Vcc
R5	PG9/IRQ9	Vcc
R17	PC7/A7/$\overline{\text{IRQOUT}}$/SCK6	Vcc
R18	PC3/A3/TIC5WS/$\overline{\text{POE3}}$	Vcc
R20	PA7/$\overline{\text{CS7}}$/IRQ3/TCLKB	Vcc
R21	Vss	—
T1	V_{CL}	—
T2	PG10/TI32I0A	Vcc
T4	PG11/TI32I0B	Vcc
T5	PG12/TI32I1A	Vcc
T17	PA6/$\overline{\text{CS6}}$/IRQ2/TCLKA	Vcc
T18	PA8/RD$\overline{\text{WR}}$/IRQ4/TCLKC	Vcc
T20	PA9/$\overline{\text{BS}}$/IRQ5/TCLKD	Vcc
T21	PA16/$\overline{\text{BACK}}$/IRQ12/$\overline{\text{ADTRG}}$	Vcc
U1	PG14/CK32	Vcc
U2	PG13/TI32I1B	Vcc
U4	PG15	Vcc
U5	PB8/$\overline{\text{CS1}}$/$\overline{\text{CS5}}$/RXD2	Vcc
U6	PB7/$\overline{\text{CS0}}$/$\overline{\text{CS4}}$/$\overline{\text{REFOUT}}$	Vcc
U7	V_{CL}	—
U8	PD17/D17/TCLKB/KEY17/$\overline{\text{COM1}}$	Vcc
U9	PD19/D19/TCLKD/KEY19/$\overline{\text{COM3}}$	Vcc
U10	PD21/D21/TIC5VS/KEY21/$\overline{\text{COM5}}$	Vcc
U11	Vss	—

Pin number	Pin name	I/O buffer power supply
U12	Vcc	—
U13	PD26/D26/DACK1/KEY26/$\overline{\text{COM2}}$/P2	Vcc
U14	PD11/D11/TXD5/KEY11	Vcc
U15	PD29/D29/TIC5U/KEY29/P1/P5	Vcc
U16	PD30/D30/TIC5V/KEY30/P2/P6	Vcc
U17	PA4/$\overline{\text{CS4}}$/CRx0/TXD1	Vcc
U18	PA5/$\overline{\text{CS5}}$/RXD1	Vcc
U20	Vcc	—
U21	Vss	—
V1	PB4/A18/$\overline{\text{POE5}}$/RXD0	Vcc
V2	Vcc	—
V4	PB6/A20/$\overline{\text{POE8}}$/SCK0	Vcc
V5	PB3/$\overline{\text{IRQOUT}}$/$\overline{\text{POE6}}$/SDA	Vcc
V6	PB9/A21/CKE/TXD2	Vcc
V7	PD16/D16/TCLKA/KEY16/$\overline{\text{COM0}}$	Vcc
V8	PD18/D18/TCLKC/KEY18/$\overline{\text{COM2}}$	Vcc
V9	PD20/D20/TIC5WS/KEY20/$\overline{\text{COM4}}$	Vcc
V10	PD22/D22/TIC5US/KEY22/$\overline{\text{COM6}}$	Vcc
V11	PD23/D23/TEND0/KEY23/$\overline{\text{COM7}}$	Vcc
V12	PD24/D24/DREQ0/KEY24/$\overline{\text{COM0}}$/P0	Vcc
V13	PD25/D25/DREQ1/KEY25/$\overline{\text{COM1}}$/P1	Vcc
V14	PD27/D27/DACK0/KEY27/$\overline{\text{COM3}}$/P3	Vcc
V15	PD12/D12/RXD5/KEY12	Vcc
V16	PD14/D14/TXD3/KEY14	Vcc
V17	PD31/D31/TIC5W/KEY31/P3/P7	Vcc
V18	PA2/$\overline{\text{CS2}}$/CTx0/SCK0	Vcc
V20	PA18/$\overline{\text{WAIT}}$/IRQ14/$\overline{\text{POE4}}$	Vcc
V21	PA19/$\overline{\text{AH}}$/IRQ15/$\overline{\text{POE0}}$	Vcc
W1	XTAL32	Vcc
W2	PB5/A19/$\overline{\text{POE4}}$/TXD0	Vcc
W20	PA3/$\overline{\text{CS3}}$/CTx0/SCK1	Vcc
W21	PA17/$\overline{\text{BREQ}}$/IRQ13/$\overline{\text{POE8}}$	Vcc

Pin number	Pin name	I/O buffer power supply
Y1	EXTAL32	Vcc
Y2	Vss	—
Y3	PB13/A25/$\overline{\text{RASU}}$/SCK6	Vcc
Y4	PB0/A16/$\overline{\text{POE0}}$	Vcc
Y5	PB2/$\overline{\text{MRES}}$/$\overline{\text{POE7}}$/SCL	Vcc
Y6	Vss	—
Y7	Vcc	—
Y8	PD2/D2/RXD1/KEY2	Vcc
Y9	PD3/D3/SCK7/KEY3	Vcc
Y10	PD4/D4/TXD7/KEY4	Vcc
Y11	PD6/D6/SCK2/KEY6	Vcc
Y12	PD7/D7/$\overline{\text{POE8}}$/KEY7	Vcc
Y13	PD9/D9/RXD2/KEY9	Vcc
Y14	PD10/D10/SCK5/KEY10	Vcc
Y15	PD28/D28/TEND1/KEY28/P0/P4	Vcc
Y16	Vss	—
Y17	Vcc	—
Y18	PA1/$\overline{\text{CS1}}$/IRQ1/TXD0	Vcc
Y19	PA11/$\overline{\text{WRHL}}$/DQMUL/IRQ7/TXD4	Vcc
Y20	PA14/$\overline{\text{RD}}$/IRQ10/TXD5	Vcc
Y21	PA13/$\overline{\text{WRL}}$/DQMLL/IRQ9/RXD5	Vcc
AA1	PB11/A23/$\overline{\text{CASU}}$/RXD6	Vcc
AA2	PB10/A22/$\overline{\text{CASL}}$/SCK2	Vcc
AA3	PB12/A24/$\overline{\text{RASL}}$/TXD6	Vcc
AA4	Vcc	—
AA5	Vss	—
AA6	PB1/A17/$\overline{\text{ADTRG}}$	Vcc
AA7	PD0/D0/SCK1/KEY0	Vcc
AA8	PD1/D1/TXD1/KEY1	Vcc
AA9	Vss	—
AA10	Vcc	—
AA11	PD5/D5/RXD7/KEY5	Vcc

Pin number	Pin name	I/O buffer power supply
AA12	PD8/D8/TXD2/KEY8	Vcc
AA13	Vss	—
AA14	Vcc	—
AA15	PD13/D13/SCK3/KEY13	Vcc
AA16	PD15/D15/RXD3/KEY15	Vcc
AA17	PA0/$\overline{\text{CS0}}$/IRQ0/RXD0	Vcc
AA18	PA10/$\overline{\text{WRHH}}$/DQMUU/IRQ6/RXD4	Vcc
AA19	Vcc	—
AA20	Vss	—
AA21	PA12/$\overline{\text{WRH}}$/DQMLU/IRQ8/SCK4	Vcc

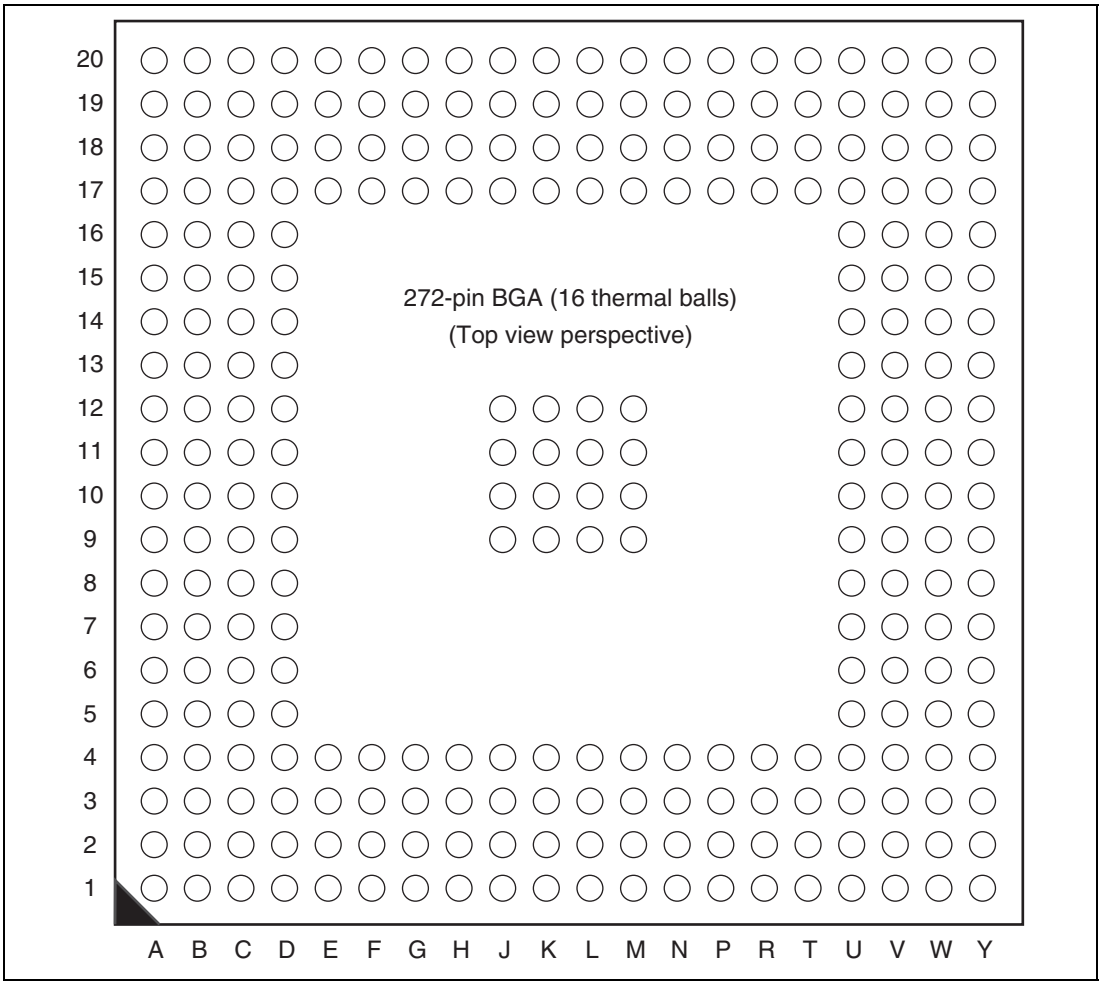


Figure 1.3 Pin Assignment Diagram (P-FBGA1717-272)

Table 1.3 Pin assignment (P-FBGA1717-272)

Pin number	Pin name	I/O buffer power supply
A1	PE19/DACK0/TIOC3DS/TXD6	Vcc
A2	Vcc	—
A3	Vss	—
A4	PE16/$\overline{\text{UBCTRG}}$/TIOC3AS/TXD0	Vcc
A5	Vss	—
A6	Vcc	—
A7	PE6/IRQ6/TIOC2A/TXD2	Vcc
A8	PE4/IRQ4/TIOC1A/RXD3	Vcc
A9	PE0/IRQ0/TIOC0A/SCK4	Vcc
A10	PF14/AN14	AVcc
A11	PF13/AN13	AVcc
A12	AVref	—
A13	AVcc	—
A14	AVss	—
A15	PF2/AN2	AVcc
A16	MD0	Vcc
A17	PH13/TIC1	Vcc
A18	PLLVss	—
A19	PLLVcc	—
A20	PH10	Vcc
B1	PJ2/IRQ12	PVcc1
B2	PJ0/IRQ10	PVcc1
B3	PE22/DACK1/TIOC4CS/TXD7	Vcc
B4	PE14/IRQ14/TIOC4C/RXD5	Vcc
B5	PE15/IRQ15/TIOC4D/SCK0	Vcc
B6	PE11/IRQ11/TIOC3D/RXD1	Vcc
B7	PE9/IRQ9/TIOC3B/SCK1	Vcc
B8	PE5/IRQ5/TIOC1B/TXD3	Vcc
B9	PE2/IRQ2/TIOC0C/RXD4	Vcc
B10	$\overline{\text{WDTOVF}}$	Vcc
B11	PF12/AN12	AVcc

Pin number	Pin name	I/O buffer power supply
B12	PF10/AN10	AVcc
B13	PF6/AN6	AVcc
B14	PF4/AN4	AVcc
B15	PF1/AN1	AVcc
B16	PH15/TOC1	Vcc
B17	PH12/TIC0	Vcc
B18	AUDATA1	Vcc
B19	Vss	—
B20	EXTAL	Vcc
C1	PJ3/SCK4	PVcc1
C2	PJ6/SCK5	PVcc1
C3	PE18/DREQ0/TIOC3CS/SCK6	Vcc
C4	PE21/DREQ1/TIOC4BS/SCK7	Vcc
C5	PE13/IRQ13/TIOC4B/TXD5	Vcc
C6	PE17/$\overline{\text{MRES}}$/TIOC3BS/RXD0	Vcc
C7	PE10/IRQ10/TIOC3C/TXD1	Vcc
C8	PE8/IRQ8/TIOC3A/SCK2	Vcc
C9	PE3/IRQ3/TIOC0D/SCK3	Vcc
C10	MD1	Vcc
C11	PF11/AN11	AVcc
C12	PF8/AN8	AVcc
C13	PF5/AN5	AVcc
C14	PF0/AN0	AVcc
C15	PH14/TOC0	Vcc
C16	AUDATA0	Vcc
C17	AUDCK	Vcc
C18	PH11	Vcc
C19	Vcc	—
C20	XTAL	Vcc
D1	PJ8/RXD5	PVcc1
D2	PJ9/RSPCK0	PVcc1
D3	PJ4/TXD4	PVcc1

Pin number	Pin name	I/O buffer power supply
D4	PJ1/IRQ11	PVcc1
D5	PE20/TEND0/TIOC4AS/RXD6	Vcc
D6	PE23/TEND1/TIOC4DS/RXD7	Vcc
D7	PE12/IRQ12/TIOC4A/SCK5	Vcc
D8	PE7/IRQ7/TIOC2B/RXD2	Vcc
D9	PE1/IRQ1/TIOC0B/TXD4	Vcc
D10	PF15/AN15	AVcc
D11	PF9/AN9	AVcc
D12	PF7/AN7	AVcc
D13	PF3/AN3	AVcc
D14	Vcc	—
D15	Vss	—
D16	AUDATA2	Vcc
D17	AUDATA3	Vcc
D18	PH8/RXD7	Vcc
D19	PH7/TXD7	Vcc
D20	Vcc	—
E1	PJ10/MOSI0	PVcc1
E2	PJ12/SSL0	PVcc1
E3	PJ5/RXD4	PVcc1
E4	PVcc1	—
E17	$\overline{\text{AUDSYNC}}$	Vcc
E18	PH6/SCK7	Vcc
E19	PH5/TIC5WS	Vcc
E20	$\overline{\text{RES}}$	Vcc
F1	PVcc2	—
F2	PJ13/SSL1	PVcc1
F3	PJ7/TXD5	PVcc1
F4	PVss1	—
F17	PH9	Vcc
F18	PH4/TIC5VS	Vcc
F19	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Vcc

Pin number	Pin name	I/O buffer power supply
F20	TMS	Vcc
G1	PVss2	—
G2	PK3/SCK6	PVcc2
G3	PJ14/SSL2	PVcc1
G4	PJ11/MISO0	PVcc1
G17	PH3/TIC5US	Vcc
G18	TCK	Vcc
G19	TDO	Vcc
G20	Vcc	—
H1	PK6	PVcc2
H2	PK4/TXD6	PVcc2
H3	PK0/IRQ13	PVcc2
H4	PJ15/SSL3	PVcc1
H17	$\overline{\text{TRST}}$	Vcc
H18	TDI	Vcc
H19	$\overline{\text{ASEMD0}}$	Vcc
H20	Vss	—
J1	PL2/RXIN0P (for SH72315A)	LVDSVcc
	PL2 (for SH72315L/SH72314L)	Vcc
J2	LVDSVss (for SH72315A)	—
	Vss (for SH72315L/SH72314L)	—
J3	PK2/IRQ15	PVcc2
J4	PK1/IRQ14	PVcc2
J9	NC	—
J10	NC	—
J11	NC	—
J12	NC	—
J17	PH2/TIC5W	Vcc
J18	NMI	Vcc
J19	PH1/TIC5V	Vcc
J20	PC15/A15/IRQ23/RXD7	Vcc

Pin number	Pin name	I/O buffer power supply
K1	PL3/RXIN0M (for SH72315A)	LVDSVcc
	PL3 (for SH72315L/SH72314L)	Vcc
K2	PL0/RXCLKINP (for SH72315A)	LVDSVcc
	PL0 (for SH72315L/SH72314L)	Vcc
K3	PK5/RXD6	PVcc2
K4	PK7	PVcc2
K9	NC	—
K10	Vss	—
K11	Vss	—
K12	NC	—
K17	PC11/A11/IRQ19/TXD4	Vcc
K18	PH0/TIC5U	Vcc
K19	PC14/A14/IRQ22/TXD7	Vcc
K20	PC13/A13/IRQ21/SCK7	Vcc
L1	PL4/RXIN1P (for SH72315A)	LVDSVcc
	PL4 (for SH72315L/SH72314L)	Vcc
L2	PL1/RXCLKINM (for SH72315A)	LVDSVcc
	PL1 (for SH72315L/SH72314L)	Vcc
L3	LVDSVcc (for SH72315A)	—
	Vcc (for SH72315L/SH72314L)	—
L4	PG0/IRQ0	Vcc
L9	NC	—
L10	Vss	—
L11	Vss	—
L12	NC	—
L17	PC1/A1/TIC5V/$\overline{\text{POE1}}$	Vcc
L18	PC12/A12/IRQ20/RXD4	Vcc
L19	Vss	—
L20	PA15/CK/IRQ11/SCK5	Vcc
M1	PL5/RXIN1M (for SH72315A)	LVDSVcc
	PL5 (for SH72315L/SH72314L)	Vcc
M2	LVDSVss (for SH72315A)	—
	Vss (for SH72315L/SH72314L)	—

Pin number	Pin name	I/O buffer power supply
M3	PG6/IRQ6	Vcc
M4	PG7/IRQ7	Vcc
M9	NC	—
M10	NC	—
M11	NC	—
M12	NC	—
M17	PC8/A8/IRQ16/TXD6	Vcc
M18	PC6/A6/$\overline{\text{UBCTRG}}$/RXD3	Vcc
M19	PC10/A10/IRQ18/SCK4	Vcc
M20	Vcc	—
N1	PG1/IRQ1	Vcc
N2	PG2/IRQ2	Vcc
N3	PG4/IRQ4	Vcc
N4	PG11/TI32I0B	Vcc
N17	V_{CL}	—
N18	PC9/A9/IRQ17/RXD6	Vcc
N19	PC5/A5/TIC5US/TXD3	Vcc
N20	PC2/A2/TIC5W/$\overline{\text{POE2}}$	Vcc
P1	PG3/IRQ3	Vcc
P2	PG5/IRQ5	Vcc
P3	PG8/IRQ8	Vcc
P4	PG12/TI32I1A	Vcc
P17	Vcc	—
P18	PA7/$\overline{\text{CS7}}$/IRQ3/TCLKB	Vcc
P19	PC4/A4/TIC5VS/SCK3	Vcc
P20	PC0/A0/TIC5U/$\overline{\text{POE0}}$	Vcc
R1	Vcc	—
R2	V_{CL}	—
R3	PG9/IRQ9	Vcc
R4	PB6/A20/$\overline{\text{POE8}}$/SCK0	Vcc
R17	Vss	—
R18	PA16/$\overline{\text{BACK}}$/IRQ12/$\overline{\text{ADTRG}}$	Vcc

Pin number	Pin name	I/O buffer power supply
R19	PC7/A7/$\overline{\text{IRQOUT}}$/SCK6	Vcc
R20	PC3/A3/TIC5WS/$\overline{\text{POE3}}$	Vcc
T1	Vss	—
T2	PG13/TI32I1B	Vcc
T3	PG10/TI32I0A	Vcc
T4	PB5/A19/$\overline{\text{POE4}}$/TXD0	Vcc
T17	PA4/$\overline{\text{CS4}}$/CRx0/TXD1	Vcc
T18	PA5/$\overline{\text{CS5}}$/RXD1	Vcc
T19	PA8/RD$\overline{\text{WR}}$/IRQ4/TCLKC	Vcc
T20	PA9/$\overline{\text{BS}}$/IRQ5/TCLKD	Vcc
U1	PG15	Vcc
U2	PB4/A18/$\overline{\text{POE5}}$/RXD0	Vcc
U3	PG14/CK32	Vcc
U4	PB11/A23/$\overline{\text{CASU}}$/RXD6	Vcc
U5	PB12/A24/$\overline{\text{RASL}}$/TXD6	Vcc
U6	Vss	—
U7	Vcc	—
U8	PD0/D0/SCK1/KEY0	Vcc
U9	Vcc	—
U10	PD22/D22/TIC5US/KEY22/$\overline{\text{COM6}}$	Vcc
U11	PD24/D24/DREQ0/KEY24/$\overline{\text{COM0}}$/P0	Vcc
U12	PD9/D9/RXD2/KEY9	Vcc
U13	PD27/D27/DACK0/KEY27/$\overline{\text{COM3}}$/P3	Vcc
U14	PD15/D15/RXD3/KEY15	Vcc
U15	PD31/D31/TIC5W/KEY31/P3/P7	Vcc
U16	PA1/$\overline{\text{CS1}}$/IRQ1/TXD0	Vcc
U17	PA17/$\overline{\text{BREQ}}$/IRQ13/$\overline{\text{POE8}}$	Vcc
U18	PA3/$\overline{\text{CS3}}$/CTx0/SCK1	Vcc
U19	PA18/$\overline{\text{WAIT}}$/IRQ14/$\overline{\text{POE4}}$	Vcc
U20	PA6/$\overline{\text{CS6}}$/IRQ2/TCLKA	Vcc
V1	XTAL32	Vcc
V2	Vcc	—

Pin number	Pin name	I/O buffer power supply
V3	PB2/$\overline{\text{MRES}}$/$\overline{\text{POE7}}$/SCL	Vcc
V4	PB0/A16/$\overline{\text{POE0}}$	Vcc
V5	PB3/$\overline{\text{IRQOUT}}$/$\overline{\text{POE6}}$/SDA	Vcc
V6	PB9/A21/CKE/TXD2	Vcc
V7	PD17/D17/TCLKB/KEY17/$\overline{\text{COM1}}$	Vcc
V8	PD18/D18/TCLKC/KEY18/$\overline{\text{COM2}}$	Vcc
V9	PD20/D20/TIC5WS/KEY20/$\overline{\text{COM4}}$	Vcc
V10	PD5/D5/RXD7/KEY5	Vcc
V11	PD7/D7/$\overline{\text{POE8}}$/KEY7	Vcc
V12	PD26/D26/DACK1/KEY26/$\overline{\text{COM2}}$/P2	Vcc
V13	PD11/D11/TXD5/KEY11	Vcc
V14	PD29/D29/TIC5U/KEY29/P1/P5	Vcc
V15	PD30/D30/TIC5V/KEY30/P2/P6	Vcc
V16	PA0/$\overline{\text{CS0}}$/IRQ0/RXD0	Vcc
V17	PA2/$\overline{\text{CS2}}$/CTx0/SCK0	Vcc
V18	PA14/$\overline{\text{RD}}$/IRQ10/TXD5	Vcc
V19	PA19/$\overline{\text{AH}}$/IRQ15/$\overline{\text{POE0}}$	Vcc
V20	Vss	—
W1	EXTAL32	Vcc
W2	Vss	—
W3	PB13/A25/$\overline{\text{RASU}}$/SCK6	Vcc
W4	PB7/$\overline{\text{CS0}}$/$\overline{\text{CS4}}$/$\overline{\text{REFOUT}}$	Vcc
W5	PB1/A17/$\overline{\text{ADTRG}}$	Vcc
W6	PD16/D16/TCLKA/KEY16/$\overline{\text{COM0}}$	Vcc
W7	PD2/D2/RXD1/KEY2	Vcc
W8	PD19/D19/TCLKD/KEY19/$\overline{\text{COM3}}$	Vcc
W9	PD21/D21/TIC5VS/KEY21/$\overline{\text{COM5}}$	Vcc
W10	PD6/D6/SCK2/KEY6	Vcc
W11	PD23/D23/TEND0/KEY23/$\overline{\text{COM7}}$	Vcc
W12	PD25/D25/DREQ1/KEY25/$\overline{\text{COM1}}$/P1	Vcc
W13	PD10/D10/SCK5/KEY10	Vcc
W14	PD28/D28/TEND1/KEY28/P0/P4	Vcc

Pin number	Pin name	I/O buffer power supply
W15	PD13/D13/SCK3/KEY13	Vcc
W16	Vcc	—
W17	PA10/$\overline{\text{WRHH}}$/DQMUU/IRQ6/RXD4	Vcc
W18	PA12/$\overline{\text{WRH}}$/DQMLU/IRQ8/SCK4	Vcc
W19	Vss	—
W20	Vcc	—
Y1	PB10/A22/$\overline{\text{CASL}}$/SCK2	Vcc
Y2	PB8/$\overline{\text{CS1}}$/$\overline{\text{CS5}}$/RXD2	Vcc
Y3	Vss	—
Y4	Vcc	—
Y5	V_{CL}	—
Y6	PD1/D1/TXD1/KEY1	Vcc
Y7	Vss	—
Y8	PD3/D3/SCK7/KEY3	Vcc
Y9	PD4/D4/TXD7/KEY4	Vcc
Y10	Vss	—
Y11	Vcc	—
Y12	PD8/D8/TXD2/KEY8	Vcc
Y13	Vss	—
Y14	Vcc	—
Y15	PD12/D12/RXD5/KEY12	Vcc
Y16	Vss	—
Y17	PD14/D14/TXD3/KEY14	Vcc
Y18	PA11/$\overline{\text{WRHL}}$/DQMUL/IRQ7/TXD4	Vcc
Y19	Vcc	—
Y20	PA13/$\overline{\text{WRL}}$/DQMLL/IRQ9/RXD5	Vcc

1.4 Pin Functions

Table 1.4 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	Vcc	I	Power supply	Power supply pins. All the Vcc pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	Vss	I	Ground	Ground pins. All the Vss pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	PVcc1	I	Power supply for I/O circuits (group 1)	Power supply pin for I/O pins (group 1).
	PVss1	I	Ground for I/O circuits (group 1)	Ground pin for I/O pins (group 1).
	PVcc2	I	Power supply for I/O circuits (group 2)	Power supply pin for I/O pins (group 2).
	PVss2	I	Ground for I/O circuits (group 2)	Ground pin for I/O pins (group 2).
	PLLVcc	I	Power supply for PLL	Power supply pin for the on-chip PLL oscillator.
	PLLVss	I	Ground for PLL	Ground pin for the on-chip PLL oscillator.
	VCL	I	Power supply for internal power step-down	For attaching an external capacitor for the internal stepped-down power supply. Connect each VCL pin to Vss via a 0.1 μF capacitor (placed close to the pin).
Clock	EXTAL	I	External clock	For connection to a crystal resonator. An external clock signal may also be input through the EXTAL pin.
	XTAL	O	Crystal	Connected to a crystal resonator.
	CK	O	System clock	Supplies the system clock to external devices.

Classification	Symbol	I/O	Name	Function
Clock	EXTAL32	I	External clock for 32kHz	Connected to a 32 kHz-crystal resonator. An external clock signal may also be input through this pin.
	XTAL32	O	Crystal for 32kHz	Connected to a 32 kHz-crystal resonator.
	CK32	O	System clock for 32kHz	Supplies the system clock signal at 32 kHz to external devices.
Operating mode control	MD1 MD0	I	Mode set	Sets the operating mode. Do not change values on these pins during operation.
	$\overline{\text{ASEMD0}}$	I	Debugging mode	Enables the E10A-USB emulator functions. Input a high level to operate the LSI in normal mode (not in debugging mode). To operate it in debugging mode, apply a low level to this pin on the user system board.
	$\overline{\text{FWE}}$	I	Flash memory write enable	Pin for flash memory. Flash memory can be protected against writing or erasure through this pin.
System control	$\overline{\text{RES}}$	I	Power-on reset	This LSI enters the power-on reset state when this signal goes low.
	$\overline{\text{MRES}}$	I	Manual reset	This LSI enters the manual reset state when this signal goes low.
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Outputs an overflow signal from the watchdog timer. If this pin needs to be pulled down, use the resistor larger than 1 MΩ to pull this pin down.
	$\overline{\text{BREQ}}$	I	Bus-mastership request	A low level is input to this pin when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	Bus-mastership request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus.

Classification	Symbol	I/O	Name	Function
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix it high when not in use.
	IRQ23 to IRQ0	I	Interrupt requests 23 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising edge, falling edge, or both edges can also be selected.
	$\overline{\text{IRQOUT}}$	O	Interrupt request output	Shows that an interrupt cause has occurred. The interrupt cause can be recognized even in the bus release state.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	Bidirectional data bus.
Bus control	$\overline{\text{CS7}}$ to $\overline{\text{CS0}}$	O	Chip select 7 to 0	Chip-select signals for external memory or devices.
	$\overline{\text{RD}}$	O	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	O	Read/write	Read/write signal.
	$\overline{\text{BS}}$	O	Bus start	Bus-cycle start signal.
	$\overline{\text{AH}}$	O	Address hold	Address hold timing signal for the device that uses the address/data-multiplexed bus.
	$\overline{\text{WAIT}}$	I	Wait	Inserts a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WRHH}}$	O	Write	Indicates a write access to bits 31 to 24 of data of external memory or device.
	$\overline{\text{WRHL}}$	O	Write	Indicates a write access to bits 23 to 16 of data of external memory or device.
	$\overline{\text{WRH}}$	O	Write	Indicates a write access to bits 15 to 8 of data of external memory or device
	$\overline{\text{WRL}}$	O	Write	Indicates a write access to bits 7 to 0 of data of external memory or device
	DQMUU	O	Byte select	Selects bits D31 to D24 when SDRAM is connected.
	DQMUL	O	Byte select	Selects bits D23 to D16 when SDRAM is connected.

Classification	Symbol	I/O	Name	Function
Bus control	DQMLU	O	Byte select	Selects bits D15 to D8 when SDRAM is connected.
	DQMLL	O	Byte select	Selects bits D7 to D0 when SDRAM is connected.
	$\overline{\text{RASU}}$, $\overline{\text{RASL}}$	O	RAS	Connected to the $\overline{\text{RAS}}$ pin when SDRAM is connected.
	$\overline{\text{CASU}}$, $\overline{\text{CASL}}$	O	CAS	Connected to the $\overline{\text{CAS}}$ pin when SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.
	$\overline{\text{REFOUT}}$	O	Refresh request output	Request signal output for refresh execution.
Direct memory access controller (DMAC)	DREQ0, DREQ1	I	DMA-transfer request	Input pin to receive external requests for DMA transfer.
	DACK0, DACK1	O	DMA-transfer request accept	Output pin for signals indicating acceptance of external requests from external devices.
	TEND0, TEND1	O	DMA-transfer end output	Output pin for DMA transfer end.
Multi-function timer pulse unit 2 (MTU2)	TCLKA, TCLKB, TCLKC, TCLKD	I	Timer clock input	External clock input pins for the timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	Input capture/ output compare (channel 0)	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A, TIOC1B	I/O	Input capture/ output compare (channel 1)	The TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A, TIOC2B	I/O	Input capture/ output compare (channel 2)	The TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	Input capture/ output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.

Classification	Symbol	I/O	Name	Function
Multi-function timer pulse unit 2 (MTU2)	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	Input capture/ output compare (channel 4)	The TGRA_4 to TGRD_4 input capture input/output compare output/PWM output pins.
	TIC5U, TIC5V, TIC5W	I	Input capture (channel 5)	The TGRU_5, TGRV_5, and TGRW_5 input capture input/dead time compensation input pins.
Multi-function timer pulse unit 2S (MTU2S)	TIOC3AS, TIOC3BS, TIOC3CS, TIOC3DS	I/O	Input capture/ output compare (channel 3)	The TGRA_3S to TGRD_3S input capture input/output compare output/PWM output pins.
	TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS	I/O	Input capture/ output compare (channel 4)	The TGRA_4S to TGRD_4S input capture input/output compare output/PWM output pins.
	TIC5US, TIC5VS, TIC5WS	I	Input capture (channel 5)	The TGRU_5S, TGRV_5S, and TGRW_5S input capture input/dead time compensation input pins.
Port output enable 2 (POE2)	$\overline{\text{POE8}}$ to $\overline{\text{POE0}}$	I	Port output control	Request signal input to place the MTU2 and MTU2S waveform output pins in the high impedance state.
Serial communication interface (SCI)	TXD3 to TXD0	O	Transmit data	Data output pins.
	RXD3 to RXD0	I	Receive data	Data input pins.
	SCK3 to SCK0	I/O	Serial clock	Clock input/output pins.
Serial communication interface with FIFO (SCIF)	TxD7 to TxD4	O	Transmit data	Data output pins.
	RxD7 to RxD4	I	Receive data	Data input pins.
	SCK7 to SCK4	I/O	Serial clock	Clock input/output pins.
I^2C bus interface 3 (IIC3)	SCL	I/O	Serial clock pin	Serial clock input/output pin.
	SDA	I/O	Serial data pin	Serial data input/output pin.
LVDS receiving interface (LVDS) for SH72315A	RXCLKINP	I	Receive clock	Input pin for the positive clock signal.
	RXCLKINM	I	Receive clock	Input pin for the negative clock signal.
	RXIN1P, RXIN0P	I	Receive data	Input pin for the positive data signal.
	RXIN1M, RXIN0M	I	Receive data	Input pin for the negative data signal.

Classification	Symbol	I/O	Name	Function
LVDS receiving interface (LVDS) for SH72315A	LVDSVcc	I	Power supply for LDVS	Power supply pin for the on-chip LVDS receiver. Connect this to the system supply (Vcc) when the LVDS is not to be used.
	LVDSVss	I	Ground for LDVS	Ground pin for the on-chip LVDS receiver. Connect each LVDSVss pin to the system power supply (Vss). If any pin is open-circuit, the module will not operate. Connect the pins to the system power supply (Vss) even if the LVDS receiver is not to be used.
Renesas serial peripheral interface (RSPI)	MOSI0	I/O	Transmit data	Data transmission pin for RSPI master operation.
	MISO0	I/O	Receive data	Data reception pin for RSPI slave operation.
	RSPCK0	I/O	Serial clock	Clock input/output pin.
	SSL0	I/O	Slave select 0	Slave selection pin 0 for the RSPI.
	SSL3 to SSL1	O	Slave select 3 to 1	Slave selection pins 3 to 1 for the RSPI.
Controller area network (RCAN-ET)	CTx0	O	CAN bus transmit data	Output pin for transmit data on the CAN bus.
	CRx0	I	CAN bus receive data	Output pin for receive data on the CAN bus.
A/D converter (ADC)	AN15 to A0	I	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$	I	A/D conversion trigger input	External trigger input pin for starting A/D conversion.
	AVcc	I	Analog power supply	Power supply pin for the A/D converter. Connect all AVcc pins to the system power supply (Vcc) when the A/D converter is not used.
	AVref	I	Analog reference power supply	Reference voltage pin for the A/D converter. Connect all the AVref pins to the system power supply when the A/D converter.
	AVss	I	Analog ground	Ground pin for the A/D converter. Connect this pin to the system power supply (V_{SS}) when the A/D converter is not used.

Classification	Symbol	I/O	Name	Function
I/O ports	PA19 to PA0	I/O	General port	20-bit general input/output port pins.
	PB13 to PB0	I/O	General port	14-bit general input/output port pins.
	PC15 to PC0	I/O	General port	16-bit general input/output port pins.
	PD31 to PD0	I/O	General port	32-bit general input/output port pins.
	PE 23 to PE0	I/O	General port	24-bit general input/output port pins.
	PF15 to PF0	I	General port	16-bit general input port pins.
	PG15 to PG0	I/O	General port	16-bit general input/output port pins.
	PH15 to PH0	I/O	General port	16-bit general input/output port pins.
	PJ15 to PJ0	I/O	General port	16-bit general input/output port pins.
	PK7 to PK0	I/O	General port	8-bit general input/output port pins.
	PL5 to PL0	I	General port	6-bit general input port pins.
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Test-mode select signal input pin.
	TDI	I	Test data input	Serial input pin for instructions and data.
	TDO	O	Test data output	Serial output pin for instructions and data.
	$\overline{\text{TRST}}$	I	Test reset	Initialization-signal input pin. Input a low level when not using the H-UDI.
Emulator interface	AUDATA3 to AUDATA0	O	AUD Data	Branch source or destination address output pins.
	AUDCK	O	AUD clock	Sync-clock output pin.
	$\overline{\text{AUDSYNC}}$	O	AUD sync signal	Data start-position acknowledge-signal output pin.
Emulator interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A-USB emulator has entered its break mode.
	$\overline{\text{ASEBRK}}$	I	Break request	E10A-USB emulator break input pin.
User break Controller (UBC)	$\overline{\text{UBCTRG}}$	O	User break trigger output	Trigger output pin for UBC condition match.
Compare match timerb 2 (CMT2)	TIC1, TIC0	I	Input capture	Input capture pins.
	TOC1, ROC0	O	Output capture	Output-compare output pins.

Classification	Symbol	I/O	Name	Function
Key scan controller (KEYC)	KEY31 to KEY0	I	Key input	Key input pins.
	$\overline{\text{COM7}}$ to $\overline{\text{COM0}}$	O	Key matrix output	Key matrix output pins.
	P7 to P0	I	Key matrix input	Key matrix input pins.
32kHz timer (TIM32C)	TI32I1A, TI32I0A, TI32I1B, TI32I0B	I	Phase counting input	Phase-counting input pins.

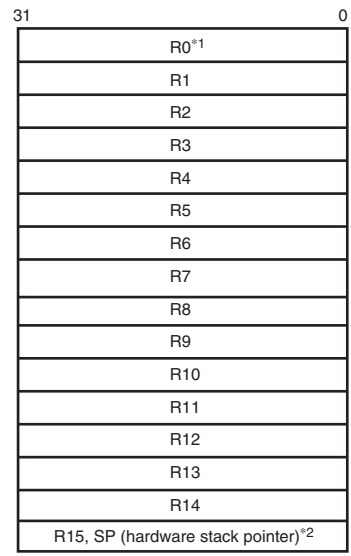
Section 2 CPU

2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, four 32-bit control registers, and four 32-bit system registers.

2.1.1 General Registers

Figure 2.1 shows the general registers. The sixteen 32-bit general registers are numbered R0 to R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and restoring the status register (SR) and program counter (PC) in exception handling is accomplished by referencing the stack using R15.



Notes: 1. R0 functions as an index register in the indexed register indirect addressing mode and indexed GBR indirect addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

2.1.2 Control Registers

The control registers consist of four 32-bit registers: the status register (SR), the global base register (GBR), the vector base register (VBR), and the jump table base register (TBR).

The status register indicates instruction processing states.

The global base register functions as a base address for the GBR indirect addressing mode to transfer data to the registers of on-chip peripheral modules.

The vector base register functions as the base address of the exception handling vector area (including interrupts).

The jump table base register functions as the base address of the function table area.

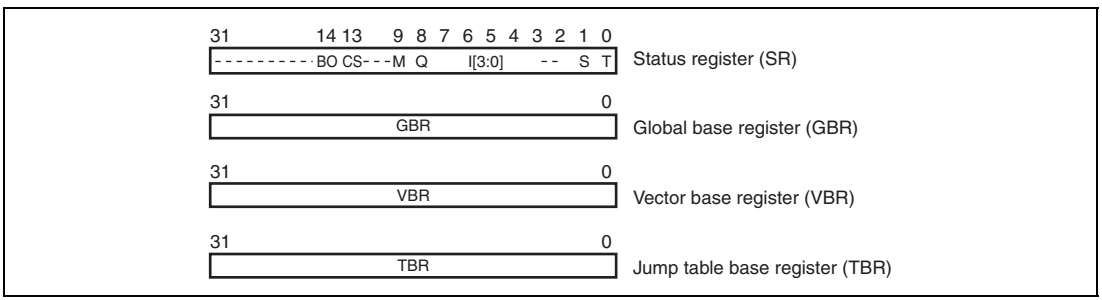


Figure 2.2 Control Registers

(1) Status Register (SR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-	S	T
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W:	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	BO	0	R/W	BO Bit Indicates that a register bank has overflowed.
13	CS	0	R/W	CS Bit Indicates that, in CLIP instruction execution, the value has exceeded the saturation upper-limit value or fallen below the saturation lower-limit value.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	I[3:0]	1111	R/W	Interrupt Mask Level
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	S Bit Specifies a saturation operation for a MAC instruction.
0	T	—	R/W	T Bit True/false condition or carry/borrow bit

(2) Global Base Register (GBR)

GBR is referenced as the base address in a GBR-referencing MOV instruction.

(3) Vector Base Register (VBR)

VBR is referenced as the branch destination base address in the event of an exception or an interrupt.

(4) Jump Table Base Register (TBR)

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

2.1.3 System Registers

The system registers consist of four 32-bit registers: the high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). MACH and MACL store the results of multiply or multiply and accumulate operations. PR stores the return address from a subroutine procedure. PC points four bytes ahead of the current instruction and controls the flow of the processing.

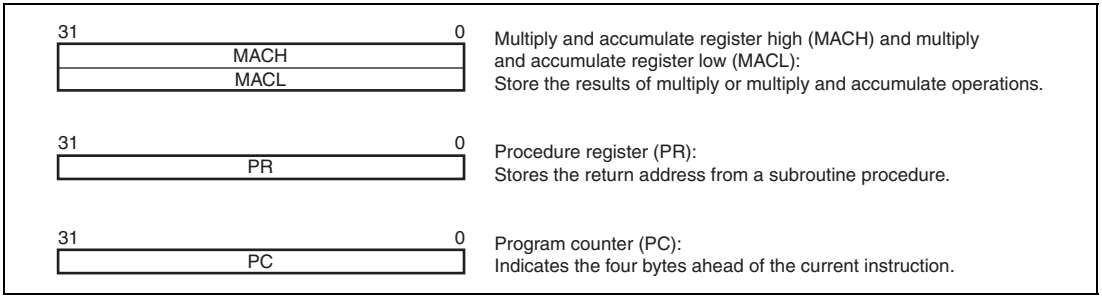


Figure 2.3 System Registers

(1) Multiply and Accumulate Register High (MACH) and Multiply and Accumulate Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the result of a MAC or MUL instruction.

(2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, and is referenced by a subroutine return instruction (RTS).

(3) Program Counter (PC)

PC points four bytes ahead of the instruction being executed.

2.1.4 Register Banks

For the nineteen 32-bit registers comprising general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR, high-speed register saving and restoration can be carried out using a register bank. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses a register bank. Restoration from the bank is executed by issuing a RESBANK instruction in an interrupt processing routine.

This LSI has 15 banks. For details, see the SH-2A, SH2A-FPU Software Manual and section 7.8, Register Banks.

2.1.5 Initial Values of Registers

Table 2.1 lists the values of the registers after a reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I[3:0] are 1111 (H'F), BO and CS are 0, reserved bits are 0, and other bits are undefined
	GBR, TBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword by expanding the sign-part when loaded into a register.

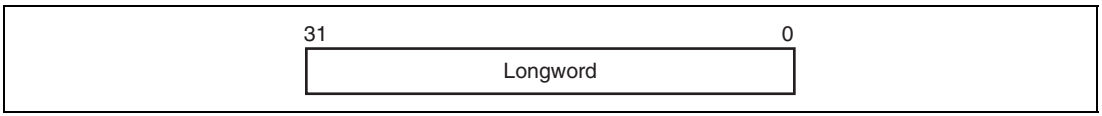


Figure 2.4 Data Format in Registers

2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.5.

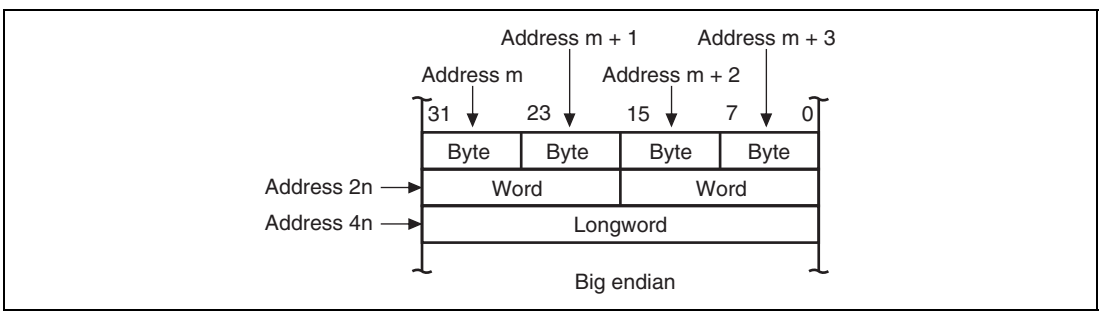


Figure 2.5 Data Formats in Memory

2.2.3 Immediate Data Format

Byte (8-bit) immediate data is located in an instruction code.

Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

20-bit immediate data is located in the code of a MOVI20 or MOVI20S 32-bit transfer instruction. The MOVI20 instruction stores immediate data in the destination register in sign-extended form. The MOVI20S instruction shifts immediate data by eight bits in the upper direction, and stores it in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is stored in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

See examples given in section 2.3.1 (10), Immediate Data.

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

Instructions are RISC type. This section details their functions.

(1) 16-Bit Fixed-Length Instructions

Basic instructions have a fixed length of 16 bits, improving program code efficiency.

(2) 32-Bit Fixed-Length Instructions

The SH-2A additionally features 32-bit fixed-length instructions, improving performance and ease of use.

(3) One Instruction per State

Each basic instruction can be executed in one cycle using the pipeline system.

(4) Data Length

Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data in memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It is also handled as longword data.

Table 2.2 Sign Extension of Word Data

SH-2A CPU		Description	Example of Other CPU
MOV.W	@(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W #H'1234,R0
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

Note: @(disp, PC) accesses the immediate data.

(5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

(6) Delayed Branch Instructions

With the exception of some instructions, unconditional branch instructions, etc., are executed as delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction immediately following the delayed branch instruction. This reduces disturbance of the pipeline control when a branch is taken.

In a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution such as register updating excluding the actual branch operation, is performed in the order of delayed branch instruction → delay slot instruction. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

Table 2.3 Delayed Branch Instructions

SH-2A CPU		Description	Example of Other CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

(7) Unconditional Branch Instructions with No Delay Slot

The SH-2A additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduces the code size.

(8) Multiply/Multiply-and-Accumulate Operations

16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

(9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

Table 2.4 T Bit

SH-2A CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1.	CMP.W	R1,R0
BT	TRGET0	The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	T bit is set when R0 = 0.	BEQ	TRGET
BT	TRGET	The program branches if R0 = 0.		

(10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data is not located in instruction codes but in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

With the SH-2A, 17- to 28-bit immediate data can be located in an instruction code. However, for 21- to 28-bit immediate data, an OR instruction must be executed after the data is transferred to a register.

Table 2.5 Immediate Data Accessing

Classification	SH-2A CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOVI20	#H'1234,R0	MOV.W	#H'1234,R0
20-bit immediate	MOVI20	#H'12345,R0	MOV.L	#H'12345,R0
28-bit immediate	MOVI20S	#H'12345,R0	MOV.L	#H'1234567,R0
	OR	#H'67,R0		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
		.....		
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

(11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.6 Absolute Address Accessing

Classification	SH-2A CPU		Example of Other CPU	
Up to 20 bits	MOVI20	#H'12345,R1	MOV.B	@H'12345,R0
	MOV.B	@R1,R0		
21 to 28 bits	MOVI20S	#H'12345,R1	MOV.B	@H'1234567,R0
	OR	#H'67,R1		
	MOV.B	@R1,R0		
29 bits or more	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,R0
	MOV.B	@R1,R0		
		.....		
	.DATA.L	H'12345678		

(12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

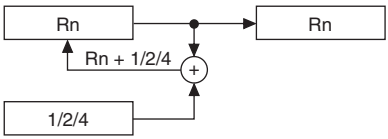
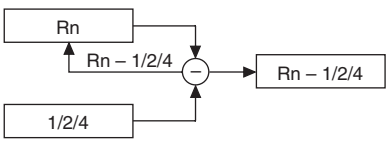
Table 2.7 Displacement Accessing

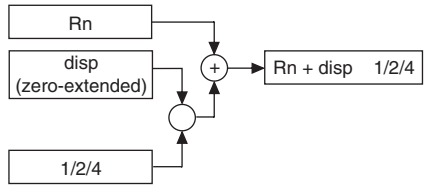
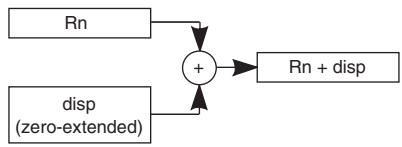
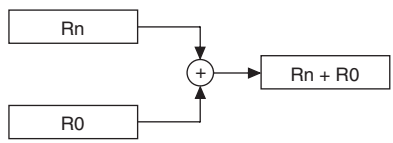
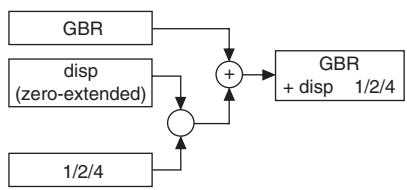
Classification	SH-2A CPU		Example of Other CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

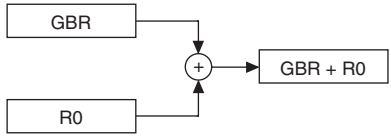
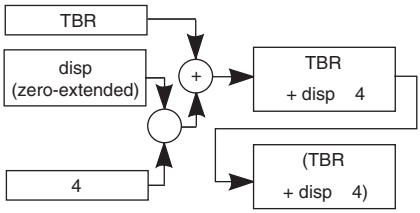
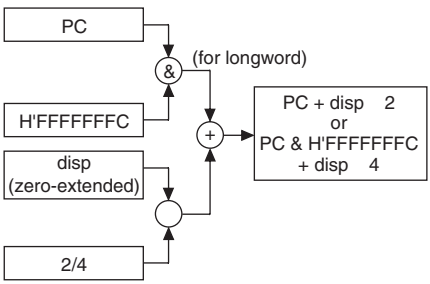
2.3.2 Addressing Modes

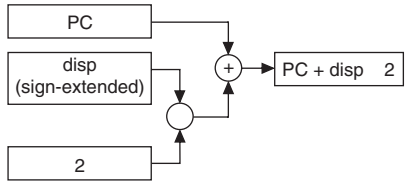
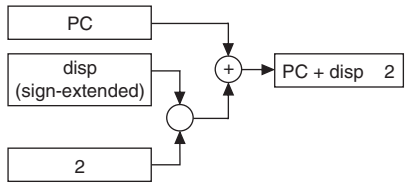
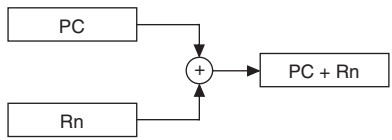
Addressing modes and effective address calculation are as follows:

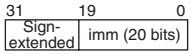
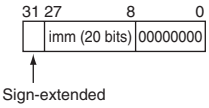
Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register direct	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Register indirect	@Rn	The effective address is the contents of register Rn. 	Rn
Register indirect with post-increment	@Rn+	The effective address is the contents of register Rn. A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After instruction execution) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Register indirect with pre-decrement	@-Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction is executed with Rn after this calculation)

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register indirect with displacement	@(disp:4, Rn)	The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Register indirect with displacement	@(disp:12, Rn)	The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended. 	Byte: Rn + disp Word: Rn + disp Longword: Rn + disp
Indexed register indirect	@(R0,Rn)	The effective address is the sum of Rn and R0. 	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Indexed GBR indirect	@(R0, GBR)	The effective address is the sum of GBR value and R0. 	GBR + R0
TBR duplicate indirect with displacement	@@ (disp:8, TBR)	The effective address is the sum of TBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and is multiplied by 4. 	Contents of address (TBR + disp × 4)
PC relative with displacement	@(disp:8, PC)	The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative	disp:8	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp). 	PC + disp × 2
	disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp). 	PC + disp × 2
	Rn	The effective address is the sum of PC value and Rn. 	PC + Rn

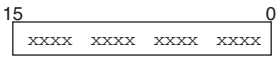
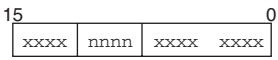
Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Immediate	#imm:20	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended. 	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, and the lower bits are padded with zero. 	—
	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—
	#imm:3	The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.	—

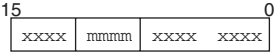
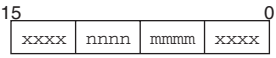
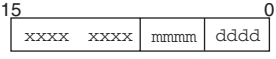
2.3.3 Instruction Format

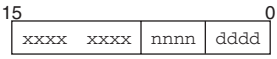
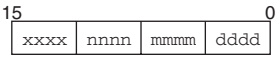
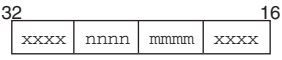
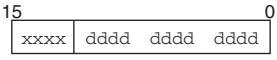
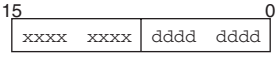
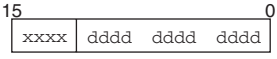
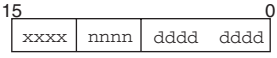
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

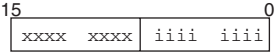
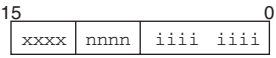
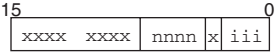
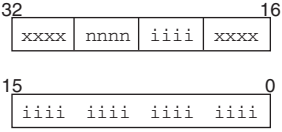
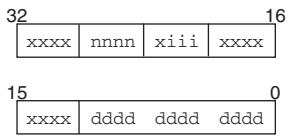
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

Table 2.9 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	`NOP`
n format 	—	nnnn: Register direct	`MOVT Rn`
	Control register or system register	nnnn: Register direct	`STS MACH,Rn`
	R0 (Register direct)	nnnn: Register direct	`DIVU R0,Rn`
	Control register or system register	nnnn: Register indirect with pre-decrement	`STC.L SR,@-Rn`
	mmmm: Register direct	R15 (Register indirect with pre-decrement)	`MOVMU.L Rm,@-R15`
	R15 (Register indirect with post-increment)	nnnn: Register direct	`MOVMU.L @R15+,Rn`
	R0 (Register direct)	nnnn: (Register indirect with post-increment)	`MOV.L R0,@Rn+`

Instruction Formats	Source Operand	Destination Operand	Example
m format 	mmmm: Register direct	Control register or system register	LDC Rm,SR
	mmmm: Register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: Register indirect	—	JMP @Rm
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	MOV.L @-Rm,R0
	mmmm: PC relative using Rm	—	BRAF Rm
nm format 	mmmm: Register direct	nnnn: Register direct	ADD Rm,Rn
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,@Rn
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @Rm+,Rn
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm,@-Rn
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm),R0

Instruction Formats	Source Operand	Destination Operand	Example
nd4 format 	R0 (Register direct)	nnnndddd: Register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd format 	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp,Rm),Rn
nmd12 format  	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp12,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp12,Rm),Rn
d format 	dddddddd: GBR indirect with displacement	R0 (Register direct)	MOV.L @(disp,GBR),R0
	R0 (Register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
	dddddddd: PC relative with displacement	R0 (Register direct)	MOVA @(disp,PC),R0
	dddddddd: TBR duplicate indirect with displacement	—	JSR/N @@(disp8,TBR)
	dddddddd: PC relative	—	BF label
d12 format 	ddddddddddddd: PC relative	—	BRA label (label = disp + PC)
nd8 format 	dddddddd: PC relative with displacement	nnnn: Register direct	MOV.L @(disp,PC),Rn

Instruction Formats	Source Operand	Destination Operand	Example
i format 	iiiiiiii: Immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: Immediate	R0 (Register direct)	AND #imm,R0
	iiiiiiii: Immediate	—	TRAPA #imm
ni format 	iiiiiiii: Immediate	nnnn: Register direct	ADD #imm,Rn
ni3 format 	nnnn: Register direct iii: Immediate	—	BLD #imm3,Rn
	—	nnnn: Register direct iii: Immediate	BST #imm3,Rn
ni20 format 	iiiiiiiiiiiiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format 	nnnnddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnnddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12,Rn)

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

2.4 Instruction Set

2.4.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their classification.

Table 2.10 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	13	MOV	Data transfer	62
			Immediate data transfer	
			Peripheral module data transfer	
			Structure data transfer	
			Reverse stack transfer	
		MOVA	Effective address transfer	
		MOVI20	20-bit immediate data transfer	
		MOVI20S	20-bit immediate data transfer	
			8-bit left-shit	
		MOVML	R0–Rn register save/restore	
		MOVMU	Rn–R14 and PR register save/restore	
		MOVRT	T bit inversion and transfer to Rn	
		MOVT	T bit transfer	
		MOVU	Unsigned data transfer	
		NOTT	T bit inversion	
		PREF	Prefetch to operand cache	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations	26	ADD	Binary addition	40
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		CLIPS	Signed saturation value comparison	
		CLIPU	Unsigned saturation value comparison	
		DIVS	Signed division (32 ÷ 32)	
		DIVU	Unsigned division (32 ÷ 32)	
		DIV1	One-step division	
		DIV0S	Initialization of signed one-step division	
		DIV0U	Initialization of unsigned one-step division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate operation	
		MUL	Double-precision multiply operation	
		MULR	Signed multiplication with result storage in Rn	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	10	BF	Conditional branch, conditional delayed branch (branch when T = 0)	15
		BT	Conditional branch, conditional delayed branch (branch when T = 1)	
		BRA	Unconditional delayed branch	
		BRAF	Unconditional delayed branch	
		BSR	Delayed branch to subroutine procedure	
		BSRF	Delayed branch to subroutine procedure	
		JMP	Unconditional delayed branch	
		JSR	Branch to subroutine procedure	
			Delayed branch to subroutine procedure	
		RTS	Return from subroutine procedure	
			Delayed return from subroutine procedure	
		RTV/N	Return from subroutine procedure with Rm → R0 transfer	

Classification	Types	Operation Code	Function	No. of Instructions
System control	14	CLRT	T bit clear	36
		CLRMAC	MAC register clear	
		LDBANK	Register restoration from specified register bank entry	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RESBANK	Register restoration from register bank	
		RTE	Return from exception handling	
		SETT	T bit set	
		SLEEP	Transition to low power consumption mode	
		STBANK	Register save to specified register bank entry	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Floating-point instructions	19	FABS	Floating-point absolute value	48
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FCNVDS	Conversion from double-precision to single-precision	
		FCNVSD	Conversion from single-precision to double-precision	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Conversion from integer to floating-point	
		FMAC	Floating-point multiply and accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	

Classification	Types	Operation Code	Function	No. of Instructions
Floating-point instructions	19	FSCHG	SZ bit inversion	48
		FSQRT	Floating-point square root	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	112			253

The table below shows the format of instruction codes, operation, and execution states. They are described by using this format according to their classification.

Instruction	Instruction Code	Operation	Execution States	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait states are inserted.*1	Value of T bit after instruction is executed.
[Legend] Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	[Legend] mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ····· 1111: R15 iiii: Immediate data dddd: Displacement	[Legend] →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift		[Legend] —: No change

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states will be increased in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.4.2 Data Transfer Instructions

Table 2.11 Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOV	#imm,Rn	1110nnnniiiiiiii	imm → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(disp,PC),Rn	1001nnnnddddddd	(disp × 2 + PC) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(disp,PC),Rn	1101nnnnddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Yes	Yes
MOV	Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Yes	Yes
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Yes	Yes
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Yes	Yes
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Yes	Yes
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	Yes	Yes
MOV.B	R0,@Rn+	0100nnnn10001011	R0 → (Rn), Rn + 1 → Rn	1	—			Yes
MOV.W	R0,@Rn+	0100nnnn10011011	R0 → (Rn), Rn + 2 → Rn	1	—			Yes
MOV.L	R0,@Rn+	0100nnnn10101011	R0 → Rn), Rn + 4 → Rn	1	—			Yes
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4 → Rm, (Rm) → R0	1	—			Yes
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm → (disp + Rn)	1	—			Yes
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm → (disp × 2 + Rn)	1	—			Yes
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm → (disp × 4 + Rn)	1	—			Yes
MOV.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp + Rm) → sign extension → Rn	1	—			Yes
MOV.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 0101dddddddddddd	(disp × 2 + Rm) → sign extension → Rn	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOV.L	@(disp12,Rm),Rn	0011nnnnmmmm0001 0110dddddddddddd	(disp × 4 + Rm) → Rn	1	—			Yes
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—	Yes	Yes	Yes
MOVI20	#imm20,Rn	0000nnnniiii0000 iiiiiiiiiiiiiiii	imm → sign extension → Rn	1	—			Yes
MOVI20S	#imm20,Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm << 8 → sign extension → Rn	1	—			Yes
MOVML.L	Rm,@-R15	0100mmmm11110001	R15-4 → R15, Rm → (R15) R15-4 → R15, Rm-1 → (R15) : R15-4 → R15, R0 → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVML.L	@R15+,Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rn as PR	1 to 16	—			Yes
MOVMU.L	Rm,@-R15	0100mmmm11110000	R15-4 → R15, PR → (R15) R15-4 → R15, R14 → (R15) : R15-4 → R15, Rm → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVMU.L	@R15+,Rn	0100nnnn11110100	(R15) → Rn, R15 + 4 → R15 (R15) → Rn + 1, R15 + 4 → R15 : (R15) → R14, R15 + 4 → R15 (R15) → PR Note: When Rn = R15, read Rn as PR	1 to 16	—			Yes
MOVRT	Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT	Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOVU.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 1000ddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 1001ddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes
NOTT		0000000001101000	~T → T	1	Operation result			Yes
PREF	@Rn	0000nnnn10000011	(Rn) → operand cache	1	—		Yes	Yes
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn Swap upper and lower two bytes the	1	—	Yes	Yes	Yes
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	Yes	Yes
XTRCT	Rm,Rn	0010nnnnmmmm1101	Middle 32 bits of Rm:Rn → Rn	1	—	Yes	Yes	Yes

2.4.3 Arithmetic Operation Instructions

Table 2.12 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—	Yes	Yes	Yes
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—	Yes	Yes	Yes
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, carry → T	1	Carry	Yes	Yes	Yes
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, overflow → T	1	Overflow	Yes	Yes	Yes
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
CLIPS.B	Rn	0100nnnn10010001	When Rn > (H'0000007F), (H'0000007F) → Rn, 1 → CS when Rn < (H'FFFFFF80), (H'FFFFFF80) → Rn, 1 → CS	1	—			Yes
CLIPS.W	Rn	0100nnnn10010101	When Rn > (H'00007FFF), (H'00007FFF) → Rn, 1 → CS When Rn < (H'FFFF8000), (H'FFFF8000) → Rn, 1 → CS	1	—			Yes
CLIPU.B	Rn	0100nnnn10000001	When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS	1	—			Yes
CLIPU.W	Rn	0100nnnn10000101	When Rn > (H'0000FFFF), (H'0000FFFF) → Rn, 1 → CS	1	—			Yes
DIV1	Rm,Rn	0011nnnnmmmm0100	1-step division (Rn ÷ Rm)	1	Calcu-lation result	Yes	Yes	Yes
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calcu-lation result	Yes	Yes	Yes
DIV0U		0000000000011001	0 → M/Q/T	1	0	Yes	Yes	Yes
DIVS	R0,Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—			Yes
DIVU	R0,Rn	0100nnnn10000100	Unsigned operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	34	—			Yes
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DT	Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Compa-rison result	Yes	Yes	Yes
EXTS.B	Rm,Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
EXTU.W	Rm,Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	4	—	Yes	Yes	Yes
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3	—	Yes	Yes	Yes
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	Yes	Yes
MULR	R0,Rn	0100nnnn10000000	R0 × Rn → Rn 32 × 32 → 32 bits	2				Yes
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
NEG	Rm,Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Yes	Yes
NEGC	Rm,Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUB	Rm,Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Yes	Yes
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Over-flow	Yes	Yes	Yes

2.4.4 Logic Operation Instructions

Table 2.13 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	Yes
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—	Yes	Yes	Yes
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—	Yes	Yes	Yes
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Yes	Yes
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Yes	Yes
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Yes	Yes
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Yes	Yes
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Yes	Yes
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST	#imm,R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm When the result is 0, 1 → T Otherwise, 0 → T	3	Test result	Yes	Yes	Yes
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	Yes
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Yes	Yes
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Yes	Yes

2.4.5 Shift Instructions

Table 2.14 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB	Yes	Yes	Yes
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB	Yes	Yes	Yes
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB	Yes	Yes	Yes
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Yes	Yes
SHAD	Rm,Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Yes	Yes
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB	Yes	Yes	Yes
SHLD	Rm,Rn	0100nnnnmmmm1101	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Yes	Yes
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB	Yes	Yes	Yes
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—	Yes	Yes	Yes
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—	Yes	Yes	Yes
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—	Yes	Yes	Yes
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—	Yes	Yes	Yes
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—	Yes	Yes	Yes
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—	Yes	Yes	Yes

2.4.6 Branch Instructions

Table 2.15 Branch Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BF	label	10001011dddddddd	When T = 0, disp × 2 + PC → PC, When T = 1, nop	3/1*	—	Yes	Yes	Yes
BF/S	label	10001111dddddddd	Delayed branch When T = 0, disp × 2 + PC → PC, When T = 1, nop	2/1*	—	Yes	Yes	Yes
BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Yes	Yes
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Yes	Yes
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Yes	Yes
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Yes	Yes
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Yes	Yes
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	Yes	Yes
JSR/N	@Rm	0100mmmm01001011	PC-2 → PR, Rm → PC	3	—			Yes
JSR/N	@@(disp8,TBR)	10000011dddddddd	PC-2 → PR, (disp × 4 + TBR) → PC	5	—			Yes
RTS		0000000000001011	Delayed branch, PR → PC	2	—	Yes	Yes	Yes
RTS/N		0000000001101011	PR → PC	3	—			Yes
RTV/N	Rm	0000mmmm01111011	Rm → R0, PR → PC	3	—			Yes

Note: * One cycle when the program does not branch.

2.4.7 System Control Instructions

Table 2.16 System Control Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
CLRT		0000000000001000	0 → T	1	0	Yes	Yes	Yes
CLRMAC		0000000000101000	0 → MACH,MACL	1	—	Yes	Yes	Yes
LDBANK	@Rm,R0	0100mmmm11100101	(Specified register bank entry) → R0	6	—			Yes
LDC	Rm,SR	0100mmmm00001110	Rm → SR	3	LSB	Yes	Yes	Yes
LDC	Rm,TBR	0100mmmm01001010	Rm → TBR	1	—			Yes
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—	Yes	Yes	Yes
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—	Yes	Yes	Yes
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Yes	Yes
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—	Yes	Yes	Yes
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—	Yes	Yes	Yes
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
NOP		0000000000001001	No operation	1	—	Yes	Yes	Yes
RESBANK		0000000001011011	Bank → R0 to R14, GBR, MACH, MACL, PR	9*	—			Yes
RTE		0000000000101011	Delayed branch, stack area → PC/SR	6	—	Yes	Yes	Yes
SETT		0000000000011000	1 → T	1	1	Yes	Yes	Yes
SLEEP		0000000000011011	Sleep	5	—	Yes	Yes	Yes
STBANK	R0,@Rn	0100nnnn11100001	R0 → (specified register bank entry)	7	—			Yes
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Yes	Yes
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—	Yes	Yes	Yes
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—	Yes	Yes	Yes
STC.L	SR,@-Rn	0100nnnn00000011	Rn-4 → Rn, SR → (Rn)	2	—	Yes	Yes	Yes
STC.L	GBR,@-Rn	0100nnnn00010011	Rn-4 → Rn, GBR → (Rn)	1	—	Yes	Yes	Yes
STC.L	VBR,@-Rn	0100nnnn00100011	Rn-4 → Rn, VBR → (Rn)	1	—	Yes	Yes	Yes
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—	Yes	Yes	Yes
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—	Yes	Yes	Yes
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—	Yes	Yes	Yes
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Yes	Yes
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Yes	Yes
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Yes	Yes
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Yes	Yes

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

* In the event of bank overflow, the number of cycles is 19.

2.4.8 Floating-Point Operation Instructions

Table 2.17 Floating-Point Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn| → FRn	1	—	Yes	Yes	Yes
FABS	DRn	1111nnn001011101	|DRn| → DRn	1	—		Yes	Yes
FADD	FRm, FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—	Yes	Yes	Yes
FADD	DRm, DRn	1111nnn0mmm00000	DRn + DRm → DRn	6	—		Yes	Yes
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result	Yes	Yes	Yes
FCMP/EQ	DRm, DRn	1111nnn0mmm00100	(DRn = DRm)? 1:0 → T	2	Comparison result		Yes	Yes
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Comparison result	Yes	Yes	Yes
FCMP/GT	DRm, DRn	1111nnn0mmm00101	(DRn > DRm)? 1:0 → T	2	Comparison result		Yes	Yes
FCNVDS	DRm, FPUL	1111mmm010111101	(float) DRm → FPUL	2	—		Yes	Yes
FCNVSD	FPUL, DRn	1111nnn010101101	(double) FPUL → DRn	2	—		Yes	Yes
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm → FRn	10	—	Yes	Yes	Yes
FDIV	DRm, DRn	1111nnn0mmm00011	DRn/DRm → DRn	23	—		Yes	Yes
FLDI0	FRn	1111nnnn10001101	0 × 00000000 → FRn	1	—	Yes	Yes	Yes
FLDI1	FRn	1111nnnn10011101	0 × 3F800000 → FRn	1	—	Yes	Yes	Yes
FLDS	FRm, FPUL	1111mmmm00011101	FRm → FPUL	1	—	Yes	Yes	Yes
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL → FRn	1	—	Yes	Yes	Yes
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL → DRn	2	—		Yes	Yes
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0 × FRm+FRn → FRn	1	—	Yes	Yes	Yes
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—	Yes	Yes	Yes
FMOV	DRm, DRn	1111nnn0mmm01100	DRm → DRn	2	—		Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FMOV.S	@(R0, Rm), FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—	Yes	Yes	Yes
FMOV.D	@(R0, Rm), DRn	1111nnn0mmmm0110	(R0 + Rm) → DRn	2	—		Yes	Yes
FMOV.S	@Rm+, FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+=4	1	—	Yes	Yes	Yes
FMOV.D	@Rm+, DRn	1111nnn0mmmm1001	(Rm) → DRn, Rm += 8	2	—		Yes	Yes
FMOV.S	@Rm, FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—	Yes	Yes	Yes
FMOV.D	@Rm, DRn	1111nnn0mmmm1000	(Rm) → DRn	2	—		Yes	Yes
FMOV.S	@(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp × 4 + Rm) → FRn	1	—			Yes
FMOV.D	@(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp × 8 + Rm) → DRn	2	—			Yes
FMOV.S	FRm, @(R0,Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @(R0,Rn)	1111nnnnmmm00111	DRm → (R0 + Rn)	2	—		Yes	Yes
FMOV.S	FRm, @-Rn	1111nnnnmmmm1011	Rn-=4, FRm → (Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @-Rn	1111nnnnmmm01011	Rn-=8, DRm → (Rn)	2	—		Yes	Yes
FMOV.S	FRm, @Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @Rn	1111nnnnmmm01010	DRm → (Rn)	2	—		Yes	Yes
FMOV.S	FRm, @(disp12,Rn)	0011nnnnmmmm0001 0011dddddddddddd	FRm → (disp × 4 + Rn)	1	—			Yes
FMOV.D	DRm, @(disp12,Rn)	0011nnnnmmm00001 0011dddddddddddd	DRm → (disp × 8 + Rn)	2	—			Yes
FMUL	FRm, FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—	Yes	Yes	Yes
FMUL	DRm, DRn	1111nnn0mmm00010	DRn × DRm → DRn	6	—		Yes	Yes
FNEG	FRn	1111nnnn01001101	-FRn → FRn	1	—	Yes	Yes	Yes
FNEG	DRn	1111nnn001001101	-DRn → DRn	1	—		Yes	Yes
FSCHG		1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		Yes	Yes
FSQRT	FRn	1111nnnn01101101	√FRn → FRn	9	—		Yes	Yes
FSQRT	DRn	1111nnn001101101	√DRn → DRn	22	—		Yes	Yes
FSTS	FPUL,FRn	1111nnnn00001101	FPUL → FRn	1	—	Yes	Yes	Yes
FSUB	FRm, FRn	1111nnnnmmmm0001	FRn-FRm → FRn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FSUB	DRm, DRn	1111nnn0mmm00001	DRn-DRm → DRn	6	—		Yes	Yes
FTRC	FRm, FPUL	1111mmmm00111101	(long)FRm → FPUL	1	—	Yes	Yes	Yes
FTRC	DRm, FPUL	1111mmm000111101	(long)DRm → FPUL	2	—		Yes	Yes

2.4.9 FPU-Related CPU Instructions

Table 2.18 FPU-Related CPU Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
LDS	Rm,FPSCR	0100mmmm01101010	Rm → FPSCR	1	—	Yes	Yes	Yes
LDS	Rm,FPUL	0100mmmm01011010	Rm → FPUL	1	—	Yes	Yes	Yes
LDS.L	@Rm+, FPSCR	0100mmmm01100110	(Rm) → FPSCR, Rm+=4	1	—	Yes	Yes	Yes
LDS.L	@Rm+, FPUL	0100mmmm01010110	(Rm) → FPUL, Rm+=4	1	—	Yes	Yes	Yes
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—	Yes	Yes	Yes
STS	FPUL,Rn	0000nnnn01011010	FPUL → Rn	1	—	Yes	Yes	Yes
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn-=4, FPCSR → (Rn)	1	—	Yes	Yes	Yes
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn-=4, FPUL → (Rn)	1	—	Yes	Yes	Yes

2.4.10 Bit Manipulation Instructions

Table 2.19 Bit Manipulation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp + Rn)) & T →	3	Operation result			Yes
BANDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of (disp + Rn)) & T → T	3	Operation result			Yes
BCLR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0000dddddddddddd	0 → (imm of (disp + Rn))	3	—			Yes
BCLR	#imm3,Rn	10000110nnnn0iii	0 → imm of Rn	1	—			Yes
BLD.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0011dddddddddddd	(imm of (disp + Rn)) →	3	Operation result			Yes
BLD	#imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result			Yes
BLDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result			Yes
BOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BORNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BSET.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1 → (imm of (disp + Rn))	3	—			Yes
BSET	#imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—			Yes
BST.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T → (imm of (disp + Rn))	3	—			Yes
BST	#imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—			Yes
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

2.5 Processing States

The CPU has five processing states: reset, exception handling, bus-released, program execution, and low power consumption. Figure 2.6 shows the transitions between the states.

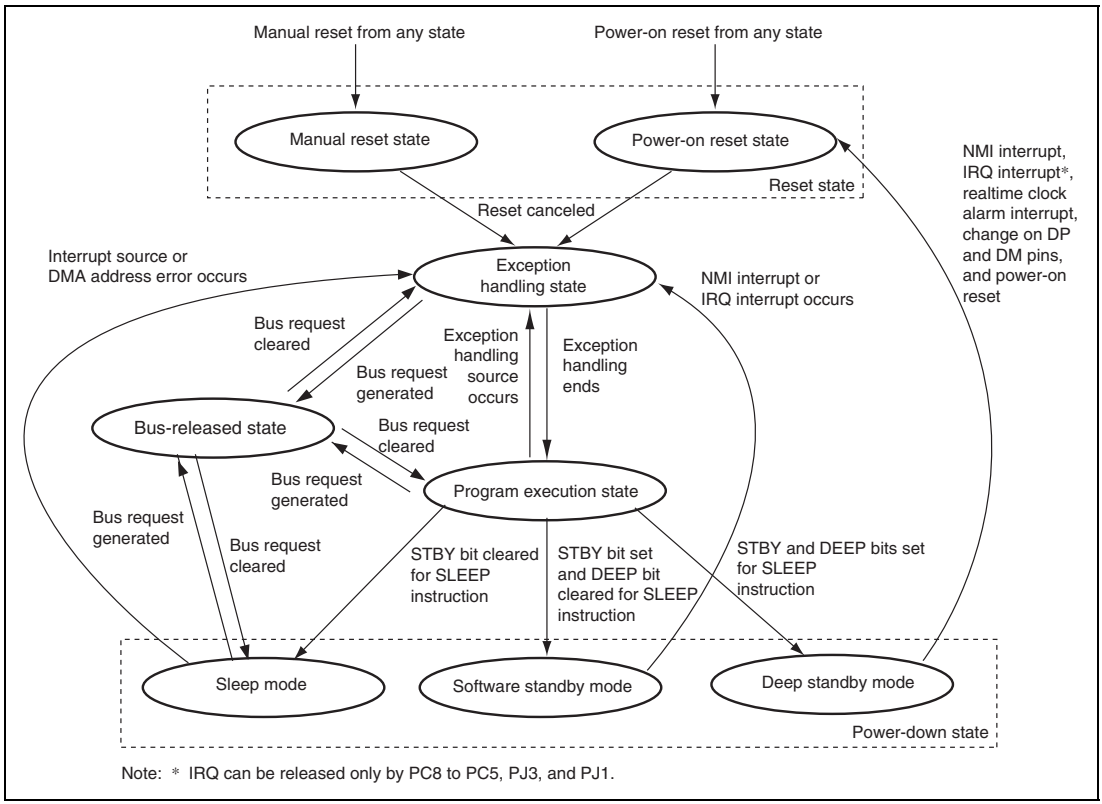


Figure 2.6 Transitions between Processing States

(1) Reset State

In the reset state, the CPU is reset. There are two kinds of reset, power-on reset and manual reset.

(2) Exception Handling State

The exception handling state is a transient state that occurs when exception handling sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception handling vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception handling vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

(3) Program Execution State

In the program execution state, the CPU sequentially executes the program.

(4) Low Power Consumption State

In the low power consumption state, the CPU stops operating to reduce power consumption. The SLEEP instruction places the CPU in sleep mode, software standby mode, or deep software standby mode.

(5) Bus-Released State

In the bus-released state, the CPU releases bus to a device that has requested it.

Section 3 Floating-Point Unit (FPU)

3.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 16 single-precision floating-point registers (can also be referenced as eight double-precision registers)
- Two rounding modes: Round to nearest and round to zero
- Denormalization modes: Flush to zero
- Five exception sources: Invalid operation, divide by zero, overflow, underflow, and inexact
- Comprehensive instructions: Single-precision, double-precision, and system control

3.2 Data Formats

3.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

This LSI can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 3.1 and 3.2.

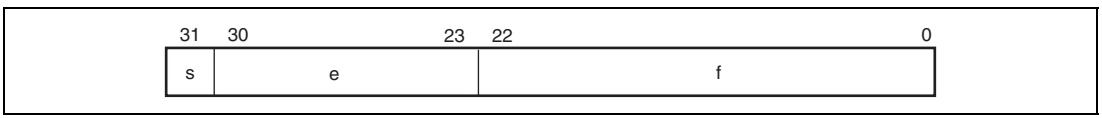


Figure 3.1 Format of Single-Precision Floating-Point Number

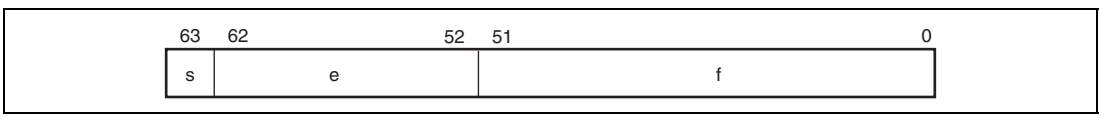


Figure 3.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). Table 3.1 shows E_{min} and E_{max} values.

Table 3.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s.
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity].
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number].
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number].
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero].

Table 3.2 shows the ranges of the various numbers in hexadecimal notation.

Table 3.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

3.2.2 Non-Numbers (NaN)

Figure 3.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

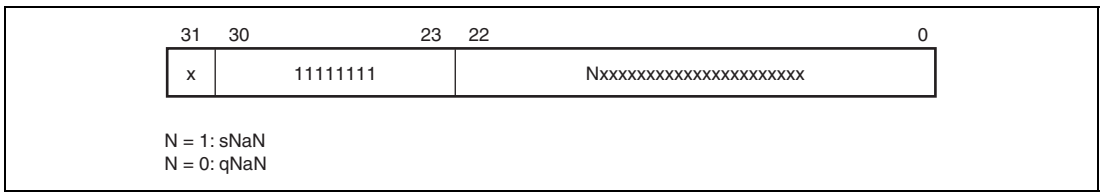


Figure 3.3 Single-Precision NaN Bit Pattern

An sNaN is input in an operation, except copy, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will generate FPU exception processing. In this case, the contents of the operation destination register are unchanged.

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results are as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

See the individual instruction descriptions for details of floating-point operations when a non-number (NaN) is input.

3.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

In the SH2A-FPU, the DN bit in the status register FPSCR is always set to 1, therefore a denormalized number (source operand or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy, FNEG, or FABS).

See the individual instruction descriptions for details of floating-point operations when a denormalized number is input.

3.3 Register Descriptions

3.3.1 Floating-Point Registers

Figure 3.4 shows the floating-point register configuration. There are sixteen 32-bit floating-point registers FPR0 to FPR15, referenced by specifying FR0 to FR15, DR0/2/4/6/8/10/12/14. The correspondence between FRPn and the reference name is determined by the PR and SZ bits in FPSCR. Refer figure 3.4.

1. Floating-point registers, FPRi (16 registers)
   FPR0, FPR1, FPR2, FPR3, FPR4, FPR5, FPR6, FPR7,
   FPR8, FPR9, FPR10, FPR11, FPR12, FPR13, FPR14, FPR15
2. Single-precision floating-point registers, FRi (16 registers)
   FR0 to FR15 indicate FPR0 to FPR15
3. Double-precision floating-point registers or single-precision floating-point registers in pairs, DRi (8 registers)
   A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

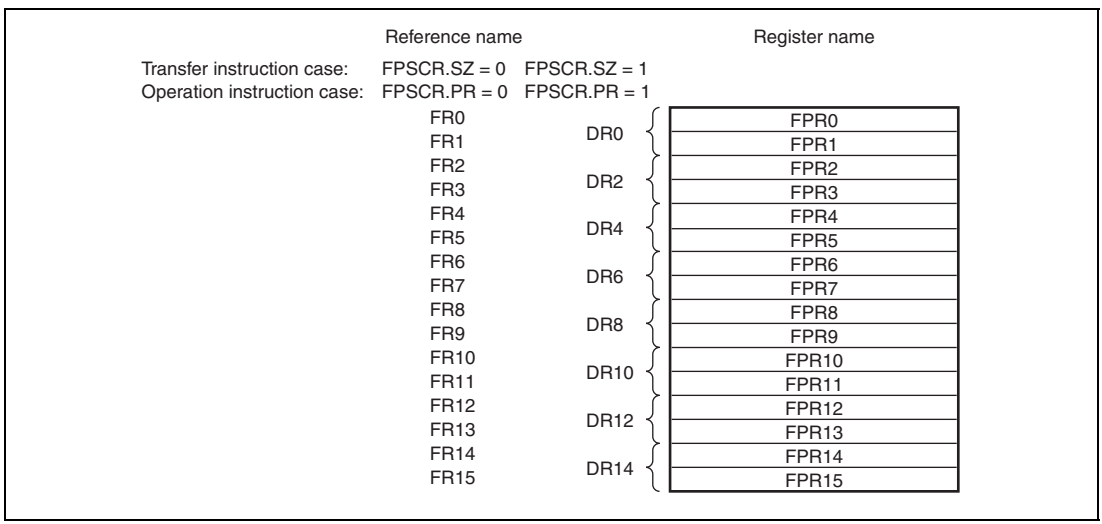


Figure 3.4 Floating-Point Registers

3.3.2 Floating-Point Status/Control Register (FPSCR)

FPSCR is a 32-bit register that controls floating-point instructions, sets FPU exceptions, and selects the rounding mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	QIS	-	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable					Flag					RM1	RM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	QIS	0	R/W	Nonnunerical Processing Mode 0: Processes qNaN or ±∞ as such 1: Treats qNaN or ±∞ as the same as sNaN (valid only when FPSCR.Enable.V = 1)
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits)
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations
18	DN	1	R	Denormalization Mode (Always fixed to 1 in SH2A-FPU) 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field
11 to 7	Enable	All 0	R/W	FPU Exception Enable Field
6 to 2	Flag	All 0	R/W	FPU Exception Flag Field Each time floating-point operation instruction is executed, the exception cause field is cleared to 0 first. When an FPU exception on floating-point operation occurs, the bits corresponding to the FPU exception cause field and FPU exception flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. As the bits corresponding to FPU exception enable filed are sets to 1, FPU exception processing occurs. For bit allocations of each field, see table 3.3.
1	RM1	0	R/W	Rounding Mode
0	RM0	1	R/W	These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved

Table 3.3 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Note: No FPU error occurs in the SH2A-FPU.

3.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system register that is accessed from the CPU side by means of LDS and STS instructions. For example, to convert the integer stored in general register R1 to a single-precision floating-point number, the processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

3.4 Rounding

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

(1) Round to Nearest

The operation result is rounded to the nearest expressible value. If there are two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax} (2 - 2^{-P})$ or more, the result will be infinity with the same sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 1023 and 53 for double-precision.

(2) Round to Zero

The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value will become the maximum expressible absolute value.

3.5 FPU Exceptions

3.5.1 FPU Exception Sources

FPU exceptions may occur on floating-point operation instruction and the exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input (No error occurs in the SH2A-FPU)
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

The FPU exception cause field in FPSCR contains bits corresponding to all of above sources E, V, Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corresponding to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the corresponding bit in the FPU exception flag field remains unchanged.

3.5.2 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input (No error occurs in the SH2A-FPU)
- Invalid operation (V): FPSCR.Enable.V = 1 and invalid operation
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation result overflow
- Underflow (U): FPSCR.Enable.U = 1 and instruction with possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

These possibilities of each exceptional handling on floating-point operation are shown in the individual instruction descriptions. All exception events that originate in the floating-point operation are assigned as the same FPU exceptional handling event. The meaning of an exception generated by floating-point operation is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed when FPU exception handling operation occurs.

Except for the above, the FPU disables exception handling. In every processing, the bit corresponding to source V, Z, O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):
  When rounding to zero = the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding to nearest = infinity with the same sign as the unrounded value is generated.
- Underflow (U):
  Zero with the same sign as the unrounded value is generated.
- Inexact exception (I): An inexact result is generated.

Section 4 MCU Operating Modes

4.1 Selection of Operating Modes

This LSI has four MCU operating modes and three on-chip flash memory programming modes. The operating mode is determined by the setting of FWE, MD1, and MD0 pins. Table 4.1 shows the allowable combinations of these pin settings; do not set these pins in the other way than the shown combinations.

Ensure that power-on reset processing proceeds whenever power is supplied to the system. Additionally, whether or not the H-UDI is in use, be sure to initialize the H-UDI by applying the low level to the $\overline{\text{TRST}}$ pin whenever power is supplied or assertion of the $\overline{\text{RES}}$ pin releases the chip from deep software standby mode.

The MCU operating mode can be selected from MCU extension modes 0 to 2 and single chip mode. For the on-chip flash memory programming mode, boot mode, user boot mode, and user program mode which are on-board programming modes are available.

Table 4.1 Selection of Operating Modes

	Pin Setting					
Mode No.	**FWE**	**MD1**	**MD0**	**Mode Name**	**On-Chip ROM**	**Bus Width of CS0 Space**
Mode 0	0	0	0	MCU extension mode 0	Not active	16
Mode 1	0	0	1	MCU extension mode 1	Not active	32
Mode 2	0	1	0	MCU extension mode 2	Active	Set by CS0BCR in BSC
Mode 3	0	1	1	Single chip mode	Active	—
Mode 4*	1	0	0	Boot mode	Active	—
Mode 5*	1	0	1	User boot mode	Active	Set by CS0BCR in BSC
Mode 6*	1	1	0	User program mode	Active	Set by CS0BCR in BSC
Mode 7*	1	1	1		Active	—

Note: * Flash memory programming mode.

4.2 Input/Output Pins

Table 4.2 describes the configuration of operating mode related pin.

Table 4.2 Pin Configuration

Pin Name	Input/Output	Function
MD0	Input	Designates operating mode through the level applied to this pin
MD1	Input	Designates operating mode through the level applied to this pin
FWE	Input	Enables, by hardware, programming/erasing of the on-chip flash memory

4.3 Operating Modes

4.3.1 Mode 0 (MCU Extension Mode 0)

In this mode, CS0 space becomes external memory spaces with 16-bit bus width.

4.3.2 Mode 1 (MCU Extension Mode 1)

In this mode, CS0 space becomes external memory spaces with 32-bit bus width.

4.3.3 Mode 2 (MCU Extension Mode 2)

The on-chip ROM is active and CS0 space can be used in this mode.

4.3.4 Mode 3 (Single Chip Mode)

All ports can be used in this mode, however the external address cannot be used.

4.4 Address Map

The address map for the operating modes is shown in figures 4.1 and 4.2.

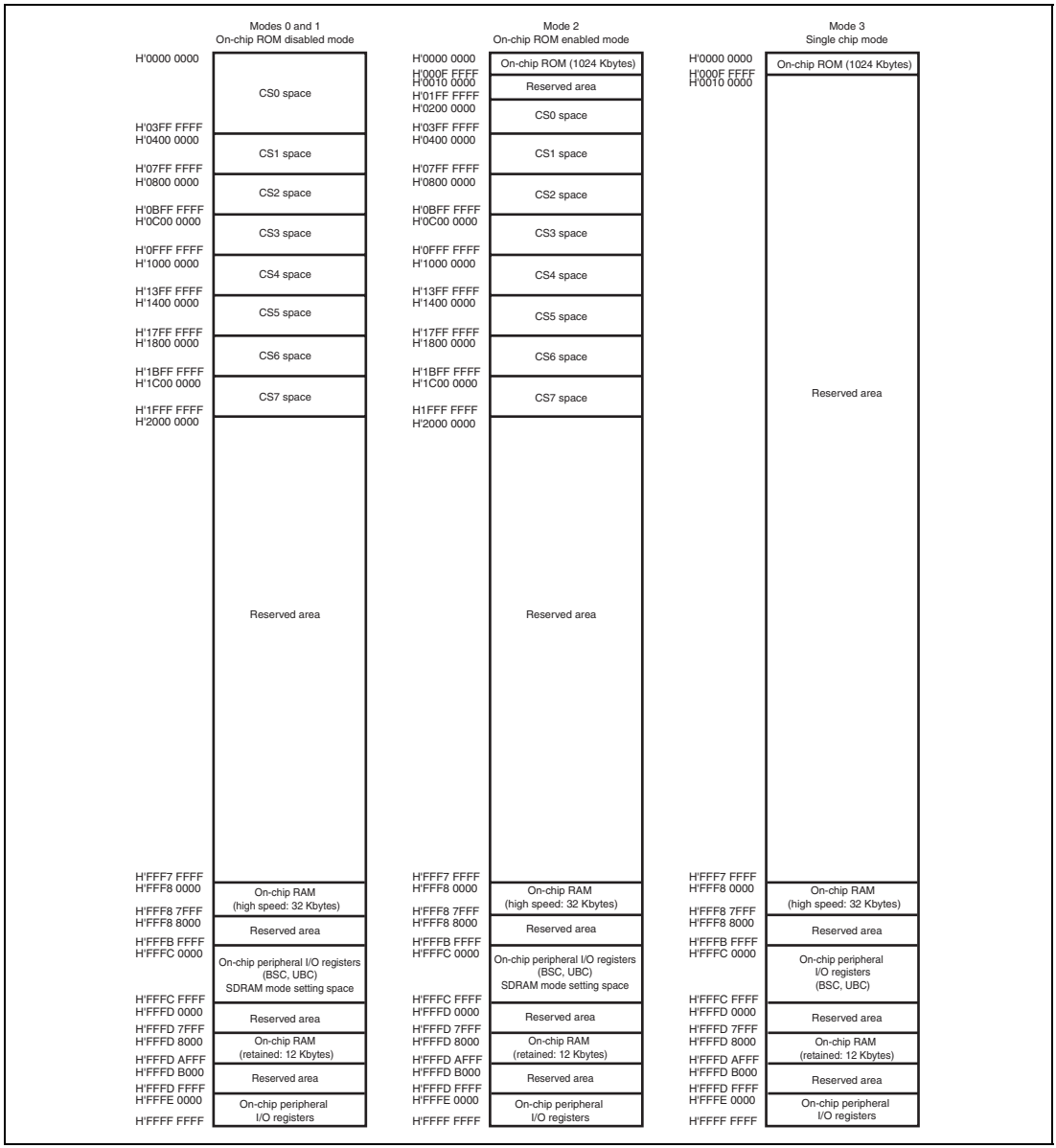


Figure 4.1 Address Map in Each Operating Mode (SH72315A/SH72315L)

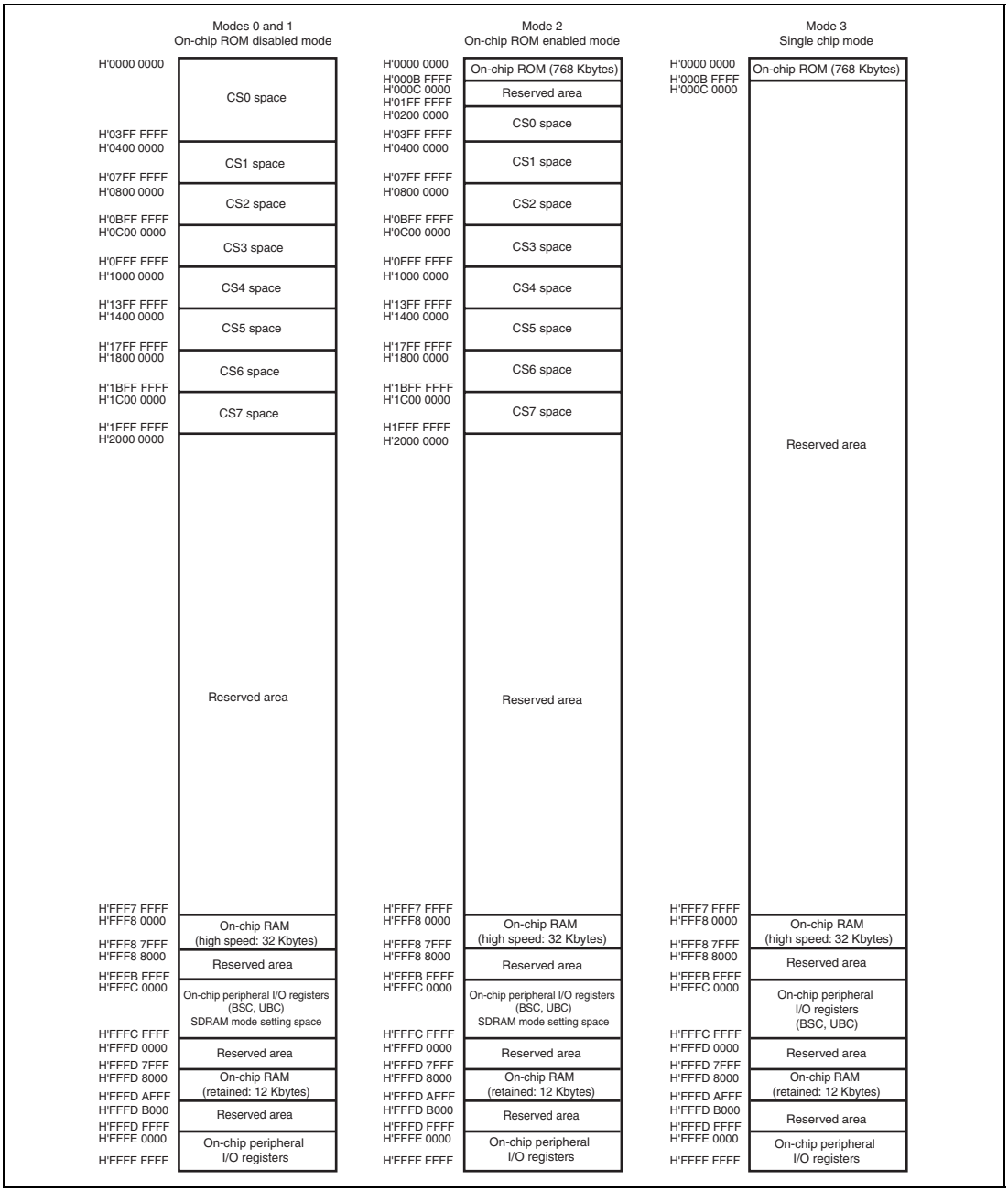


Figure 4.2 Address Map in Each Operating Mode (SH72314L)

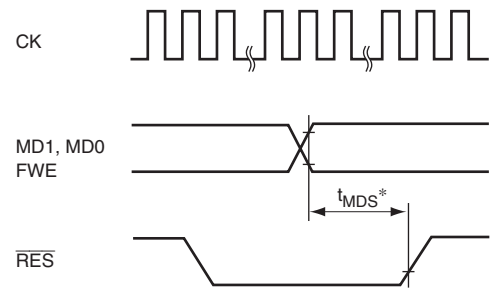
4.5 Initial State in this LSI

In the initial state of this LSI, some of on-chip modules are set in module standby state for saving power. When operating these modules, clear module standby state according to the procedure in section 32, Power-Down Modes.

4.6 Usage Notes

4.6.1 Note on Changing Operating Mode

If the operating mode is to be changed while power is being supplied to this LSI, be sure this is done while it is in the power-on reset state (that is, the low level is being applied to the $\overline{\text{RES}}$ pin). However, when the operating mode is changed between MCU extension mode 2 (mode 2) and user program mode (mode 6) or between single chip mode (mode 3) and user program mode (mode 7), switching the level on the FWE pin immediately changes the operating mode.



Note: * See section 35.4.2, Control Signal Timing.

Figure 4.3 Reset Input Timing when Changing Operating Mode

4.6.2 Note when Power is Applied

Ensure that power-on reset processing proceeds whenever power is supplied to the system. Additionally, whether or not the H-UDI is in use, be sure to initialize the H-UDI by applying the low level to the $\overline{\text{TRST}}$ pin whenever power is supplied or assertion of the $\overline{\text{RES}}$ pin releases the chip from deep software standby mode.

Section 5 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates an internal clock (Iϕ), a peripheral clock (Pϕ), a bus clock (Bϕ), an MTU2S clock (Mϕ), and an AD clock (Aϕ). The CPG consists of a crystal oscillator, a PLL circuit, and a divider circuit. The CPG also has a 32-kHz clock crystal oscillator (CPG32) for the KEYC and TIM32C modules.

5.1 Features

- Five clocks generated independently

  An internal clock (Iϕ) for the CPU, a peripheral clock (Pϕ) for the peripheral modules, a bus clock (Bϕ = CK) for the external bus interface, an MTU2S clock (Mϕ) for the MTU2S module, and an AD clock (Aϕ) for the ADC module can be generated independently.
- Frequency change function

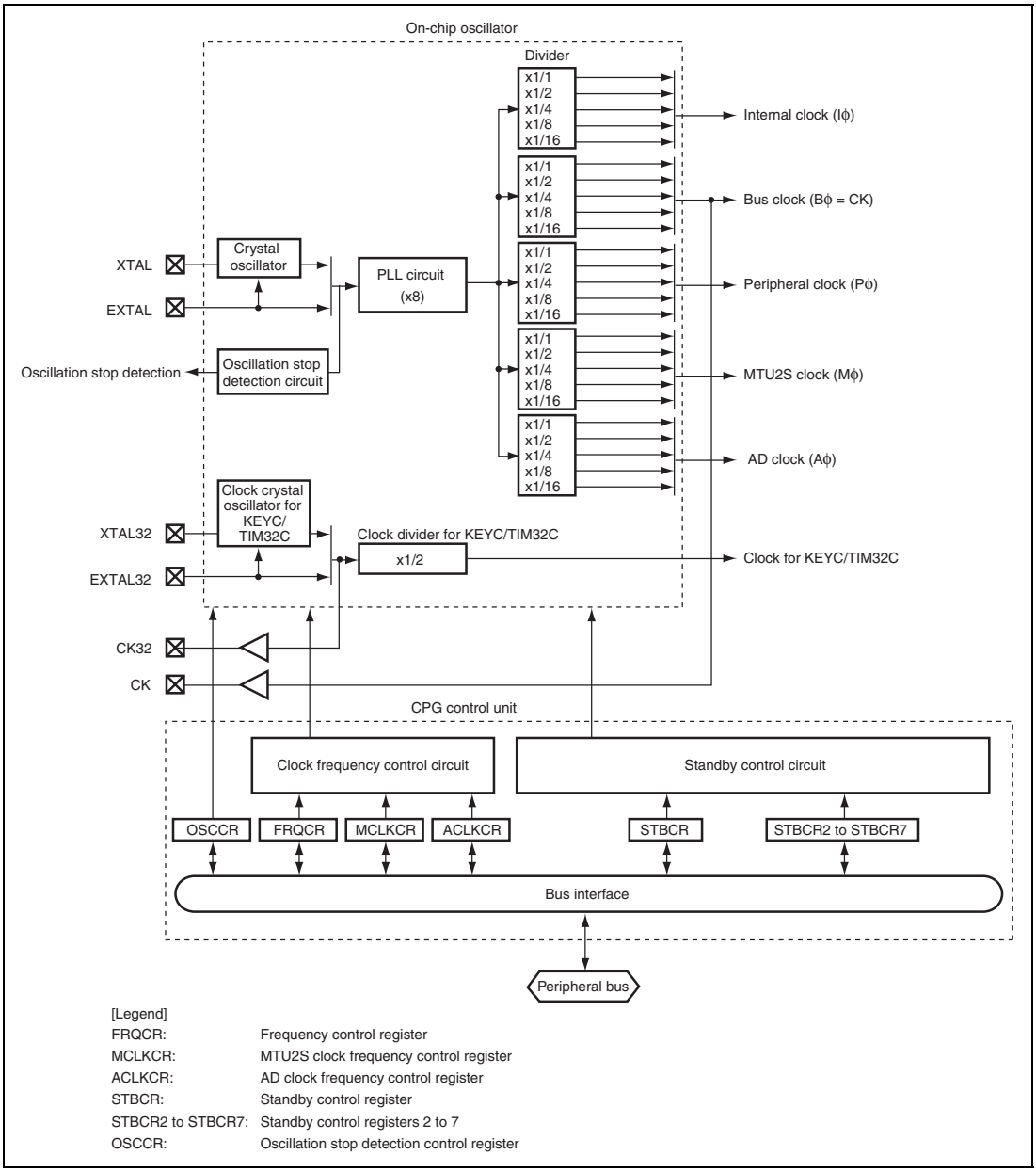
  The respective clock frequencies can be changed independently by using the divider circuits within the CPG. Changes to frequencies are made by software through the frequency control register (FRQCR) settings.
- Low power consumption mode control

  The clocks can be stopped in sleep mode, software standby mode, and deep software standby mode, and specific modules can be stopped by using the module standby function. For details on clock control in low power consumption modes, see section 32, Power-Down Modes.
- Oscillation stop detection function

  If the clock supply from the clock input pins is stopped for any reason, the timer pins are automatically driven to the high-impedance state.
- Clock generation for the KEYC and TIM32C modules

  The CPG can generate the clocks to be used by the key can controller (KEYC) and the 32-kHz timer (TIM32C). In addition, the supply of a 32-kHz system clock for external devices can be set up by pin function controller (PFC) settings.

Figure 5.1 shows a block diagram of the clock pulse generator.



[Legend]

FRQCR:	Frequency control register
MCLKCR:	MTU2S clock frequency control register
ACLKCR:	AD clock frequency control register
STBCR:	Standby control register
STBCR2 to STBCR7:	Standby control registers 2 to 7
OSCCR:	Oscillation stop detection control register

Figure 5.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

(1) PLL Circuit

The PLL circuit multiplies the input clock frequency from the crystal oscillator or EXTAL pin by 8.

(2) Crystal Oscillator

The crystal oscillator is an oscillation circuit in which a crystal resonator is connected to the XTAL pin or EXTAL pin.

(3) Divider

The divider generates a clock signal at the operating frequency used by the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (Mϕ), or AD clock (Aϕ). The operating frequency can be 1, 1/2, 1/4, 1/8, or 1/16 times the output frequency of the PLL circuit. The division ratio is set in the frequency control register (FRQCR).

(4) Oscillation Stop Detection Circuit

This circuit detects an abnormal condition in the crystal oscillator.

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the frequency control register (FRQCR).

(6) Standby Control Circuit

The standby control circuit controls the state of the on-chip oscillation circuit and other modules in sleep or software standby mode.

(7) Frequency Control Register (FRQCR)

The frequency control register (FRQCR) has control bits assigned for the following functions: the frequency division ratios of the internal clock (Iϕ), bus clock (Bϕ), and peripheral clock (Pϕ).

(8) MTU2S Clock Frequency Control Register (MCLKCR)

The MTU2S clock frequency control register (MCLKCR) has control bits assigned for the following functions: enabling or disabling the MTU2S clock (Mϕ) output and setting the frequency division ratio of the clock.

(9) AD Clock Frequency Control Register (ACLKCR)

The AD clock frequency control register (ACLKCR) has control bits assigned for the following functions: enabling or disabling the AD clock (Aϕ) output and setting the frequency division ratio of the clock.

(10) Standby Control Register

Bits for controlling the low-power consumption modes are assigned to the standby control register. See section 32, Power-Down Modes, for more information.

(11) Oscillation Stop Detection Control Register (OSCCR)

The oscillation stop detection control register (OSCCR) has an oscillation stop detection flag and a bit for selecting flag status output through an external pin.

(12) Crystal Oscillator for the KEYC and TIM32C Clock

This crystal oscillator is an oscillation circuit in which a 32.768-kHz crystal resonator is connected to the XTAL32 and EXTAL32 pins.

(13) Frequency Divider for the KEYC and TIM32C Clock

This divider generates a clock signal at the operating frequency used by the key scan controller (KEYC) and the 32-kHz timer (TIM32C). The division ratio is fixed to 1/2.

Table 5.1 lists the operating clock for each module.

Table 5.1 Operating Clock for Each Module

Operating Clock	Corresponding Modules
Internal clock (Iϕ)	CPU
	FPU
	UBC
	ROM
	RAM (high speed)
Bus clock (Bϕ)	BSC
	DMAC
	DTC
Peripheral clock (Pϕ)	MTU2
	POE2
	SCI
	SCIF
	IIC3
	CMT
	CMT2
	WDT
	LVDS (SH72315A only)
	RAM (for retention)
	H-UDI
	RSPI
	RCAN-ET
	Low power consumption mode control
MTU2S clock (Mϕ)	MTU2S
AD clock (Aϕ)	A/D converter
Clock for the KEYC and TIM32C modules	KEYC
	TIM32C

5.2 Input/Output Pins

Table 5.2 lists the clock pulse generator pins and their functions.

Table 5.2 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function
Crystal input/output pins (clock input pins)	XTAL	Output	Connected to the crystal resonator. (Leave this pin open when the crystal resonator is not in use.)
	EXTAL	Input	Connected to the crystal resonator or used to input an external clock.
Clock output pin	CK	Output	Clock output pin.
Crystal input/output pins for the KEYC and TIM32C modules (input pins for the KEYC/TIM32C clock)	XTAL32	Output	Connected to the crystal resonator of 32.768 kHz. Leave this pin open when the crystal resonator is not in use.
	EXTAL32	Input	Connected to the crystal resonator of 32.768 kHz or used to input an external clock. Connect this pin to Vss when the crystal resonator is not in use.
32-kHz clock output pin	CK32	Output	Supplies the 32-kHz system clock for external devices.

To use the clock output (CK) pin and 32-kHz clock output (CK32) pin, appropriate settings may be needed in the pin function controller (PFC) in some cases. For details, refer to section 22, Pin Function Controller (PFC).

5.3 Clock Operating Modes

Table 5.3 shows the clock operating modes of this LSI.

Table 5.3 Clock Operating Modes

Mode	Clock I/O		PLL Circuit	Input to Divider
	Source	Output		
1	EXTAL input or crystal resonator	CK* (= Bϕ)	On (× 8)	× 8
	EXTAL32 input or crystal resonator	CK32* (= EXTAL32)	None	× 1

Note: * To output the clock through the CK pin or CK32 pin, appropriate settings should be made in the PFC. For details, refer to section 22, Pin Function Controller (PFC).

The frequency of the external clock input from the EXTAL pin is multiplied by 8 in the PLL circuit before it is supplied to the on-chip modules in this LSI, which eliminates the need to generate a high-frequency clock outside the LSI. Since frequencies of the clock signal input from the EXTAL pin in the range from 10 MHz to 12.5 MHz can be used, the frequency of the internal clock (Iϕ) is in the range from 5 MHz to 100 MHz. The bus clock (Bϕ) is output from the CK pin.

Maximum operating frequencies:
Iϕ = 100 MHz, Bϕ = 50 MHz, Pϕ = 50 MHz, Mϕ = 100 MHz, Aϕ = 50 MHz

Table 5.4 shows an example of a range for the frequency division ratios that can be specified with FRQCR.

The clock input from the EXTAL32 pin does not pass through the PLL circuit before input to the frequency divider (fixed to 1/2) and is supplied to the KEYC and TIM32C modules. A clock signal with the same frequency as that input through the EXTAL32 pin is output from the CK32 pin.

Table 5.4 Example of Relationship between Clock Operating Mode and Frequency Range

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1/16	1/16	1/16	1/16	1/16	1/2	1/2	1/2	1/2	1/2	12.5	6.25	6.25	6.25	6.25	6.25
	1/8	1/16	1/16	1/16	1/16	1	1/2	1/2	1/2	1/2		12.5	6.25	6.25	6.25	6.25
	1/8	1/16	1/16	1/16	1/8	1	1/2	1/2	1/2	1		12.5	6.25	6.25	6.25	12.5
	1/8	1/16	1/16	1/8	1/16	1	1/2	1/2	1	1/2		12.5	6.25	6.25	12.5	6.25
	1/8	1/16	1/16	1/8	1/8	1	1/2	1/2	1	1		12.5	6.25	6.25	12.5	12.5
	1/8	1/8	1/16	1/16	1/16	1	1	1/2	1/2	1/2		12.5	12.5	6.25	6.25	6.25
	1/8	1/8	1/16	1/16	1/8	1	1	1/2	1/2	1		12.5	12.5	6.25	6.25	12.5
	1/8	1/8	1/16	1/8	1/16	1	1	1/2	1	1/2		12.5	12.5	6.25	12.5	6.25
	1/8	1/8	1/16	1/8	1/8	1	1	1/2	1	1		12.5	12.5	6.25	12.5	12.5
	1/8	1/8	1/8	1/8	1/8	1	1	1	1	1		12.5	12.5	12.5	12.5	12.5
	1/4	1/16	1/16	1/16	1/16	2	1/2	1/2	1/2	1/2		25	6.25	6.25	6.25	6.25
	1/4	1/16	1/16	1/16	1/8	2	1/2	1/2	1/2	1		25	6.25	6.25	6.25	12.5
	1/4	1/16	1/16	1/16	1/4	2	1/2	1/2	1/2	2		25	6.25	6.25	6.25	25
	1/4	1/16	1/16	1/8	1/16	2	1/2	1/2	1	1/2		25	6.25	6.25	12.5	6.25
	1/4	1/16	1/16	1/8	1/8	2	1/2	1/2	1	1		25	6.25	6.25	12.5	12.5
	1/4	1/16	1/16	1/8	1/4	2	1/2	1/2	1	2		25	6.25	6.25	12.5	25
	1/4	1/16	1/16	1/4	1/16	2	1/2	1/2	2	1/2		25	6.25	6.25	25	6.25
	1/4	1/16	1/16	1/4	1/8	2	1/2	1/2	2	1		25	6.25	6.25	25	12.5
	1/4	1/16	1/16	1/4	1/4	2	1/2	1/2	2	2		25	6.25	6.25	25	25
	1/4	1/8	1/16	1/16	1/16	2	1	1/2	1/2	1/2		25	12.5	6.25	6.25	6.25
	1/4	1/8	1/16	1/16	1/8	2	1	1/2	1/2	1		25	12.5	6.25	6.25	12.5
	1/4	1/8	1/16	1/16	1/4	2	1	1/2	1/2	2		25	12.5	6.25	6.25	25
	1/4	1/8	1/16	1/8	1/16	2	1	1/2	1	1/2		25	12.5	6.25	12.5	6.25
	1/4	1/8	1/16	1/8	1/8	2	1	1/2	1	1		25	12.5	6.25	12.5	12.5
	1/4	1/8	1/16	1/8	1/4	2	1	1/2	1	2		25	12.5	6.25	12.5	25
	1/4	1/8	1/16	1/4	1/16	2	1	1/2	2	1/2		25	12.5	6.25	25	6.25
	1/4	1/8	1/16	1/4	1/8	2	1	1/2	2	1		25	12.5	6.25	25	12.5
	1/4	1/8	1/16	1/4	1/4	2	1	1/2	2	2		25	12.5	6.25	25	25

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1/4	1/8	1/8	1/8	1/8	2	1	1	1	1	12.5	25	12.5	12.5	12.5	12.5
	1/4	1/8	1/8	1/8	1/4	2	1	1	1	2		25	12.5	12.5	12.5	25
	1/4	1/8	1/8	1/4	1/8	2	1	1	2	1		25	12.5	12.5	25	12.5
	1/4	1/8	1/8	1/4	1/4	2	1	1	2	2		25	12.5	12.5	25	25
	1/4	1/4	1/16	1/16	1/16	2	2	1/2	1/2	1/2		25	25	6.25	6.25	6.25
	1/4	1/4	1/16	1/16	1/8	2	2	1/2	1/2	1		25	25	6.25	6.25	12.5
	1/4	1/4	1/16	1/16	1/4	2	2	1/2	1/2	2		25	25	6.25	6.25	25
	1/4	1/4	1/16	1/8	1/16	2	2	1/2	1	1/2		25	25	6.25	12.5	6.25
	1/4	1/4	1/16	1/8	1/8	2	2	1/2	1	1		25	25	6.25	12.5	12.5
	1/4	1/4	1/16	1/8	1/4	2	2	1/2	1	2		25	25	6.25	12.5	25
	1/4	1/4	1/16	1/4	1/16	2	2	1/2	2	1/2		25	25	6.25	25	6.25
	1/4	1/4	1/16	1/4	1/8	2	2	1/2	2	1		25	25	6.25	25	12.5
	1/4	1/4	1/16	1/4	1/4	2	2	1/2	2	2		25	25	6.25	25	25
	1/4	1/4	1/8	1/8	1/8	2	2	1	1	1		25	25	12.5	12.5	12.5
	1/4	1/4	1/8	1/8	1/4	2	2	1	1	2		25	25	12.5	12.5	25
	1/4	1/4	1/8	1/4	1/8	2	2	1	2	1		25	25	12.5	25	12.5
	1/4	1/4	1/8	1/4	1/4	2	2	1	2	2		25	25	12.5	25	25
	1/4	1/4	1/4	1/4	1/4	2	2	2	2	2		25	25	25	25	25
	1/2	1/16	1/16	1/16	1/16	4	1/2	1/2	1/2	1/2		50	6.25	6.25	6.25	6.25
	1/2	1/16	1/16	1/16	1/8	4	1/2	1/2	1/2	1		50	6.25	6.25	6.25	12.5
	1/2	1/16	1/16	1/16	1/4	4	1/2	1/2	1/2	2		50	6.25	6.25	6.25	25
	1/2	1/16	1/16	1/16	1/2	4	1/2	1/2	1/2	4		50	6.25	6.25	6.25	50
	1/2	1/16	1/16	1/8	1/16	4	1/2	1/2	1	1/2		50	6.25	6.25	12.5	6.25
	1/2	1/16	1/16	1/8	1/8	4	1/2	1/2	1	1		50	6.25	6.25	12.5	12.5
	1/2	1/16	1/16	1/8	1/4	4	1/2	1/2	1	2		50	6.25	6.25	12.5	25
	1/2	1/16	1/16	1/8	1/2	4	1/2	1/2	1	4		50	6.25	6.25	12.5	50
	1/2	1/16	1/16	1/4	1/16	4	1/2	1/2	2	1/2		50	6.25	6.25	25	6.25
	1/2	1/16	1/16	1/4	1/8	4	1/2	1/2	2	1		50	6.25	6.25	25	12.5
	1/2	1/16	1/16	1/4	1/4	4	1/2	1/2	2	2		50	6.25	6.25	25	25
	1/2	1/16	1/16	1/4	1/2	4	1/2	1/2	2	4		50	6.25	6.25	25	50

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1/2	1/16	1/16	1/2	1/16	4	1/2	1/2	4	1/2	12.5	50	6.25	6.25	50	6.25
	1/2	1/16	1/16	1/2	1/8	4	1/2	1/2	4	1		50	6.25	6.25	50	12.5
	1/2	1/16	1/16	1/2	1/4	4	1/2	1/2	4	2		50	6.25	6.25	50	25
	1/2	1/16	1/16	1/2	1/2	4	1/2	1/2	4	4		50	6.25	6.25	50	50
	1/2	1/8	1/16	1/16	1/16	4	1	1/2	1/2	1/2		50	12.5	6.25	6.25	6.25
	1/2	1/8	1/16	1/16	1/8	4	1	1/2	1/2	1		50	12.5	6.25	6.25	12.5
	1/2	1/8	1/16	1/16	1/4	4	1	1/2	1/2	2		50	12.5	6.25	6.25	25
	1/2	1/8	1/16	1/16	1/2	4	1	1/2	1/2	4		50	12.5	6.25	6.25	50
	1/2	1/8	1/16	1/8	1/16	4	1	1/2	1	1/2		50	12.5	6.25	12.5	6.25
	1/2	1/8	1/16	1/8	1/8	4	1	1/2	1	1		50	12.5	6.25	12.5	12.5
	1/2	1/8	1/16	1/8	1/4	4	1	1/2	1	2		50	12.5	6.25	12.5	25
	1/2	1/8	1/16	1/8	1/2	4	1	1/2	1	4		50	12.5	6.25	12.5	50
	1/2	1/8	1/16	1/4	1/16	4	1	1/2	2	1/2		50	12.5	6.25	25	6.25
	1/2	1/8	1/16	1/4	1/8	4	1	1/2	2	1		50	12.5	6.25	25	12.5
	1/2	1/8	1/16	1/4	1/4	4	1	1/2	2	2		50	12.5	6.25	25	25
	1/2	1/8	1/16	1/4	1/2	4	1	1/2	2	4		50	12.5	6.25	25	50
	1/2	1/8	1/16	1/2	1/16	4	1	1/2	4	1/2		50	12.5	6.25	50	6.25
	1/2	1/8	1/16	1/2	1/8	4	1	1/2	4	1		50	12.5	6.25	50	12.5
	1/2	1/8	1/16	1/2	1/4	4	1	1/2	4	2		50	12.5	6.25	50	25
	1/2	1/8	1/16	1/2	1/2	4	1	1/2	4	4		50	12.5	6.25	50	50
	1/2	1/8	1/8	1/8	1/8	4	1	1	1	1		50	12.5	12.5	12.5	12.5
	1/2	1/8	1/8	1/8	1/4	4	1	1	1	2		50	12.5	12.5	12.5	25
	1/2	1/8	1/8	1/8	1/2	4	1	1	1	4		50	12.5	12.5	12.5	50
	1/2	1/8	1/8	1/4	1/8	4	1	1	2	1		50	12.5	12.5	25	12.5
	1/2	1/8	1/8	1/4	1/4	4	1	1	2	2		50	12.5	12.5	25	25
	1/2	1/8	1/8	1/4	1/2	4	1	1	2	4		50	12.5	12.5	25	50
	1/2	1/8	1/8	1/2	1/8	4	1	1	4	1		50	12.5	12.5	50	12.5
	1/2	1/8	1/8	1/2	1/4	4	1	1	4	2		50	12.5	12.5	50	25
	1/2	1/8	1/8	1/2	1/2	4	1	1	4	4		50	12.5	12.5	50	50
	1/2	1/4	1/16	1/16	1/16	4	2	1/2	1/2	1/2		50	25	6.25	6.25	6.25

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1/2	1/4	1/16	1/16	1/8	4	2	1/2	1/2	1	12.5	50	25	6.25	6.25	12.5
	1/2	1/4	1/16	1/16	1/4	4	2	1/2	1/2	2		50	25	6.25	6.25	25
	1/2	1/4	1/16	1/16	1/2	4	2	1/2	1/2	4		50	25	6.25	6.25	50
	1/2	1/4	1/16	1/8	1/16	4	2	1/2	1	1/2		50	25	6.25	12.5	6.25
	1/2	1/4	1/16	1/8	1/8	4	2	1/2	1	1		50	25	6.25	12.5	12.5
	1/2	1/4	1/16	1/8	1/4	4	2	1/2	1	2		50	25	6.25	12.5	25
	1/2	1/4	1/16	1/8	1/2	4	2	1/2	1	4		50	25	6.25	12.5	50
	1/2	1/4	1/16	1/4	1/16	4	2	1/2	2	1/2		50	25	6.25	25	6.25
	1/2	1/4	1/16	1/4	1/8	4	2	1/2	2	1		50	25	6.25	25	12.5
	1/2	1/4	1/16	1/4	1/4	4	2	1/2	2	2		50	25	6.25	25	25
	1/2	1/4	1/16	1/4	1/2	4	2	1/2	2	4		50	25	6.25	25	50
	1/2	1/4	1/16	1/2	1/16	4	2	1/2	4	1/2		50	25	6.25	50	6.25
	1/2	1/4	1/16	1/2	1/8	4	2	1/2	4	1		50	25	6.25	50	12.5
	1/2	1/4	1/16	1/2	1/4	4	2	1/2	4	2		50	25	6.25	50	25
	1/2	1/4	1/16	1/2	1/2	4	2	1/2	4	4		50	25	6.25	50	50
	1/2	1/4	1/8	1/8	1/8	4	2	1	1	1		50	25	12.5	12.5	12.5
	1/2	1/4	1/8	1/8	1/4	4	2	1	1	2		50	25	12.5	12.5	25
	1/2	1/4	1/8	1/8	1/2	4	2	1	1	4		50	25	12.5	12.5	50
	1/2	1/4	1/8	1/4	1/8	4	2	1	2	1		50	25	12.5	25	12.5
	1/2	1/4	1/8	1/4	1/4	4	2	1	2	2		50	25	12.5	25	25
	1/2	1/4	1/8	1/4	1/2	4	2	1	2	4		50	25	12.5	25	50
	1/2	1/4	1/8	1/2	1/8	4	2	1	4	1		50	25	12.5	50	12.5
	1/2	1/4	1/8	1/2	1/4	4	2	1	4	2		50	25	12.5	50	25
	1/2	1/4	1/8	1/2	1/2	4	2	1	4	4		50	25	12.5	50	50
	1/2	1/4	1/4	1/4	1/4	4	2	2	2	2		50	25	25	25	25
	1/2	1/4	1/4	1/4	1/2	4	2	2	2	4		50	25	25	25	50
	1/2	1/4	1/4	1/2	1/4	4	2	2	4	2		50	25	25	50	25
	1/2	1/4	1/4	1/2	1/2	4	2	2	4	4		50	25	25	50	50
	1/2	1/2	1/16	1/16	1/16	4	4	1/2	1/2	1/2		50	50	6.25	6.25	6.25
	1/2	1/2	1/16	1/16	1/8	4	4	1/2	1/2	1		50	50	6.25	6.25	12.5

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1/2	1/2	1/16	1/16	1/4	4	4	1/2	1/2	2	12.5	50	50	6.25	6.25	25
	1/2	1/2	1/16	1/16	1/2	4	4	1/2	1/2	4		50	50	6.25	6.25	50
	1/2	1/2	1/16	1/8	1/16	4	4	1/2	1	1/2		50	50	6.25	12.5	6.25
	1/2	1/2	1/16	1/8	1/8	4	4	1/2	1	1		50	50	6.25	12.5	12.5
	1/2	1/2	1/16	1/8	1/4	4	4	1/2	1	2		50	50	6.25	12.5	25
	1/2	1/2	1/16	1/8	1/2	4	4	1/2	1	4		50	50	6.25	12.5	50
	1/2	1/2	1/16	1/4	1/16	4	4	1/2	2	1/2		50	50	6.25	25	6.25
	1/2	1/2	1/16	1/4	1/8	4	4	1/2	2	1		50	50	6.25	25	12.5
	1/2	1/2	1/16	1/4	1/4	4	4	1/2	2	2		50	50	6.25	25	25
	1/2	1/2	1/16	1/4	1/2	4	4	1/2	2	4		50	50	6.25	25	50
	1/2	1/2	1/16	1/2	1/16	4	4	1/2	4	1/2		50	50	6.25	50	6.25
	1/2	1/2	1/16	1/2	1/8	4	4	1/2	4	1		50	50	6.25	50	12.5
	1/2	1/2	1/16	1/2	1/4	4	4	1/2	4	2		50	50	6.25	50	25
	1/2	1/2	1/16	1/2	1/2	4	4	1/2	4	4		50	50	6.25	50	50
	1/2	1/2	1/8	1/8	1/8	4	4	1	1	1		50	50	12.5	12.5	12.5
	1/2	1/2	1/8	1/8	1/4	4	4	1	1	2		50	50	12.5	12.5	25
	1/2	1/2	1/8	1/8	1/2	4	4	1	1	4		50	50	12.5	12.5	50
	1/2	1/2	1/8	1/4	1/8	4	4	1	2	1		50	50	12.5	25	12.5
	1/2	1/2	1/8	1/4	1/4	4	4	1	2	2		50	50	12.5	25	25
	1/2	1/2	1/8	1/4	1/2	4	4	1	2	4		50	50	12.5	25	50
	1/2	1/2	1/8	1/2	1/8	4	4	1	4	1		50	50	12.5	50	12.5
	1/2	1/2	1/8	1/2	1/4	4	4	1	4	2		50	50	12.5	50	25
	1/2	1/2	1/8	1/2	1/2	4	4	1	4	4		50	50	12.5	50	50
	1/2	1/2	1/4	1/4	1/4	4	4	2	2	2		50	50	25	25	25
	1/2	1/2	1/4	1/4	1/2	4	4	2	2	4		50	50	25	25	50
	1/2	1/2	1/4	1/2	1/4	4	4	2	4	2		50	50	25	50	25
	1/2	1/2	1/4	1/2	1/2	4	4	2	4	4		50	50	25	50	50
	1/2	1/2	1/2	1/2	1/2	4	4	4	4	4		50	50	50	50	50
	1	1/8	1/16	1/16	1/16	8	1	1/2	1/2	1/2		100	12.5	6.25	6.25	6.25
	1	1/8	1/16	1/16	1/8	8	1	1/2	1/2	1		100	12.5	6.25	6.25	12.5

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1	1/8	1/16	1/16	1/4	8	1	1/2	1/2	2	12.5	100	12.5	6.25	6.25	25
	1	1/8	1/16	1/16	1/2	8	1	1/2	1/2	4		100	12.5	6.25	6.25	50
	1	1/8	1/16	1/8	1/16	8	1	1/2	1	1/2		100	12.5	6.25	12.5	6.25
	1	1/8	1/16	1/8	1/8	8	1	1/2	1	1		100	12.5	6.25	12.5	12.5
	1	1/8	1/16	1/8	1/4	8	1	1/2	1	2		100	12.5	6.25	12.5	25
	1	1/8	1/16	1/8	1/2	8	1	1/2	1	4		100	12.5	6.25	12.5	50
	1	1/8	1/16	1/4	1/16	8	1	1/2	2	1/2		100	12.5	6.25	25	6.25
	1	1/8	1/16	1/4	1/8	8	1	1/2	2	1		100	12.5	6.25	25	12.5
	1	1/8	1/16	1/4	1/4	8	1	1/2	2	2		100	12.5	6.25	25	25
	1	1/8	1/16	1/4	1/2	8	1	1/2	2	4		100	12.5	6.25	25	50
	1	1/8	1/16	1/2	1/16	8	1	1/2	4	1/2		100	12.5	6.25	50	6.25
	1	1/8	1/16	1/2	1/8	8	1	1/2	4	1		100	12.5	6.25	50	12.5
	1	1/8	1/16	1/2	1/4	8	1	1/2	4	2		100	12.5	6.25	50	25
	1	1/8	1/16	1/2	1/2	8	1	1/2	4	4		100	12.5	6.25	50	50
	1	1/8	1/16	1	1/16	8	1	1/2	8	1/2		100	12.5	6.25	100	6.25
	1	1/8	1/16	1	1/8	8	1	1/2	8	1		100	12.5	6.25	100	12.5
	1	1/8	1/16	1	1/4	8	1	1/2	8	2		100	12.5	6.25	100	25
	1	1/8	1/16	1	1/2	8	1	1/2	8	4		100	12.5	6.25	100	50
	1	1/8	1/8	1/8	1/8	8	1	1	1	1		100	12.5	12.5	12.5	12.5
	1	1/8	1/8	1/8	1/4	8	1	1	1	2		100	12.5	12.5	12.5	25
	1	1/8	1/8	1/8	1/2	8	1	1	1	4		100	12.5	12.5	12.5	50
	1	1/8	1/8	1/4	1/8	8	1	1	2	1		100	12.5	12.5	25	12.5
	1	1/8	1/8	1/4	1/4	8	1	1	2	2		100	12.5	12.5	25	25
	1	1/8	1/8	1/4	1/2	8	1	1	2	4		100	12.5	12.5	25	50
	1	1/8	1/8	1/2	1/8	8	1	1	4	1		100	12.5	12.5	50	12.5
	1	1/8	1/8	1/2	1/4	8	1	1	4	2		100	12.5	12.5	50	25
	1	1/8	1/8	1/2	1/2	8	1	1	4	4		100	12.5	12.5	50	50
	1	1/8	1/8	1	1/8	8	1	1	8	1		100	12.5	12.5	100	12.5
	1	1/8	1/8	1	1/4	8	1	1	8	2		100	12.5	12.5	100	25
	1	1/8	1/8	1	1/2	8	1	1	8	4		100	12.5	12.5	100	50

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1	1/4	1/16	1/16	1/16	8	2	1/2	1/2	1/2	12.5	100	25	6.25	6.25	6.25
	1	1/4	1/16	1/16	1/8	8	2	1/2	1/2	1		100	25	6.25	6.25	12.5
	1	1/4	1/16	1/16	1/4	8	2	1/2	1/2	2		100	25	6.25	6.25	25
	1	1/4	1/16	1/16	1/2	8	2	1/2	1/2	4		100	25	6.25	6.25	50
	1	1/4	1/16	1/8	1/16	8	2	1/2	1	1/2		100	25	6.25	12.5	6.25
	1	1/4	1/16	1/8	1/8	8	2	1/2	1	1		100	25	6.25	12.5	12.5
	1	1/4	1/16	1/8	1/4	8	2	1/2	1	2		100	25	6.25	12.5	25
	1	1/4	1/16	1/8	1/2	8	2	1/2	1	4		100	25	6.25	12.5	50
	1	1/4	1/16	1/4	1/16	8	2	1/2	2	1/2		100	25	6.25	25	6.25
	1	1/4	1/16	1/4	1/8	8	2	1/2	2	1		100	25	6.25	25	12.5
	1	1/4	1/16	1/4	1/4	8	2	1/2	2	2		100	25	6.25	25	25
	1	1/4	1/16	1/4	1/2	8	2	1/2	2	4		100	25	6.25	25	50
	1	1/4	1/16	1/2	1/16	8	2	1/2	4	1/2		100	25	6.25	50	6.25
	1	1/4	1/16	1/2	1/8	8	2	1/2	4	1		100	25	6.25	50	12.5
	1	1/4	1/16	1/2	1/4	8	2	1/2	4	2		100	25	6.25	50	25
	1	1/4	1/16	1/2	1/2	8	2	1/2	4	4		100	25	6.25	50	50
	1	1/4	1/16	1	1/16	8	2	1/2	8	1/2		100	25	6.25	100	6.25
	1	1/4	1/16	1	1/8	8	2	1/2	8	1		100	25	6.25	100	12.5
	1	1/4	1/16	1	1/4	8	2	1/2	8	2		100	25	6.25	100	25
	1	1/4	1/16	1	1/2	8	2	1/2	8	4		100	25	6.25	100	50
	1	1/4	1/8	1/8	1/8	8	2	1	1	1		100	25	12.5	12.5	12.5
	1	1/4	1/8	1/8	1/4	8	2	1	1	2		100	25	12.5	12.5	25
	1	1/4	1/8	1/8	1/2	8	2	1	1	4		100	25	12.5	12.5	50
	1	1/4	1/8	1/4	1/8	8	2	1	2	1		100	25	12.5	25	12.5
	1	1/4	1/8	1/4	1/4	8	2	1	2	2		100	25	12.5	25	25
	1	1/4	1/8	1/4	1/2	8	2	1	2	4		100	25	12.5	25	50
	1	1/4	1/8	1/2	1/8	8	2	1	4	1		100	25	12.5	50	12.5
	1	1/4	1/8	1/2	1/4	8	2	1	4	2		100	25	12.5	50	25
	1	1/4	1/8	1/2	1/2	8	2	1	4	4		100	25	12.5	50	50
	1	1/4	1/8	1	1/8	8	2	1	8	1		100	25	12.5	100	12.5

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1	1/4	1/8	1	1/4	8	2	1	8	2	12.5	100	25	12.5	100	25
	1	1/4	1/8	1	1/2	8	2	1	8	4		100	25	12.5	100	50
	1	1/4	1/4	1/4	1/4	8	2	2	2	2		100	25	25	25	25
	1	1/4	1/4	1/4	1/2	8	2	2	2	4		100	25	25	25	50
	1	1/4	1/4	1/2	1/4	8	2	2	4	2		100	25	25	50	25
	1	1/4	1/4	1/2	1/2	8	2	2	4	4		100	25	25	50	50
	1	1/4	1/4	1	1/4	8	2	2	8	2		100	25	25	100	25
	1	1/4	1/4	1	1/2	8	2	2	8	4		100	25	25	100	50
	1	1/2	1/16	1/16	1/16	8	4	1/2	1/2	1/2		100	50	6.25	6.25	6.25
	1	1/2	1/16	1/16	1/8	8	4	1/2	1/2	1		100	50	6.25	6.25	12.5
	1	1/2	1/16	1/16	1/4	8	4	1/2	1/2	2		100	50	6.25	6.25	25
	1	1/2	1/16	1/16	1/2	8	4	1/2	1/2	4		100	50	6.25	6.25	50
	1	1/2	1/16	1/8	1/16	8	4	1/2	1	1/2		100	50	6.25	12.5	6.25
	1	1/2	1/16	1/8	1/8	8	4	1/2	1	1		100	50	6.25	12.5	12.5
	1	1/2	1/16	1/8	1/4	8	4	1/2	1	2		100	50	6.25	12.5	25
	1	1/2	1/16	1/8	1/2	8	4	1/2	1	4		100	50	6.25	12.5	50
	1	1/2	1/16	1/4	1/16	8	4	1/2	2	1/2		100	50	6.25	25	6.25
	1	1/2	1/16	1/4	1/8	8	4	1/2	2	1		100	50	6.25	25	12.5
	1	1/2	1/16	1/4	1/4	8	4	1/2	2	2		100	50	6.25	25	25
	1	1/2	1/16	1/4	1/2	8	4	1/2	2	4		100	50	6.25	25	50
	1	1/2	1/16	1/2	1/16	8	4	1/2	4	1/2		100	50	6.25	50	6.25
	1	1/2	1/16	1/2	1/8	8	4	1/2	4	1		100	50	6.25	50	12.5
	1	1/2	1/16	1/2	1/4	8	4	1/2	4	2		100	50	6.25	50	25
	1	1/2	1/16	1/2	1/2	8	4	1/2	4	4		100	50	6.25	50	50
	1	1/2	1/16	1	1/16	8	4	1/2	8	1/2		100	50	6.25	100	6.25
	1	1/2	1/16	1	1/8	8	4	1/2	8	1		100	50	6.25	100	12.5
	1	1/2	1/16	1	1/4	8	4	1/2	8	2		100	50	6.25	100	25
	1	1/2	1/16	1	1/2	8	4	1/2	8	4		100	50	6.25	100	50
	1	1/2	1/8	1/8	1/8	8	4	1	1	1		100	50	12.5	12.5	12.5
	1	1/2	1/8	1/8	1/4	8	4	1	1	2		100	50	12.5	12.5	25

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×8	1	1/2	1/8	1/8	1/2	8	4	1	1	4	12.5	100	50	12.5	12.5	50
	1	1/2	1/8	1/4	1/8	8	4	1	2	1		100	50	12.5	25	12.5
	1	1/2	1/8	1/4	1/4	8	4	1	2	2		100	50	12.5	25	25
	1	1/2	1/8	1/4	1/2	8	4	1	2	4		100	50	12.5	25	50
	1	1/2	1/8	1/2	1/8	8	4	1	4	1		100	50	12.5	50	12.5
	1	1/2	1/8	1/2	1/4	8	4	1	4	2		100	50	12.5	50	25
	1	1/2	1/8	1/2	1/2	8	4	1	4	4		100	50	12.5	50	50
	1	1/2	1/8	1	1/8	8	4	1	8	1		100	50	12.5	100	12.5
	1	1/2	1/8	1	1/4	8	4	1	8	2		100	50	12.5	100	25
	1	1/2	1/8	1	1/2	8	4	1	8	4		100	50	12.5	100	50
	1	1/2	1/4	1/4	1/4	8	4	2	2	2		100	50	25	25	25
	1	1/2	1/4	1/4	1/2	8	4	2	2	4		100	50	25	25	50
	1	1/2	1/4	1/2	1/4	8	4	2	4	2		100	50	25	50	25
	1	1/2	1/4	1/2	1/2	8	4	2	4	4		100	50	25	50	50
	1	1/2	1/4	1	1/4	8	4	2	8	2		100	50	25	100	25
	1	1/2	1/4	1	1/2	8	4	2	8	4		100	50	25	100	50
	1	1/2	1/2	1/2	1/2	8	4	4	4	4		100	50	50	50	50
	1	1/2	1/2	1	1/2	8	4	4	8	4		100	50	50	100	50

Notes: * Clock frequencies when the input clock frequency is assumed to be the shown value.

1. The PLL multiplication ratio is fixed at ×8. The division ratio can be selected from ×1, ×1/2, ×1/4, ×1/8, and ×1/16 for each clock by the setting in the frequency control register.
2. The output frequency of the PLL circuit is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin and the multiplication ratio (×8) of the PLL circuit. This output frequency must be 100 MHz or lower.
3. The input to the divider is always the output from the PLL circuit.
4. The frequency for each clock is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, and the division ratio of the divider.
5. The frequency of the internal clock (Iϕ) must be 100 MHz or lower.
6. The frequency of the bus clock (Bϕ) must be 50 MHz or lower and also lower than or equal to the frequency of the internal clock (Iϕ).
7. The frequency of the peripheral clock (Pϕ) must be 50 MHz or lower and also lower than or equal to the frequency of the bus clock (Bϕ).

8. The frequency of the MTU2S clock (Mϕ) must be lower than or equal to the frequency of the internal clock (Iϕ) and also higher than or equal to the frequency of the peripheral clock (Pϕ).
9. The frequency of the AD clock (Aϕ) must be 50 MHz or lower, lower than or equal to the frequency of the internal clock (Iϕ), and also higher than or equal to the frequency of the peripheral clock (Pϕ).
10. The frequency of the CK pin output is always equal to the frequency of the bus clock (Bϕ).
11. Settings that lead to the following relation are prohibited—internal clock (Iϕ) : bus clock (Bϕ) = 1 : 1/16.
12. Any combination not listed in the above table is not settable.

5.4 Register Descriptions

The clock pulse generator has the following registers.

Table 5.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0666	H'FFFE0010	16
MTU2S clock frequency control register	MCLKCR	R/W	H'46	H'FFFE0410	8
AD clock frequency control register	ACLKCR	R/W	H'46	H'FFFE0414	8
Oscillation stop detection control register	OSCCR	R/W	H'00	H'FFFE001C	8

5.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register used to specify the frequency division ratios for the internal clock (Iϕ), bus clock (Bϕ), and peripheral clock (Pϕ). FRQCR is only accessible in word units. Additionally, make settings for individual modules after setting FRQCR.

FRQCR is initialized to H'0666 only by a power-on reset. FRQCR retains its previous value by a manual reset or in software standby mode. The previous value is also retained when an internal reset is triggered by an overflow of the WDT.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	STC[2:0]			-	IFC[2:0]			-	PFC[2:0]		
Initial value:	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	STC[2:0]	110	R/W	Bus Clock (Bϕ) Frequency Division Ratio These bits specify the frequency division ratio of the bus clock. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 110: × 1/16 Other than above: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	IFC[2:0]	110	R/W	Internal Clock (Iϕ) Frequency Division Ratio These bits specify the frequency division ratio of the internal clock. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 110: × 1/16 Other than above: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PFC[2:0]	110	R/W	Peripheral Clock (Pϕ) Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 110: × 1/16 Other than above: Setting prohibited

5.4.2 MTU2S Clock Frequency Control Register (MCLKCR)

MCLKCR is an 8-bit readable/writable register. MCLKCR can be accessed only in byte units.

MCLKCR is initialized to H'46 only by a power-on reset. MCLKCR retains its previous value by a manual reset or in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	MSSCS[1:0]		-	-	-	MSDIVS[2:0]		
Initial value:	0	1	0	0	0	1	1	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MSSCS[1:0]	01	R/W	Source Clock Select These bits select the source clock. 00: Clock stopped 01: PLL output clock 10: Reserved (setting prohibited) 11: Reserved (setting prohibited)
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	MSDIVS[2:0]	110	R/W	Division Ratio Select These bits specify the frequency division ratio of the source clock. Set these bits so that the output clock is lower than or equal to the frequency of the internal clock (Iϕ) and also an integer multiple of the frequency of the peripheral clock (Pϕ). 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 110: × 1/16 Other than above: Setting prohibited

5.4.3 AD Clock Frequency Control Register (ACLKCR)

ACLKCR is an 8-bit readable/writable register that can be accessed only in byte units.

ACLKCR is initialized to H'46 only by a power-on reset, but retains its previous value by a manual reset or in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ASSCS[1:0]		-	-	-	ASDIVS[2:0]		
Initial value:	0	1	0	0	0	1	1	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	ASSCS[1:0]	01	R/W	Source Clock Select These bits select the source clock. 00: Clock stopped 01: PLL output clock 10: Reserved (setting prohibited) 11: Reserved (setting prohibited)
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	ASDIVS[2:0]	110	R/W	Division Ratio Select These bits specify the frequency division ratio of the source clock. Set these bits so that the output clock is 50 MHz or less, lower than or equal to the frequency of the internal clock (Iϕ), and also an integer multiple of the frequency of the peripheral clock (Pϕ). 000: $\times$ 1 001: $\times$ 1/2 010: Setting prohibited 011: $\times$ 1/4 100: Setting prohibited 101: $\times$ 1/8 110: $\times$ 1/16 Other than above: Setting prohibited

5.4.4 Oscillation Stop Detection Control Register (OSCCR)

OSCCR is an 8-bit readable/writable register that has an oscillation stop detection flag and selects flag status output to an external pin. OSCCR can be accessed only in byte units.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	OSC STOP	-	OSC ERS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	OSCSTOP	0	R	Oscillation Stop Detection Flag [Setting conditions] • When a stop in the clock input is detected during normal operation • When software standby mode is entered [Clearing conditions] • By a power-on reset input through the $\overline{\text{RES}}$ pin • When software standby mode is canceled
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	OSCERS	0	R/W	Oscillation Stop Detection Flag Output Select Selects whether to output the oscillation stop detection flag signal through the $\overline{\text{WDTOVF}}$ pin. 0: Outputs only the WDT overflow signal through the $\overline{\text{WDTOVF}}$ pin 1: Outputs the WDT overflow signal and oscillation stop detection flag signal through the $\overline{\text{WDTOVF}}$ pin

5.5 Changing the Frequency

Selecting division ratios for the frequency divider can change the frequencies of the internal clock, bus clock, peripheral clock, MTU2S clock, and AD clock under the software control through the frequency control register (FRQCR). The following describes how to specify the frequencies.

1. In the initial state, IFC2 to IFC0 = B'110 (×1/16), STC2 to STC0 = B'110 (×1/16), PFC2 to PFC0 = B'110 (×1/16), MSDIVS2 to MSDIVS0 = B'110 (×1/16), and ASDIVS2 to ASDIVS0 = B'110 (×1/16).
2. Stop all modules except the CPU, on-chip ROM, and on-chip RAM.
3. Set the desired values in bits IFC2 to IFC0, STC2 to STC0, PFC2 to PFC0, MSDIVS2 to MSDIVS0, and ASDIVS2 to ASDIVS0. When specifying the frequencies, satisfy the following condition: internal clock (Iϕ) ≥ bus clock (Bϕ) ≥ peripheral clock (Pϕ). Moreover, when using the MTU2S clock, specify the frequencies to satisfy the following condition: internal clock (Iϕ) ≥ MTU2S clock (Mϕ) ≥ peripheral clock (Pϕ). When using the AD clock, specify the frequencies to satisfy the following condition: internal clock (Iϕ) ≥ AD clock (Aϕ) ≥ peripheral clock (Pϕ).
4. After an instruction to rewrite FRQCR has been issued, the actual clock frequencies will change after (1 to 16) cyc + 3Bϕ + 6Pϕ.
   cyc: Clock obtained by using the PLL to multiply the clock signal from the EXTAL pin by 8.
   Note: (1 to 16) depends on the internal state.

5.6 Oscillator

The source of clock supply can be selected from a connected crystal resonator or an external clock input through a pin.

5.6.1 Connecting Crystal Resonator

A crystal resonator can be connected as shown in figure 5.2. Use the damping resistance (R_d) shown in table 5.6. Use a crystal resonator that has a resonance frequency of 10 to 12.5 MHz.

It is recommended to consult the crystal resonator manufacturer concerning the compatibility of the crystal resonator and the LSI.

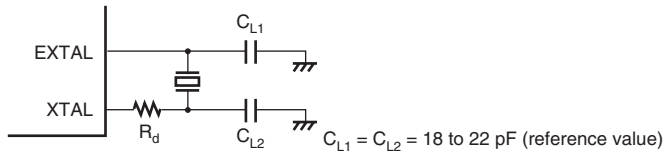


Figure 5.2 Example of Crystal Resonator Connection

Table 5.6 Damping Resistance Values (Reference Values)

Frequency (MHz)	10	12.5
R_d (Ω) (reference value)	0	0

Figure 5.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics shown in table 5.7.

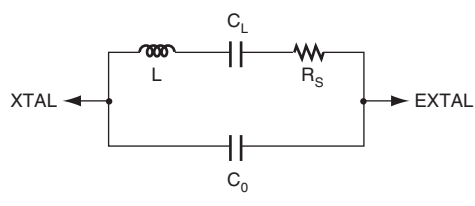


Figure 5.3 Crystal Resonator Equivalent Circuit

Table 5.7 Crystal Resonator Characteristics

Frequency (MHz)	10	12.5
R_s max. (Ω) (reference value)	60	50
C_0 max. (pF) (reference value)	7	7

5.6.2 External Clock Input Method

Figure 5.4 shows an example of an external clock input connection. Drive the external clock high when it is stopped in software standby mode or deep software standby mode. During operation, input an external clock with a frequency of 10 to 12.5 MHz. Make sure the parasitic capacitance of the XTAL pin is 10 pF or less.

Even when inputting an external clock, be sure to wait at least for the oscillation settling time in power-on sequence or in canceling software standby mode or deep software standby mode, in order to ensure the PLL settling time.

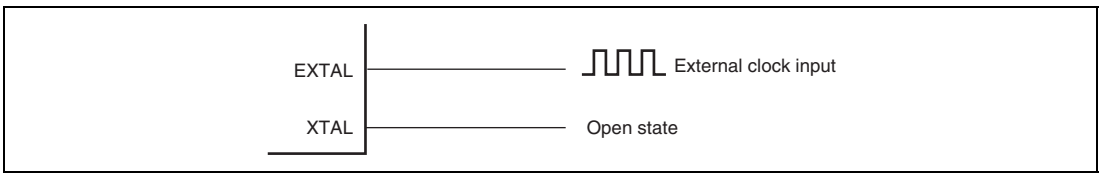


Figure 5.4 Example of External Clock Connection

5.7 Oscillation Stop Detection

The CPG detects a stop in the clock input if any system abnormality halts the clock supply.

When no change has been detected in the EXTAL input for a certain period, the OSCSTOP bit in OSCCR is set to 1 and this state is retained until a power-on reset is input through the $\overline{\text{RES}}$ pin or the software standby mode is canceled. If the OSCERS bit is 1 at this time, an oscillation stop detection flag signal is output through the $\overline{\text{WDTOVF}}$ pin. In addition, the high-current ports (multiplexed pins to which the TIOC3B, TIOC3D, and TIOC4A to TIOC4D signals in the MTU2, and the TIOC3BS, TIOC3DS, and TIOC4AS to TIOC4DS signals in the MTU2S are assigned) can be placed in the high-impedance state regardless of the PFC setting. Said pins can be placed in the high-impedance state even when the chip is in software standby mode. For details, refer to section 22.1.29, Large Current Port Control Register (HCPCR) and appendix A, Pin States.

These pins enter the normal state after release from software standby mode. In addition, under an abnormal condition where oscillation stops while the LSI is not in software standby mode, LSI operations other than the oscillation stop detection function become unpredictable. In this case, even after oscillation is restarted, LSI operations including the above high-current pins become unpredictable.

Even while no change is detected in the EXTAL input, the PLL circuit in this LSI continues oscillating at a frequency range from 100 kHz to 10 MHz (depending on the temperature and operating voltage).

5.8 Clock Oscillator for the KEYC and TIM32C Modules

The clock supply for the KEYC and TIM32C modules is selectable as a connected 32.768-kHz crystal resonator or an external clock input.

5.8.1 Connecting a 32.768-kHz Crystal Resonator

Figure 5.5 shows an example of the connections for a 32.768-kHz crystal resonator. When a crystal resonator is connected, be sure to set both the MSTP74 and MSTP75 bits in the standby control register 7 (STBCR7) to 0.

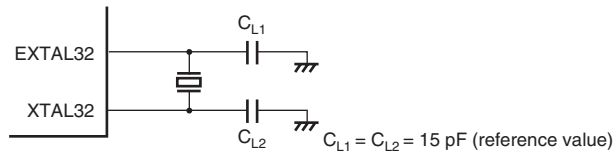


Figure 5.5 Example of Connecting a 32.768-kHz Crystal Resonator

5.8.2 Input of an External Clock Signal

Figure 5.6 shows an example of the connections for input of an external clock signal. The XTAL32 pin must be left open. When an external clock signal is input, be sure to set the MSTP74 and MSTP75 bits in the standby control register 7 (STBCR7) to 1 and 0, respectively.

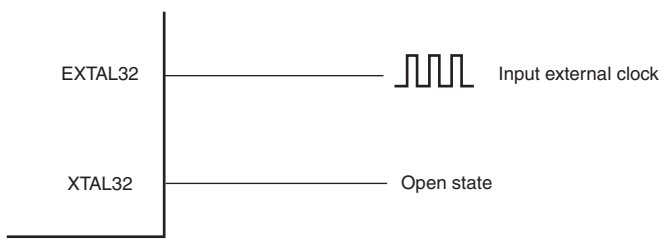


Figure 5.6 Example of Connecting an External Clock

5.8.3 Handling of Pins when the Clock for the KEYC and TIM32C Modules is Not in Use

When the clock for the KEYC and TIM32C modules is not in use, connect the EXTAL32 pin to ground (Vss) and leave the XTAL32 pin open-circuit as shown in figure 5.7. In addition, be sure to set both the MSTP74 and MSTP75 bits in the standby control register 7 (STBCR7) to 1.

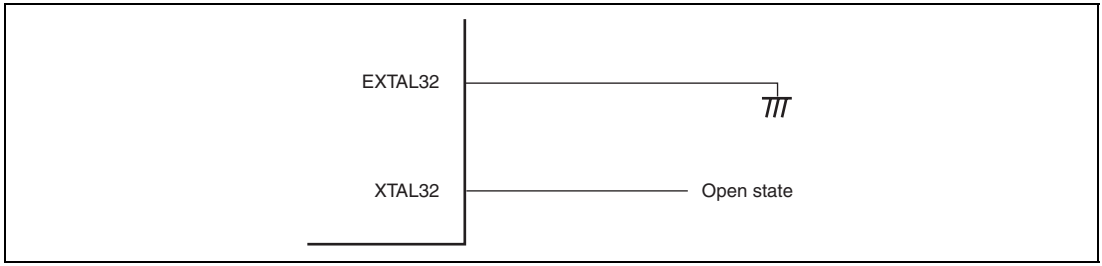


Figure 5.7 Handling of Pins when the Clock for the KEYC and TIM32C Modules is Not in Use

5.9 Usage Notes

5.9.1 Note on Crystal Resonator

A sufficient evaluation at the user's site is necessary to use the LSI, by referring the resonator connection examples shown in this section, because various characteristics related to the crystal resonator are closely linked to the user's board design. As the oscillator circuit's circuit constant will depend on the resonator and the floating capacitance of the mounting circuit, the value of each external circuit's component should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

5.9.2 Notes on Board Design

Measures against radiation noise are taken in this LSI. If further reduction in radiation noise is needed, it is recommended to use a multiple layer board and provide a layer exclusive to the system ground.

When using a crystal resonator, place the crystal resonator and capacitors CL1 and CL2 as close as possible to the XTAL and EXTAL pins. In addition, to minimize induction and thus obtain oscillation at the correct frequency, the capacitors to be attached to the resonator must be grounded to the same ground. Do not bring wiring patterns close to these components.

The XTAL32 and EXTAL32 pins should be handled in a similar manner.

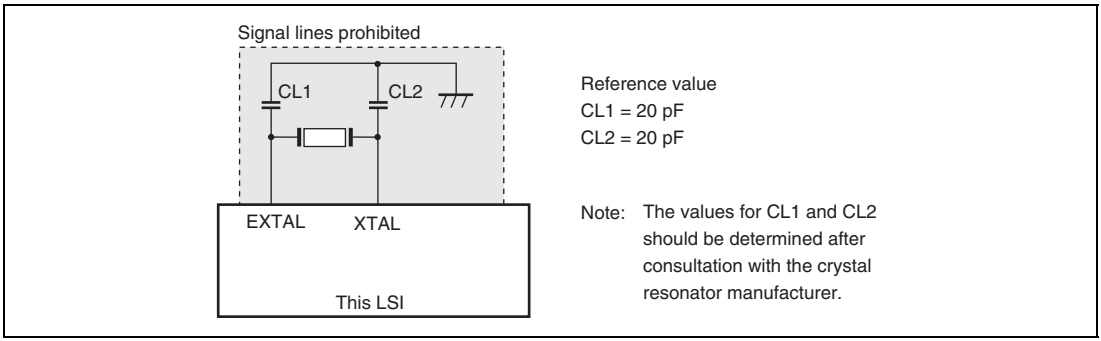


Figure 5.8 Note on Using a Crystal Resonator

A circuitry shown in figure 5.9 is recommended as an external circuitry around the PLL. The PLL power lines (PLLVcc, PLLVss) and the system power lines (Vcc, Vss) must be separated from the board power supply source to avoid an influence from power supply noise. Be sure to insert bypass capacitors CB and CPB close to the pins.

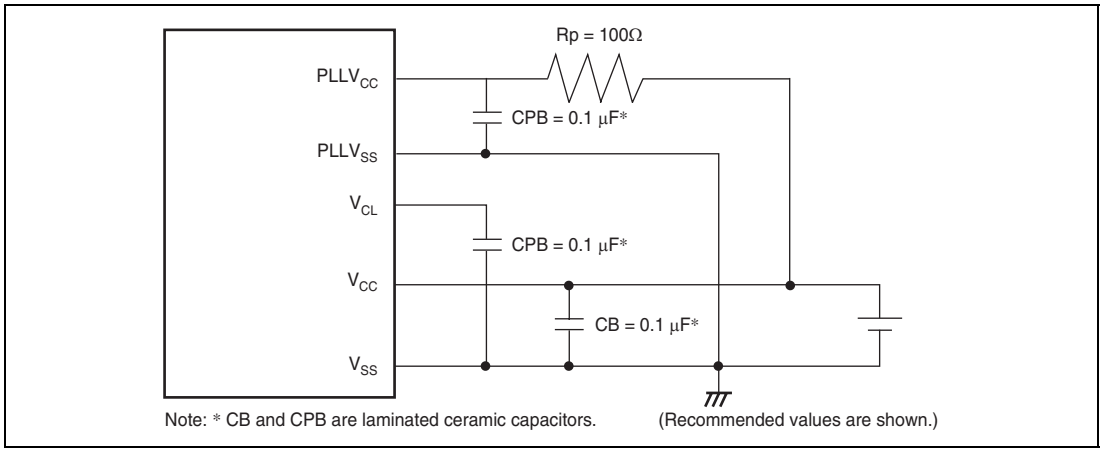


Figure 5.9 Recommended External Circuitry around PLL

Section 6 Exception Handling

6.1 Overview

6.1.1 Types of Exception Handling and Priority

Exception handling is started by sources, such as resets, address errors, register bank errors, interrupts, and instructions. Table 6.1 shows their priorities. When several exception handling sources occur at once, they are processed according to the priority shown.

Table 6.1 Types of Exception Handling and Priority Order

Type	Exception Handling		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error		
	DMAC/DTC address error		
Instruction	FPU exception		
	Integer division exception (division by zero)		
	Integer division exception (overflow)		
Register bank error	Bank underflow		
	Bank overflow		
Interrupt	NMI		
	User break		
	H-UDI		
	IRQ		
	FCU interrupt (FIF)		
	On-chip peripheral modules	32-kHz timer (TIM32C)	
		Key scan controller (KEYC)	
		A/D converter (ADC)	
		Direct memory access controller (DMAC)	
		Serial communication interface with FIFO (SCIF)	Low

Type	Exception Handling		Priority
Interrupt	On-chip peripheral modules	Compare match timer (CMT)	High
		Compare match timer 2 (CMT2)	
		Low power consumption functions	
		Bus state controller (BSC)	
		Renesas serial peripheral interface (RSPI)	
		Watchdog timer (WDT)	
		Multi-function timer pulse unit 2 (MTU2)	
		Port output enable 2 (POE2)	
		Multi-function timer pulse unit 2S (MTU2S)	
		LVDS receive interface (LVDS) (SH72315A only)	
		I^2C bus interface 3 (IIC3)	
		Controller area network (RCAN-ET)	
		Serial communication interface (SCI)	
Instruction	Trap instruction (TRAPA instruction)		
	General illegal instructions (undefined code)		
	Slot illegal instructions (undefined code placed directly after a delayed branch instruction*1 (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby), instructions that rewrite the PC*2, 32-bit instructions*3, RESBANK instruction, DIVS instruction, and DIVU instruction)		Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.

2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N.

3. 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.

6.1.2 Exception Handling Operations

The exception handling sources are detected and begin processing according to the timing shown in table 6.2.

Table 6.2 Timing of Exception Source Detection and Start of Exception Handling

Exception	Source	Timing of Source Detection and Start of Handling
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high, when the H-UDI reset negate command is set after the H-UDI reset assert command has been set, or when the WDT overflows.
	Manual reset	Starts when the $\overline{\text{MRES}}$ pin changes from low to high or when the WDT overflows.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Register bank error	Bank underflow	Starts upon attempted execution of a RESBANK instruction when saving has not been performed to register banks.
	Bank overflow	In the state where saving has been performed to all register bank areas, starts when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby) anytime except immediately after a delayed branch instruction (delay slot).
	Slot illegal instructions	Starts from the decoding of undefined code placed immediately after a delayed branch instruction (delay slot) (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction.

Exception	Source	Timing of Source Detection and Start of Handling
Instructions	Integer division instructions	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by –1.
	FPU exception	Starts on detection of an invalid operation exception (as prescribed by the IEEE 754 standard), division-by-zero exception, overflow, underflow, or inexact exception. Also starts when qNaN or ±∞ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.

When exception handling starts, the CPU operates as follows:

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 6.1.3, Exception Handling Vector Table, for more information. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the interrupt controller (INTC) is also initialized to 0. Additionally, the values of FPSCR are initialized to H'00040001 for power-on resets. The program begins running from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI or user break with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, user break interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error or instruction, the I3 to I0 bits are not affected. The start address is then fetched from the exception handling vector table and the program begins running from that address.

6.1.3 Exception Handling Vector Table

Before exception handling begins running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address. Table 6.3 shows the vector numbers and vector table address offsets. Table 6.4 shows how vector table addresses are calculated.

Table 6.3 Exception Handling Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
	SP	1	H'00000004 to H'00000007
Manual reset	PC	2	H'00000008 to H'0000000B
	SP	3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Slot illegal instruction		6	H'00000018 to H'0000001B
(Reserved by system)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
DMAC/DTC address error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
FPU exception		13	H'00000034 to H'00000037
H-UDI		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043
Integer division exception (division by zero)		17	H'00000044 to H'00000047

Exception Sources	Vector Numbers	Vector Table Address Offset
Integer division exception (overflow)	18	H'00000048 to H'0000004B
(Reserved by system)	19	H'0000004C to H'0000004F
	:	:
	26	H'00000068 to H'0000006B
FCU interrupt (FIF)	27	H'0000006C to H'0000006F
(Reserved by system)	28	H'00000070 to H'00000073
	:	:
	31	H'0000007C to H'0000007F
Trap instruction (user vector)	32	H'00000080 to H'00000083
	:	:
	63	H'000000FC to H'000000FF
External interrupts (IRQ), on-chip peripheral module interrupts*	64	H'00000100 to H'00000103
	:	:
	511	H'000007FC to H'000007FF

Note: * The vector numbers and vector table address offsets for each external interrupt and on-chip peripheral module interrupt are given in table 7.4 in section 7, Interrupt Controller (INTC).

Table 6.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, register bank errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. Vector table address offset: See table 6.3.
2. Vector number: See table 6.3.

6.2 Resets

6.2.1 Input/Output Pins

Table 6.5 lists the configuration of reset related pins.

Table 6.5 Pin Configuration

Reset Name	Pin Name	Input/Output	Function
Power-on reset	$\overline{\text{RES}}$	Input	Low level input on this pin transits this LSI to power-on reset processing.
Manual reset	$\overline{\text{MRES}}$	Input	Low level input on this pin transits this LSI to manual reset processing.

6.2.2 Types of Reset

A reset is the highest-priority exception handling source. There are two kinds of reset, power-on and manual. As shown in table 6.6, the CPU state is initialized in both a power-on reset and a manual reset. The FPU state is initialized in a power-on reset, but not in a manual reset. On-chip peripheral module registers except some registers are initialized by a power-on reset, but not by a manual reset.

Table 6.6 Reset States

	Conditions for Transition to Reset States				Internal States	
Type	**$\overline{\text{RES}}$**	**H-UDI Command**	**$\overline{\text{MRES}}$**	**WDT Overflow**	**CPU**	**Modules other than CPU**
Power-on reset	Low	—	—	—	Initialized	Initialized
	High	H-UDI reset assert command is set	—	—	Initialized	Initialized
	High	Command other than H-UDI reset assert is set	—	Power-on reset	Initialized	*
	High	Return from deep software standby mode due to an interrupt			Initialized	*
Manual reset	High	Command other than H-UDI reset assert is set	Low	—	Initialized	*
	High	Command other than H-UDI reset assert is set	High	Manual reset	Initialized	*

Note: * See section 34.3, Register States in Each Operating Mode.

6.2.3 Power-On Reset

(1) Power-On Reset by Means of $\overline{\text{RES}}$ Pin

When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept at the low level for the duration of the oscillation settling time at power-on or when in software standby mode (when the clock is halted), and at least 20 t_{cyc} when the clock is running. In the power-on reset state, the internal state of the CPU and all the on-chip peripheral module registers are initialized. See appendix A, Pin States, for the status of individual pins during the power-on reset state.

In the power-on reset state, power-on reset exception handling starts when the $\overline{\text{RES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the INTC is also initialized to 0. Additionally, the values of FPSCR are initialized to H'00040001.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

Be sure to keep the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins at the low level over the duration of the oscillation settling time when the system power is turned on.

(2) Power-On Reset by Means of H-UDI Reset Assert Command

When the H-UDI reset assert command is set, this LSI enters the power-on reset state. Power-on reset by means of an H-UDI reset assert command is equivalent to power-on reset by means of the $\overline{\text{RES}}$ pin. Setting the H-UDI reset negate command cancels the power-on reset state. The time required between an H-UDI reset assert command and H-UDI reset negate command is the same as the time to keep the $\overline{\text{RES}}$ pin low to initiate a power-on reset. In the power-on reset state generated by an H-UDI reset assert command, setting the H-UDI reset negate command starts power-on reset exception handling. The CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

(3) Power-On Reset Initiated by the WDT

If the settings for the generation of a power-on reset by the WDT in its watchdog timer mode were made, this LSI enters the power-on reset state on an overflow of the WTCNT of the WDT.

At this time, the reset signal from the WDT does not initialize registers WRCSR of the WDT and FRQCR of the CPG.

Furthermore, when a reset due to the $\overline{\text{RES}}$ pin or H-UDI reset assert command and a reset due to an overflow of the WDT are generated at the same time, the reset due to the $\overline{\text{RES}}$ pin or H-UDI reset assert command takes priority, and the WOVF bit in WRCSR is cleared to 0. CPU operation when exception processing for a power-on reset due to the WDT has started is the same as when a power-on reset is applied through the $\overline{\text{RES}}$ pin.

(4) Power-on Reset When an Interrupt Leads to Return from Deep Software Standby Mode

A power-on reset is generated within this LSI when an interrupt leads to return from deep software standby mode. At this time, modules required for continued operation and registers for which the values must be retained are not initialized. For details, see section 32, Power-Down Modes and section 34, List of Registers.

Furthermore, if a reset due to the $\overline{\text{RES}}$ pin and a reset when an interrupt leads to return from deep software standby mode are generated at the same time, the reset due to the $\overline{\text{RES}}$ pin takes priority. When an interrupt leads to return from deep software standby mode, CPU operation once exception processing for the power-on reset has started is the same as when a power-on reset is applied through the $\overline{\text{RES}}$ pin.

6.2.4 Manual Reset

(1) Manual Reset by Means of $\overline{\text{MRES}}$ Pin

When the $\overline{\text{MRES}}$ pin is driven low, this LSI enters the manual reset state. To reset this LSI without fail, the $\overline{\text{MRES}}$ pin should be kept at the low level for at least 20 t_{cyc}. In the manual reset state, the CPU's internal state is initialized, but all the on-chip peripheral module registers are not initialized. In the manual reset state, manual reset exception handling starts when the $\overline{\text{MRES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

(2) Manual Reset Initiated by WDT

When a setting is made for a manual reset to be generated in the WDT's watchdog timer mode, and WTCNT of the WDT overflows, this LSI enters the manual reset state.

When manual reset exception processing is started by the WDT, the CPU operates in the same way as when a manual reset was caused by the $\overline{\text{MRES}}$ pin.

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs while the bus is released or during DMAC/DTC burst transfer, manual reset exception handling will be deferred until the CPU acquires the bus. However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than the internal manual reset interval cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception handling is not executed. A manual reset initializes the CPU and the BN bit in IBNR of the INTC but not the FPU and other modules.

6.3 Address Errors

6.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 6.7.

Table 6.7 Bus Cycles and Address Errors

Bus Cycle			
Type	**Bus Master**	**Bus Cycle Description**	**Address Errors**
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space	None (normal)
		Instruction fetched from on-chip peripheral module space	Address error occurs
		Instruction fetched from external memory space in single-chip mode	Address error occurs
Data read/write	CPU, DMAC or DTC	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space	None (normal)
		Longword data accessed in 16-bit on-chip peripheral module space	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space	None (normal)
		External memory space accessed when in single chip mode	Address error occurs

6.3.2 Address Error Exception Handling

When an address error occurs, the bus cycle in which the address error occurred ends*. When the executing instruction then finishes, address error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the address error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the start address for an exception service routine fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

Note: * In the case of an address error caused by instruction fetching when data is read or written, if the bus cycle on which the address error occurred is not completed by the end of the above operation, the CPU will recommence address error exception processing until the end of that bus cycle.

6.4 Register Bank Errors

6.4.1 Register Bank Error Sources

(1) Bank Overflow

In the state where saving has already been performed to all register bank areas, bank overflow occurs when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is set to 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

(2) Bank Underflow

Bank underflow occurs when an attempt is made to execute a RESBANK instruction while saving has not been performed to register banks.

6.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a bank overflow, and the start address of the executed RESBANK instruction for a bank underflow.

   To prevent multiple interrupts from occurring at a bank overflow, the interrupt priority level that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. After jumping to the start address for an exception service routine fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.5 Interrupts

6.5.1 Interrupt Sources

Table 6.8 shows the sources that start up interrupt exception handling. These are divided into NMI, user breaks, H-UDI, IRQ, and on-chip peripheral modules.

Table 6.8 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	User debugging interface (H-UDI)	1
FCU interrupt (FIF)	Flash sequencer	1
IRQ	IRQ0 to IRQ23 pins (external input)	24
On-chip peripheral module	32-kHz timer (TIM32C)	5
	Key scan controller (KEYC)	1
	A/D converter (ADC)	2
	Direct memory access controller (DMAC)	8
	Compare match timer (CMT)	2
	Compare match timer 2 (CMT2)	5
	Low power consumption functions	1
	Bus state controller (BSC)	1
	Renesas serial peripheral interface (RSPI)	3
	Watchdog timer (WDT)	1
	Multi-function timer pulse unit 2 (MTU2)	28
	Port output enable 2 (POE2)	3
	Multi-function timer pulse unit 2S (MTU2S)	13
	LVDS receive interface (LVDS) (SH72315A only)	4
	I^2C bus interface 3 (IIC3)	5
	Controller area network (RCAN-ET)	4
	Serial communication interface (SCI)	16

Each interrupt source is allocated a different vector number and vector table offset. See table 7.4 in section 7, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

6.5.2 Interrupt Priority Level

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller (INTC) determines their relative priorities and starts processing according to the results.

The priority order of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break and H-UDI interrupt priority level is 15. Priority levels of IRQ interrupts, and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01 to 04, 06, and 08 to 18 (IPR01 to IPR04, IPR06, and IPR08 to IPR18) of the INTC as shown in table 6.9. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 7.3.1, Interrupt Priority Registers 01 to 04, 06, 08 to 18 (IPR01 to IPR04, IPR06, IPR 08 to IPR18), for details of IPR01 to IPR04, IPR06, and IPR08 to IPR18.

Table 6.9 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
H-UDI	15	Fixed priority level.
FCU interrupt (FIF)	0 to 15	Set with the interrupt priority registers 01 to 04, 06, and 08 to 18 (IPR01 to IPR04, IPR06, and IPR 08 to IPR18)
IRQ		
On-chip peripheral module		

6.5.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, interrupt exception handling begins. In interrupt exception handling, the CPU fetches the exception service routine start address which corresponds to the accepted interrupt from the exception handling vector table, and saves SR and the program counter (PC) to the stack. In the case of interrupt exception handling other than NMI or user break with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved in the register banks. In the case of exception handling due to an address error, NMI interrupt, user break interrupt, or instruction, saving is not performed to the register banks. If saving has been performed to all register banks (0 to 14), automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception occurs. Next, the priority level value of the accepted interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, but the value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address of the interrupt exception service routine fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 7.6, Operation, for further details of interrupt exception handling.

6.6 Exceptions Triggered by Instructions

6.6.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, slot illegal instructions, general illegal instructions, integer division exceptions, and FPU exceptions, as shown in table 6.10.

Table 6.10 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	
Slot illegal instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.
General illegal instructions	Undefined code anywhere besides in a delay slot (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby)	
Integer division exceptions	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (−1)	DIVS
FPU exceptions	Starts when detecting invalid operation exception defined by IEEE754, division-by-zero exception, overflow, underflow, or inexact exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

6.6.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the vector number specified in the TRAPA instruction is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
4. After jumping to the start address of the exception service routine fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.6.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby), an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. Furthermore, floating point operation instructions and CPU instructions related to the FPU are treated as undefined codes while the FPU is on module standby. If such an instruction is placed in a delay slot, exception handling as a slot illegal instruction starts when it is decoded. At this time, the CPU operates as follows:

1. The exception service routine start address is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code, the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
4. After jumping to the start address of the exception service routine fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.6.4 General Illegal Instructions

When an undefined code (including FPU instructions and CPU instructions related to the FPU when the FPU module is on standby) placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) is decoded, general illegal instruction exception handling starts. Floating point operation instructions and CPU instructions related to the FPU are also treated as undefined codes when the FPU module is on standby. When these codes are placed anywhere other than immediately after a delayed branch instruction (i.e., other than in a delay slot), general illegal instruction exception handling starts after decoding of the codes. The CPU handles general illegal instruction exceptions in the same way as slot illegal instruction exceptions. Unlike in the exception processing of slot illegal instructions, however, the stored program counter value is the address where the undefined code starts.

6.6.5 Integer Division Exceptions

When an integer division instruction performs division by zero or the result of integer division overflows, integer division instruction exception handling starts. The instructions that may become the source of division-by-zero exception are DIVU and DIVS. The only source instruction of overflow exception is DIVS, and overflow exception occurs only when the negative maximum value is divided by −1. The CPU operates as follows:

1. The exception service routine start address which corresponds to the integer division instruction exception that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the integer division instruction at which the exception occurred.
4. After jumping the start address of the exception service routine fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.6.6 FPU Instructions

FPU exception handling is generated when the V, Z, O, U or I bit in the FPU enable field (Enable) of the floating point status/control register (FPSCR) is set. This indicates the occurrence of an invalid operation exception defined by the IEEE standard 754, a division-by-zero exception, overflow (in the case of an instruction for which this is possible), underflow (in the case of an instruction for which this is possible), or inexact exception (in the case of an instruction for which this is possible).

The floating-point operation instructions that may cause FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

FPU exception handling is generated only when the corresponding FPU exception enable bit (Enable) is set. When the FPU detects an exception source due to floating-point operations, FPU operation is suspended and generation of the FPU exception handling is reported to the CPU. When exception handling is started, the CPU operations are as follows.

1. The start address of the exception service routine corresponding to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not FPU exception handling has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time a floating-point operation instruction is executed.

When the V bit in the FPU exception enable field (Enable) of FPSCR is set and the QIS bit in FPSCR is also set, FPU exception handling is generated when qNaN or ±∞ is input to a floating point operation instruction source.

6.7 When Exception Sources are not Accepted

When an address error, an FPU exception, a register bank error (overflow), or an interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 6.11. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 6.11 Exception Source Generation Immediately after Delayed Branch Instruction

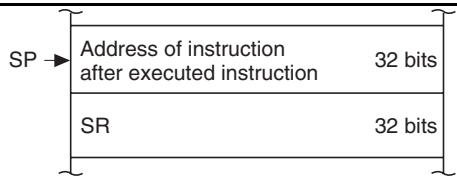
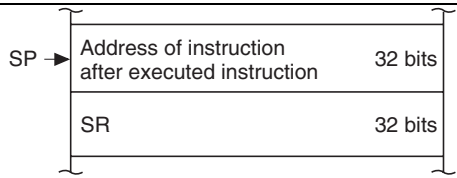
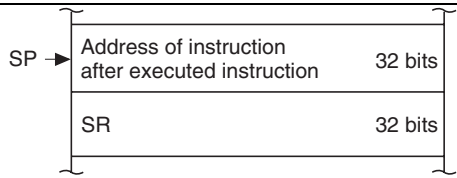
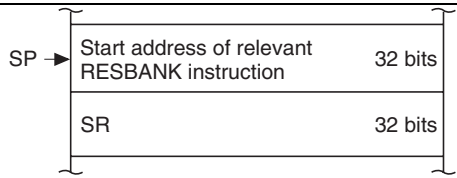
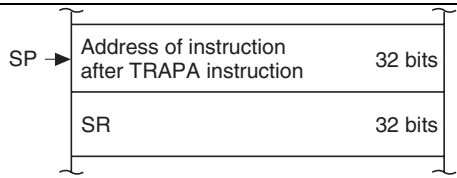
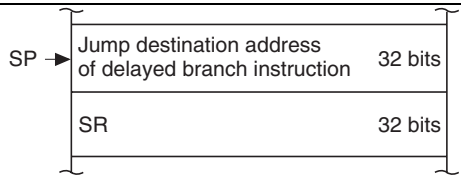
	Exception Source			
Point of Occurrence	**Address Error**	**FPU Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted

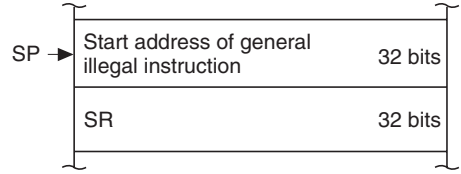
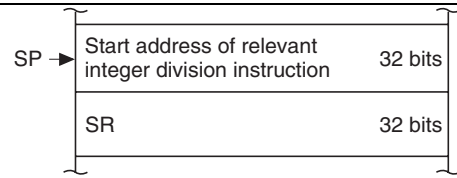
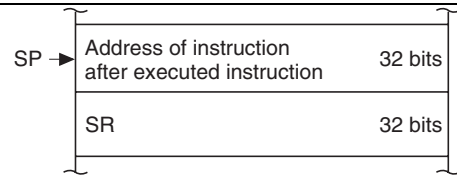
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

6.8 Stack Status after Exception Handling Ends

The status of the stack after exception handling ends is as shown in table 6.12.

Table 6.12 Stack Status after Exception Handling Ends

Exception Type	Stack Status
Address error	
Interrupt	
Register bank error (overflow)	
Register bank error (underflow)	
Trap instruction	
Slot illegal instruction	

Exception Type	Stack Status
General illegal instruction	
Integer division instruction	
FPU exception	

6.9 Usage Notes

6.9.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

6.9.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the vector table is accessed during exception handling.

6.9.3 Address Errors Caused by Stacking of Address Error Exception Handling

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception handling (interrupts, etc.) and address error exception handling will start up as soon as the first exception handling is ended. Address errors will then also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

6.9.4 Note on Changes to the Interrupt Mask Level (IMASK Bits) in the Status Register of the CPU

When using an LDC or LDC.L instruction to manipulate the interrupt mask level in the status register (SR) of the CPU, place five instructions between instructions to enable and disable interrupts whenever such control is applied. An example is given below.

[Example of the Measure to Apply in Programs]

```
LDC Rx, SR   \\interrupt enabled
NOP
NOP
NOP
NOP
NOP
LDC Ry, SR   \\interrupt disabled
```

Section 7 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

7.1 Features

- 16 levels of interrupt priority can be set

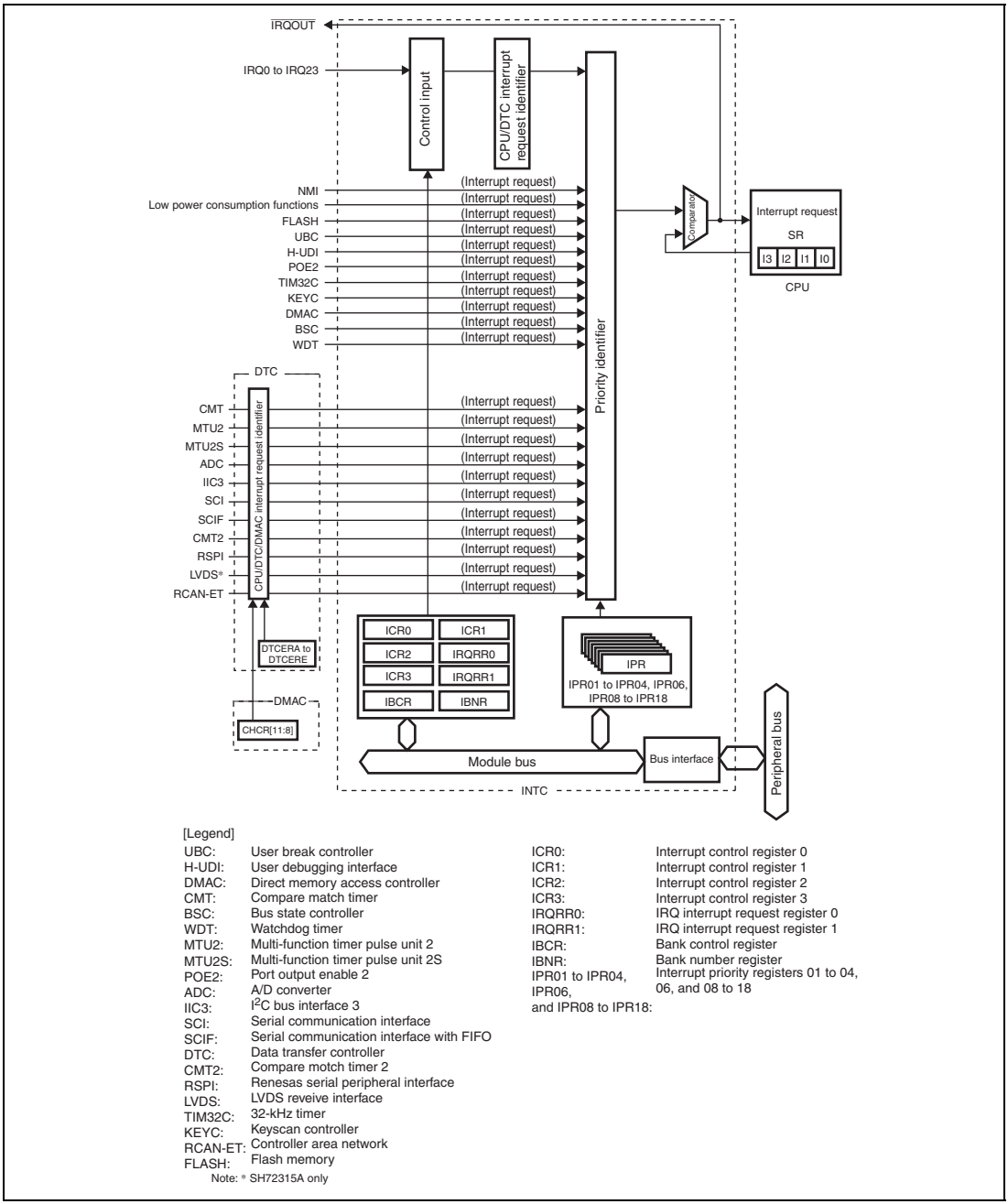
  By setting the 16 interrupt priority registers, the priority of IRQ interrupts and on-chip peripheral module interrupts can be selected from 16 levels for request sources.
- NMI noise canceler function

  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as the noise canceler function.
- Occurrence of interrupt can be reported externally ($\overline{\text{IRQOUT}}$ pin)

  For example, when this LSI has released the bus mastership, this LSI can inform the external bus master of occurrence of an on-chip peripheral module interrupt and request for the bus mastership.
- Register banks

  This LSI has register banks that enable register saving and restoration required in the interrupt processing to be performed at high speed.

Figure 7.1 shows a block diagram of the INTC.



[Legend]

UBC:	User break controller
H-UDI:	User debugging interface
DMAC:	Direct memory access controller
CMT:	Compare match timer
BSC:	Bus state controller
WDT:	Watchdog timer
MTU2:	Multi-function timer pulse unit 2
MTU2S:	Multi-function timer pulse unit 2S
POE2:	Port output enable 2
ADC:	A/D converter
IIC3:	I^2C bus interface 3
SCI:	Serial communication interface
SCIF:	Serial communication interface with FIFO
DTC:	Data transfer controller
CMT2:	Compare motch timer 2
RSPI:	Renesas serial peripheral interface
LVDS:	LVDS reveive interface
TIM32C:	32-kHz timer
KEYC:	Keyscan controller
RCAN-ET:	Controller area network
FLASH:	Flash memory

Note: * SH72315A only

ICR0:	Interrupt control register 0
ICR1:	Interrupt control register 1
ICR2:	Interrupt control register 2
ICR3:	Interrupt control register 3
IRQRR0:	IRQ interrupt request register 0
IRQRR1:	IRQ interrupt request register 1
IBCR:	Bank control register
IBNR:	Bank number register
IPR01 to IPR04, IPR06, and IPR08 to IPR18:	Interrupt priority registers 01 to 04, 06, and 08 to 18

Figure 7.1 Block Diagram of INTC

7.2 Input/Output Pins

Table 7.1 shows the pin configuration of the INTC.

Table 7.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
Interrupt request input pins	IRQ23 to IRQ0	Input	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	Output	Output of signal to report occurrence of interrupt source

7.3 Register Descriptions

The INTC has the following registers. These registers are used to set the interrupt priorities and control detection of the external interrupt input signal. For the states of these registers in each processing status, refer to section 34, List of Registers.

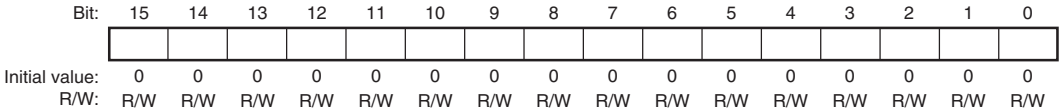
Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*	H'FFFE0800	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFE0802	16
Interrupt control register 2	ICR2	R/W	H'0000	H'FFFE0804	16, 32
Interrupt control register 3	ICR3	R/W	H'0000	H'FFFE0806	16
IRQ interrupt request register 0	IRQRR0	R/W	H'0000	H'FFFE0808	16, 32
IRQ interrupt request register 1	IRQRR1	R/W	H'0000	H'FFFE080A	16
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	16
Interrupt priority register 01	IPR01	R/W	H'0000	H'FFFE0818	16, 32
Interrupt priority register 02	IPR02	R/W	H'0000	H'FFFE081A	16
Interrupt priority register 03	IPR03	R/W	H'0000	H'FFFE081C	16, 32
Interrupt priority register 04	IPR04	R/W	H'0000	H'FFFE081E	16
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFE0C00	16
Interrupt priority register 08	IPR08	R/W	H'0000	H'FFFE0C04	16, 32
Interrupt priority register 09	IPR09	R/W	H'0000	H'FFFE0C06	16
Interrupt priority register 10	IPR10	R/W	H'0000	H'FFFE0C08	16, 32
Interrupt priority register 11	IPR11	R/W	H'0000	H'FFFE0C0A	16
Interrupt priority register 12	IPR12	R/W	H'0000	H'FFFE0C0C	16, 32
Interrupt priority register 13	IPR13	R/W	H'0000	H'FFFE0C0E	16
Interrupt priority register 14	IPR14	R/W	H'0000	H'FFFE0C10	16, 32
Interrupt priority register 15	IPR15	R/W	H'0000	H'FFFE0C12	16
Interrupt priority register 16	IPR16	R/W	H'0000	H'FFFE0C14	16, 32
Interrupt priority register 17	IPR17	R/W	H'0000	H'FFFE0C16	16
Interrupt priority register 18	IPR18	R/W	H'0000	H'FFFE0C18	16

Note: * When the NMI pin is high, becomes H'8000; when low, becomes H'0000.

7.3.1 Interrupt Priority Registers 01 to 04, 06, 08 to 18 (IPR01 to IPR04, IPR06, IPR08 to IPR18)

IPR01 to IPR04, IPR06, and IPR08 to IPR18 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts and on-chip peripheral module interrupts. Table 7.3 shows the correspondence between the interrupt request sources and the bits in IPR01 to IPR04, IPR06, and IPR08 to IPR18.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.3 Interrupt Request Sources and IPR01 to IPR04, IPR06, and IPR08 to IPR18

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 01	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register 02	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register 03	IRQ8 to IRQ15	IRQ16 to IRQ23	Low power consumption functions (SSRI)	FCU (FIF)
Interrupt priority register 04	ADI0	ADI1	Reserved	Reserved
Interrupt priority register 06	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 08	CMT0	CMT1	BSC	WDT
Interrupt priority register 09	MTU2_0 (TGI0A to TGI0D)	MTU2_0 (TCI0V, TGI0E, TGI0F)	MTU2_1 (TGI1A, TGI1B)	MTU2_1 (TCI1V, TCI1U)
Interrupt priority register 10	MTU2_2 (TGI2A, TGI2B)	MTU2_2 (TCI2V, TCI2U)	MTU2_3 (TGI3A to TGI3D)	MTU2_3 (TCI3V)
Interrupt priority register 11	MTU2_4 (TGI4A to TGI4D)	MTU2_4 (TCI4V)	MTU2_5 (TGI5U, TGI5V, TGI5W)	POE2
Interrupt priority register 12	MTU2S_3 (TGI3AS to TGI3DS)	MTU2S_3 (TCI3VS)	MTU2S_4 (TGI4AS to TGI4DS)	MTU2S_4 (TCI4VS)

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 13	MTU2S_5 (TGI5US, TGI5VS, TGI5WS)	CMT2 (CM2I)	IIC3	CMT2 (IC0I, IC1I, OC0I, OC1I)
Interrupt priority register 14	Reserved	Reserved	LVDS (SH72315A only)	Reserved
Interrupt priority register 15	TIM32C (CH0UF, CH0DF, CH1UF, CH1DF)	TIM32C (CH2F)	KEYC	Reserved
Interrupt priority register 16	SCI0	SCI1	SCI2	SCI3
Interrupt priority register 17	SCIF4	SCIF5	SCIF6	SCIF7
Interrupt priority register 18	Reserved	Reserved	RSPI	RCAN-ET

As shown in table 7.3, by setting the 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interrupt is set. Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (the highest level).

IPR01 to IPR04, IPR06, and IPR08 to IPR18 are initialized to H'0000 by a power-on reset.

7.3.2 Interrupt Control Register 0 (ICR0)

ICR0 is a 16-bit register that sets the input signal detection mode for the external interrupt input pin NMI, and indicates the input level at the NMI pin.

ICR0 except for the NMIL bit is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	-	-	-	-	-	-	NMIE	-	-	-	-	-	-	-	-
Initial value:	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Note: * 1 when the NMI pin is high, and 0 when the NMI pin is low.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*	R	NMI Input Level Sets the level of the signal input at the NMI pin. The NMI pin level can be obtained by reading this bit. This bit cannot be modified. 0: Low level is input to NMI pin 1: High level is input to NMI pin
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal on the NMI pin is detected. 0: Interrupt request is detected on falling edge of NMI input 1: Interrupt request is detected on rising edge of NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

7.3.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ7 to IRQ0 individually: low level, falling edge, rising edge, or both edges.

ICR1 is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select
14	IRQ70S	0	R/W	These bits select whether interrupt signals corresponding to pins IRQ7 to IRQ0 are detected by a low level, falling edge, rising edge, or both edges.
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	00: Interrupt request is detected on low level of IRQn input
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	01: Interrupt request is detected on falling edge of IRQn input
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	10: Interrupt request is detected on rising edge of IRQn input
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	11: Interrupt request is detected on both edges of IRQn input
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

[Legend]

n = 7 to 0

Note: When changing the condition for detection of an IRQn interrupt, IRQxF flag in register IRQRR0 is automatically cleared to 0.

7.3.4 Interrupt Control Register 2 (ICR2)

ICR2 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ15 to IRQ8 individually: low level, falling edge, rising edge, or both edges.

ICR2 is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ151S	IRQ150S	IRQ141S	IRQ140S	IRQ131S	IRQ130S	IRQ121S	IRQ120S	IRQ111S	IRQ110S	IRQ101S	IRQ100S	IRQ91S	IRQ90S	IRQ81S	IRQ80S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ151S	0	R/W	IRQ Sense Select
14	IRQ150S	0	R/W	These bits select whether interrupt signals corresponding to pins IRQ15 to IRQ8 are detected by a low level, falling edge, rising edge, or both edges.
13	IRQ141S	0	R/W	
12	IRQ140S	0	R/W	00: Interrupt request is detected on low level of IRQn input
11	IRQ131S	0	R/W	
10	IRQ130S	0	R/W	01: Interrupt request is detected on falling edge of IRQn input
9	IRQ121S	0	R/W	
8	IRQ120S	0	R/W	10: Interrupt request is detected on rising edge of IRQn input
7	IRQ111S	0	R/W	
6	IRQ110S	0	R/W	11: Interrupt request is detected on both edges of IRQn input
5	IRQ101S	0	R/W	
4	IRQ100S	0	R/W	
3	IRQ91S	0	R/W	
2	IRQ90S	0	R/W	
1	IRQ81S	0	R/W	
0	IRQ80S	0	R/W	

[Legend]

n = 15 to 8

Note: When changing the condition for detection of an IRQn interrupt, IRQxF flag in register IRQRR0 is automatically cleared to 0.

7.3.5 Interrupt Control Register 3 (ICR3)

ICR3 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ23 to IRQ16 individually: low level, falling edge, rising edge, or both edges.

ICR3 is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ231S	IRQ230S	IRQ221S	IRQ220S	IRQ211S	IRQ210S	IRQ201S	IRQ200S	IRQ191S	IRQ190S	IRQ181S	IRQ180S	IRQ171S	IRQ170S	IRQ161S	IRQ160S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ231S	0	R/W	IRQ Sense Select
14	IRQ230S	0	R/W	These bits select whether interrupt signals corresponding to pins IRQ23 to IRQ16 are detected by a low level, falling edge, rising edge, or both edges.
13	IRQ221S	0	R/W	
12	IRQ220S	0	R/W	00: Interrupt request is detected on low level of IRQn input
11	IRQ211S	0	R/W	
10	IRQ210S	0	R/W	01: Interrupt request is detected on falling edge of IRQn input
9	IRQ201S	0	R/W	
8	IRQ200S	0	R/W	10: Interrupt request is detected on rising edge of IRQn input
7	IRQ191S	0	R/W	
6	IRQ190S	0	R/W	11: Interrupt request is detected on both edges of IRQn input
5	IRQ181S	0	R/W	
4	IRQ180S	0	R/W	
3	IRQ171S	0	R/W	
2	IRQ170S	0	R/W	
1	IRQ161S	0	R/W	
0	IRQ160S	0	R/W	

[Legend]

n = 23 to 16

Note: When changing the condition for detection of an IRQn interrupt, IRQxF flag in register IRQRR0 is automatically cleared to 0.

7.3.6 IRQ Interrupt Request Register 0 (IRQRR0)

IRQRR0 is a 16-bit register that indicates interrupt requests from external input pins IRQ15 to IRQ0. If edge detection is set for the IRQ15 to IRQ0 interrupts, writing 0 to the IRQ15F to IRQ0F bits after reading IRQ15F to IRQ0F = 1 cancels the retained interrupts.

IRQRR0 is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15F	0	R/(W)*	IRQ Interrupt Request
14	IRQ14F	0	R/(W)*	These bits indicate the status of the IRQ15 to IRQ0 interrupt requests.
13	IRQ13F	0	R/(W)*	Level detection:
12	IRQ12F	0	R/(W)*	0: IRQn interrupt request has not occurred
11	IRQ11F	0	R/(W)*	[Clearing condition]
10	IRQ10F	0	R/(W)*	• IRQn input is high
9	IRQ9F	0	R/(W)*	1: IRQn interrupt has occurred
8	IRQ8F	0	R/(W)*	[Setting condition]
7	IRQ7F	0	R/(W)*	• IRQn input is low
6	IRQ6F	0	R/(W)*	Edge detection:
5	IRQ5F	0	R/(W)*	0: IRQn interrupt request is not detected
4	IRQ4F	0	R/(W)*	[Clearing conditions]
3	IRQ3F	0	R/(W)*	• Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF
2	IRQ2F	0	R/(W)*	• Cleared by executing IRQn interrupt exception handling
1	IRQ1F	0	R/(W)*	• The settings of the IRQn1S and IRQn0S bits in ICR1 and ICR2 are changed
0	IRQ0F	0	R/(W)*	1: IRQn interrupt request is detected
				[Setting condition]
				• Edge corresponding to IRQn1S or IRQn0S of ICR1 or ICR2 has occurred at IRQn pin

[Legend]

n = 15 to 0

7.3.7 IRQ Interrupt Request Register 1 (IRQRR1)

IRQRR1 is a 16-bit register that indicates interrupt requests from external input pins IRQ23 to IRQ16. If edge detection is set for the IRQ23 to IRQ16 interrupts, writing 0 to the IRQ23F to IRQ16F bits after reading IRQ23F to IRQ16F = 1 cancels the retained interrupts.

IRQRR1 is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ23F	IRQ22F	IRQ21F	IRQ20F	IRQ19F	IRQ18F	IRQ17F	IRQ16F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	IRQ23F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ23 to IRQ16 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred [Clearing condition] • IRQn input is high 1: IRQn interrupt request has occurred [Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected [Clearing conditions] • Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF • Cleared by executing IRQn interrupt exception handling • The settings of the IRQn1S and IRQn0S bits in ICR3 are changed 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR3 has occurred at IRQn pin
6	IRQ22F	0	R/(W)*	
5	IRQ21F	0	R/(W)*	
4	IRQ20F	0	R/(W)*	
3	IRQ19F	0	R/(W)*	
2	IRQ18F	0	R/(W)*	
1	IRQ17F	0	R/(W)*	
0	IRQ16F	0	R/(W)*	

7.3.8 Bank Control Register (IBCR)

IBCR is a 16-bit register that enables or disables use of register banks for each interrupt priority level.

IBCR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15	E15	0	R/W	Enable
14	E14	0	R/W	These bits enable or disable use of register banks for interrupt priority levels 15 to 1. However, use of register banks is always disabled for the user break interrupts. 0: Use of register banks is disabled 1: Use of register banks is enabled
13	E13	0	R/W	
12	E12	0	R/W	
11	E11	0	R/W	
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

7.3.9 Bank Number Register (IBNR)

IBNR is a 16-bit register that enables or disables use of register banks and register bank overflow exception. IBNR also indicates the bank number to which saving is performed next through the bits BN3 to BN0.

IBNR is initialized to H'0000 by a power-on reset. Additionally, the BN[3:0] bits are initialized by a manual reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BE[1:0]		BOVE	-	-	-	-	-	-	-	-	-	BN[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BE[1:0]	00	R/W	Register Bank Enable These bits enable or disable use of register banks. 00: Use of register banks is disabled for all interrupts. The setting of IBCR is ignored. 01: Use of register banks is enabled for all interrupts except NMI and user break. The setting of IBCR is ignored. 10: Reserved (setting prohibited) 11: Use of register banks is controlled by the setting of IBCR.
13	BOVE	0	R/W	Register Bank Overflow Enable Enables of disables register bank overflow exception. 0: Generation of register bank overflow exception is disabled 1: Generation of register bank overflow exception is enabled

Bit	Bit Name	Initial Value	R/W	Description
12 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	BN[3:0]	0000	R	Bank Number These bits indicate the bank number to which saving is performed next. When an interrupt using register banks is accepted, saving is performed to the register bank indicated by these bits, and BN is incremented by 1. After BN is decremented by 1 due to execution of a RESBANK (restore from register bank) instruction, restoration from the register bank is performed.

7.4 Interrupt Sources

There are five types of interrupt sources: NMI, user break, H-UDI, IRQ, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 0 the lowest and 16 the highest. When set to level 0, that interrupt is masked at all times.

7.4.1 NMI Interrupt

The NMI interrupt has a priority level of 16 and is accepted at all times. NMI interrupt requests are edge-detected, and the NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) selects whether the rising edge or falling edge is detected.

Though the priority level of the NMI interrupt is 16, the NMI interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15.

7.4.2 User Break Interrupt

A user break interrupt which occurs when a break condition set in the user break controller (UBC) matches has a priority level of 15. The user break interrupt exception handling sets the I3 to I0 bits in SR to level 15. For user break interrupts, see section 8, User Break Controller (UBC).

7.4.3 H-UDI Interrupt

The user debugging interface (H-UDI) interrupt has a priority level of 15, and occurs at serial input of an H-UDI interrupt instruction. H-UDI interrupt requests are edge-detected and retained until they are accepted. The H-UDI interrupt exception handling sets the I3 to I0 bits in SR to level 15. For H-UDI interrupts, see section 33, User Debugging Interface (H-UDI).

7.4.4 IRQ Interrupts

IRQ interrupts are input from pins IRQ23 to IRQ0. For the IRQ interrupts, low-level, falling-edge, rising-edge, or both-edge detection can be selected individually for each pin by the IRQ sense select bits (IRQ71S to IRQ01S and IRQ70S to IRQ00S) in the interrupt control register 1 (ICR1), the IRQ sense select bits (IRQ151S to IRQ81S and IRQ150S to IRQ80S) in the interrupt control register 2 (ICR2), and the IRQ sense select bits (IRQ231S to IRQ161S and IRQ230S to IRQ160S) in the interrupt control register 3 (ICR3). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01, 02, and 03 (IPR01, IPR02, and IPR03).

When using low-level setting for IRQ interrupts, an interrupt request signal is sent to the INTC while the IRQ23 to IRQ0 pins are low. An interrupt request signal is stopped being sent to the INTC when the IRQ23 to IRQ0 pins are driven high. The status of the interrupt requests can be checked by reading the IRQ interrupt request bits (IRQ15F to IRQ0F) in the IRQ interrupt request register 1 (IRQRR1) or the IRQ interrupt request bits (IRQ23F to IRQ16F) in the IRQ interrupt request register 2 (IRQRR2).

When using edge-sensing for IRQ interrupts, an interrupt request is detected due to change of the IRQ23 to IRQ0 pin states, and an interrupt request signal is sent to the INTC. The result of IRQ interrupt request detection is retained until that interrupt request is accepted. Whether IRQ interrupt requests have been detected or not can be checked by reading the IRQ15F to IRQ0F bits in IRQRR1 or the IRQ23F to IRQ16F bits in IRQRR2. Writing 0 to these bits after reading them as 1 clears the result of IRQ interrupt request detection.

The IRQ interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted IRQ interrupt.

When execution is to return from an IRQ interrupt exception service routine, confirm that the interrupt request bit in the IRQ interrupt request register 0 (IRQRR0) or the IRQ interrupt request register 1 (IRQRR1) has been cleared so that the request is not accepted again accidentally, and only then execute the RTE instruction.

7.4.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following on-chip peripheral modules:

- A/D converter (ADC)
- Flash-memory related (ROM/FLD)
- Direct memory access controller (DMAC)
- Compare match timer (CMT)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Multi-function timer pulse unit 2 (MTU2)
- Port output enable 2 (POE2)
- Multi-function timer pulse unit 2S (MTU2S)
- Compare match timer 2 (CMT2)
- I^2C bus interface 3 (IIC3)
- Renesas serial peripheral interface (RSPI)
- LVDS receive interface (LVDS) (SH72315A only)
- Controller area network (RCAN-ET)
- 32-kHz timer (TIM32C)
- Key scan controller (KEYC)
- Serial communication interface (SCI)
- Serial communication interface with FIFO (SCIF)
- Low power consumption functions

As every source is assigned a different interrupt vector, the sources need not be identified in the exception service routines except in the case of return from software standby due to a TIM32C or KEYC interrupt. However, return from software standby due to a TIM32C or KEYC interrupt is treated in common as return due to an SSRI interrupt, so the various flags must be checked to identify the source that led to return. Regarding the respective flags, refer to sections 27.3.2, Timer 32 Status Register (TI32SR), 28.3.5, Key Scan Status Register (KSSR), and 32.2.13, Standby Interrupt Flag Register (SIFR).

A priority level in the range from 0 to 15 can be set for each module in the interrupt priority registers 04, 06, and 08 to 18 (IPR04, IPR06, and IPR08 to IPR18). The I3 to I0 bits in the SR are set to the priority level of an accepted on-chip peripheral module interrupt in exception handling for the on-chip peripheral module interrupt.

7.5 Interrupt Exception Handling Vector Table and Priority

Table 7.4 lists interrupt sources and their vector numbers, vector table address offsets, and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from the vector numbers and vector table address offsets. In interrupt exception handling, the interrupt exception service routine start address is fetched from the vector table indicated by the vector table address. For details of calculation of the vector table address, see table 6.4 in section 6, Exception Handling.

The priorities of IRQ interrupts and on-chip peripheral module interrupts can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers 01 to 04, 06, and 08 to 18 (IPR01 to IPR04, IPR06, and IPR08 to IPR18). However, if two or more interrupts specified by the same IPR among IPR01 to IPR04, IPR06, and IPR08 to IPR18 occur, the priorities are defined as shown in the IPR setting unit internal priority of table 7.4, and the priorities cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priorities indicated in table 7.4.

Table 7.4 Interrupt Exception Handling Vectors and Priorities

Interrupt Source Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
NMI		11	H'0000002C to H'0000002F	16	—	—	High
UBC		12	H'00000030 to H'00000033	15	—	—	
H-UDI		14	H'00000038 to H'0000003B	15	—	—	
FCU (FIF)		27	H'0000006C to H'0000006F	0 to 15 (0)	IPR03 (3 to 0)	—	
IRQ	IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPR01 (15 to 12)	—	
	IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPR01 (11 to 8)	—	
	IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPR01 (7 to 4)	—	
	IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPR01 (3 to 0)	—	
	IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPR02 (15 to 12)	—	
	IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPR02 (11 to 8)	—	
	IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPR02 (7 to 4)	—	
	IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPR02 (3 to 0)	—	Low

Interrupt	Source Number	Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
		Vector	Vector Table Address Offset				
IRQ	IRQ8	72	H'00000120 to H'00000123	0 to 15 (0)	IPR03 (15 to 12)	1	High
	IRQ9	73	H'00000124 to H'00000127			2	
	IRQ10	74	H'00000128 to H'0000012B			3	
	IRQ11	75	H'0000012C to H'0000012F			4	
	IRQ12	76	H'00000130 to H'00000133			5	
	IRQ13	77	H'00000134 to H'00000137			6	
	IRQ14	78	H'00000138 to H'0000013B			7	
	IRQ15	79	H'0000013C to H'0000013F			8	
	IRQ16	80	H'00000140 to H'00000143	0 to 15 (0)	IPR03 (11 to 8)	1	
	IRQ17	81	H'00000144 to H'00000147			2	
	IRQ18	82	H'00000148 to H'0000014B			3	
	IRQ19	83	H'0000014C to H'0000014F			4	
	IRQ20	84	H'00000150 to H'00000153			5	
	IRQ21	85	H'00000154 to H'00000157			6	
	IRQ22	86	H'00000158 to H'0000015B			7	
	IRQ23	87	H'0000015C to H'0000015F			8	Low

Interrupt Source		Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
TIM32C	CH0D		88	H'00000160 to H'00000163	0 to 15 (0)	IPR15 (15 to 12)	1	High
	CH0U		89	H'00000164 to H'00000167			2	
	CH1D		90	H'00000168 to H'0000016B			3	
	CH1U		91	H'0000016C to H'0000016F			4	
	CH2		92	H'00000170 to H'00000173	0 to 15 (0)	IPR15 (11 to 8)	—	
KEYC	KSI		95	H'0000017C to H'0000017F	0 to 15 (0)	IPR15 (7 to 4)	—	
ADC	ADI0		104	H'000001A0 to H'000001A3	0 to 15 (0)	IPR04 (15 to 12)	—	
	ADI1		105	H'000001A4 to H'000001A7	0 to 15 (0)	IPR04 (11 to 8)	—	
DMAC	DMAC0	DEI0	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR06 (15 to 12)	1	
		HEI0	109	H'000001B4 to H'000001B7			2	
	DMAC1	DEI1	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPR06 (11 to 8)	1	
		HEI1	113	H'000001C4 to H'000001C7			2	
	DMAC2	DEI2	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR06 (7 to 4)	1	
		HEI2	117	H'000001D4 to H'000001D7			2	
	DMAC3	DEI3	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR06 (3 to 0)	1	
		HEI3	121	H'000001E4 to H'000001E7			2	Low

Interrupt Source		Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SCIF	SCIF4	BRI4	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR17 (15 to 12)	1	High
		ERI4	125	H'000001F4 to H'000001F7			2	
		RXI4	126	H'000001F8 to H'000001FB			3	
		TXI4	127	H'000001FC to H'000001FF			4	
	SCIF5	BRI5	128	H'00000200 to H'00000203	0 to 15 (0)	IPR17 (11 to 8)	1	
		ERI5	129	H'00000204 to H'00000207			2	
		RXI5	130	H'00000208 to H'0000020B			3	
		TXI5	131	H'0000020C to H'0000020F			4	
	SCIF6	BRI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR17 (7 to 4)	1	
		ERI6	133	H'00000214 to H'00000217			2	
		RXI6	134	H'00000218 to H'0000021B			3	
		TXI6	135	H'0000021C to H'0000021F			4	
	SCIF7	BRI7	136	H'00000220 to H'00000223	0 to 15 (0)	IPR17 (3 to 0)	1	
		ERI7	137	H'00000224 to H'00000227			2	
		RXI7	138	H'00000228 to H'0000022B			3	
		TXI7	139	H'0000022C to H'0000022F			4	Low

Interrupt Source	Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
CMT	CMI0	140	H'00000230 to H'00000233	0 to 15 (0)	IPR08 (15 to 12)	—	High
	CMI1	141	H'00000234 to H'00000237	0 to 15 (0)	IPR08 (11 to 8)	—	
CMT2	CM2I	142	H'00000238 to H'0000023B	0 to 15 (0)	IPR13 (11 to 8)	—	
	IC0I	143	H'0000023C to H'0000023F	0 to 15 (0)	IPR13 (3 to 0)	1	
	IC1I	144	H'00000240 to H'00000243			2	
	OC0I	145	H'00000244 to H'00000247			3	
	OC1I	146	H'00000248 to H'0000024B			4	
Low power consumption	SSRI	147	H'0000024C to H'0000024F	0 to 15 (0)	IPR03 (7 to 4)	—	
BSC	CMI	148	H'00000250 to H'00000253	0 to 15 (0)	IPR08 (7 to 4)	—	
RSPI	SPEI_0	149	H'00000254 to H'00000257	0 to 15 (0)	IPR18 (7 to 4)	1	
	SPRXI_0	150	H'00000258 to H'0000025B			2	
	SPTXI_0	151	H'0000025C to H'0000025F			3	
WDT	ITI	152	H'00000260 to H'00000263	0 to 15 (0)	IPR08 (3 to 0)	—	Low

Interrupt Source	Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU2_0	TGIA_0	156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (15 to 12)	1	High
		TGIB_0	157	H'00000274 to H'00000277			2	
		TGIC_0	158	H'00000278 to H'0000027B			3	
		TGID_0	159	H'0000027C to H'0000027F			4	
		TCIV_0	160	H'00000280 to H'00000283	0 to 15 (0)	IPR09 (11 to 8)	1	
		TGIE_0	161	H'00000284 to H'00000287			2	
		TGIF_0	162	H'00000288 to H'0000028B			3	
	MTU2_1	TGIA_1	164	H'00000290 to H'00000293	0 to 15 (0)	IPR09 (7 to 4)	1	
		TGIB_1	165	H'00000294 to H'00000297			2	
		TCIV_1	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPR09 (3 to 0)	1	
		TCIU_1	169	H'000002A4 to H'000002A7			2	
	MTU2_2	TGIA_2	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR10 (15 to 12)	1	
		TGIB_2	173	H'000002B4 to H'000002B7			2	
		TCIV_2	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPR10 (11 to 8)	1	
		TCIU_2	177	H'000002C4 to H'000002C7			2	Low

Interrupt Source	Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU2_3	TGIA_3	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR10 (7 to 4)	1	High
		TGIB_3	181	H'000002D4 to H'000002D7			2	
		TGIC_3	182	H'000002D8 to H'000002DB			3	
		TGID_3	183	H'000002DC to H'000002DF			4	
		TCIV_3	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR10 (3 to 0)	—	
	MTU2_4	TGIA_4	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR11 (15 to 12)	1	
		TGIB_4	189	H'000002F4 to H'000002F7			2	
		TGIC_4	190	H'000002F8 to H'000002FB			3	
		TGID_4	191	H'000002FC to H'000002FF			4	
		TCIV_4	192	H'00000300 to H'00000303	0 to 15 (0)	IPR11 (11 to 8)	—	
	MTU2_5	TGIU_5	196	H'00000310 to H'00000313	0 to 15 (0)	IPR11 (7 to 4)	1	
		TGIV_5	197	H'00000314 to H'00000317			2	
		TGIW_5	198	H'00000318 to H'0000031B			3	
POE2	OEI1		200	H'00000320 to H'00000323	0 to 15 (0)	IPR11 (3 to 0)	1	
	OEI2		201	H'00000324 to H'00000327			2	
	OEI3		202	H'00000328 to H'0000032B			3	Low

Interrupt Source	Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2S	MTU2S_3	TGIA_3S	204	H'00000330 to H'00000333	0 to 15 (0)	IPR12 (15 to 12)	1	High
		TGIB_3S	205	H'00000334 to H'00000337			2	
		TGIC_3S	206	H'00000338 to H'0000033B			3	
		TGID_3S	207	H'0000033C to H'0000033F			4	
		TCIV_3S	208	H'00000340 to H'00000343	0 to 15 (0)	IPR12 (11 to 8)	—	
	MTU2S_4	TGIA_4S	212	H'00000350 to H'00000353	0 to 15 (0)	IPR12 (7 to 4)	1	
		TGIB_4S	213	H'00000354 to H'00000357			2	
		TGIC_4S	214	H'00000358 to H'0000035B			3	
		TGID_4S	215	H'0000035C to H'0000035F			4	
		TCIV_4S	216	H'00000360 to H'00000363	0 to 15 (0)	IPR12 (3 to 0)	—	
	MTU2S_5	TGIU_5S	220	H'00000370 to H'00000373	0 to 15 (0)	IPR13 (15 to 12)	1	
		TGIV_5S	221	H'00000374 to H'00000377			2	
		TGIW_5S	222	H'00000378 to H'0000037B			3	
LVDS (SH72315A only)	LVOREI		224	H'00000380 to H'00000383	0 to 15 (0)	IPR14 (7 to 4)	1	
	LVUREI		225	H'00000384 to H'00000387			2	
	LVRXI		226	H'00000388 to H'0000038B			3	
	LVECI		227	H'0000038C to H'0000038F			4	Low

Interrupt Source Number			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
IIC3	IISTPI		228	H'00000390 to H'00000393	0 to 15 (0)	IPR13 (7 to 4)	1	High
	IINAKI		229	H'00000394 to H'00000397			2	
	IIRXI		230	H'00000398 to H'0000039B			3	
	IITXI		231	H'0000039C to H'0000039F			4	
	IITEI		232	H'000003A0 to H'000003A3			5	
RCAN-ET	ERS_0		236	H'000003B0 to H'000003B3	0 to 15 (0)	IPR18 (3 to 0)	1	
	OVR_0		237	H'000003B4 to H'000003B7			2	
	RM_0, RM_1		238	H'000003B8 to H'000003BB			3	
	SLE_0		239	H'000003BC to H'000003BF			4	
SCI	SCI0	ERI0	240	H'000003C0 to H'000003C3	0 to 15 (0)	IPR16 (15 to 12)	1	
		RXI0	241	H'000003C4 to H'000003C7			2	
		TXI0	242	H'000003C8 to H'000003CB			3	
		TEI0	243	H'000003CC to H'000003CF			4	
	SCI1	ERI1	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR16 (11 to 8)	1	
		RXI1	245	H'000003D4 to H'000003D7			2	
		TXI1	246	H'000003D8 to H'000003DB			3	
		TEI1	247	H'000003DC to H'000003DF			4	Low

Interrupt Source		Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SCI	SCI2	ERI2	248	H'000003E0 to H'000003E3	0 to 15 (0)	IPR16 (7 to 4)	1	High
		RXI2	249	H'000003E4 to H'000003E7			2	
		TXI2	250	H'000003E8 to H'000003EB			3	
		TEI2	251	H'000003EC to H'000003EF			4	
	SCI3	ERI3	252	H'000003F0 to H'000003F3	0 to 15 (0)	IPR16 (3 to 0)	1	
		RXI3	253	H'000003F4 to H'000003F7			2	
		TXI3	254	H'000003F8 to H'000003FB			3	
		TEI3	255	H'000003FC to H'000003FF			4	Low

7.6 Operation

7.6.1 Interrupt Operation Sequence

The sequence of interrupt operations is described below. Figure 7.2 shows the operation flow.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers 01 to 04, 06, and 08 to 18 (IPR01 to IPR04, IPR06, and IPR08 to IPR18). Lower priority interrupts are ignored*. If two of more interrupts have the same priority level or if multiple interrupts occur within a single IPR, the interrupt with the highest priority is selected, according to the default priority and IPR setting unit internal priority shown in table 7.4.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt level mask bits (I3 to I0) in the status register (SR) of the CPU. If the interrupt request priority level is equal to or less than the level set in bits I3 to I0, the interrupt request is ignored. If the interrupt request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU detects the interrupt request sent from the interrupt controller when the CPU decodes the instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (figure 7.4).
6. The interrupt exception service routine start address is fetched from the exception handling vector table corresponding to the accepted interrupt.
7. The status register (SR) is saved onto the stack, and the priority level of the accepted interrupt is copied to bits I3 to I0 in SR.
8. The program counter (PC) is saved onto the stack.
9. The CPU jumps to the fetched interrupt exception service routine start address and starts executing the program. The jump that occurs is not a delayed branch.
10. A high level is output from the $\overline{\text{IRQOUT}}$ pin. However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just being accepted, the $\overline{\text{IRQOUT}}$ pin holds low level.

Notes: The interrupt source flag should be cleared in the interrupt handler. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" shown in table 7.5 is required before the interrupt source sent to the CPU is actually cancelled. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then execute an RTE instruction.

* Interrupt requests that are designated as edge-sensing are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ interrupt request registers 0 and 1 (IRQRR0 and IRQRR1). For details, see section 7.4.4, IRQ Interrupts.
  Interrupts held pending due to edge-sensing are cleared by a power-on reset.

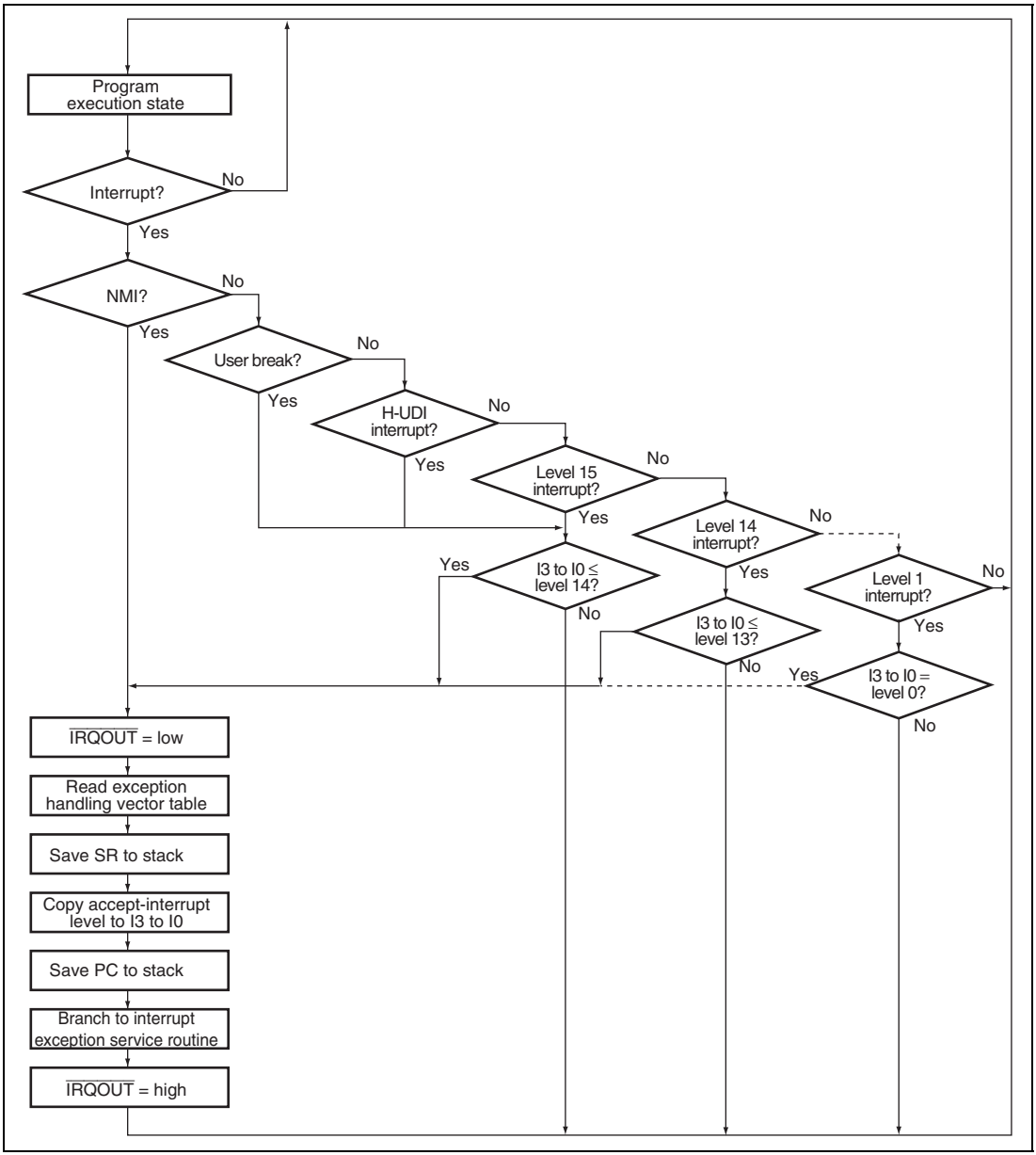
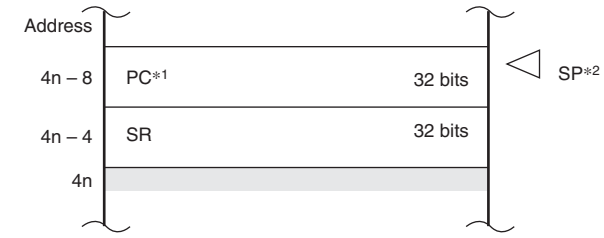


Figure 7.2 Interrupt Operation Flow

7.6.2 Stack after Interrupt Exception Handling

Figure 7.3 shows the stack after interrupt exception handling.



Notes: 1. PC: Start address of the next instruction (return destination instruction) after the executed instruction
2. Always make sure that SP is a multiple of 4.

Figure 7.3 Stack after Interrupt Exception Handling

7.7 Interrupt Response Time

Table 7.5 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction in the interrupt exception service routine begins. The interrupt processing operations differ in the cases when banking is disabled, when banking is enabled without register bank overflow, and when banking is enabled with register bank overflow. Figures 7.4 and 7.5 show examples of pipeline operation when banking is disabled. Figures 7.6 and 7.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 7.8 and 7.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Table 7.5 Interrupt Response Time

			Number of States					
Item			**NMI**	**User Break**	**H-UDI**	**IRQ**	**Peripheral Module**	**Remarks**
Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU			2 Icyc + 2 Bcyc + 1 Pcyc	3 Icyc	2 Icyc + 1 Pcyc	2 Icyc + 3 Bcyc + 1 Pcyc	2 Icyc + 1 Bcyc + 2 Pcyc	Interrupts with the DTC activation sources
							2 Icyc + 1 Bcyc + 1 Bcyc	Interrupts without the DTC activation sources.
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in exception service routine is fetched	No register banking	Min.	3 Icyc + m1 + m2					Min. is when the interrupt wait time is zero. Max. is when a higher-priority interrupt request has occurred during interrupt exception handling.
		Max.	4 Icyc + 2(m1 + m2) + m3					
	Register banking without register bank overflow	Min.	—	—	3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	—	12 Icyc + m1 + m2			
	Register banking with register bank overflow	Min.	—	—	3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	—	3 Icyc + m1 + m2 + 19(m4)			

Item			Number of States					Remarks
			NMI	User Break	H-UDI	IRQ	Peripheral Module	
Interrupt response time	No register banking	Min.	5 Icyc + 2 Bcyc +1 Pcyc + m1 + m2	6 Icyc + m1 + m2	5 Icyc + 1Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	100-MHz operation*1*2: 0.080 to 0.150 µs
		Max.	6 Icyc + 2 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	7 Icyc + 2(m1 + m2) + m3	6 Icyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 3 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 1 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	100-MHz operation*1*2: 0.120 to 0.190 µs
	Register banking without register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	100-MHz operation*1*2: 0.090 to 0.150 µs
		Max.	—	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	100-MHz operation*1*2: 0.180 to 0.240 µs
	Register banking with register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	100-MHz operation*1*2: 0.090 to 0.150 µs
		Max.	—	—	5 Icyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	100-MHz operation*1*2: 0.280 to 0.340 µs

Notes: m1 to m4 are the number of states needed for the following memory accesses.

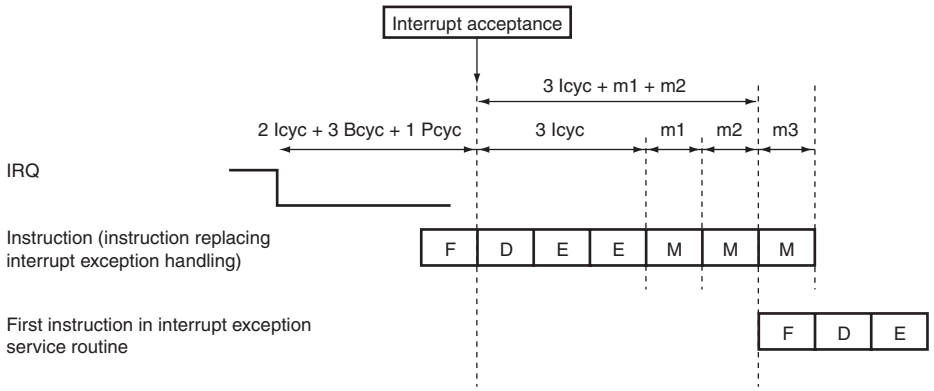
m1: Vector address read (longword read)

m2: SR save (longword write)

m3: PC save (longword write)

m4: Banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack.

1. In the case that m1 = m2 = m3 = m4 = 1 Icyc.
2. In the case that (Iϕ, Bϕ, Pϕ) = (100 MHz, 50 MHz, 50 MHz).



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
F: Instruction fetch. Instruction is fetched from memory in which program is stored.
D: Instruction decoding. Fetched instruction is decoded.
E: Instruction execution. Data operation or address calculation is performed in accordance with the result of decoding.
M: Memory access. Memory data access is performed.

Figure 7.4 Example of Pipeline Operation when IRQ Interrupt is Accepted (No Register Banking)

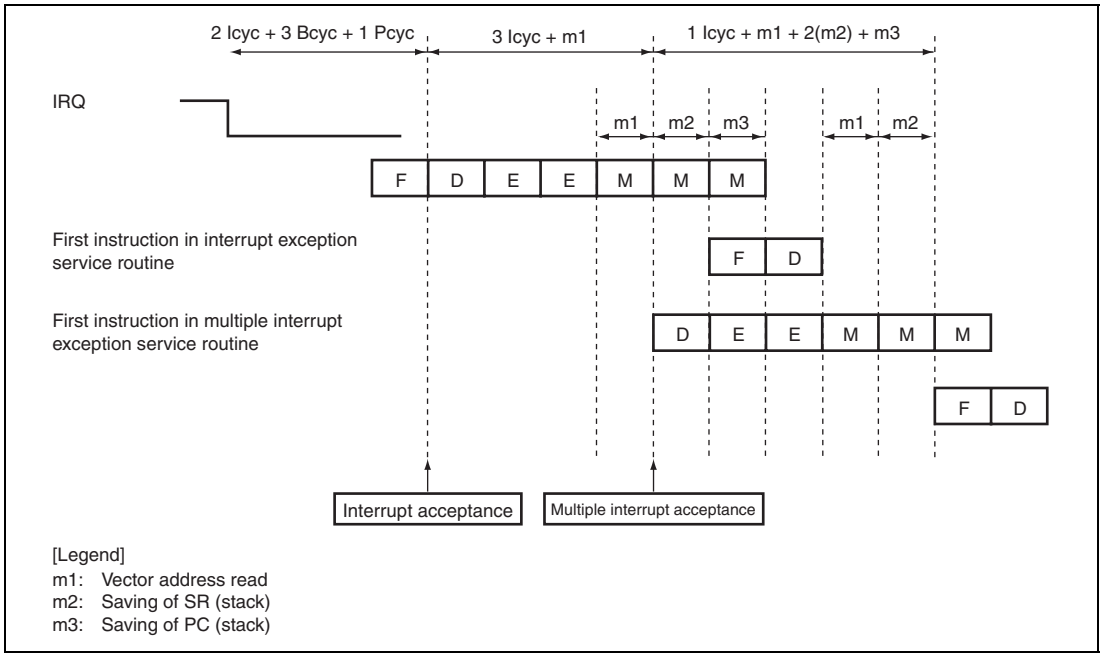


Figure 7.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

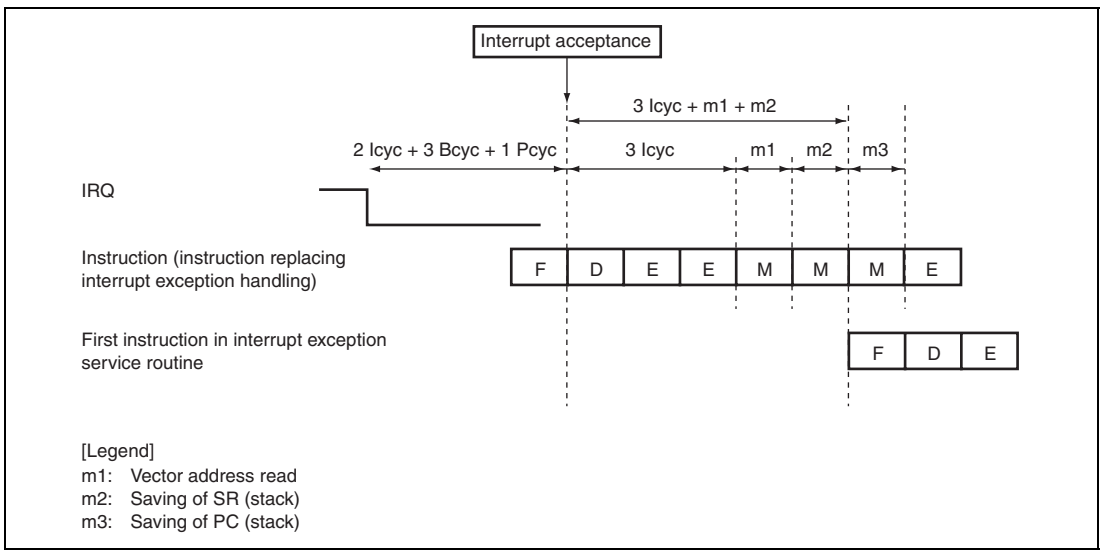


Figure 7.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

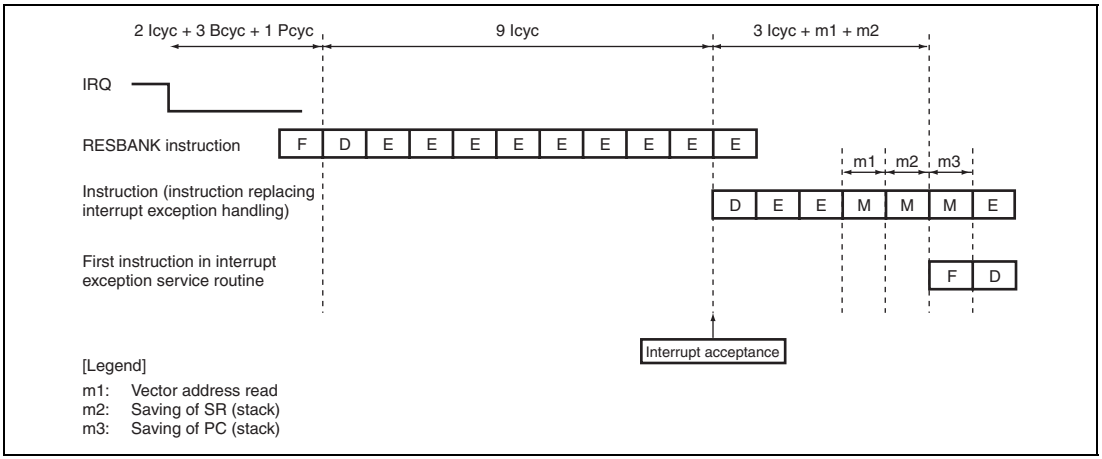


Figure 7.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

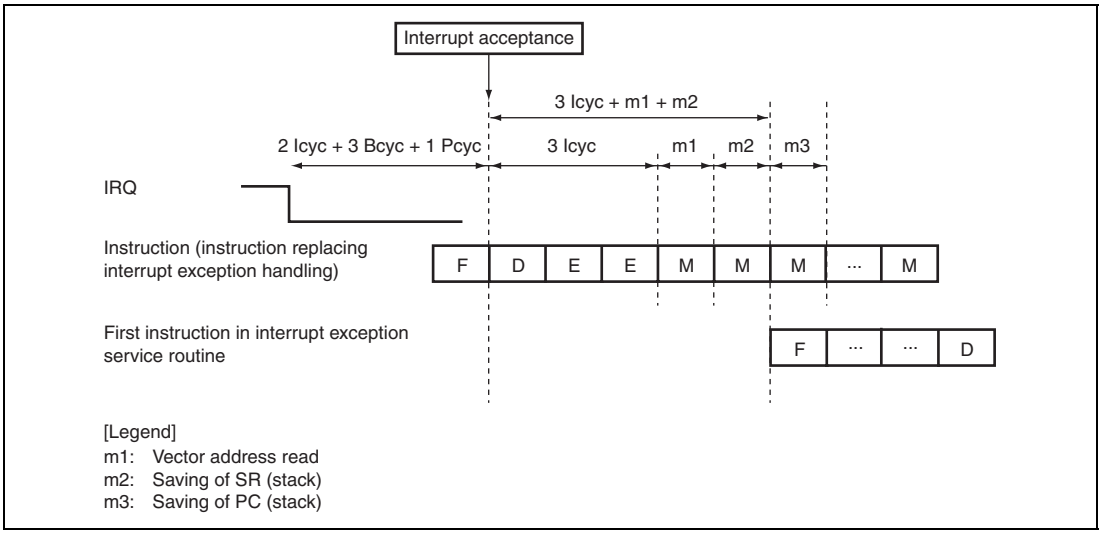


Figure 7.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

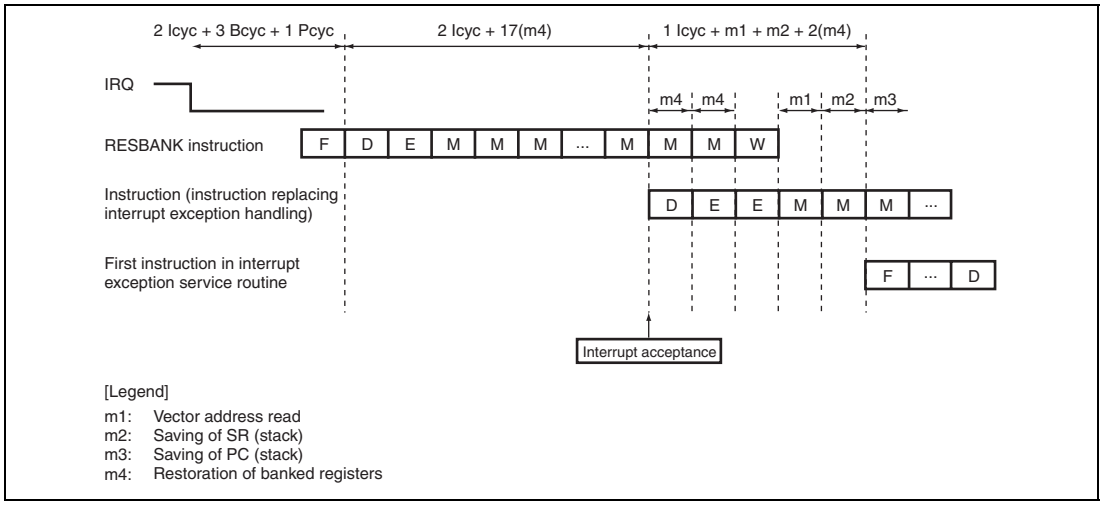


Figure 7.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

7.8 Register Banks

This LSI has fifteen register banks used to perform register saving and restoration required in the interrupt processing at high speed. Figure 7.10 shows the register bank configuration.

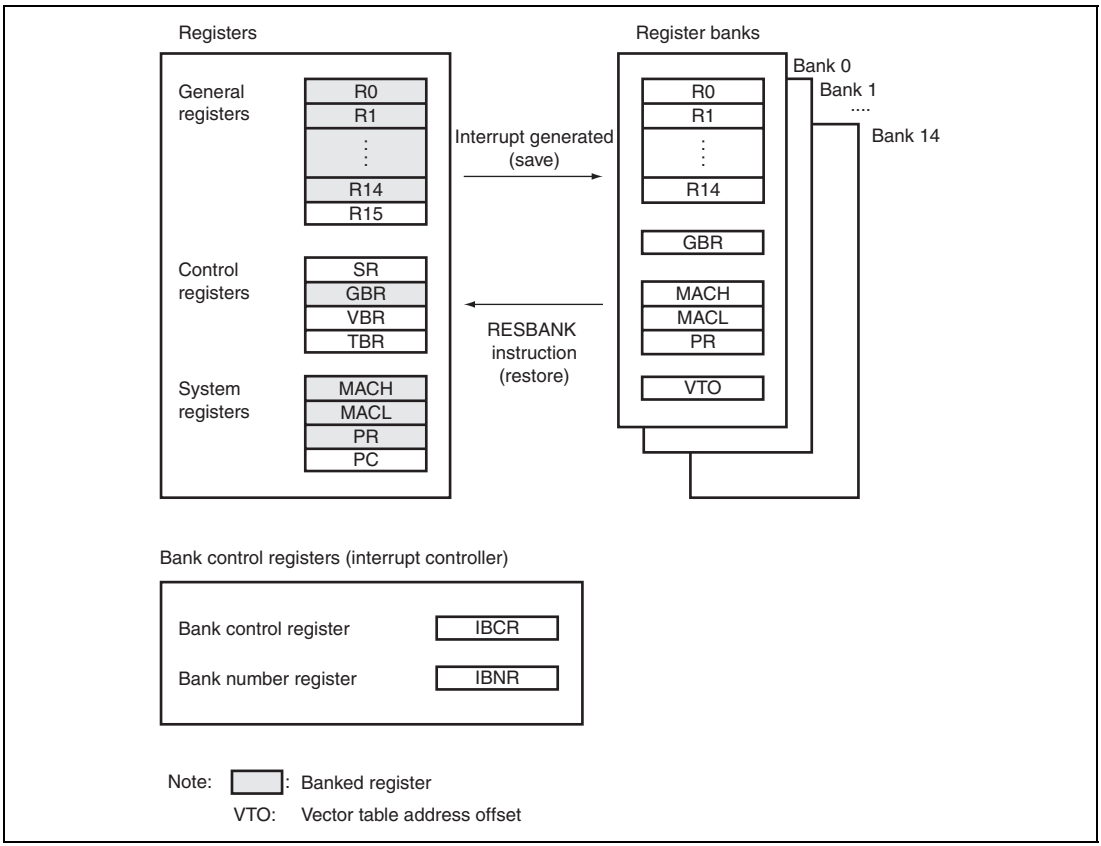


Figure 7.10 Overview of Register Bank Configuration

7.8.1 Banked Register and Input/Output of Banks

(1) Banked Register

The contents of the general registers (R0 to R14), global base register (GBR), multiply and accumulate registers (MACH and MACL), and procedure register (PR), and the vector table address offset are banked.

(2) Input/Output of Banks

This LSI has fifteen register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in order, beginning from bank 0, and restoration takes place in the reverse order, beginning from the last bank saved to.

7.8.2 Bank Save and Restore Operations

(1) Saving to Bank

Figure 7.11 shows register bank save operations. The following operations are performed when an interrupt for which usage of register banks is allowed is accepted by the CPU:

a. Assume that the bank number bit value in the bank number register (IBNR), BN, is "i" before the interrupt is generated.
b. The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the interrupt vector table address offset (VTO) of the accepted interrupt are saved in the bank indicated by BN, bank i.
c. The BN value is incremented by 1.

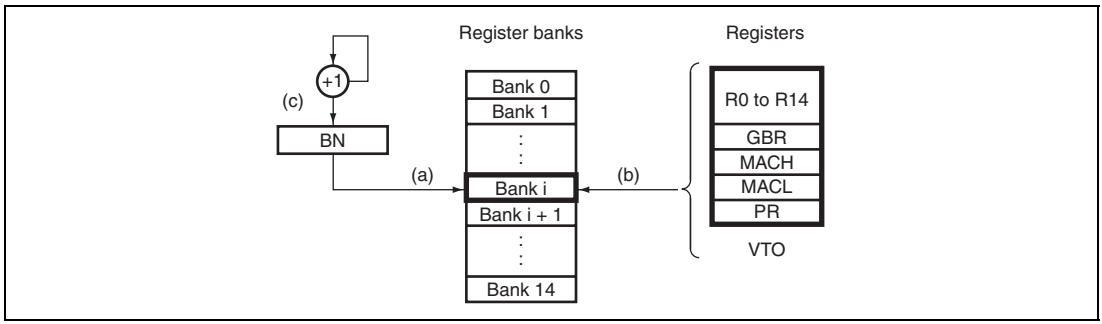


Figure 7.11 Bank Save Operations

Figure 7.12 shows the timing for saving to a register bank. Saving to a register bank takes place between the start of interrupt exception handling and the start of fetching the first instruction in the interrupt exception service routine.

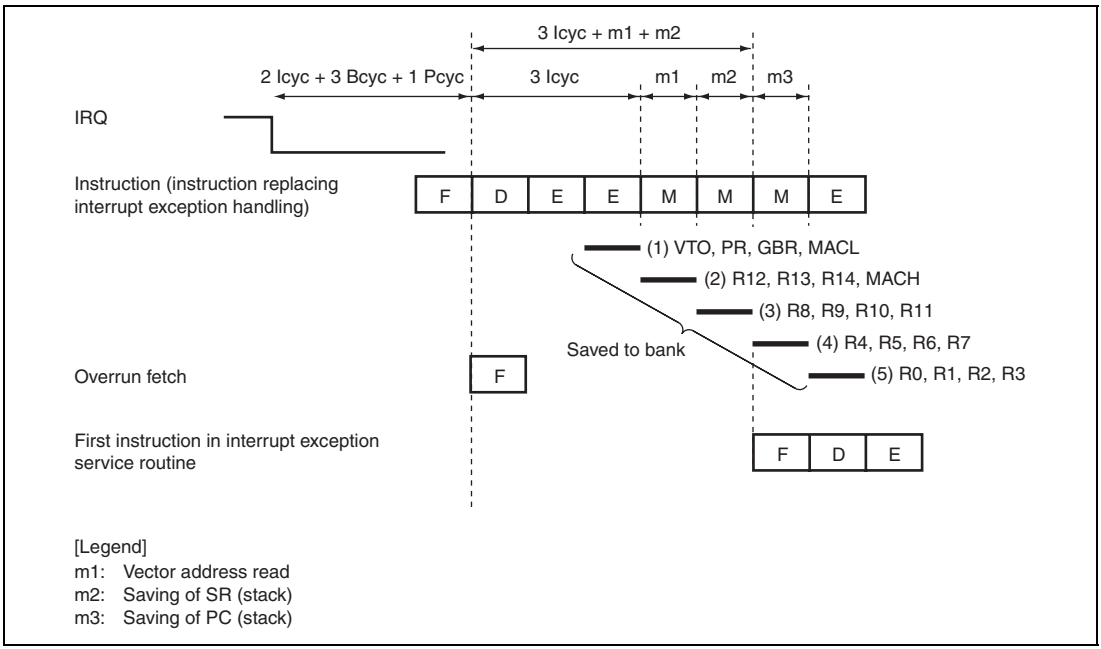


Figure 7.12 Bank Save Timing

(2) Restoration from Bank

The RESBANK (restore from register bank) instruction is used to restore data saved in a register bank. After restoring data from the register banks with the RESBANK instruction at the end of the interrupt service routine, execute the RTE instruction to return from the exception handling.

7.8.3 Save and Restore Operations after Saving to All Banks

If an interrupt occurs and usage of the register banks is enabled for the interrupt accepted by the CPU in a state where saving has been performed to all register banks, automatic saving to the stack is performed instead of register bank saving if the BOVE bit in the bank number register (IBNR) is cleared to 0. If the BOVE bit in IBNR is set to 1, register bank overflow exception occurs and data is not saved to the stack.

Save and restore operations when using the stack are as follows:

(1) Saving to Stack

1. The status register (SR) and program counter (PC) are saved to the stack during interrupt exception handling.
2. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are saved to the stack. The registers are saved to the stack in the order of MACL, MACH, GBR, PR, R14, R13, …, R1, and R0.
3. The register bank overflow bit (BO) in SR is set to 1.
4. The bank number bit (BN) value in the bank number register (IBNR) remains set to the maximum value of 15.

(2) Restoration from Stack

When the RESBANK (restore from register bank) instruction is executed with the register bank overflow bit (BO) in SR set to 1, the CPU operates as follows:

1. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack. The registers are restored from the stack in the order of R0, R1, …, R13, R14, PR, GBR, MACH, and MACL.
2. The bank number bit (BN) value in the bank number register (IBNR) remains set to the maximum value of 15.

7.8.4 Register Bank Exception

There are two register bank exceptions (register bank errors): register bank overflow and register bank underflow.

(1) Register Bank Overflow

This exception occurs if, after data has been saved to all of the register banks, an interrupt for which register bank use is allowed is accepted by the CPU, and the BOVE bit in the bank number register (IBNR) is set to 1. In this case, the bank number bit (BN) value in the bank number register (IBNR) remains set to the bank count of 15 and saving is not performed to the register bank.

(2) Register Bank Underflow

This exception occurs if the RESBANK (restore from register bank) instruction is executed when no data has been saved to the register banks. In this case, the values of R0 to R14, GBR, MACH, MACL, and PR do not change. In addition, the bank number bit (BN) value in the bank number register (IBNR) remains set to 0.

7.8.5 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When this happens, the CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a register bank overflow, and the start address of the executed RESBANK instruction for a register bank underflow. To prevent multiple interrupts from occurring at a register bank overflow, the interrupt priority level that caused the register bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. Program execution starts from the exception service routine start address.

7.9 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to trigger the following data transfer.

- Only the DMAC is activated and no CPU interrupt occurs.
- Only the DTC is activated and a CPU interrupt may occur depending on the DTC setting.

Interrupt sources that are designated to activate the DMAC are masked without being input to the INTC. The mask condition is as follows:

Mask condition = DME • (DE0 • interrupt source select 0 + DE1 • interrupt source select 1 + DE2 • interrupt source select 2 + DE3 • interrupt source select 3

Here, DME is bit 0 in DMAOR of the DMAC, and DEn (n = 0 to 3) is bit 0 in CHCR0 to CHCR3 of the DMAC.

The INTC masks a CPU interrupt when the DTCE bits corresponding to the DTC are 1. The DTCE clearing condition and interrupt source flag clearing condition are as follows:

DTCE clearing condition = DTC transfer end • DTCECLR

Interrupt source flag clearing condition = DTC transfer end • $\overline{\text{DTCECLR}}$ + DMAC transfer end

However, DTCECLR = DISEL + counter value of 0

Figures 7.13 and 7.14 show block diagrams of interrupt control.

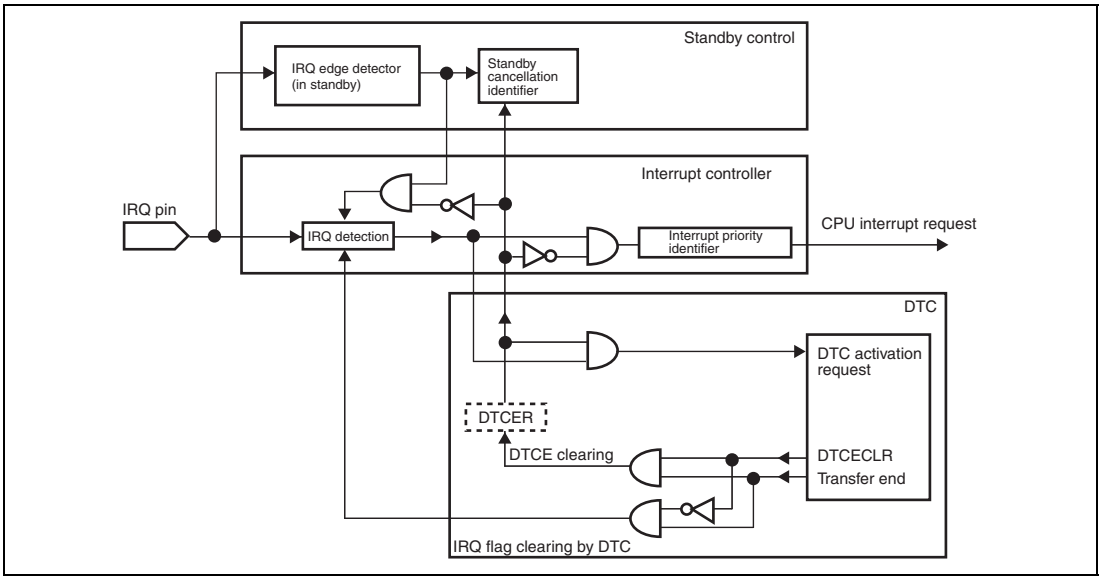


Figure 7.13 Block Diagram of Interrupt Control for an IRQ Signal

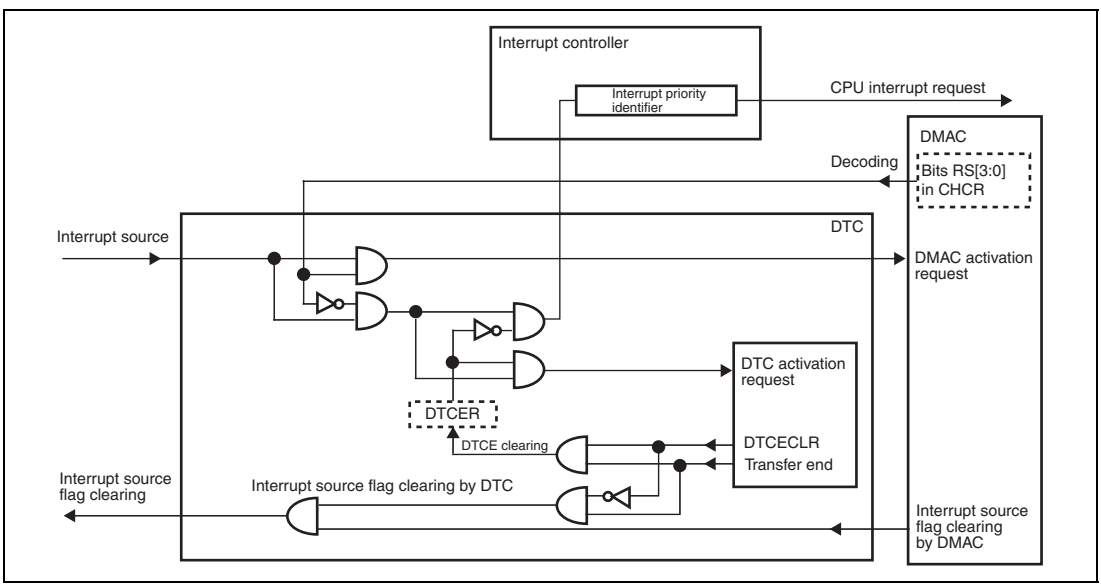


Figure 7.14 Block Diagram of Controlling an On-Chip Peripheral Module Interrupt

7.9.1 Handling an Interrupt Request Signal as a Source for DTC Activation and a Source for CPU Interrupts but not as a DMAC Activating Source

1. Do not select DMAC activating sources or clear the DME bit to 0. If DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. Set both the corresponding DTCE bit and DISEL bit to 1 in the DTC.
3. Activating sources are applied to the DTC when interrupts occur.
4. The DTC clears the DTCE bit to 0 and sends interrupt requests to the CPU when starting data transfer. The DTC does not clear the activating sources.
5. The CPU clears the interrupt sources in the interrupt exception handling routine, and then confirms the transfer counter value. If the transfer counter value is not 0, the DTCE bit is set to 1 and the next data transfer enabled. If the transfer counter value is 0, the CPU performs the necessary termination processing in the interrupt exception handling routine.

7.9.2 Handling an Interrupt Request Signal as a Source for DMAC Activation but for Neither CPU Interrupts nor DTC Activation

1. Select DMAC activating sources and set both the DE and DME bits to 1. This masks CPU interrupt sources and DTC activating sources regardless of the interrupt priority register and DTC register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears the activating sources when starting data transfer.

7.9.3 Handling an Interrupt Request Signal as a Source for DTC Activation but for Neither CPU Interrupts nor DMAC Activation

1. Do not select DMAC activating sources or clear the DME bit to 0. If DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. Set the corresponding DTCE bit to 1 and clear the DISEL bit to 0 in the DTC.
3. Activating sources are applied to the DTC when interrupts occur.
4. The DTC clears the activating sources when starting data transfer. Interrupt requests are not sent to the CPU because the DTCE bit remains set to 1.
5. However, when the transfer counter value is 0, the DTCE bit is cleared to 0 and interrupt requests are sent to the CPU.
6. The CPU performs the necessary termination processing in the interrupt exception handling routine.

7.9.4 Handling an Interrupt Request Signal as a Source for CPU Interrupts but for Neither DTC nor DMAC Activation

1. Do not select DMAC activating sources or clear the DME bit to 0. If DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. Clear the corresponding DTCE bit to 0 in the DTC.
3. Interrupt requests are sent to the CPU when interrupts occur.
4. The CPU clears the interrupt sources and performs the necessary termination processing in the interrupt exception handling routine.

7.10 Usage Note

7.10.1 Timing to Clear an Interrupt Source

The interrupt source flags should be cleared in the interrupt exception service routine. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" shown in table 7.5 is required before the interrupt source sent to the CPU is actually cancelled. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then execute an RTE instruction.

7.10.2 Timing for Negation of the $\overline{\text{IRQOUT}}$ Pin

The low level is output on the $\overline{\text{IRQOUT}}$ pin when the interrupt controller accepts an interrupt request, and the high level is output on the $\overline{\text{IRQOUT}}$ pin after the jump to the address where the interrupt exception service routine starts.

However, when an interrupt request is withdrawn after the low level has been output on the $\overline{\text{IRQOUT}}$ pin due to acceptance of the interrupt request by the controller but before the jump to the address where the interrupt exception service routine starts, the low level continues to be output on the $\overline{\text{IRQOUT}}$ pin until the jump to the address where the interrupt exception service routine starts for the next interrupt request.

7.10.3 Note on IRQx Interrupt Requests Initiating Release from Software Standby

When an IRQx interrupt request initiates release from software standby, change the IRQ sense selection setting in IRQx under the condition that the IRQ interrupt request will not be generated or execute the IRQx interrupt to clear the IRQxF flag in IRQRRx automatically.

When an IRQxF flag in IRQ interrupt request register x (IRQRRx) is 1, changing the IRQ sense selection setting in IRQx or clearing the IRQxF flag in the IRQRRx to 0 clears the corresponding IRQx interrupt request but not the signal requesting release from software standby.

Section 8 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Instruction fetch or data read/write (bus master (CPU, DMAC, or DTC) selection in the case of data read/write), data size, address value, and stop timing in the case of instruction fetch are break conditions that can be set in the UBC. Since this LSI uses a Harvard architecture, instruction fetch on the CPU bus (C bus) is performed by issuing bus cycles on the instruction fetch bus (F bus), and data access on the C bus is performed by issuing bus cycles on the memory access bus (M bus). The UBC monitors the C bus and internal bus (I bus).

8.1 Features

1. The following break comparison conditions can be set.
   Number of break channels: eight channels (channels 0 to 7)
   User break can be requested as the independent condition on channels 0 to 7
   The only break condition for channels 2 to 7 is the F bus instruction-fetch address break.

- Address
  Comparison of the 32-bit address is maskable in 1-bit units. (Channels 0 and 1)
  One of the three address buses (F address bus (FAB), M address bus (MAB), and I address bus (IAB)) can be selected.
- Bus master when I bus is selected
  Selection of CPU cycles, DMAC cycles, or DTC cycles
- Bus cycle
  Instruction fetch (only when C bus is selected) or data access
- Read/write
- Operand size
  Byte, word, and longword

2. Exception handling routine for user-specified break conditions can be executed.
3. In an instruction fetch cycle, it can be selected whether PC breaks are set before or after an instruction is executed.
4. When a break condition is satisfied, a trigger signal is output from the $\overline{\text{UBCTRG}}$ pin.

Figure 8.1 shows a block diagram of the UBC.

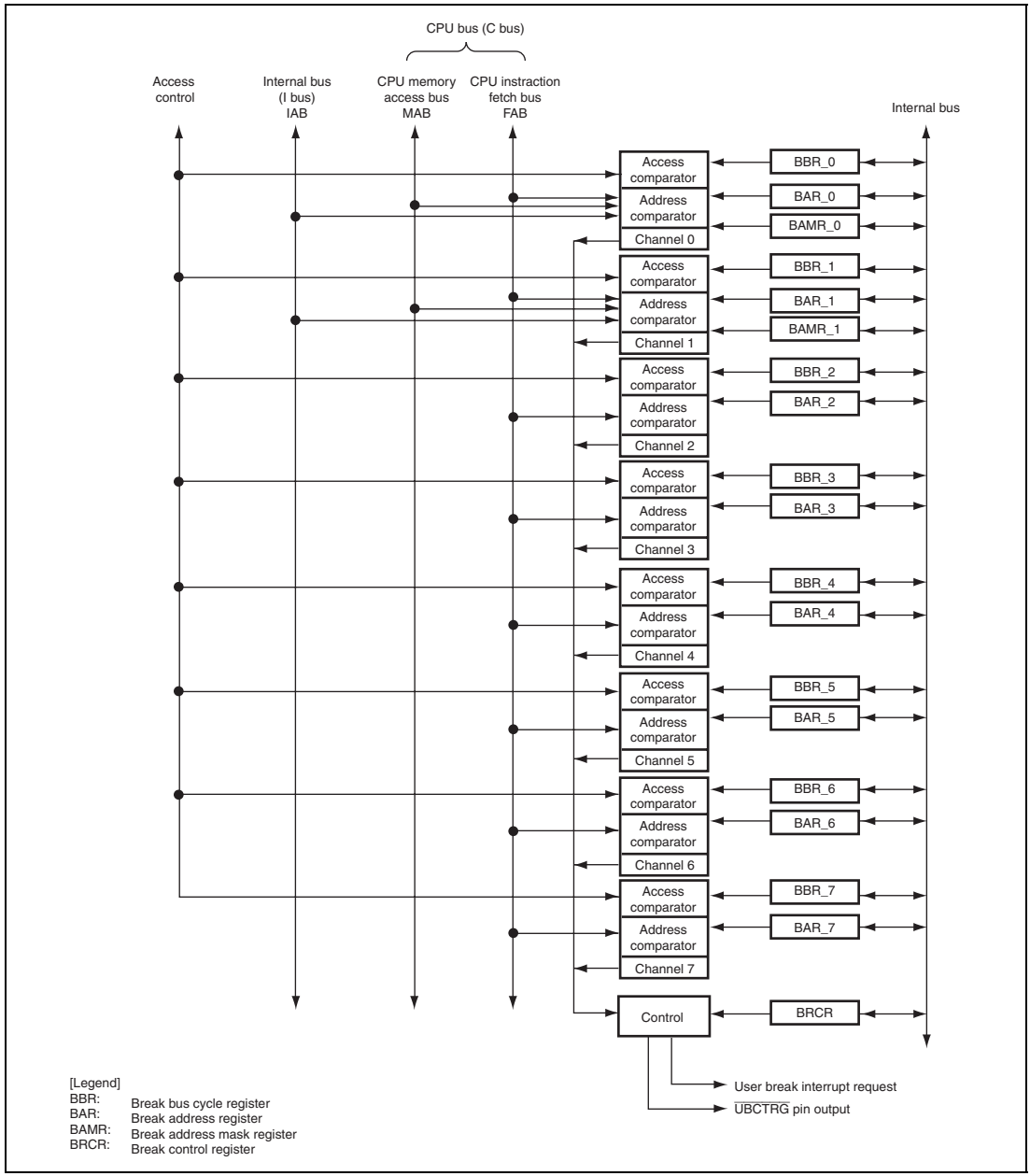


Figure 8.1 Block Diagram of UBC

8.2 Input/Output Pin

Table 8.1 shows the pin configuration of the UBC.

Table 8.1 Pin Configuration

Pin Name	Symbol	I/O	Function
UBC trigger	$\overline{\text{UBCTRG}}$	Output	Indicates that a setting condition has been satisfied on any of channels 0 to 7 of the UBC.

8.3 Register Descriptions

The UBC has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 8.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Break address register_0	BAR_0	R/W	H'00000000	H'FFFC0400	32
	Break address mask register_0	BAMR_0	R/W	H'00000000	H'FFFC0404	32
	Break bus cycle register_0	BBR_0	R/W	H'0000	H'FFFC04A0	16
1	Break address register_1	BAR_1	R/W	H'00000000	H'FFFC0410	32
	Break address mask register_1	BAMR_1	R/W	H'00000000	H'FFFC0414	32
	Break bus cycle register_1	BBR_1	R/W	H'0000	H'FFFC04A4	16
2	Break address register_2	BAR_2	R/W	H'00000000	H'FFFC0420	32
	Break bus cycle register_2	BBR_2	R/W	H'0000	H'FFFC04A8	16
3	Break address register_3	BAR_3	R/W	H'00000000	H'FFFC0430	32
	Break bus cycle register_3	BBR_3	R/W	H'0000	H'FFFC04AC	16
4	Break address register_4	BAR_4	R/W	H'00000000	H'FFFC0440	32
	Break bus cycle register_4	BBR_4	R/W	H'0000	H'FFFC04B0	16
5	Break address register_5	BAR_5	R/W	H'00000000	H'FFFC0450	32
	Break bus cycle register_5	BBR_5	R/W	H'0000	H'FFFC04B4	16
6	Break address register_6	BAR_6	R/W	H'00000000	H'FFFC0460	32
	Break bus cycle register_6	BBR_6	R/W	H'0000	H'FFFC04B8	16
7	Break address register_7	BAR_7	R/W	H'00000000	H'FFFC0470	32
	Break bus cycle register_7	BBR_7	R/W	H'0000	H'FFFC04BC	16
Common	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0	32

8.3.1 Break Address Registers_0 and 1(BAR_0 and BAR_1)

BAR_0 and BAR_1 are 32-bit readable/writable registers. BAR_0 and BAR_1 specify the address used as a break condition in channel 1. The control bits CD[1:0] and CP[2:0] in the break bus cycle register_0 and 1 (BBR_0 and BBR_1) select one of the three address buses for a break condition. BAR_0 and BAR_1 are initialized to H'00000000 by a power-on reset, but retain its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA31 to BA0	All 0	R/W	Break Address Store an address on the CPU address bus (FAB or MAB) or the internal address bus (IAB) specifying break conditions. When the C bus and instruction fetch cycle are selected by BBR, specify an FAB address in bits BA31 to BA0. When the C bus and data access cycle are selected by BBR, specify an MAB address in bits BA31 to BA0. When the I bus is selected by BBR, specify an IAB address in bits BA31 to BA0.

Note: When setting the instruction fetch cycle as a break condition, clear the BA0 bit in BAR to 0.

8.3.2 Break Address Mask Registers_0 and 1 (BAMR_0 and BAMR_1)

BAMR_0 and BAMR_1 are 32-bit readable/writable registers. BAMR_0 and BAMR_1 specify bits masked in the break address bits specified by BAR_0 and BAR_1. Channels 2 to 7 do not have these registers. BAMR_0 and BAMR_1 are initialized to H'00000000 by a power-on reset, but retain its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24	BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8	BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM31 to BAM0	All 0	R/W	Break Address Mask Specify bits masked in break address bits specified by BAR (BA31 to BA0). 0: Break address bit BAn is included in the break condition. 1: Break address bit BAn is masked and not included in the break condition.

Note: n = 31 to 0

8.3.3 Break Bus Cycle Registers_0 and 1 (BBR_0 and BBR_1)

BBR_0 and BBR_1 are 16-bit readable/writable registers, which specify (1) disabling or enabling of user break interrupt requests, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channels 0 and 1. BBR_0 and BBR_1 are initialized to H'0000 by a power-on reset, but retain its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID	-	-	CP[2:0]			CD[1:0]		ID[1:0]		RW[1:0]		SZ[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID	0	R/W	User Break Interrupt Disable Disables or enables user break interrupt requests when a condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP[2:0]	000	R/W	I-Bus Bus Master Select Select the bus master when the bus cycle of the break condition is the I bus cycle. However, when the C bus cycle is selected, these bits are invalidated (only the CPU cycle). 000: Condition comparison is not performed xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: DTC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select Select the C bus cycle or I bus cycle as the bus cycle of the break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID[1:0]	00	R/W	Instruction Fetch/Data Access Select Select the instruction fetch cycle or data access cycle as the bus cycle of the break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW[1:0]	00	R/W	Read/Write Select Select the read cycle or write cycle as the bus cycle of the break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ[1:0]	00	R/W	Operand Size Select Select the operand size of the bus cycle for the break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

8.3.4 Break Address Registers_2 to 7 (BAR_2 to BAR_7)

BAR_2 to BAR_7 are 32-bit readable/writable registers. BAR_2 to BAR_7 specify the address used as a break condition in channels 2 to 7. The only break condition is an F bus instruction-fetch address break. BAR_2 to BAR_7 are initialized to H'00000000 by a power-on reset, but retain its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA31 to BA0	All 0	R/W	Break Address Store an address on the CPU address bus (FAB) specifying break conditions.

Note: Set the BA0 bit in BAR to 0.

8.3.5 Break Bus Cycle Registers_2 to 7 (BBR_2 to BBR_7)

BBR_2 to BBR_7 are 16-bit readable/writable registers, which specify (1) disabling or enabling of user break interrupt requests and (2) selecting break conditions as break conditions in channels 2 to 7. BBR_2 to BBR_7 are initialized to H'0000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID	-	-	-	-	-	CD[1:0]		ID[1:0]		-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID	0	R/W	User Break Interrupt Disable Disables or enables user break interrupt requests when a condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	CD[1:0]	00	R/W	F Bus Cycle Select Select the F bus cycle as the bus cycle of the break condition. 00: Condition comparison is not performed 01: Break condition is the F bus cycle 1x: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
5, 4	ID[1:0]	00	R/W	Instruction Fetch Select the instruction fetch cycle as the bus cycle of the break condition. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 1x: Setting prohibited
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

[Legend]

x: Don't care

8.3.6 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether exception handling of user break interrupts due to instruction fetch cycles starts before or after the instruction is executed.
2. Specifies the pulse width of the $\overline{\text{UBCTRG}}$ output when a break condition is satisfied.

BRCR is a 32-bit readable/writable register that has break condition match flags and bits for setting other break conditions. For the condition match flags of bits 25, 24, and 15 to 8, writing 1 is invalid (previous values are retained) and writing 0 is only possible. To clear the flag, write 0 to the flag bit to be cleared and 1 to all other flag bits. BRCR is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SCMFD1	SCMFD0	-	-	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCMFC7	SCMFC6	SCMFC5	SCMFC4	SCMFC3	SCMFC2	SCMFC1	SCMFC0	PCB7	PCB6	PCB5	PCB4	PCB3	PCB2	PCB1	PCB0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	SCMFD1	0	R/W	I Bus Cycle Condition Match Flag 1 When the I bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 1 does not match 1: The I bus cycle condition for channel 1 matches

Bit	Bit Name	Initial Value	R/W	Description
24	SCMFD0	0	R/W	I Bus Cycle Condition Match Flag 0 When the I bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 0 does not match 1: The I bus cycle condition for channel 0 matches
23 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	CKS[1:0]	00	R/W	Clock Select These bits specify the pulse width output to the $\overline{\text{UBCTRG}}$ pin when a condition is satisfied. 00: Reserved (setting prohibited) 01: Reserved (setting prohibited) 10: Pulse width of $\overline{\text{UBCTRG}}$ is four bus cycles 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight bus cycles
15	SCMFC7	0	R/W	C Bus Cycle Condition Match Flag 7 When the C bus cycle condition in the break conditions set for channel 7 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 7 does not match 1: The C bus cycle condition for channel 7 matches
14	SCMFC6	0	R/W	C Bus Cycle Condition Match Flag 6 When the C bus cycle condition in the break conditions set for channel 6 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 6 does not match 1: The C bus cycle condition for channel 6 matches

Bit	Bit Name	Initial Value	R/W	Description
13	SCMFC5	0	R/W	C Bus Cycle Condition Match Flag 5 When the C bus cycle condition in the break conditions set for channel 5 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 5 does not match 1: The C bus cycle condition for channel 5 matches
12	SCMFC4	0	R/W	C Bus Cycle Condition Match Flag 4 When the C bus cycle condition in the break conditions set for channel 4 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 4 does not match 1: The C bus cycle condition for channel 4 matches
11	SCMFC3	0	R/W	C Bus Cycle Condition Match Flag 3 When the C bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 3 does not match 1: The C bus cycle condition for channel 3 matches
10	SCMFC2	0	R/W	C Bus Cycle Condition Match Flag 2 When the C bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 2 does not match 1: The C bus cycle condition for channel 2 matches
9	SCMFC1	0	R/W	C Bus Cycle Condition Match Flag 1 When the C bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 1 does not match 1: The C bus cycle condition for channel 1 matches

Bit	Bit Name	Initial Value	R/W	Description
8	SCMFC0	0	R/W	C Bus Cycle Condition Match Flag 0 When the C bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 0 does not match 1: The C bus cycle condition for channel 0 matches
7	PCB7	0	R/W	PC Break Select 7 Selects the break timing of the instruction fetch cycle for channel 7 as before or after instruction execution. 0: PC break of channel 7 is generated before instruction execution 1: PC break of channel 7 is generated after instruction execution
6	PCB6	0	R/W	PC Break Select 6 Selects the break timing of the instruction fetch cycle for channel 6 as before or after instruction execution. 0: PC break of channel 6 is generated before instruction execution 1: PC break of channel 6 is generated after instruction execution
5	PCB5	0	R/W	PC Break Select 5 Selects the break timing of the instruction fetch cycle for channel 5 as before or after instruction execution. 0: PC break of channel 5 is generated before instruction execution 1: PC break of channel 5 is generated after instruction execution
4	PCB4	0	R/W	PC Break Select 4 Selects the break timing of the instruction fetch cycle for channel 4 as before or after instruction execution. 0: PC break of channel 4 is generated before instruction execution 1: PC break of channel 4 is generated after instruction execution

Bit	Bit Name	Initial Value	R/W	Description
3	PCB3	0	R/W	PC Break Select 3 Selects the break timing of the instruction fetch cycle for channel 3 as before or after instruction execution. 0: PC break of channel 3 is generated before instruction execution 1: PC break of channel 3 is generated after instruction execution
2	PCB2	0	R/W	PC Break Select 2 Selects the break timing of the instruction fetch cycle for channel 2 as before or after instruction execution. 0: PC break of channel 2 is generated before instruction execution 1: PC break of channel 2 is generated after instruction execution
1	PCB1	0	R/W	PC Break Select 1 Selects the break timing of the instruction fetch cycle for channel 1 as before or after instruction execution. 0: PC break of channel 1 is generated before instruction execution 1: PC break of channel 1 is generated after instruction execution
0	PCB0	0	R/W	PC Break Select 0 Selects the break timing of the instruction fetch cycle for channel 0 as before or after instruction execution. 0: PC break of channel 0 is generated before instruction execution 1: PC break of channel 0 is generated after instruction execution

8.4 Operation

8.4.1 Flow of the User Break Operation

The flow from setting of break conditions to user break interrupt exception handling is described below:

1. The break address is set in a break address register (BAR). The masked address bits are set in a break address mask register (BAMR). The bus break conditions are set in a break bus cycle register (BBR). Three control bit groups of BBR (C bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set to 00. The relevant break control conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBR, and branch after reading from the last written register. The newly written register values become valid from the instruction at the branch destination.
2. In the case where the break conditions have been satisfied and user break interrupt requests are enabled, the UBC notifies the INTC of the user break request, sets the C bus condition match flag (SCMFC) or I bus condition match flag (SCMFD) for the appropriate channel, and outputs a pulse on the $\overline{\text{UBCTRG}}$ pin with the width set by the CKS1 and CKS0 bits. Setting the UBID bit in BBR to 1 enables external monitoring of the trigger output without requesting user break interrupts.
3. On receiving a user break interrupt request signal, the INTC determines its priority. Since the user break interrupt has a priority level of 15, it is accepted when the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR) is 14 or lower. If the I3 to I0 bits are set to a priority level of 15, the user break interrupt is not accepted, but the conditions are checked, and condition match flags are set if the conditions match. For details on ascertaining the priority, see section 7, Interrupt Controller (INTC).
4. Condition match flags (SCMFC and SCMFD) can be used to check which condition has been satisfied. Clear the condition match flag bit from within the exception handling routine for the user break interrupt. Without this operation, the interrupt will be generated again.
5. A single break interrupt request to the INTC may set multiple break channel match flags if multiple sets of conditions are matched.

6. When selecting the I bus as the break condition, note as follows:
   — Several bus masters, including the CPU and DMAC, are connected to the I bus. The UBC monitors bus cycles generated by the bus master specified by BBR, and determines the condition match.
   — I bus cycles resulting from instruction fetches on the C bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.
   — The DTC and DMAC only issue data access cycles for I bus cycles.
   — If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the user break is to be accepted cannot be clearly defined.

8.4.2 Break on Instruction Fetch Cycle

1. When C bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBR), the break condition is the FAB bus instruction fetch cycle. Whether exception handling of a user break interrupt starts before or after the instruction is executed can be selected by the PCB0 or PCB1 bit for the given channel in the break control register (BRCR). If an instruction fetch cycle is set as a break condition, clear the BA0 bit in the break address register (BAR) to 0. A break cannot be generated as long as this bit is set to 1.
2. A break for instruction fetch which is set as a break before instruction execution occurs when it is confirmed that the instruction has been fetched and will be executed. This means a break does not occur for instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delayed branch instruction, the user break interrupt request is not accepted until the execution of the first instruction at the branch destination.

   Note: If a branch does not occur at a delayed branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When setting a break condition for break after instruction execution, the instruction set with the break condition is executed and then the break is generated prior to execution of the next instruction. As with pre-execution breaks, a break does not occur with overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, the user break interrupt request is not accepted until the first instruction at the branch destination.
4. If the I bus is set for a break of an instruction fetch cycle, the setting is invalidated.

8.4.3 Break on Data Access Cycle

1. If the C bus is specified as a break condition for data access break, condition comparison is performed for the address accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the address of the data access cycles that are issued by the bus master specified by the bits to select the bus master of the I bus, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 6 in section 8.4.1, Flow of the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 8.3.

Table 8.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits 31 to 2
Word	Compares break address register bits 31 to 1 to address bus bits 31 to 1
Byte	Compares break address register bits 31 to 0 to address bus bits 31 to 0

This means that when address H'00001003 is set in the break address register (BAR), for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. If the data access cycle is selected, the instruction at which the break will occur cannot be determined.

8.4.4 Value of Saved Program Counter

When a user break interrupt request is accepted, the address of the instruction from where execution is to be resumed is saved to the stack, and the exception handling state is entered. If the C bus (FAB)/instruction fetch cycle is specified as a break condition, the instruction at which the break should occur can be uniquely determined. If the C bus/data access cycle or I bus/data access cycle is specified as a break condition, the instruction at which the break should occur cannot be uniquely determined.

1. When C bus (FAB)/instruction fetch (before instruction execution) is specified as a break condition:

   The address of the instruction that matched the break condition is saved to the stack. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
2. When C bus (FAB)/instruction fetch (after instruction execution) is specified as a break condition:

   The address of the instruction following the instruction that matched the break condition is saved to the stack. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
3. When C bus/data access cycle or I bus/data access cycle is specified as a break condition:

   The address after executing several instructions of the instruction that matched the break condition is saved to the stack.

8.4.5 Usage Examples

(1) Break Condition Specified for C Bus Instruction Fetch Cycle

(Example 1-1)

- Register specifications:
  BAR_0 = H'00000404, BAMR_0 = H'00000000, BBR_0 = H'0054, BAR_1 = H'00008010, BAMR_1 = H'00000006, BBR_1 = H'0054, BRCR = H'00000001
  <Channel 0>
  Address: H'00000404, Address mask: H'00000000
  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)
  <Channel 1>
  Address: H'00008010, Address mask: H'00000006
  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)
  A user break occurs after an instruction of address H'00000404 is executed or before instructions of addresses H'00008010 to H'00008016 is executed.

(Example 1-2)

- Register specifications:
  BAR_0 = H'00027128, BAMR_0 = H'00000000, BBR_0 = H'005A, BAR_1= H'00031415, BAMR_1 = H'00000000, BBR_1 = H'0054, BRCR = H'00000000
  <Channel 0>
  Address: H'00027128, Address mask: H'00000000
  Bus cycle: C bus/instruction fetch (before instruction execution)/write/word
  <Channel 1>
  Address: H'00031415, Address mask: H'00000000
  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)
  On channel 0, a user break does not occur since instruction fetch is not a write cycle. On channel 1, a user break does not occur since instruction fetch is performed for an even address.

(Example 1-3)

- Register specifications:
  BBR_0 = H'0054, BAR_0 = H'00008404, BAMR_0 = H'00000FFF, BBR_1 = H'0054,
  BAR_1 = H'00008010, BAMR_1 = H'00000006, BRCR = H'00000001
  <Channel 0>
  Address: H'00008404, Address mask: H'00000FFF
  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)
  <Channel 1>
  Address: H'00008010, Address mask: H'00000006
  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)
  A user break occurs after an instruction with addresses H'00008000 to H'00008FFE is executed or before an instruction with addresses H'00008010 to H'00008016 are executed.

(2) Break Condition Specified for C Bus Data Access Cycle

(Example 2-1)

- Register specifications:
  BBR_0 = H'0064, BAR_0 = H'00123456, BAMR_0 = H'00000000,
  BBR_1 = H'006A, BAR_1 = H'000ABCDE, BAMR_1 = H'000000FF, BRCR = H'00000000
  <Channel 0>
  Address: H'00123456, Address mask: H'00000000
  Bus cycle: C bus/data access/read (operand size is not included in the condition)
  <Channel 1>
  Address: H'000ABCDE, Address mask: H'000000FF
  Bus cycle: C bus/data access/write/word
  On channel 0, a user break occurs with longword read from address H'00123456, word read from address H'00123456, or byte read from address H'00123456. On channel 1, a user break occurs when word is written in addresses H'000ABC00 to H'000ABCFE.

(3) Break Condition Specified for I Bus Data Access Cycle

(Example 3-1)

- Register specifications:
  BBR_0 = H'0094, BAR_0 = H'00314156, BAMR_0 = H'00000000,
  BBR_1 = H'02A9, BAR_1 = H'00055555, BAMR_1 = H'00000000, BRCR = H'00000000
  <Channel 0>
  Address: H'00314156, Address mask: H'00000000
  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)
  <Channel 1>
  Address: H'00055555, Address mask: H'00000000
  Bus cycle: I bus/data access/write/byte
  On channel 0, the setting of I bus/instruction fetch is ignored.
  On channel 1, a user break occurs when the DMAC writes byte data in address H'00055555 on the I bus (write by the CPU does not generate a user break).

8.5 Interrupt Source

Table 8.4 gives details on this interrupt source.

A user break interrupt is generated when one of the compare match flags (SCMFD0 and SCMFD1, and SCMFC7 to SCMFC0) in the break control register (BRCR) is set to 1. Clearing the interrupt flag bit to 0 cancels the interrupt request.

Table 8.4 Interrupt Source

Abbreviation	Interrupt Source	Interrupt Enable Bit	Interrupt Flag	Interrupt Level
User break	User break interrupt	—	SCMFD0, SCMFD1, SCMFC7 to SCMFC0	Fixed to 15

8.6 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. The UBC cannot monitor access to the C bus and I bus cycles in the same channel.
3. When a user break interrupt request and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 6.1 in section 6, Exception Handling. If an exception with a higher priority occurs, the user break interrupt request is held pending without acceptance.
4. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at a delay slot instruction, the user break interrupt request is not accepted until immediately before execution of the branch destination.
5. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.
6. Do not set an address within an interrupt exception handling routine whose interrupt priority level is at least 15 (including user break interrupts) as a break address.
7. Do not set break after instruction execution for the SLEEP instruction or for the delayed branch instruction where the SLEEP instruction is placed at its delay slot.
8. When setting a break for a 32-bit instruction, set the address where the upper 16 bits are placed. If the address of the lower 16 bits is set and a break before instruction execution is set as a break condition, the break is handled as a break after instruction execution.
9. Do not set a pre-execution break for an instruction that immediately follows a DIVU or DIVS instruction. If such a break is set and an interrupt or other exception occurs during execution of the DIVU or DIVS instruction, the pre-execution break will still occur even though execution of the DIVU or DIVS instruction is suspended.

Section 9 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated to transfer data by an interrupt request.

9.1 Features

- Transfer possible over any number of channels
- Chain transfer

  Multiple rounds of data transfer is executed in response to a single activation source

  Chain transfer is only possible after data transfer has been done for the specified number of times (i.e. when the transfer counter is 0)
- Three transfer modes

  Normal/repeat/block transfer modes selectable

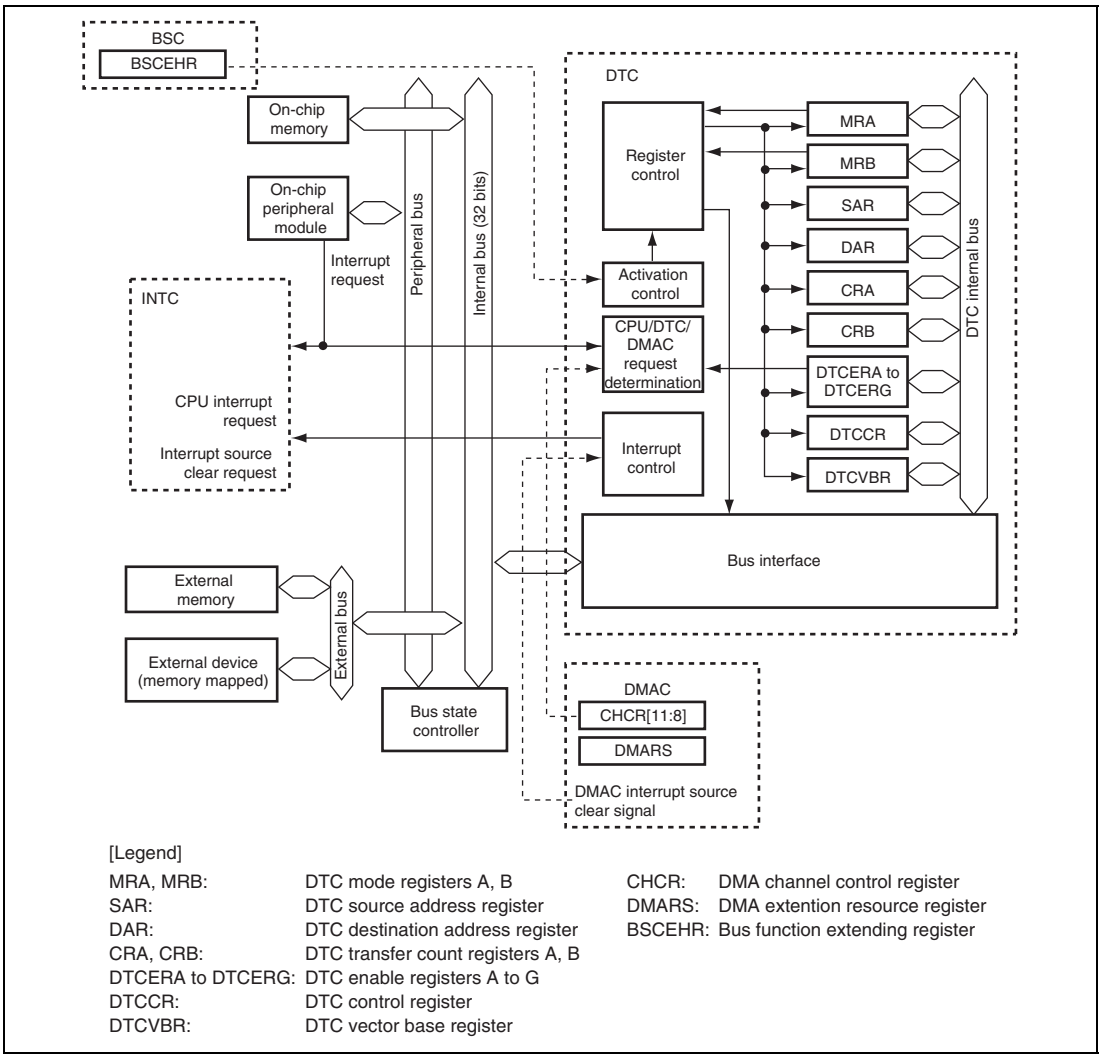
  Transfer source and destination addresses can be selected from increment/decrement/fixed
- The transfer source and destination addresses can be specified by 32 bits to select a 4-Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword
- A CPU interrupt can be requested for the interrupt that activated the DTC

  A CPU interrupt can be requested after one data transfer completion

  A CPU interrupt can be requested after the specified data transfer completion
- Read skip of the transfer information specifiable
- Write-back skip executed for the fixed transfer source and destination addresses
- Module stop mode settable
- Short address mode settable
- Bus release timing selectable: Three types
- DTC activation priority selectable: Two types

Figure 9.1 shows a block diagram of the DTC. The DTC transfer information can be allocated to the data area*.

Note: When the transfer information is stored in the on-chip RAM (high speed), the RAME0 to RAME3 bits in SYSCR1 and the RAMWE0 to RAMWE3 bits in SYSCR2 must be set to 1.



[Legend]

MRA, MRB:	DTC mode registers A, B	CHCR:	DMA channel control register
SAR:	DTC source address register	DMARS:	DMA extention resource register
DAR:	DTC destination address register	BSCEHR:	Bus function extending register
CRA, CRB:	DTC transfer count registers A, B		
DTCERA to DTCERG:	DTC enable registers A to G		
DTCCR:	DTC control register		
DTCVBR:	DTC vector base register		

Figure 9.1 Block Diagram of DTC

9.2 Register Descriptions

DTC has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 34, List of Registers.

These six registers MRA, MRB, SAR, DAR, CRA, and CRB cannot be directly accessed by the CPU. The contents of these registers are stored in the data area as transfer information. When a DTC activation request occurs, the DTC reads a start address of transfer information that is stored in the data area according to the vector address, reads the transfer information, and transfers data. After the data transfer is complete, it writes a set of updated transfer information back to the data area.

On the other hand, DTCERA to DTCERG, DTCCR, and DTCVBR can be directly accessed by the CPU.

Table 9.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
DTC enable register A	DTCERA	R/W	H'0000	H'FFFE6000	8, 16
DTC enable register B	DTCERB	R/W	H'0000	H'FFFE6002	8, 16
DTC enable register C	DTCERC	R/W	H'0000	H'FFFE6004	8, 16
DTC enable register D	DTCERD	R/W	H'0000	H'FFFE6006	8, 16
DTC enable register E	DTCERE	R/W	H'0000	H'FFFE6008	8, 16
DTC enable register F	DTCERF	R/W	H'0000	H'FFFE600A	8, 16
DTC enable register G	DTCERG	R/W	H'0000	H'FFFE600C	8, 16
DTC control register	DTCCR	R/W	H'00	H'FFFE6010	8
DTC vector base register	DTCVBR	R/W	H'00000000	H'FFFE6014	8, 16, 32
Bus function extending register	BSCEHR	R/W	H'0000	H'FFFE3C1A	8, 16

9.2.1 DTC Mode Register A (MRA)

MRA selects DTC operating mode. MRA cannot be accessed directly by the CPU.

Bit:	7	6	5	4	3	2	1	0
	MD[1:0]		Sz[1:0]		SM[1:0]		-	-
Initial value:	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-

*: Undefined

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MD[1:0]	Undefined	—	DTC Mode 1 and 0 Specify DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
5, 4	Sz[1:0]	Undefined	—	DTC Data Transfer Size 1 and 0 Specify the size of data to be transferred. 00: Byte-size transfer 01: Word-size transfer 10: Longword-size transfer 11: Setting prohibited
3, 2	SM[1:0]	Undefined	—	Source Address Mode 1 and 0 Specify an SAR operation after a data transfer. 0x: SAR is fixed (SAR write-back is skipped) 10: SAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
x: Don't care

9.2.2 DTC Mode Register B (MRB)

MRB selects DTC operating mode. MRB cannot be accessed directly by the CPU.

Bit:	7	6	5	4	3	2	1	0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		-	-
Initial value:	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-

*: Undefined

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see section 9.5.6, Chain Transfer. The chain transfer condition is selected by the CHNS bit. 0: Disables the chain transfer 1: Enables the chain transfer
6	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. If the following transfer is a chain transfer, the completion check of the specified transfer count is not performed and activation source flag or DTCER is not cleared. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0
5	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, an interrupt request is generated to the CPU every time a data transfer or a block data transfer ends. When this bit is set to 0, a CPU interrupt request is only generated when the specified number of data transfers ends. Note: Set this bit to 0 when the IIC3 is selected as the activation source.

Bit	Bit Name	Initial Value	R/W	Description
4	DTS	Undefined	—	DTC Transfer Mode Select Specifies either the source or destination as repeat or block area during repeat or block transfer mode. 0: Specifies the destination as repeat or block area 1: Specifies the source as repeat or block area
3, 2	DM[1:0]	Undefined	—	Destination Address Mode 1 and 0 Specify a DAR operation after a data transfer. 0x: DAR is fixed (DAR write-back is skipped) 10: DAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
x: Don't care

9.2.3 DTC Source Address Register (SAR)

SAR is a 32-bit register that designates the source address of data to be transferred by the DTC.

SAR cannot be accessed directly from the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

* : Undefined

9.2.4 DTC Destination Address Register (DAR)

DAR is a 32-bit register that designates the destination address of data to be transferred by the DTC.

DAR cannot be accessed directly from the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*: Undefined

9.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal transfer mode, CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and bit DTCEn (n = 15 to 0) corresponding to the activation source is cleared and then an interrupt is requested to the CPU when the count reaches H'0000. The transfer count is 1 when CRA = H'0001, 65,535 when CRA = H'FFFF, and 65,536 when CRA = H'0000.

In repeat transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and the lower eight bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent to CRAL when the count reaches H'00. The transfer count is 1 when CRAH = CRAL = H'01, 255 when CRAH = CRAL = H'FF, and 256 when CRAH = CRAL = H'00.

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and the lower eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block-size counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a byte (word or longword) data is transferred, and the contents of CRAH are sent to CRAL when the count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL =H'01, 255 bytes (words or longwords) when CRAH = CRAL = H'FF, and 256 bytes (words or longwords) when CRAH = CRAL =H'00.

CRA cannot be accessed directly from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

* : Undefined

9.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time a block of data is transferred, and bit DTCEn (n = 15 to 0) corresponding to the activation source is cleared and then an interrupt is requested to the CPU when the count reaches H'0000. The transfer count is 1 when CRB = H'0001, 65,535 when CRB = H'FFFF, and 65,536 when CRB = H'0000.

CRB is not available in normal and repeat modes and cannot be accessed directly by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*: Undefined

9.2.7 DTC Enable Registers A to G (DTCERA to DTCERG)

DTCER which is comprised of eight registers, DTCERA to DTCERG, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 9.2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	DTCE15	0	R/W	DTC Activation Enable 15 to 0 Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source. [Clearing conditions] • When writing 0 to the bit to be cleared after reading 1 • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended [Setting condition] • Writing 1 to the bit after reading 0
14	DTCE14	0	R/W	
13	DTCE13	0	R/W	
12	DTCE12	0	R/W	
11	DTCE11	0	R/W	
10	DTCE10	0	R/W	
9	DTCE9	0	R/W	
8	DTCE8	0	R/W	
7	DTCE7	0	R/W	
6	DTCE6	0	R/W	
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

9.2.8 DTC Control Register (DTCCR)

DTCCR specifies transfer information read skip.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	RRS	RCHNE	-	-	ERR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R	R/(W)*

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRS	0	R/W	DTC Transfer Information Read Skip Enable Controls the vector address read and transfer information read. A DTC vector number is always compared with the vector number for the previous activation. If the vector numbers match and this bit is set to 1, the DTC data transfer is started without reading a vector address and transfer information. If the previous DTC activation is a chain transfer, the vector address read and transfer information read are always performed. However, while the DTPR bit in the bus function extending register (BSCEHR) is set to 1, the transfer information read skip is not performed regardless of this bit setting. 0: Transfer information read skip is not performed. 1: Transfer information read skip is performed when the vector numbers match.
3	RCHNE	0	R/W	Chain Transfer Enable After DTC Repeat Transfer Enables/disables the chain transfer while transfer counter (CRAL) is 0 in repeat transfer mode. In repeat transfer mode, the CRAH value is written to CRAL when CRAL is 0. Accordingly, chain transfer may not occur when CRAL is 0. If this bit is set to 1, the chain transfer is enabled when CRAH is written to CRAL. 0: Disables the chain transfer after repeat transfer 1: Enables the chain transfer after repeat transfer

Bit	Bit Name	Initial Value	R/W	Description
2, 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.
0	ERR	0	R/(W)*	Transfer Stop Flag This flag indicates that a DTC address error or NMI interrupt request has occurred. If a DTC address error or NMI interrupt occurs while the DTC is active, handling of the DTC address error or NMI interrupt is executed after the DTC has released bus mastership. The DTC halts after data transfer or the writing of transfer information. Note that the DTC will not be activated while ERR = 1. If attempting to reactivate the DTC by clearing the ERR flag after the value of the ERR bit was changed to 1 while the DTC was active causes the DTC to halt, set the transfer information once again. 0: A DTC address error or NMI interrupt request has not occurred. 1: A DTC address error or NMI interrupt request has occurred. [Clearing condition] • When writing 0 after reading 1

Note: * Only 0 can be written to clear the flag after 1 is read.

9.2.9 DTC Vector Base Register (DTCVBR)

DTCVBR is a 32-bit register that specifies the base address for vector table address calculation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12		All 0	R/W	Bits 11 to 0 are always read as 0. The write value should always be 0.
11 to 0	—	All 0	R	

9.2.10 Bus Function Extending Register (BSCEHR)

BSCEHR is a 16-bit register that specifies the timing of bus release by the DTC and other functions. This register should be used to give priority to the DTC transfer or reduce the number of cycles in which the DTC is active. For more details, see section 10.4.8, Bus Function Extending Register (BSCEHR).

9.3 Activation Sources

The DTC is activated by an interrupt request. The interrupt source is selected by DTCER. A DTC activation source can be selected by setting the corresponding bit in DTCER; the CPU interrupt source can be selected by clearing the corresponding bit in DTCER. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source interrupt flag or corresponding DTCER bit is cleared.

9.4 Location of Transfer Information and DTC Vector Table

Locate the transfer information in the data area. The start address of transfer information should be located at the address that is a multiple of four (4n). Otherwise, the lower two bits are ignored during access ([1:0] = B'00.) Transfer information located in the data area is shown in figure 9.2.

Short address mode can be selected by setting the DTSA bit in the bus function extending register (BSCEHR) to 1 only when all DTC transfer sources and destinations are located in the on-chip RAM (high speed/for retention) and on-chip peripheral module areas (see section 10.4.8, Bus Function Extending Register (BSCEHR)).

In normal transfer, four longwords should be read as the transfer information; in short address mode, the transfer information is reduced to three longwords and the DTC active period becomes shorter.

The DTC reads the start address of transfer information from the vector table according to the activation source, and then reads the transfer information from the start address. Figure 9.3 shows correspondences between the DTC vector address and transfer information.

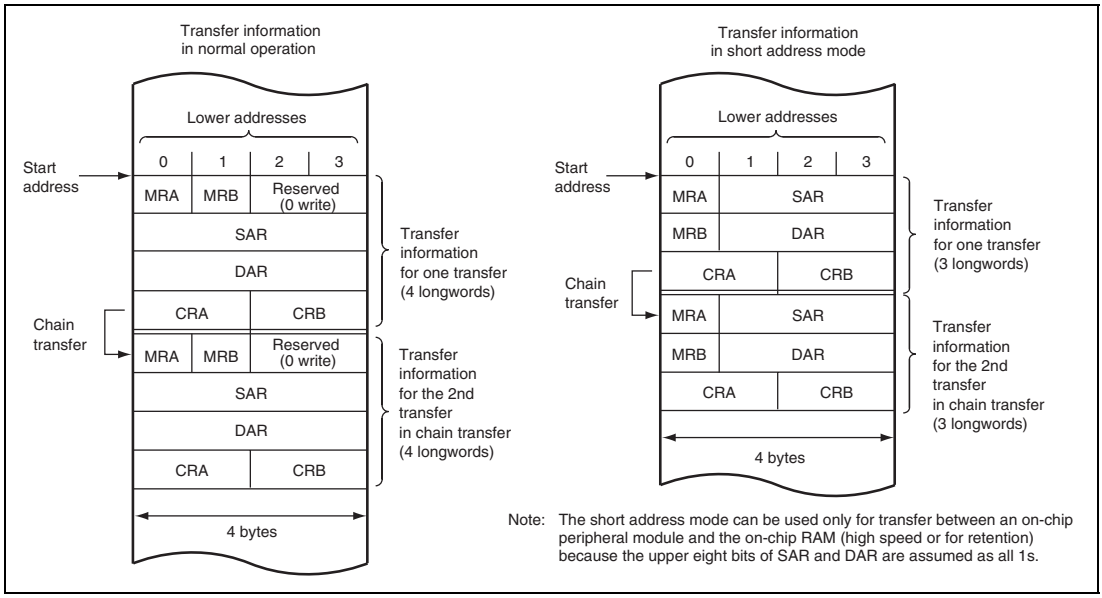


Figure 9.2 Transfer Information on Data Area

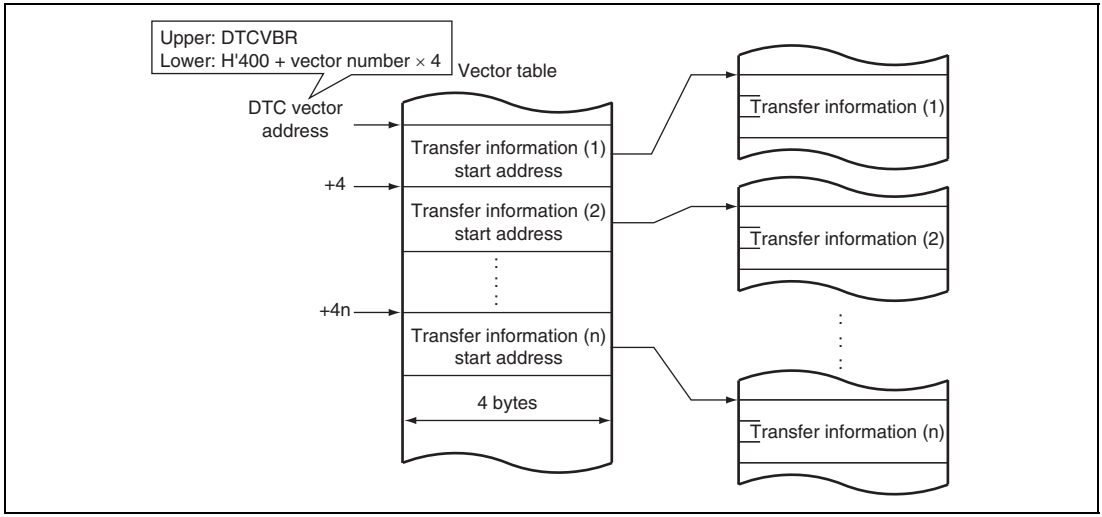


Figure 9.3 Correspondence between DTC Vector Address and Transfer Information

Table 9.2 shows correspondence between the DTC activation source and vector address.

Table 9.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE	Transfer Source	Transfer Destination	Priority
External pin	IRQ0	64	H'00000500	DTCERA15	Any location*2	Any location*2	High
	IRQ1	65	H'00000504	DTCERA14	Any location*2	Any location*2	
	IRQ2	66	H'00000508	DTCERA13	Any location*2	Any location*2	
	IRQ3	67	H'0000050C	DTCERA12	Any location*2	Any location*2	
	IRQ4	68	H'00000510	DTCERA11	Any location*2	Any location*2	
	IRQ5	69	H'00000514	DTCERA10	Any location*2	Any location*2	
	IRQ6	70	H'00000518	DTCERA9	Any location*2	Any location*2	
	IRQ7	71	H'0000051C	DTCERA8	Any location*2	Any location*2	
	IRQ8	72	H'00000520	DTCERF15	Any location*2	Any location*2	
	IRQ9	73	H'00000524	DTCERF14	Any location*2	Any location*2	
	IRQ10	74	H'00000528	DTCERF13	Any location*2	Any location*2	
	IRQ11	75	H'0000052C	DTCERF12	Any location*2	Any location*2	
	IRQ12	76	H'00000530	DTCERF11	Any location*2	Any location*2	
	IRQ13	77	H'00000534	DTCERF10	Any location*2	Any location*2	
	IRQ14	78	H'00000538	DTCERF9	Any location*2	Any location*2	
	IRQ15	79	H'0000053C	DTCERF8	Any location*2	Any location*2	
	IRQ16	80	H'00000540	DTCERF7	Any location*2	Any location*2	
	IRQ17	81	H'00000544	DTCERF6	Any location*2	Any location*2	
	IRQ18	82	H'00000548	DTCERF5	Any location*2	Any location*2	
	IRQ19	83	H'0000054C	DTCERF4	Any location*2	Any location*2	
	IRQ20	84	H'00000550	DTCERF3	Any location*2	Any location*2	
	IRQ21	85	H'00000554	DTCERF2	Any location*2	Any location*2	
	IRQ22	86	H'00000558	DTCERF1	Any location*2	Any location*2	
	IRQ23	87	H'0000055C	DTCERF0	Any location*2	Any location*2	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE	Transfer Source	Transfer Destination	Priority
A/D	ADI0	104	H'000005A0	DTCERA7	ADDR0 to ADDR7, ADSDR	Any location*2	High
	ADI1	105	H'000005A4	DTCERA6	ADDR8 to ADDR15, ADSDR	Any location*2	
SCIF4	RXI4	126	H'000005F8	DTCERG15	SCFRDR_4	Any location*2	
	TXI4	127	H'000005FC	DTCERG14	Any location*2	SCFTDR_4	
SCIF5	RXI5	130	H'00000608	DTCERG13	SCFRDR_5	Any location*2	
	TXI5	131	H'0000060C	DTCERG12	Any location*2	SCFTDR_5	
SCIF6	RXI6	134	H'00000618	DTCERG11	SCFRDR_6	Any location*2	
	TXI6	135	H'0000061C	DTCERG10	Any location*2	SCFTDR_6	
SCIF7	RXI7	138	H'00000628	DTCERG9	SCFRDR_7	Any location*2	
	TXI7	139	H'0000062C	DTCERG8	Any location*2	SCFTDR_7	
CMT	CMI0	140	H'00000630	DTCERA3	Any location*2	Any location*2	
	CMI1	141	H'00000634	DTCERA2	Any location*2	Any location*2	
CMT2	CM2I	142	H'00000638	DTCERE4	Any location*2	Any location*2	
	IC0I	143	H'0000063C	DTCERE3	Any location*2	Any location*2	
	IC1I	144	H'00000640	DTCERE2	Any location*2	Any location*2	
	OC0I	145	H'00000644	DTCERE1	Any location*2	Any location*2	
	OC1I	146	H'00000648	DTCERE0	Any location*2	Any location*2	
RSPI0	SPRXI_0	150	H'00000658	DTCERG2	SPRX	Any location*2	
	SPTXI_0	151	H'0000065C	DTCERG1	Any location*2	SPTX	
MTU2_0	TGIA_0	156	H'00000670	DTCERB15	Any location*2	Any location*2	
	TGIB_0	157	H'00000674	DTCERB14	Any location*2	Any location*2	
	TGIC_0	158	H'00000678	DTCERB13	Any location*2	Any location*2	
	TGID_0	159	H'0000067C	DTCERB12	Any location*2	Any location*2	
MTU2_1	TGIA_1	164	H'00000690	DTCERB11	Any location*2	Any location*2	
	TGIB_1	165	H'00000694	DTCERB10	Any location*2	Any location*2	
MTU2_2	TGIA_2	172	H'000006B0	DTCERB9	Any location*2	Any location*2	
	TGIB_2	173	H'000006B4	DTCERB8	Any location*2	Any location*2	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE	Transfer Source	Transfer Destination	Priority
MTU2_3	TGIA_3	180	H'000006D0	DTCERB7	Any location*2	Any location*2	High
	TGIB_3	181	H'000006D4	DTCERB6	Any location*2	Any location*2	
	TGIC_3	182	H'000006D8	DTCERB5	Any location*2	Any location*2	
	TGID_3	183	H'000006DC	DTCERB4	Any location*2	Any location*2	
MTU2_4	TGIA_4	188	H'000006F0	DTCERB3	Any location*2	Any location*2	
	TGIB_4	189	H'000006F4	DTCERB2	Any location*2	Any location*2	
	TGIC_4	190	H'000006F8	DTCERB1	Any location*2	Any location*2	
	TGID_4	191	H'000006FC	DTCERB0	Any location*2	Any location*2	
	TCIV_4	192	H'00000700	DTCERC15	Any location*2	Any location*2	
MTU2_5	TGIU_5	196	H'00000710	DTCERC14	Any location*2	Any location*2	
	TGIV_5	197	H'00000714	DTCERC13	Any location*2	Any location*2	
	TGIW_5	198	H'00000718	DTCERC12	Any location*2	Any location*2	
MTU2S_3	TGIA_3S	204	H'00000730	DTCERC3	Any location*2	Any location*2	
	TGIB_3S	205	H'00000734	DTCERC2	Any location*2	Any location*2	
	TGIC_3S	206	H'00000738	DTCERC1	Any location*2	Any location*2	
	TGID_3S	207	H'0000073C	DTCERC0	Any location*2	Any location*2	
MTU2S_4	TGIA_4S	212	H'00000750	DTCERD15	Any location*2	Any location*2	
	TGIB_4S	213	H'00000754	DTCERD14	Any location*2	Any location*2	
	TGIC_4S	214	H'00000758	DTCERD13	Any location*2	Any location*2	
	TGID_4S	215	H'0000075C	DTCERD12	Any location*2	Any location*2	
	TCIV_4S	216	H'00000760	DTCERD11	Any location*2	Any location*2	
MTU2S_5	TGIU_5S	220	H'00000770	DTCERD10	Any location*2	Any location*2	
	TGIV_5S	221	H'00000774	DTCERD9	Any location*2	Any location*2	
	TGIW_5S	222	H'00000778	DTCERD8	Any location*2	Any location*2	
LVDS (SH72315A only)	LVRXI	226	H'00000788	DTCERG5	LVFRDR	Any location*2	
IIC3	IIRXI	230	H'00000798	DTCERD7	ICDRR	Any location*2	
	IITXI	231	H'0000079C	DTCERD6	Any location*2	ICDRT	
RCAN-ET	RM_0	238	H'000007B8	DTCERG0	CONTROL0H to CONTROL1L*3	Any location*2	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE	Transfer Source	Transfer Destination	Priority
SCI0	RXI0	241	H'000007C4	DTCERE15	SCRDR_0	Any location*2	High
	TXI0	242	H'000007C8	DTCERE14	Any location*2	SCTDR_0	
SCI1	RXI1	245	H'000007D4	DTCERE13	SCRDR_1	Any location*2	
	TXI1	246	H'000007D8	DTCERE12	Any location*2	SCTDR_1	
SCI2	RXI2	249	H'000007E4	DTCERE11	SCRDR_2	Any location*2	
	TXI2	250	H'000007E8	DTCERE10	Any location*2	SCTDR_2	
SCI3	RXI3	253	H'000007F4	DTCERE9	SCRDR_3	Any location*2	
	TXI3	254	H'000007F8	DTCERE8	Any location*2	SCTDR_3	Low

Notes: 1. The DTCE bits with no corresponding interrupt are reserved, and the write value for these bits should always be 0. When this LSI is to be released from software standby in response to an interrupt signal from an external pin, write 0 to the corresponding DTCE bit.

2. An external memory, a memory-mapped external device, an on-chip memory, or an on-chip peripheral module (except for DMAC, DTC, BSC, UBC, and FLASH) can be selected as the source or destination. Note that at least either the source or destination must be an on-chip peripheral module; transfer cannot be done among an external memory, a memory-mapped external device, and an on-chip memory.

3. Read to a message field 1 (CONTROL1L) in mailbox 0 by using a block transfer or etc.

9.5 Operation

There are three transfer modes: normal, repeat, and block. Since transfer information is in the data area, it is possible to transfer data over any required number of channels. When activated, the DTC reads the transfer information stored in the data area and transfers data according to the transfer information. After the data transfer is complete, it writes updated transfer information back to the data area.

The DTC specifies the source address and destination address in SAR and DAR, respectively. After a transfer, SAR and DAR are incremented, decremented, or fixed independently.

Table 9.3 shows the DTC transfer modes.

Table 9.3 DTC Transfer Modes

Transfer Mode	Size of Data Transferred at One Transfer Request	Memory Address Increment or Decrement	Transfer Count
Normal	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1 to 65536
Repeat*1	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1 to 256*3
Block*2	Block size specified by CRAH (1 to 256 bytes/words/longwords)	Incremented/decremented by 1, 2, or 4, or fixed	1 to 65536*4

Notes: 1. Either source or destination is specified to repeat area.
2. Either source or destination is specified to block area.
3. After transfer of the specified transfer count, initial state is recovered to continue the operation.
4. Number of transfers of the specified block size of data

Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers with a single activation (chain transfer). Setting the CHNS bit in MRB to 1 can also be made to have chain transfer performed only when the transfer counter value is 0.

Figure 9.4 shows a flowchart of DTC operation, and table 9.4 summarizes the conditions for DTC transfers including chain transfer (combinations for performing the second and third transfers are omitted).

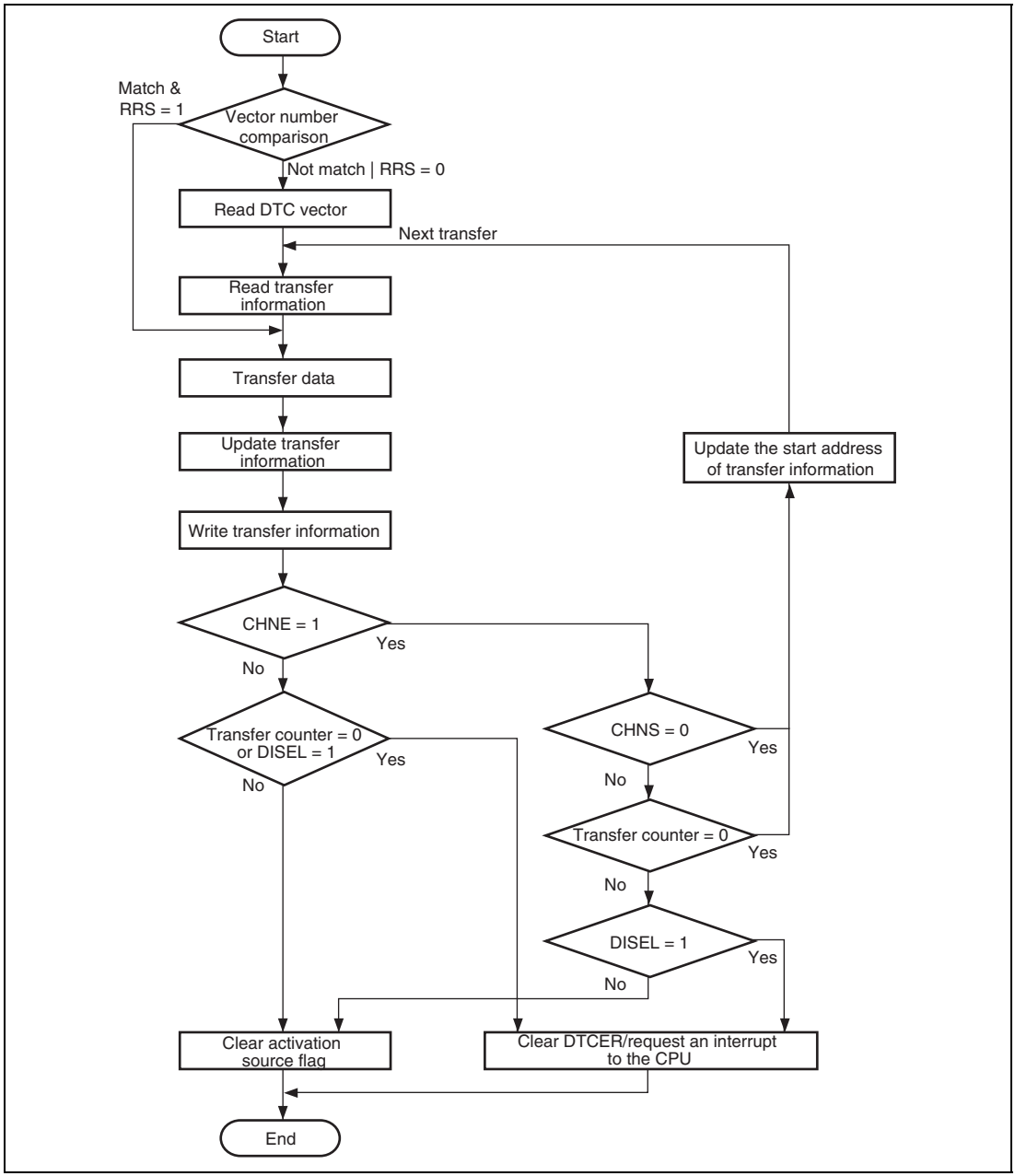


Figure 9.4 Flowchart of DTC Operation

Table 9.4 DTC Transfer Conditions (Chain Transfer Conditions Included)

	1st Transfer					2nd Transfer					
Transfer Mode	CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	DTC Transfer
Normal	0	—	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	0	—	—	0	0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	
	1	1	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	1	1	—	1	Not 0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	—	—	0	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	

Transfer Mode	1st Transfer CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	2nd Transfer CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	DTC Transfer
Repeat	0	—	—	0	—	—	—	—	—	—	Ends at 1st transfer
	0	—	—	1	—	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	0	—	—	—	0	—	—	0	—	Ends at 2nd transfer
						0	—	—	1	—	Ends at 2nd transfer Interrupt request to CPU
	1	1	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	1	1	—	1	Not 0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	0	0	0*2	—	—	—	—	—	Ends at 1st transfer
	1	1	0	1	0*2	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	1	—	0*2	0	—	—	0	—	Ends at 2nd transfer
						0	—	—	1	—	Ends at 2nd transfer Interrupt request to CPU

	1st Transfer					2nd Transfer					
Transfer Mode	**CHNE**	**CHNS**	**RCHNE**	**DISEL**	**Transfer Counter*1**	**CHNE**	**CHNS**	**RCHNE**	**DISEL**	**Transfer Counter*1**	**DTC Transfer**
Block	0	—	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	0	—	—	0	0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	
	1	1	—	0	—	—	—	—	—	—	Ends at 1st transfer
	1	1	—	1	Not 0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	—	1	0	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	

Notes: 1. CRA in normal mode transfer, CRAL in repeat transfer mode, or CRB in block transfer mode
2. When the contents of the CRAH is written to the CRAL

9.5.1 Transfer Information Read Skip Function

By setting the RRS bit of DTCCR, the vector address read and transfer information read can be skipped. The current DTC vector number is always compared with the vector number of previous activation. If the vector numbers match when RRS = 1, a DTC data transfer is performed without reading the vector address and transfer information. If the previous activation is a chain transfer, the vector address read and transfer information read are always performed. Figure 9.5 shows the transfer information read skip timing.

To modify the vector table and transfer information, temporarily clear the RRS bit to 0, modify the vector table and transfer information, and then set the RRS bit to 1 again. When the RRS bit is cleared to 0, the stored vector number is deleted, and the updated vector table and transfer information are read at the next activation.

However, this function always becomes invalid when the DTPR bit in the bus function extending register (BSCEHR) is 1.

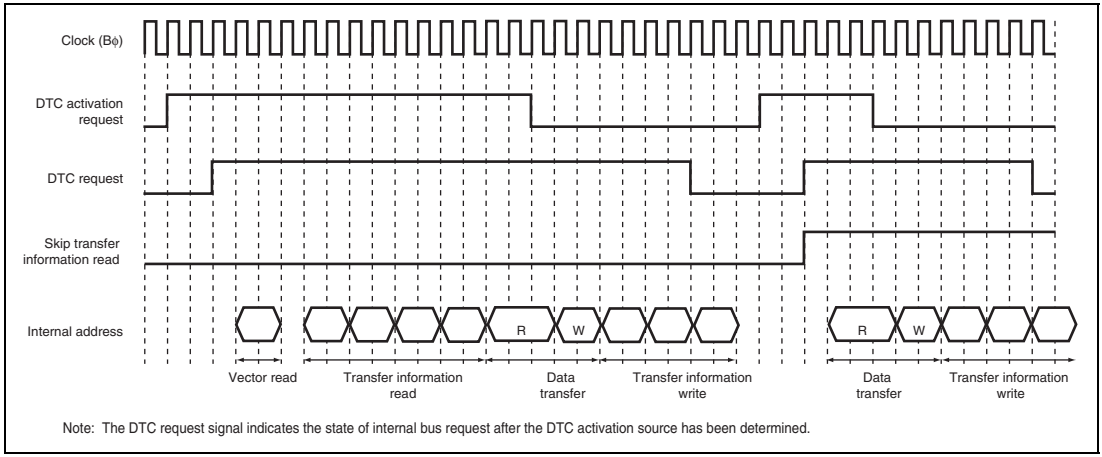


Figure 9.5 Transfer Information Read Skip Timing
(Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2;
Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed);
Transfer Information is Written in 3 Cycles)

9.5.2 Transfer Information Write-Back Skip Function

By specifying bit SM1 in MRA and bit DM1 in MRB to the fixed address mode, a part of transfer information will not be written back. Table 9.5 shows the transfer information write-back skip condition and write-back skipped registers. Note that the CRA and CRB are always written back. The write-back of the MRA and MRB are always skipped.

Table 9.5 Transfer Information Write-Back Skip Condition and Write-Back Skipped Registers

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

9.5.3 Normal Transfer Mode

In normal transfer mode, data are transferred in one byte, one word, or one longword units in response to a single activation request. From 1 to 65,536 transfers can be specified. The transfer source and destination addresses can be specified as incremented, decremented, or fixed. When the specified number of transfers ends, an interrupt can be requested to the CPU.

Table 9.6 lists the register function in normal transfer mode. Figure 9.6 shows the memory map in normal transfer mode.

Table 9.6 Register Function in Normal Transfer Mode

Register	Function	Written Back Value
SAR	Source address	Incremented/decremented/fixed*
DAR	Destination address	Incremented/decremented/fixed*
CRA	Transfer count A	CRA – 1
CRB	Transfer count B	Not updated

Note: * Transfer information write-back is skipped.

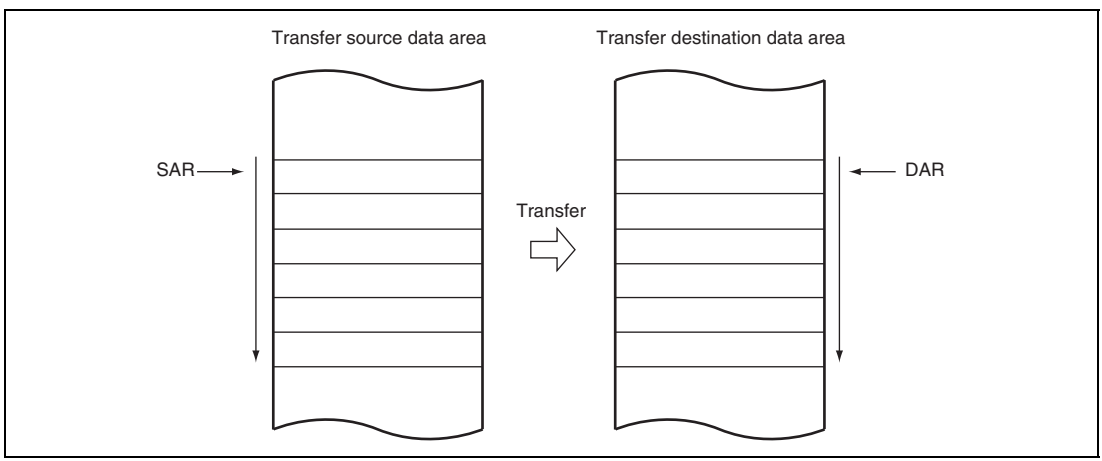


Figure 9.6 Memory Map in Normal Transfer Mode

9.5.4 Repeat Transfer Mode

In repeat transfer mode, data are transferred in one byte, one word, or one longword units in response to a single activation request. By the DTS bit in MRB, either the source or destination can be specified as a repeat area. From 1 to 256 transfers can be specified. When the specified number of transfers ends, the transfer counter and address register specified as the repeat area is restored to the initial state, and transfer is repeated. The other address register is then incremented, decremented, or left fixed. In repeat transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH when CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefore a CPU interrupt cannot be requested when DISEL = 0.

Table 9.7 lists the register function in repeat transfer mode. Figure 9.7 shows the memory map in repeat transfer mode.

Table 9.7 Register Function in Repeat Transfer Mode

		Written Back Value	
Register	**Function**	**CRAL is not 1**	**CRAL is 1**
SAR	Source address	Incremented/decremented/fixed*	DTS = 0: Incremented/ decremented/fixed* DTS = 1: SAR initial value
DAR	Destination address	Incremented/decremented/fixed*	DTS = 0: DAR initial value DTS = 1: Incremented/ decremented/fixed*
CRAH	Transfer count storage	CRAH	CRAH
CRAL	Transfer count A	CRAL – 1	CRAH
CRB	Transfer count B	Not updated	Not updated

Note: * Transfer information write-back is skipped.

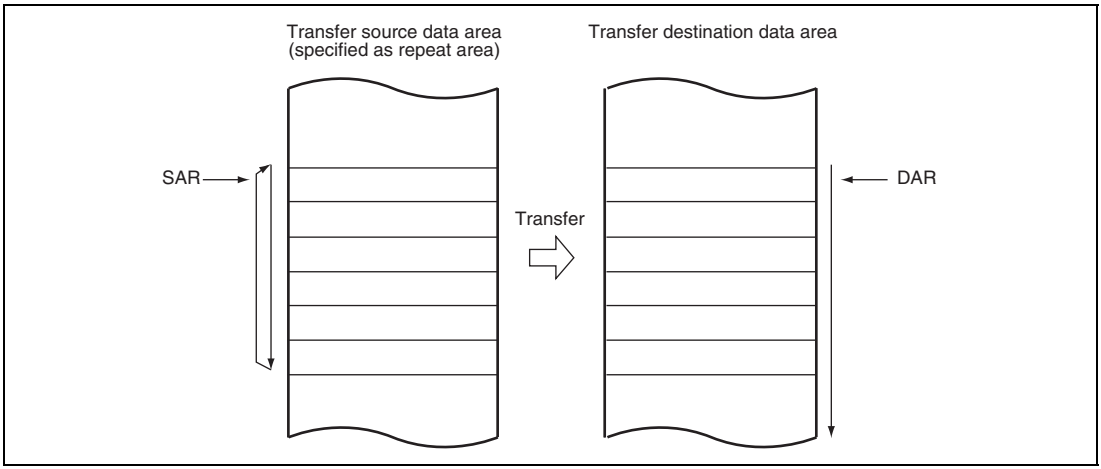


Figure 9.7 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

9.5.5 Block Transfer Mode

In block transfer mode, data are transferred in block units in response to a single activation request. Either the transfer source or the transfer destination is designated as a block area by the DTS bit in MRB.

The block size is 1 to 256 bytes (1 to 256 words, or 1 to 256 longwords). When transfer of one block of data ends, the block size counter (CRAL) and address register (SAR when DTS = 1 or DAR when DTS = 0) for the area specified as the block area are initialized. The other address register is then incremented, decremented, or left fixed. From 1 to 65,536 transfers can be specified. When the specified number of transfers ends, an interrupt is requested to the CPU.

Table 9.8 lists the register function in block transfer mode. Figure 9.8 shows the memory map in block transfer mode.

Table 9.8 Register Function in Block Transfer Mode

Register	Function	Written Back Value
SAR	Source address	DTS = 0: Incremented/decremented/fixed*
		DTS = 1: SAR initial value
DAR	Destination address	DTS = 0: DAR initial value
		DTS = 1: Incremented/decremented/fixed*
CRAH	Block size storage	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB – 1

Note: * Transfer information write-back is skipped.

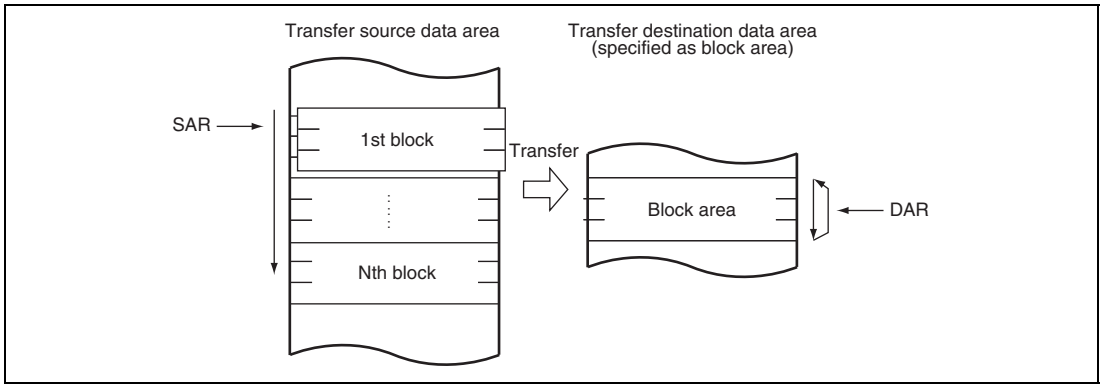


Figure 9.8 Memory Map in Block Transfer Mode (When Transfer Destination is Specified as Block Area)

9.5.6 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. Setting the CHNE and CHNS bits in MRB set to 1 enables a chain transfer only when the transfer counter reaches 0. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently. Figure 9.9 shows the chain transfer operation.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting the DISEL bit to 1, and the interrupt source flag for the activation source and DTCER are not affected.

In repeat transfer mode, setting the RCHNE bit in DTCCR and the CHNE and CHNS bits in MRB to 1 enables a chain transfer after transfer with transfer counter = 1 has been completed.

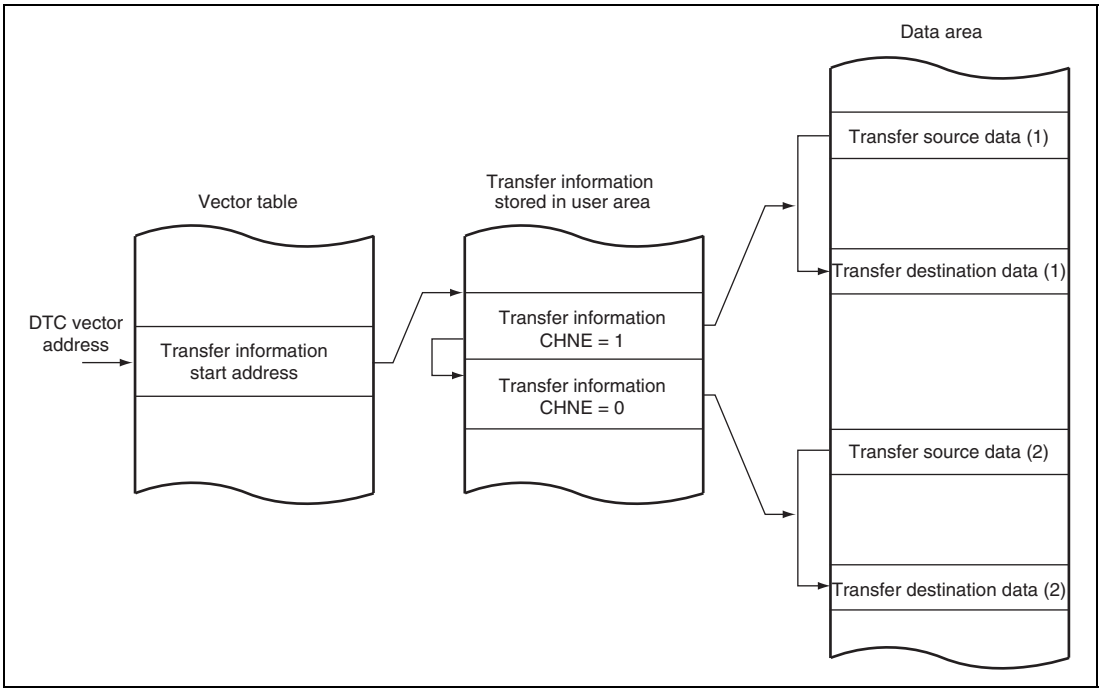
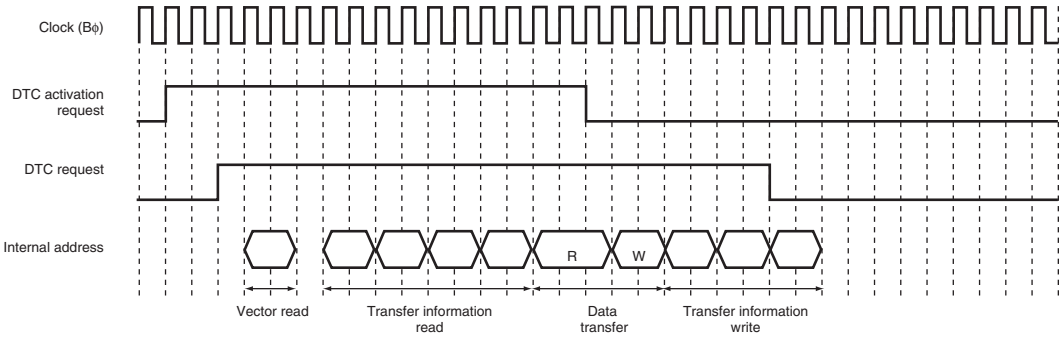


Figure 9.9 Operation of Chain Transfer

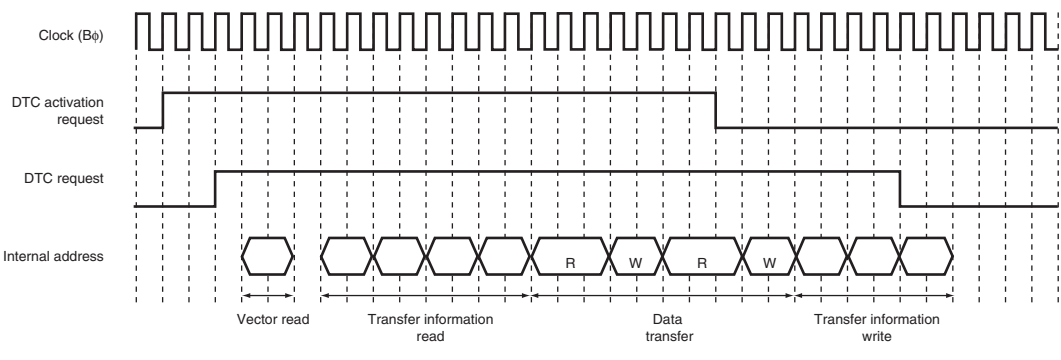
9.5.7 Operation Timing

Figures 9.10 to 9.15 show the DTC operation timings.



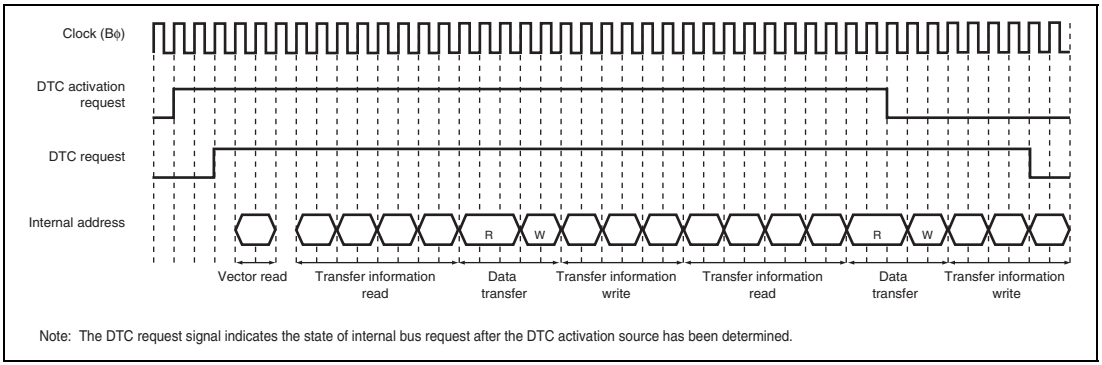
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 9.10 Example of DTC Operation Timing: Normal Transfer Mode or Repeat Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)



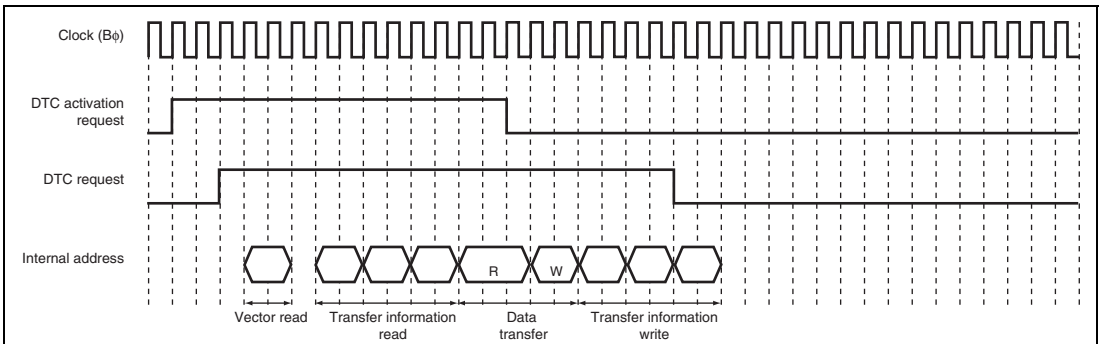
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 9.11 Example of DTC Operation Timing: Block Transfer Mode with Block Size = 2 (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)



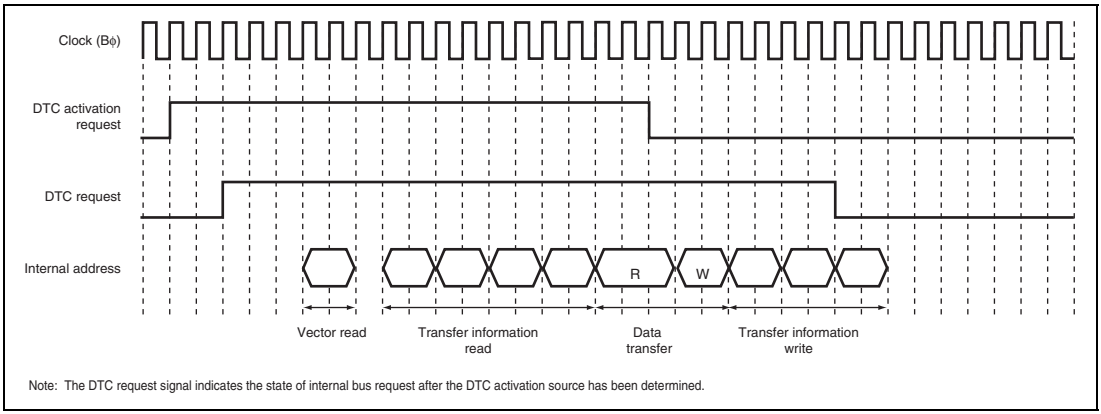
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 9.12 Example of DTC Operation Timing: Chain Transfer (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)



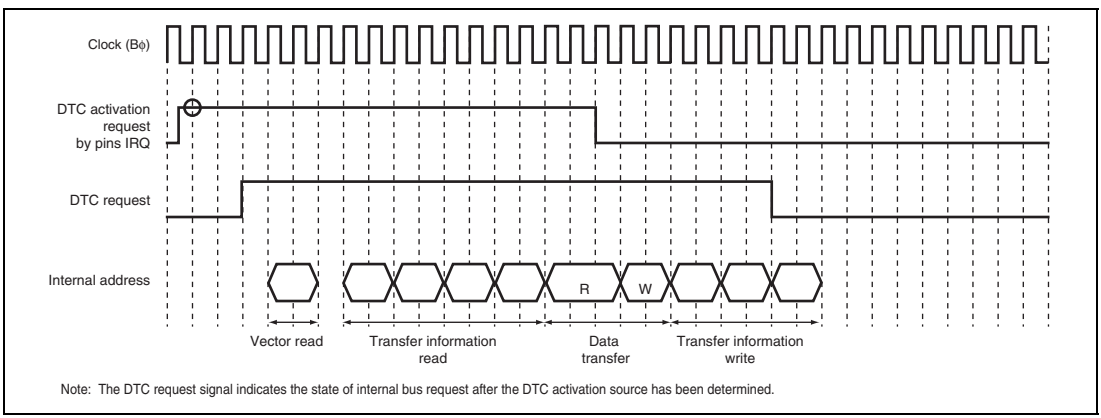
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 9.13 Example of DTC Operation Timing: Short Address Mode and Normal Transfer Mode or Repeat Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)



Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 9.14 Example of DTC Operation Timing: Normal Transfer, Repeat Transfer, DTPR=1 (Activated by On-Chip Peripheral Module; Iϕ: Bϕ: Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)



Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 9.15 Example of DTC Operation Timing: Normal Transfer, Repeat Transfer, (Activated by IRQ; Iϕ: Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)

9.5.8 Number of DTC Execution Cycles

Table 9.9 shows the number of states of execution in a single data transfer by the DTC. For numbers of cycles for access in each state, see section 10, Bus State Controller (BSC).

Table 9.9 Number of States to Execute a Single Data Transfer

Mode	Vector Read I		Transfer Information Read J			Transfer Information Write K			Data Read L	Data Write M	Internal Operation N	
Normal	1	0*1	4	3*4	0*1	3	2*2	1*3	1	1	1	0*1
Repeat	1	0*1	4	3*4	0*1	3	2*2	1*3	1	1	1	0*1
Block transfer	1	0*1	4	3*4	0*1	3	2*2	1*3	1•P	1•P	1	0*1

[Legend]

P: Block size (CRAH and CRAL value)

Notes: 1. When reading of transfer information is skipped
2. When the SAR or DAR is in fixed mode
3. When the SAR and DAR are in fixed mode
4. When short address mode is specified

9.5.9 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occurs. The DTC releases the bus after a vector read, transfer information read, a single data transfer, or transfer information write-back. The DTC does not release the bus mastership during transfer information read, a single data transfer, or write-back of transfer information.

The bus release timing can be specified through the bus function extending register (BSCEHR). For details see section 10.4.8, Bus Function Extending Register (BSCEHR). The difference in bus release timing according to the register setting is summarized in table 9.10. Settings other than shown in the table are prohibited. The value of BSCEHR must not be modified while the DTC is active.

Figure 9.16 is a timing chart showing an example of bus release timing.

Table 9.10 DTC Bus Release Timing

	Bus Function Extending Register (BSCEHR) Setting		Bus Release Timing (O: Bus must be released; x: Bus is not released)				
						After Write-Back of Transfer Information	
	DTLOCK	DTBST	After Vector Read	After Transfer Information Read	After a Single data Transfer	Normal Transfer	Continuous Transfer
Setting 1	0	0	×	×	×	O	O
Setting 2*	0	1	×	×	×	O	×
Setting 3	1	0	O	O	O	O	O

Note: * The following restrictions apply to setting 2.

- The clock setting through the frequency control register (FROCR) must be Iϕ : Bϕ : Pϕ = 8 : 4 : 4, 4 : 2 : 2, 2 : 1 : 1, 4 : 4 : 4, 2 : 2 : 2, or 1 : 1 : 1.
- The vector information must be stored in the on-chip ROM or on-chip RAM (high speed).
- The transfer information must be stored in the on-chip RAM (high speed).
- Transfer must be between the on-chip RAM (high speed/for retention) and an on-chip peripheral module or between the external memory and an on-chip peripheral module.

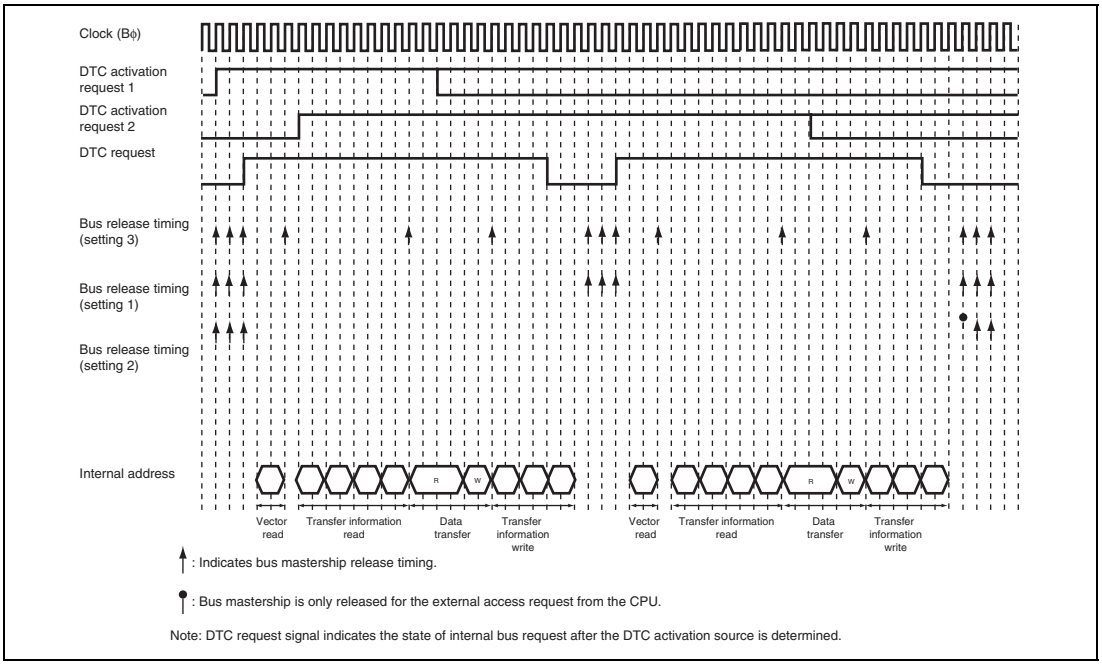


Figure 9.16 Example of DTC Operation Timing: Conflict of Two Activation Requests in Normal Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM (High Speed); Transfer Information is Written in 3 Cycles)

9.5.10 DTC Activation Priority Order

If multiple DTC activation requests are generated while the DTC is inactive, whether to start the transfer from the first activation request* or according to the DTC activation priority can be selected through the DTPR bit setting in the bus function extending register (BSCEHR). If multiple activation requests are generated while the DTC is active, transfer is performed according to the DTC activation priority. Figure 9.17 shows an example of DTC activation according to the priority.

Note: * When one DTC-activation request is generated before another, transfer starts with the first request. When an activation request with a higher priority is generated before a pending DTC request is accepted, transfer starts for the request with higher priority. Timing of DTC request generation varies according to the operating state of internal buses.

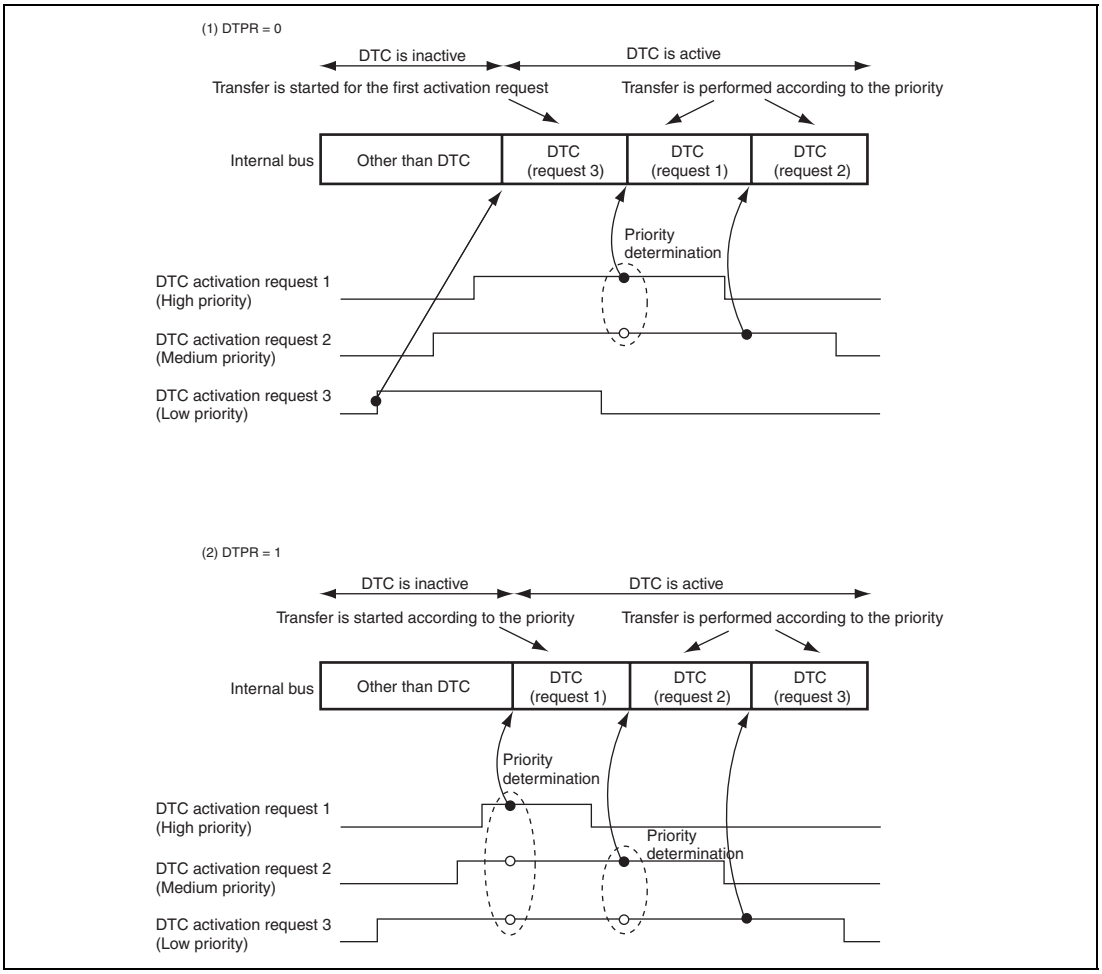
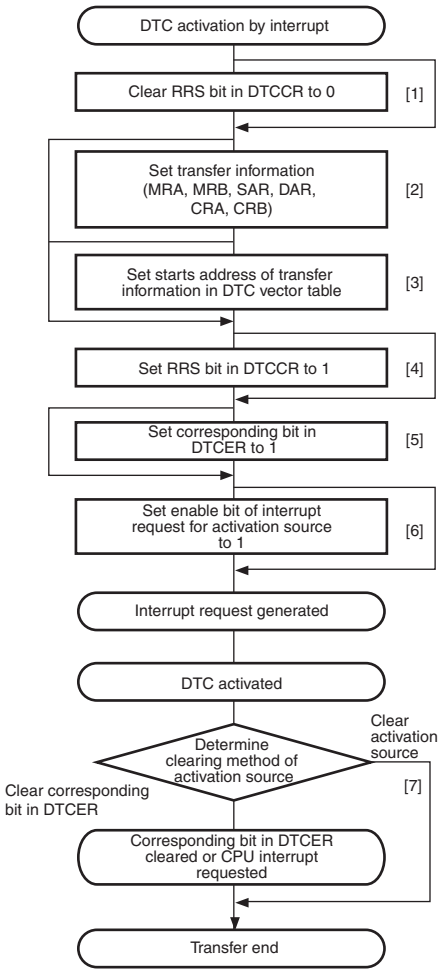


Figure 9.17 Example of DTC Activation According to Priority

9.6 DTC Activation by Interrupt

The procedure for using the DTC with interrupt activation is shown in figure 9.18.



[1] Clearing the RRS bit in DTCCR to 0 clears the read skip flag of transfer information. Read skip is not performed when the DTC is activated after clearing the RRS bit. When updating transfer information, the RRS bit must be cleared.

[2] Set the MRA, MRB, SAR, DAR, CRA, and CRB transfer information in the data area. For details on setting transfer information, see section 9.2, Register Descriptions. For details on location of transfer information, see section 9.4, Location of Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the DTC vector table. For details on setting DTC vector table, see section 9.4, Location of Transfer Information and DTC Vector Table.

[4] Setting the RRS bit to 1 performs a read skip of second time or later transfer information when the DTC is activated consecutively by the same interrupt source. Setting the RRS bit to 1 is always allowed. However, the value set during transfer will be valid from the next transfer.

[5] Set the bit in DTCER corresponding to the DTC activation interrupt source to 1. For the correspondence of interrupts and DTCER, refer to table 9.2. The bit in DTCER may be set to 1 on the second or later transfer. In this case, setting the bit is not needed.

[6] Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated. For details on the settings of the interrupt enable bits, see the corresponding descriptions of the corresponding module.

[7] After the end of one data transfer, the DTC clears the activation source flag or clears the corresponding bit in DTCER and requests an interrupt to the CPU. The operation after transfer depends on the transfer information. For details, see section 9.2, Register Descriptions and figure 9.4.

Figure 9.18 DTC Activation by Interrupt

9.7 Examples of Use of the DTC

9.7.1 Normal Transfer Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal transfer mode (MD1 = MD0 = 0), and byte size (Sz1 = Sz0 = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the RDR address of the SCI in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the transfer information for an RXI interrupt at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the receive end (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

9.7.2 Chain Transfer when Transfer Counter = 0

By executing a second data transfer and performing re-setting of the first data transfer only when the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-Kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 9.19 shows the chain transfer when the counter value is 0.

1. For the first transfer, set the normal transfer mode for input data. Set the fixed transfer source address, CRA = H'0000 (65,536 times), CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for 65,536-transfer units for the first data transfer in a separate area (in the ROM, etc.). For example, if the input buffer is configured at addresses H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat transfer mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper eight bits of DAR in the first transfer information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.
4. Execute the first data transfer 65536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, no interrupt request is sent to the CPU.

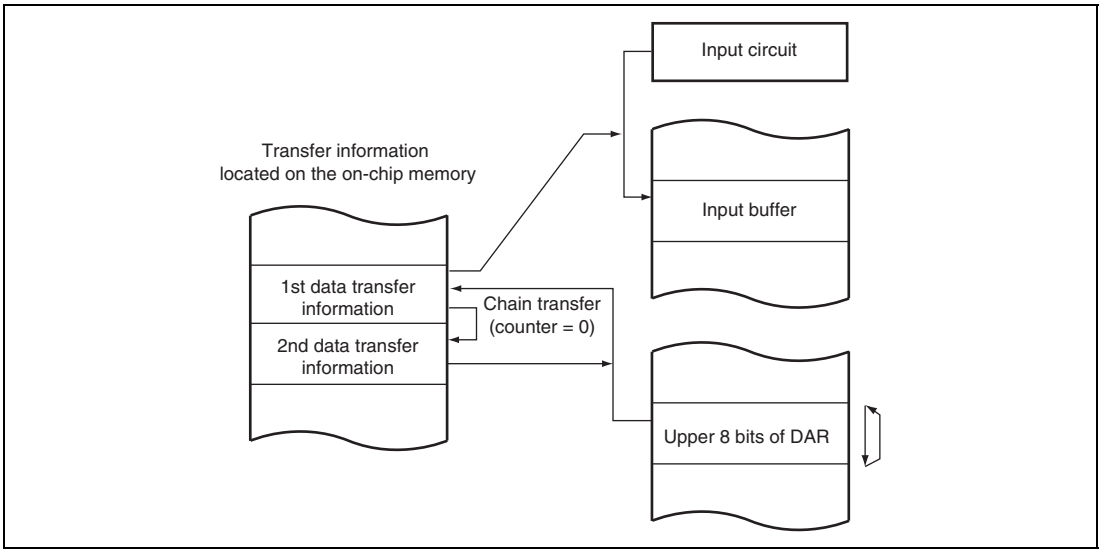


Figure 9.19 Chain Transfer when Transfer Counter = 0

9.8 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or on completion of a single data transfer or a single block data transfer with the DISEL bit set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and priority level control in the interrupt controller. For details, refer to section 7.9, Data Transfer with Interrupt Request Signals.

9.9 Usage Notes

9.9.1 Module Standby Mode Setting

Operation of the DTC can be disabled or enabled by using the standby control register. The initial setting is for operation of the DTC to be disabled. Release from module standby mode enables access to registers. Do not place the DTC in module standby mode while it is active. Before entering software standby mode or module standby mode, all DTCER registers must be cleared. For details, refer to section 32, Power-Down Modes.

9.9.2 On-Chip RAM

Transfer information can be located in on-chip RAM (high speed). In this case, the corresponding RAME bits in RAMCR must not be cleared to 0.

Additionally, do not locate transfer information in the on-chip RAM (for retention).

9.9.3 DTCE Bit Setting

To set a DTCE bit, disable the corresponding interrupt, read 0 from the bit, and then write 1 to it. While DTC transfer is in progress, do not modify the DTCE bits.

9.9.4 Chain Transfer

When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI, SCIF, IIC3, LVDS (SH72315A only), RSPI, RCAN-ET, and A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the relevant register.

Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

9.9.5 Transfer Information Start Address, Source Address, and Destination Address

The transfer information start address to be specified in the vector table should be address 4n. Locate transfer information in the on-chip RAM (high speed) or external memory space. On the other hand, do not locate transfer information in the on-chip RAM (for retention).

9.9.6 Access to DTC Registers through DTC

Do not access the DMAC or DTC registers by using DTC operation. Do not access the DTC registers by using DMAC operation.

9.9.7 Notes on Using an IRQ Interrupt as a DTC Activation Source

- Do not use the given IRQ interrupt to release this LSI from software standby.
- Do not allow attempts at DTC transfer on edges of the IRQ signal generated while this LSI is on software standby.
- When a low level on the IRQ pin is to be detected, if the end of DTC transfer is used to request an interrupt to the CPU (transfer counter = 0 or DISEL = 1), the IRQ signal must be held low until the CPU accepts the interrupt.

9.9.8 Note on SCI or SCIF as DTC Activation Sources

When the TXI interrupt from the SCI is specified as a DTC activation source, the TEND flag in the SCI must not be used as the transfer end flag.

When the TXI interrupt from the SCIF is specified as a DTC activation source, the TEND flag in the SCIF must not be used as the transfer end flag.

9.9.9 Clearing Interrupt Source Flag

The interrupt source flag set when the DTC transfer is completed should be cleared in the interrupt handler in the same way as for general interrupt source flags. For details, refer to section 7.10, Usage Note.

9.9.10 Conflict between NMI Interrupt and DTC Activation

When a conflict occurs between the generation of the NMI interrupt and the DTC activation, the NMI interrupt has priority. Thus the ERR bit is set to 1 and the DTC is not activated.

It takes 3Bϕ + 2Pϕ for checking DTC stop by the NMI, 3Bϕ + 2Pϕ for checking DTC activation by the IRQ, and 1Bϕ + 1Pϕ to 4Bϕ + 1Pϕ for checking DTC activation by the peripheral module.

9.9.11 Operation when a DTC Activation Request has been Cancelled

Once DTC has accepted an activation request, the next activation request will not be accepted until the sequence of the DTC transaction has finished up to the end of write-back.

Section 10 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for various types of memory that is connected to the external address space and external devices. BSC functions enable this LSI to connect directly with SRAM, SDRAM, and other memory storage devices, and external devices.

10.1 Features

The BSC has the following features.

1. External address space
   - A maximum of 64 Mbytes for each of areas CS0 to CS7.
   - Can specify the normal space interface, SRAM interface with byte selection, burst ROM (clock synchronous or asynchronous), MPX-I/O, and SDRAM for each address space.
   - Can select the data bus width (8, 16, or 32 bits) for each address space.
   - Controls insertion of wait cycles for each address space.
   - Controls insertion of wait cycles for each read access and write access.
   - Can set independent idle cycles during the continuous access for five cases: read-write (in same space/different spaces), read-read (in same space/different spaces), the first cycle is a write access.
2. Normal space interface
   - Supports the interface that can directly connect to the SRAM.
3. Burst ROM interface (clock asynchronous)
   - High-speed access to the ROM that has the page mode function.
4. MPX-I/O interface
   - Can directly connect to a peripheral LSI that needs an address/data multiplexing.
5. SDRAM interface
   - Can set the SDRAM in up to two CS areas.
   - Multiplex output for row address/column address.
   - Efficient access by single read/single write.
   - High-speed access in bank-active mode.
   - Supports an auto-refresh and self-refresh.
   - Supports low-frequency and power-down modes.
   - Issues MRS and EMRS commands.

6. SRAM interface with byte selection
   — Can connect directly to a SRAM with byte selection.
7. Burst ROM interface (clock synchronous)
   — Can connect directly to a ROM of the clock-synchronous type.
8. Bus arbitration
   — Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.
9. Refresh function
   — Supports the auto-refresh and self-refresh functions.
   — Specifies the refresh interval using the refresh counter and clock selection.
   — Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).
10. Usage as interval timer for refresh counter
   — Generates an interrupt request at compare match.

Figure 10.1 shows a block diagram of the BSC.

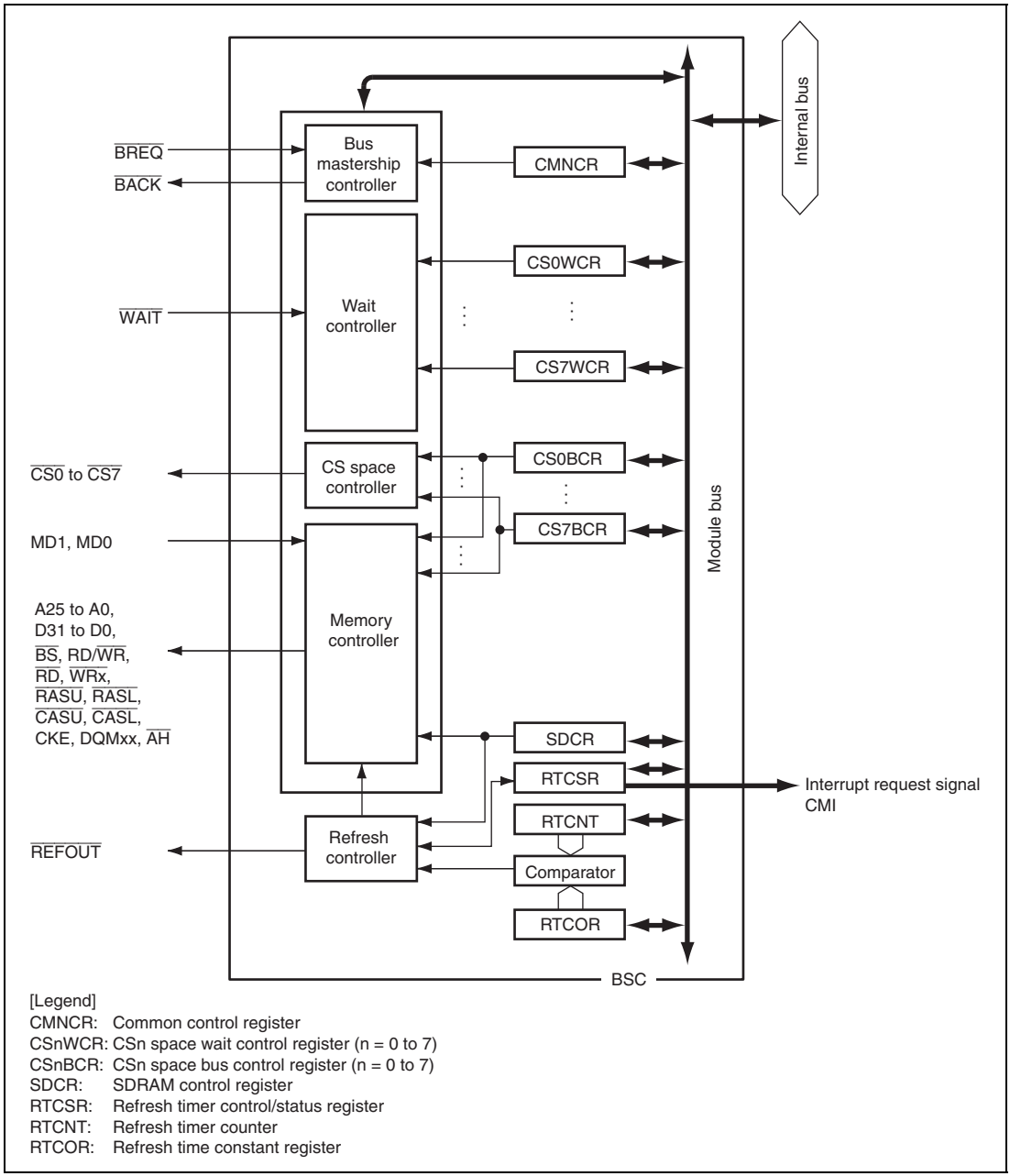


Figure 10.1 Block Diagram of BSC

10.2 Input/Output Pins

Table 10.1 shows the pin configuration of the BSC.

Table 10.1 Pin Configuration

Name	I/O	Function
A25 to A0	Output	Address bus
D31 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$ to $\overline{CS7}$	Output	Chip select
RD/$\overline{WR}$	Output	Read/write Connects to $\overline{WE}$ pins when SDRAM or SRAM with byte selection is connected.
$\overline{RD}$	Output	Read pulse signal (read data output enable signal) Functions as a strobe signal for indicating memory read cycles when PCMCIA is used.
$\overline{AH}$	Output	A signal used to hold an address when MPX-I/O is in use
$\overline{WRHH}$/DQMUU	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when SRAM with byte selection is connected. Functions as the select signals for D31 to D24 when SDRAM is connected.
$\overline{WRHL}$/DQMUL	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when SRAM with byte selection is connected. Functions as the select signals for D23 to D16 when SDRAM is connected.
$\overline{WRH}$/DQMLU	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D15 to D8 when SDRAM is connected.

Name	I/O	Function
$\overline{\text{WRL}}$/DQMLL	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D7 to D0 when SDRAM is connected.
$\overline{\text{RASL}}$, $\overline{\text{RASU}}$	Output	Connected to $\overline{\text{RAS}}$ pin when SDRAM is connected.
$\overline{\text{CASL}}$, $\overline{\text{CASU}}$	Output	Connected to $\overline{\text{CAS}}$ pin when SDRAM is connected.
CKE	Output	Connected to CKE pin when SDRAM is connected.
$\overline{\text{WAIT}}$	Input	External wait input
$\overline{\text{BREQ}}$	Input	Bus request input
$\overline{\text{BACK}}$	Output	Bus enable output
$\overline{\text{REFOUT}}$	Output	Refresh request output in bus-released state
MD0, MD1	Input	Selects bus width (16 or 32 bits) of CS0 space. Selects initial bus width for CS1 to CS7 spaces. It also selects the on-chip ROM enabled or disabled mode and external bus access enabled or disabled mode.

10.3 Area Overview

10.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into external address space and on-chip spaces (on-chip ROM, on-chip RAM, on-chip peripheral modules, and reserved areas) according to the upper bits of the address.

The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is listed below.

Table 10.2 Address Map in On-Chip ROM-Enabled Mode (SH72315A and SH72315L)

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'000F FFFF	On-chip ROM	On-chip ROM	1 Mbytes
H'0010 0000 to H'01FF FFFF	Other	Reserved area	—
H'0200 0000 to H'03FF FFFF	CS0	Normal space, burst ROM (asynchronous or synchronous)	32 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM (high-speed), reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip RAM (for data retention), on-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 31, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 34, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

Table 10.3 Address Map in On-Chip ROM-Enabled Mode (SH72314L)

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'000B FFFF	On-chip ROM	On-chip ROM	768 Kbytes
H'00C0 0000 to H'01FF FFFF	Other	Reserved area	—
H'0200 0000 to H'03FF FFFF	CS0	Normal space, burst ROM (asynchronous or synchronous)	32 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM (high-speed), reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip RAM (for data retention), on-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 31, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 34, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

Table 10.4 Address Map in On-Chip ROM-Disabled Mode

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'03FF FFFF	CS0	Normal space, burst ROM (asynchronous or synchronous)	64 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM (high-speed), reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip RAM (for data retention), on-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 31, On-Chip RAM. For the on-chip I/O register space, access the addresses shown in section 34, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

10.3.2 Setting Operating Modes

This LSI can set the following modes of operation at the time of power-on reset using the external pins.

- Single-Chip Mode

  In single-chip mode, the LSI is activated by the on-chip ROM program after a power-on reset. The external bus cannot be accessed. The BSC module enters the module standby state to reduce power consumption.

  The address, data, bus control pins used in external bus accessible mode (MCU extended mode 0, 1, or 2) can be used as the port function pins in single-chip mode.

- On-Chip ROM-Enabled Mode (MCU Extended Mode 2)/On-Chip ROM-Disabled Mode (MCU Extended Mode 0 or 1)

  In on-chip ROM-enabled mode (MCU extended mode 2), the LSI is activated by the on-chip ROM program after a power-on reset.

  In on-chip ROM-disabled mode (MCU Extended mode 1 or 0), the LSI is activated by the program stored in the external memory allocated to the CS 0 space. In this case, a ROM is assumed for the external memory of the CS0 space. Therefore, minimum functions are provided for the pins including address bus, data bus, CS0, and RD. Although $\overline{BS}$, RD/$\overline{WR}$, $\overline{WRxx}$, and other pins are shown in the examples of access waveforms in this section, these are examples when pin settings are performed by the pin function controller. For details, see section 22, Pin Function Controller (PFC). Do not perform any operation except for the CS0 space read access until the pin settings by the program is completed.

- Initial Settings of Data Bus Widths for CS0 to CS7 Spaces

  In on-chip ROM-disabled mode (MCU extended mode 0 or 1), the data bus width of CS0 space and the initial data bus width of CS1 to CS7 spaces can be selected at a power-on reset. The data bus width can be set using the MD0 pin as 16 bits (MD0 = 0) or 32 bits (MD1 = 1). The data bus width of CS0 area cannot be changed after a power-on reset, but the data bus widths of CS1 to CS7 areas can be changed by register settings in the program.

  In on-chip ROM-enabled mode (MCU extended mode 2), the initial data bus width of CS0 to CS7 spaces is 16 bits. The data bus width of CS0 to CS7 spaces can be changed by register settings in the program. Note that data bus widths will be restricted depending on memory types.

For details of mode settings, see section 4, MCU Operating Modes.

10.4 Register Descriptions

The BSC has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Do not access spaces other than CS0 space until settings of the connected memory interface are completed.

Table 10.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	CMNCR	R/W	H'00001010	H'FFFC0000	32
CS0 space bus control register	CS0BCR	R/W	H'36DB0400*	H'FFFC 0004	32
CS1 space bus control register	CS1BCR	R/W	H'36DB0400*	H'FFFC 0008	32
CS2 space bus control register	CS2BCR	R/W	H'36DB0400*	H'FFFC 000C	32
CS3 space bus control register	CS3BCR	R/W	H'36DB0400*	H'FFFC 0010	32
CS4 space bus control register	CS4BCR	R/W	H'36DB0400*	H'FFFC 0014	32
CS5 space bus control register	CS5BCR	R/W	H'36DB0400*	H'FFFC 0018	32
CS6 space bus control register	CS6BCR	R/W	H'36DB0400*	H'FFFC 001C	32
CS7 space bus control register	CS7BCR	R/W	H'36DB0400*	H'FFFC 0020	32
CS0 space wait control register	CS0WCR	R/W	H'00000500	H'FFFC0028	32
CS1 space wait control register	CS1WCR	R/W	H'00000500	H'FFFC002C	32
CS2 space wait control register	CS2WCR	R/W	H'00000500	H'FFFC0030	32
CS3 space wait control register	CS3WCR	R/W	H'00000500	H'FFFC0034	32
CS4 space wait control register	CS4WCR	R/W	H'00000500	H'FFFC0038	32
CS5 space wait control register	CS5WCR	R/W	H'00000500	H'FFFC003C	32
CS6 space wait control register	CS6WCR	R/W	H'00000500	H'FFFC0040	32
CS7 space wait control register	CS7WCR	R/W	H'00000500	H'FFFC0044	32
SDRAM control register	SDCR	R/W	H'00000000	H'FFFC004C	32
Refresh timer control/status register	RTCSR	R/W	H'00000000	H'FFFC0050	32
Refresh timer counter	RTCNT	R/W	H'00000000	H'FFFC0054	32
Refresh time constant register	RTCOR	R/W	H'00000000	H'FFFC0058	32
Bus function extending register	BSCEHR	R/W	H'0000	H'FFFE3C1A	16

Note: * Indicates the initial value when the LSI is activated in MCU extended mode 0 or 2. When activated in MCU extended mode 1, the initial value will be H'36DB 0600.

10.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each CS space. This register is initialized to H'00001010 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	BLOCK	DPRTY[1:0]		DMAIW[2:0]			DMA IWA	-	-	HIZCK	HIZ MEM	HIZ CNT
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
11	BLOCK	0	R/W	Bus Lock Specifies whether or not the $\overline{\text{BREQ}}$ signal is received. 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.
10, 9	DPRTY[1:0]	00	R/W	DMA Burst Transfer Priority Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer. 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer. 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer. 11: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	DMAIW[2:0]	000	R/W	Wait states between access cycles when DMA single address transfer is performed. Specify the number of idle cycles to be inserted after an access to an external device with DACK when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
5	DMAIWA	0	R/W	Method of inserting wait states between access cycles when DMA single address transfer is performed. Specifies the method of inserting the idle cycles specified by the DMAIW[2:0] bit. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with DACK drove it. However, when the external device with DACK drives the data bus continuously, idle cycles are not inserted. Setting this bit will make this LSI insert the idle cycles after an access to an external device with DACK, even when the continuous access cycles to an external device with DACK are performed. 0: Idle cycles inserted when another device drives the data bus after an external device with DACK drove it. 1: Idle cycles always inserted after an access to an external device with DACK
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	HIZCK	0	R/W	High-Z CK Control Specifies the state in CK software standby mode and when bus mastership is released. 0: CK is in high impedance state in software standby mode and bus-released state. 1: CK is driven in software standby mode and bus-released state.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in software standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WRxx}$/DQMxx, $\overline{AH}$, and $\overline{RD}$. At bus-released state, these pins are in high-impedance state regardless of the setting value of the HIZMEM bit. 0: High impedance in software standby mode. 1: Driven in software standby mode
0	HIZCNT	0	R/W	High-Z Control Specifies the state in software standby mode and bus-released state for CKE, $\overline{RASL}$, $\overline{CASL}$, $\overline{RASU}$, and $\overline{CASU}$. 0: CKE, $\overline{RASL}$, $\overline{CASL}$, $\overline{RASU}$, and $\overline{CASU}$ are in high-impedance state in software standby mode and bus-released state. 1: CKE, $\overline{RASL}$, $\overline{CASL}$, $\overline{RASU}$, and $\overline{CASU}$ are driven in software standby mode and bus-released state.

10.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7)

CSnBCR is a 32-bit readable/writable register that specifies the type of memory connected to a space, data bus width of an area, and the number of waits between access cycles. This register is initialized to H'36DB0x00 by a power-on reset and retains the value by a manual reset and in software standby mode.

Do not access external memory other than CS0 space until CSnBCR initial setting is completed.

Idle cycles may be inserted even when they are not specified. For details, see section 10.5.10, Wait between Access Cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Initial value:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TYPE[2:0]			-	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1*	0*	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW[2:0]	011	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27 to 25	IWRWD[2:0]	011	R/W	Idle Cycles for Another Space Read-Write Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous access cycles switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
24 to 22	IWRWS[2:0]	011	R/W	Idle Cycles for Read-Write in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	IWRRD[2:0]	011	R/W	Idle Cycles for Read-Read in Another Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
18 to 16	IWRRS[2:0]	011	R/W	Idle Cycles for Read-Read in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	TYPE[2:0]	000	R/W	Specify the type of memory connected to a space. 000: Normal space 001: Burst ROM (clock asynchronous) 010: MPX-I/O 011: SRAM with byte selection 100: SDRAM 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Burst ROM (clock synchronous) For details of memory type in each CS space, see tables 10.2 to 10.4. Note: To connect a burst ROM to CS0 space, set TYPE[2:0] to 001 after modifying CS0WCR as required for the used ROM.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10, 9	BSZ[1:0]	10*	R/W	Data Bus Width Specification Specify the data bus widths of spaces. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: 32-bit size For MPX-I/O, selects bus width by address. Notes: 1. If CS5 space is specified as MPX-I/O, the bus width can be specified as 8 bits or 16 bits by the address according to the SZSEL bit in CS5WCR by specifying the BSZ[1:0] bits to 11. The fixed bus width can be specified as 8 bits or 16 bits. 2. The initial data bus widths for CS0 to CS7 spaces are specified by external pins. In on-chip ROM-disabled mode (MCU extended mode 0 or 1), writing to the BSZ1 and BSZ0 bits in CS0BCR is ignored, but the bus width settings in CS1BCR to CS7BCR can be modified. In on-chip ROM-enabled mode (MCU extended mode 2), the bus width settings in CS0BCR to CS7BCR can be modified. 3. When CS2 or CS3 space is specified as SDRAM space, the bus width can be set as 16 or 32 bits. 4. If CS0 space is specified as clock-synchronous burst ROM space, the bus width can be set as 16 or 32 bits. 5. If the bus width of at least one space is specified as 8 bits, CS7 space cannot be used. To use CS7 space, the bus widths of all the CS spaces should be specified as 16 bits or 32 bits.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Initial value in MCU extended mode 0 or 2. In MCU extended mode 1, the initial value is 11.

10.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7)

CSnWCR specifies various wait cycles for memory access. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target CS space. Specify CSnBCR first, then specify CSnWCR.

CSnWCR is initialized to H'00000500 by a power-on reset and retains the value by a manual reset and in software standby mode.

(1) Normal Space, SRAM with Byte Selection, MPX-I/O

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-*	-*	-	-	-*	-*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	—*	All 0	R/W	Reserved Set these bits to 0 when the normal space is selected.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	—*	All 0	R/W	Reserved Set these bits to 0 when the normal space is selected.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS0}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address and $\overline{CS0}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS0}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Note * To connect the burst ROM to the CS0 space and switch to the burst ROM interface after activation in ROM-disabled mode (MCU extended mode 0 or 1), set the TYPE[2:0] bits in CS0BCR after setting the burst number by the bits 20 and 21 and the burst wait cycle number by the bits 16 and 17. Do not write 1 to the reserved bits other than above bits.

- CS1WCR, CS4WCR, CS7WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WRxx}}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{\text{WRxx}}$ Negation to Address, $\overline{\text{CSn}}$ Negation Specify the number of delay cycles from RD and $\overline{\text{WRxx}}$ negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS2WCR, CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	WR[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS5WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SZSEL	MPXW/ BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SZSEL	0	R/W	MPX-I/O Interface Bus Width Specification Specifies an address to select the bus width when the BSZ[1:0] of CS5BCR are specified as 11. This bit is valid only when CS5 space is specified as MPX-I/O. 0: Selects the bus width by address A14 1: Selects the bus width by address A21 The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized below.

SZSEL	A14	A21	Bus Width
0	0	Not affected	8 bits
0	1	Not affected	16 bits
1	Not affected	0	8 bits
1	Not affected	1	16 bits

Bit	Bit Name	Initial Value	R/W	Description
20	MPXW	0	R/W	MPX-I/O Interface Address Wait This bit setting is valid only when CS5 space is specified as MPX-I/O. Specifies the address cycle insertion wait for MPX-I/O interface. 0: Inserts no wait cycle 1: Inserts 1 wait cycle
	BAS	0	R/W	SRAM with Byte Selection Byte Access Select This bit setting is valid only when CS5 space is specified as MPX-I/O. Specifies the address is specified as SRAM with byte selection. Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS5}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS5}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS5}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS6WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WRxx}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WRxx}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WRxx}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS6}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address, $\overline{CS6}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WN	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification of this bit is valid even when the number of access wait cycles is 0. 0: The external wait input is valid 1: The external wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRxx}$ negation to address, and $\overline{CS6}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(2) Burst ROM (Clock Asynchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11.

Bus Width	BST[1:0]	Burst count
8 bits	x0	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst

Bit	Bit Name	Initial Value	R/W	Description
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		W[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11.

Bus Width	BST[1:0]	Burst count
8 bits	x0	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst

Bit	Bit Name	Initial Value	R/W	Description
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(3) SDRAM*

- CS2WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	A2CL[1:0]		-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A2CL[1:0]	10	R/W	CAS Latency for CS2 Space Specify the CAS latency for CS2 space. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * If only one CS space is connected to the SDRAM, specify CS3 space. In this case, specify CS2 space as normal space or SRAM with byte selection.

- CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	WTRP[1:0]*		-	WTRCD[1:0]*		-	A3CL[1:0]		-	-	TRWL[1:0]*		-	WTRC[1:0]*	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14, 13	WTRP[1:0]*	00	R/W	Number of Auto-Precharge Completion Wait Cycles Specify the number of minimum precharge completion wait cycles as shown below. • From the start of auto-precharge and issuing of ACTV command for the same bank • From issuing of the PRE/PALL command to issuing of the ACTV command for the same bank • Till entering power-down mode or deep power-down mode • From the issuing of PALL command to issuing REF command in auto-refresh mode • From the issuing of PALL command to issuing SELF command in self-refresh mode The setting for CS2 and CS3 spaces is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11, 10	WTRCD[1:0]	01	R/W	Number of Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command Specify the minimum number of wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. The setting for CS2 and CS3 spaces is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A3CL[1:0]	10	R/W	CAS Latency for CS3 Space Specify the CAS latency for CS3 space. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	TRWL[1:0]*	00	R/W	Number of Auto-Precharge Startup Wait Cycles Specify the number of minimum auto-precharge startup wait cycles as shown below. • Cycle number from the issuance of the WRITA command by this LSI until the completion of auto-precharge in the SDRAM. Equivalent to the cycle number from the issuance of the WRITA command until the issuance of the ACTV command. Confirm that how many cycles are required between the WRITE command receive in the SDRAM and the auto-precharge activation, referring to each SDRAM data sheet. And set the cycle number so as not to exceed the cycle number specified by this bit. • Cycle number from the issuance of the WRITA command until the issuance of the PRE command. This is the case when accessing another low address in the same bank in bank active mode. The setting for CS2 and CS3 spaces is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	WTRC[1:0]*	00	R/W	Number of Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the periods shown below. • From the issuance of the REF command until the issuance of the ACTV/REF/MRS command • From releasing self-refresh until the issuance of the ACTV/REF/MRS command. The setting for CS2 and CS3 spaces is common. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

Note: * If both CS2 and CS3 spaces are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both spaces in common.
If only one space is connected to the SDRAM, specify CS3 space. In this case, specify CS2 space as normal space or SRAM with byte selection.

(4) Burst ROM (Clock Synchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

SDCR is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	A2ROW[1:0]		-	A2COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV	-	-	-	A3ROW[1:0]		-	A3COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20, 19	A2ROW[1:0]	00	R/W	Number of Bits of Row Address for CS2 Space Specify the number of bits of row address for CS2 space. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for CS2 Space Specify the number of bits of column address for CS2 space. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RFSH or RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address, and write data for SDRAM and the latch timing of read data from SDRAM. Setting this bit makes the hold time for command, address, write and read data extended for half cycle (output or read at the falling edge of CK). This mode is suitable for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM is output at the rising edge of CK. Read data from SDRAM is latched at the rising edge of CK. 1: Command, address, and write data for SDRAM is output at the falling edge of CK. Read data from SDRAM is latched at the falling edge of CK.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh

Bit	Bit Name	Initial Value	R/W	Description
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed
9	PDOWN	0	R/W	Power-Down Mode Specifies whether the SDRAM will enter power-down mode after the access to the SDRAM. With this bit being set to 1, after the SDRAM is accessed, the CKE signal is driven low and the SDRAM enters power-down mode. 0: The SDRAM does not enter power-down mode after being accessed. 1: The SDRAM enters power-down mode after being accessed.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be set only in CS3 space, and the bus width can be set as 16 or 32 bits. When both the CS2 and CS3 spaces are set to SDRAM, specify auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	A3ROW[1:0]	00	R/W	Number of Bits of Row Address for CS3 Space Specify the number of bits of the row address for CS3 space. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	A3COL[1:0]	00	R/W	Number of Bits of Column Address for CS3 Space Specify the number of bits of the column address for CS3 space. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

10.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM. RTCSR is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

When RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

The phase of the clock for incrementing the count in the refresh timer counter (RTCNT) is adjusted only by a power-on reset. Note that there is an error in the time until the compare match flag is set for the first time after the timer is started with the CKS[2:0] bits being set to a value other than B'000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	CKS[2:0]			RRC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables CMF interrupt requests when the CMF bit in RTCSR is set to 1. 0: Disables CMF interrupt requests. 1: Enables CMF interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	CKS[2:0]	000	R/W	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2 to 0	RRC[2:0]	000	R/W	Refresh Count Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long. 000: 1 time 001: 2 times 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

10.4.6 Refresh Timer Counter (RTCNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS[2:0] in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This counter is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Counter

10.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

The $\overline{\text{REFOUT}}$ signal can be asserted when a refresh request is generated while the bus is released. For details, see the description of Relationship between Refresh Requests and Bus Cycles in section 10.5.6 (9), Relationship between Refresh Requests and Bus Cycles, and section 10.5.11, Bus Arbitration.

When the CMIE bit in RTCSR is set to 1, an interrupt request is issued by this matching signal. The request continues to be output until the CMF bit in RTCSR is cleared. Clearing the CMF bit only affects the interrupt request and does not clear the refresh request. Therefore, a combination of refresh request and interval timer interrupt can be specified so that the number of refresh requests are counted by using timer interrupts while refresh is performed periodically.

When RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This register is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Register

10.4.8 Bus Function Extending Register (BSCEHR)

BSCEHR is a 16-bit register that specifies the timing of DTC bus release. It is used to give priority to DTC transfer or reduce the number of cycles in which the DTC is active.

For the differences in DTC operation according to the combinations of the DTLOCK and DTBST bit settings, refer to section 9.5.9, DTC Bus Release Timing.

Setting the DTSA bit enables DTC short address mode. For details of the short address mode, see section 9.4, Location of Transfer Information and DTC Vector Table.

The DTPR bit selects the DTC activation priority used when multiple DTC activation requests are generated before DTC activation.

Do not modify this register while the DTC is active.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT LOCK	-	-	-	DTBST	DTSA	-	DTPR	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	DTLOCK	0	R/W	DTC Lock Enable Specifies the timing of DTC bus release. 0: The DTC releases the bus after write-back of transfer information is completed. 1: The DTC releases the bus after vector read, after transfer information read, after a single data transfer, or after write-back of transfer information.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11	DTBST	0	R/W	DTC Burst Enable Selects whether the DTC continues operation without releasing the bus when multiple DTC activation requests are generated. 0: The DTC releases the bus every time a DTC activation request has been processed. 1: The DTC continues operation without releasing the bus until all DTC activation requests have been processed. Notes: When this bit is set to 1, the following restrictions apply. 1. Clock setting through the frequency control register (FRQCR) must be Iϕ : Bϕ : Pϕ = 8 : 4 : 4, 4 : 2 : 2, 2 : 1 : 1, 4 : 4 : 4, 2 : 2 : 2, or 1 : 1 : 1 2. The vector information must be stored in the on-chip ROM or on-chip RAM (high-speed). 3. The transfer information must be stored in the on-chip RAM (high-speed). 4. Transfer must be between the on-chip RAM (high-speed/for data retention) and an on-chip peripheral module or between the external memory and an on-chip peripheral module.
10	DTSA	0	R/W	DTC Short Address Mode Selects the short address mode in which only three longwords are required for DTC transfer information read. 0: Four longwords are read as the transfer information. The transfer information is arranged as shown in the figure for normal mode in figure 9.2. 1: Three longwords are read as the transfer information. The transfer information is arranged as shown in the figure for short address mode in figure 9.2. Note: The short address mode can be used only for transfer between an on-chip peripheral module and the on-chip RAM (high-speed/for data retention) because the upper eight bits of SAR and DAR are assumed as all 1s.

Bit	Bit Name	Initial Value	R/W	Description
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	DTPR	0	R/W	DTC Activation Priority Selects whether to start transfer from the first DTC activation request or according to the DTC activation priority when multiple DTC activation requests are generated before the DTC is activated. For details, see section 9.5.10, DTC Activation Priority Order. 0: Starts transfer from the DTC activation request generated first. 1: Starts transfer according to the DTC activation priority. Notes: When this bit is set to 1, the following restrictions apply. 1. The vector information must be stored in the on-chip ROM or on-chip RAM (high-speed). 2. The transfer information must be stored in the on-chip RAM (high-speed). 3. The function for skipping the transfer information read step is always disabled. 4. The DTLOCK bit should not be set to 1.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.5 Operation

10.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian in which the 0 address is the most significant byte (MSB) in the byte data.

For normal memory, SRAM with byte selection, and clock asynchronous burst ROM, the data bus width can be selected from three widths (8, 16, and 32 bits). For SDRAM and clock synchronous burst ROM, the data bus width can be selected from two widths (16 and 32 bits). For MPX-I/O, the data bus width is fixed at 8 bits or 16 bits, or 8 bits or 16 bits can be selected by the access address. Data alignment is performed in accordance with the data bus width of the device. This also means that when longword data is read from a byte-width device, the read operation must be done four times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 10.6 to 10.8 show the relationship between device data width and access unit.

Table 10.6 32-Bit External Device Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WRHH}}$, DQMUU	$\overline{\text{WRHL}}$, DQMUL	$\overline{\text{WRH}}$, DQMLU	$\overline{\text{WRL}}$, DQMLL
Byte access at 0	Data 7 to 0	—	—	—	Assert	—	—	—
Byte access at 1	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 2	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Word access at 2	—		Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 10.7 16-Bit External Device Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WRHH}}$, DQMUU	$\overline{\text{WRHL}}$, DQMUL	$\overline{\text{WRH}}$, DQMLU	$\overline{\text{WRL}}$, DQMLL
Byte access at 0		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert
	2nd time at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert

Table 10.8 8-Bit External Device Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WRHH}}$, DQMUU	$\overline{\text{WRHL}}$, DQMUL	$\overline{\text{WRH}}$, DQMLU	$\overline{\text{WRL}}$, DQMLL
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	1st time at 0	—	—	—	Data 15 to 8	—	—	—	Assert
	2nd time at 1	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 2	1st time at 2	—	—	—	Data 15 to 8	—	—	—	Assert
	2nd time at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Longword access at 0	1st time at 0	—	—	—	Data 31 to 24	—	—	—	Assert
	2nd time at 1	—	—	—	Data 23 to 16	—	—	—	Assert
	3rd time at 2	—	—	—	Data 15 to 8	—	—	—	Assert
	4th time at 3	—	—	—	Data 7 to 0	—	—	—	Assert

10.5.2 Normal Space Interface

(1) Basic Timing

For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 10.5.8, SRAM Interface with Byte Selection. Figure 10.2 shows the basic timings of normal space access. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle.

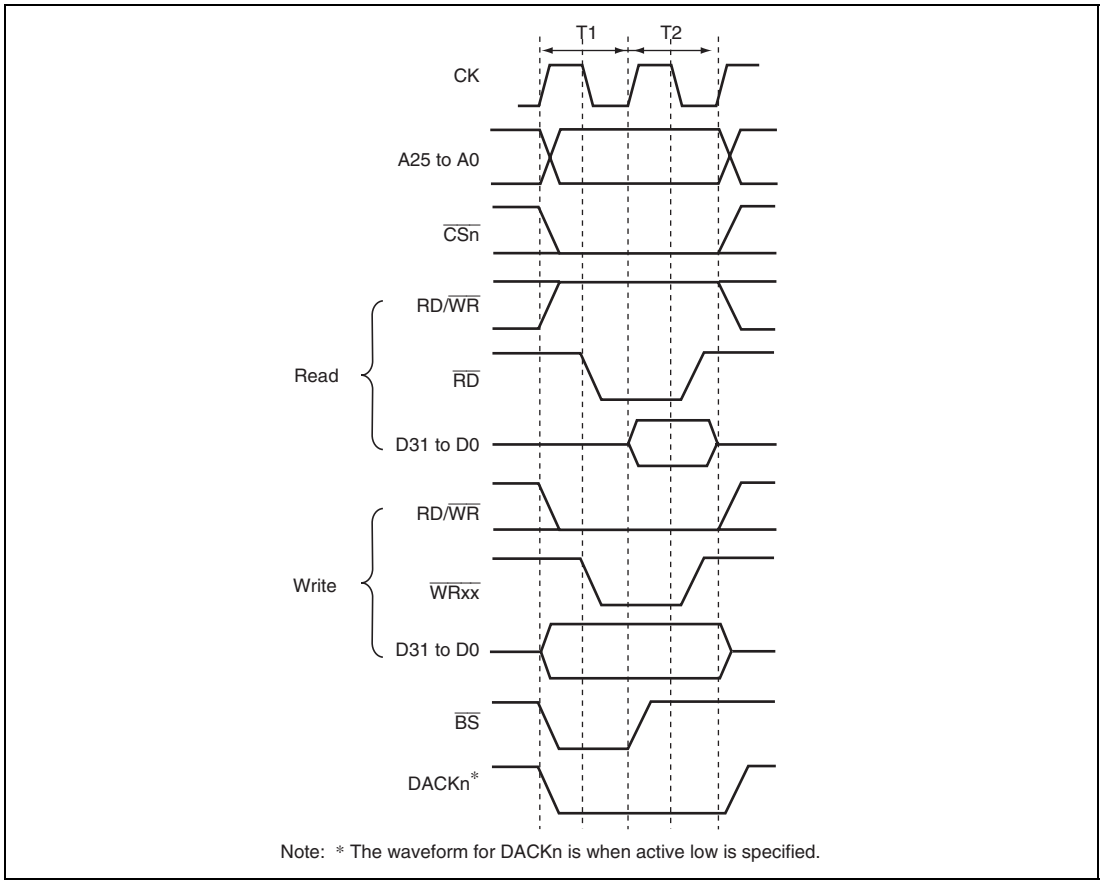


Figure 10.2 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device and 16 bits are always read in case of a 16-bit device. When writing, only the $\overline{\text{WRxx}}$ signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{\text{RD}}$ when a buffer is established in the data bus. The RD/$\overline{\text{WR}}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 10.3 and 10.4 show the basic timings of normal space access. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to evaluate the external wait (figure 10.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 10.4).

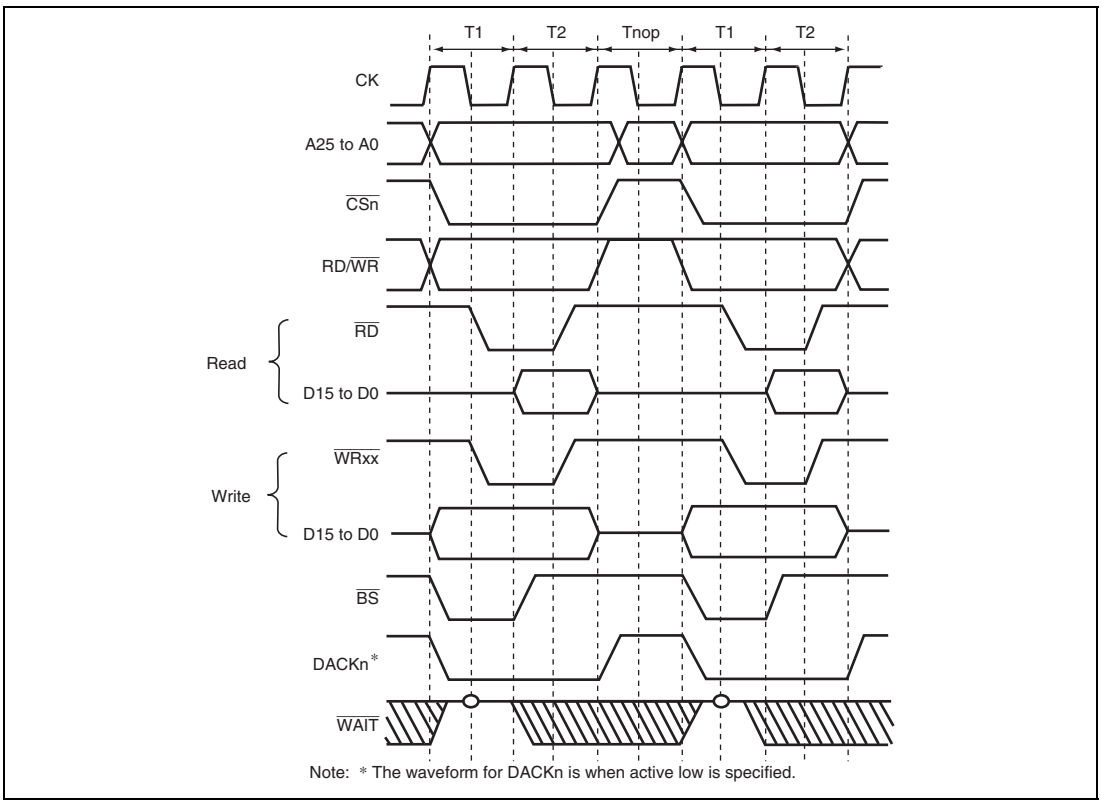


Figure 10.3 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

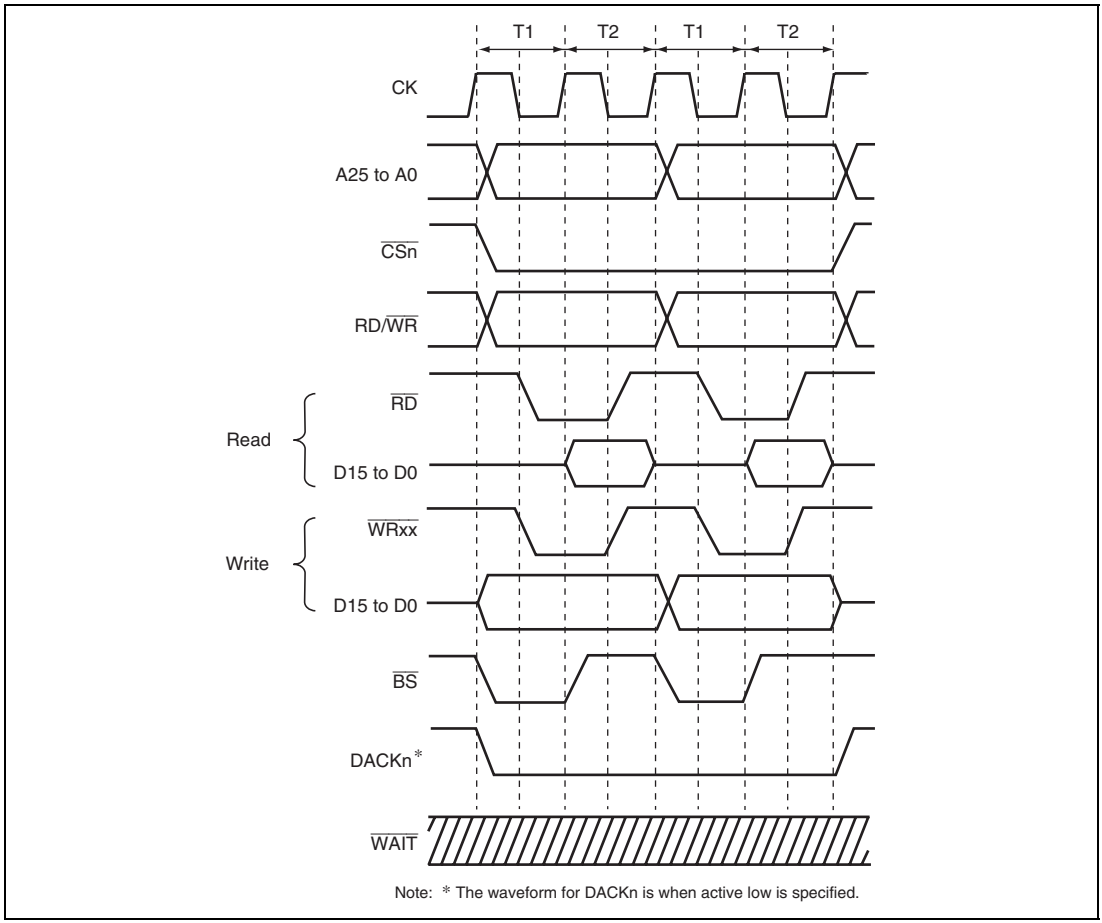


Figure 10.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

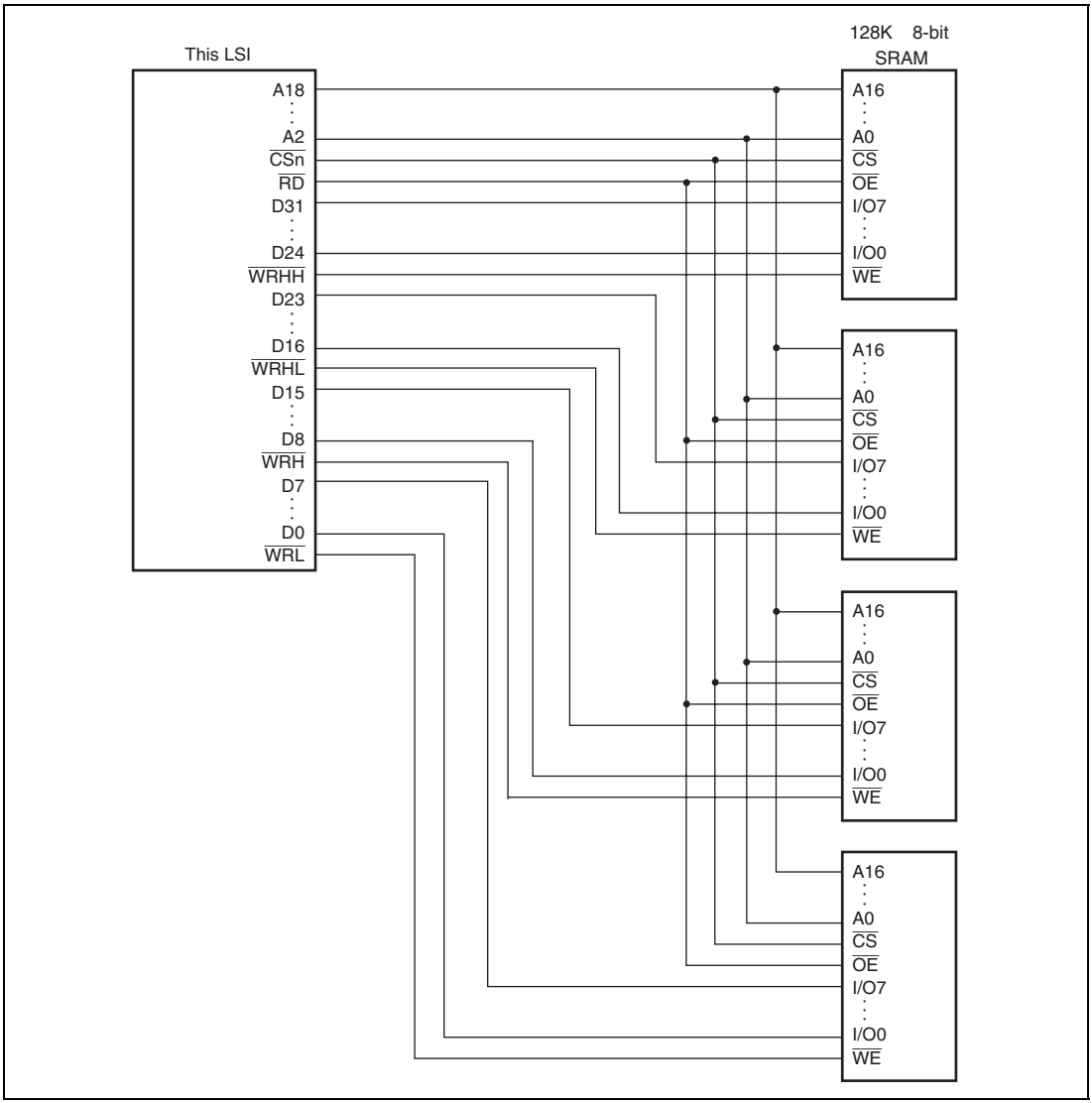


Figure 10.5 Example of 32-Bit Data-Width SRAM Connection

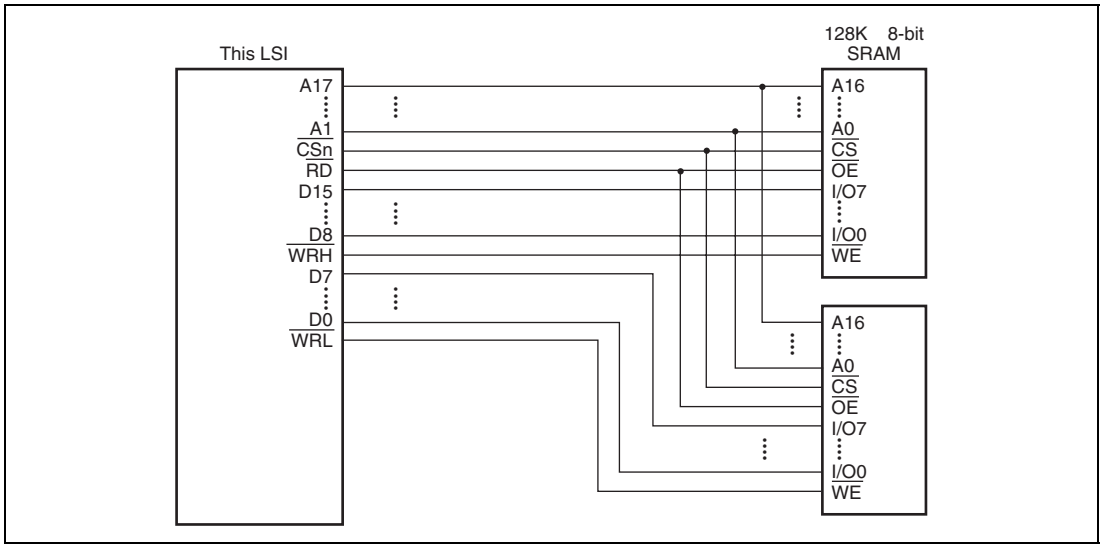


Figure 10.6 Example of 16-Bit Data-Width SRAM Connection

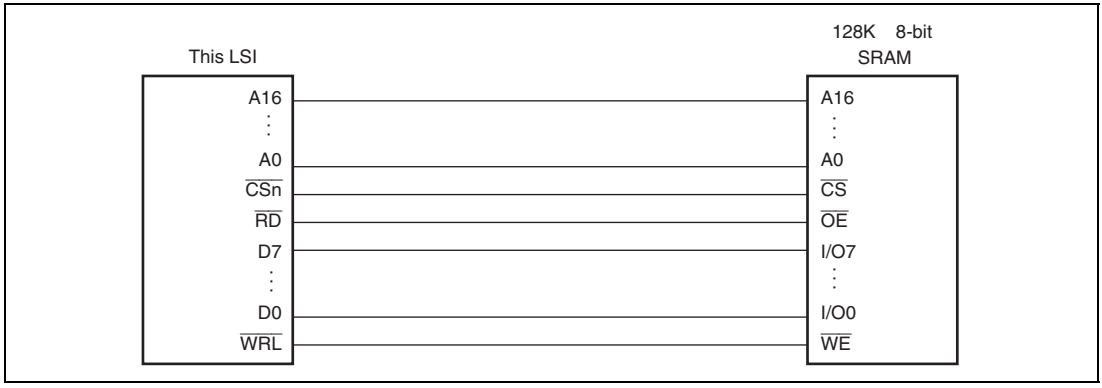
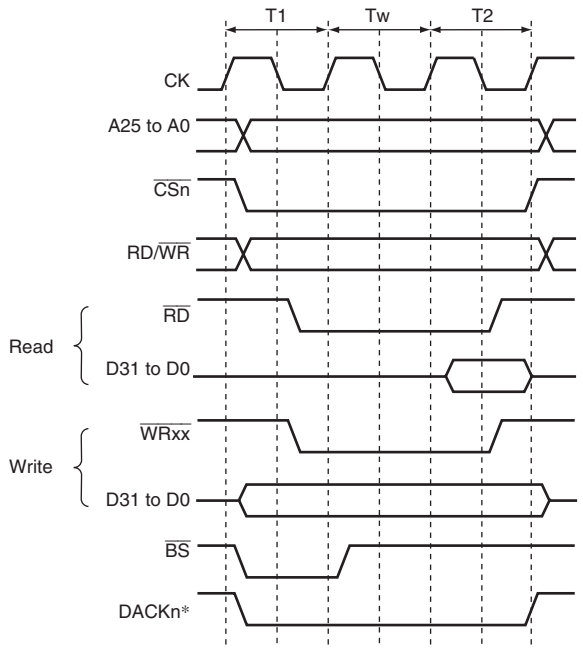


Figure 10.7 Example of 8-Bit Data-Width SRAM Connection

10.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for CS1, CS4, CS5, and CS7 spaces to insert wait cycles independently in read access and in write access. CS0, CS2, CS3, and CS6 spaces have common access wait for read cycle and write cycle. The specified number of Tw cycles are inserted as wait cycles in a normal space access shown in figure 10.8.



Note: * The waveform for DACKn is when active low is specified.

Figure 10.8 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 10.9. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CK at the transition from the T1 or Tw cycle to the T2 cycle.

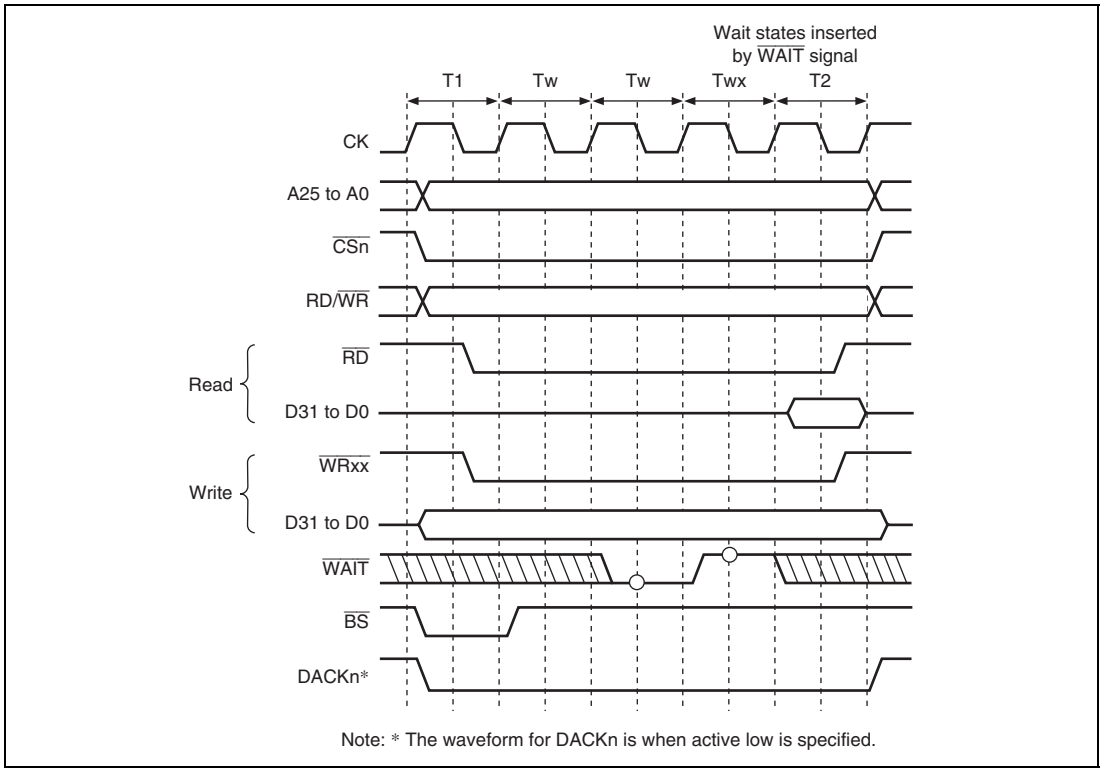


Figure 10.9 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

10.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 10.10 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WRxx}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

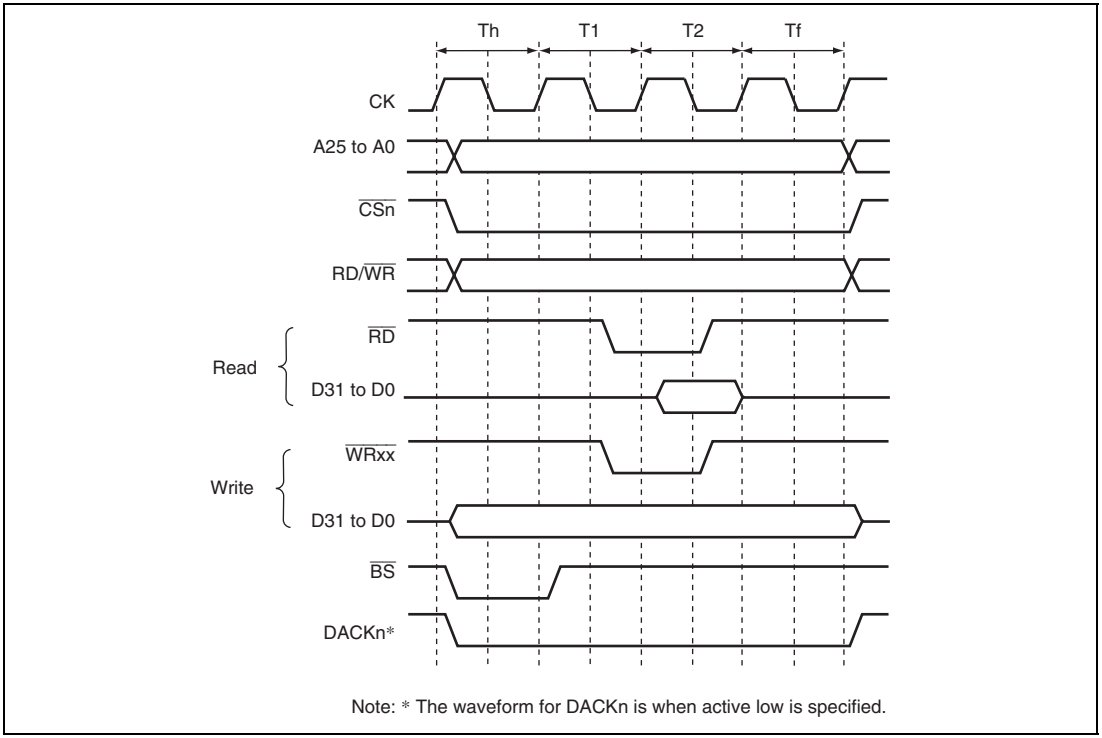


Figure 10.10 $\overline{\text{CSn}}$ Assert Period Expansion

10.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WRxx}$ signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to a normal space. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits. Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in CS5WCR to 1.

The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5}$ signal; it is high in the read cycle and low in the write cycle.

The data cycle is the same as that in a normal space access.

Timing charts are shown in figures 10.11 to 10.13.

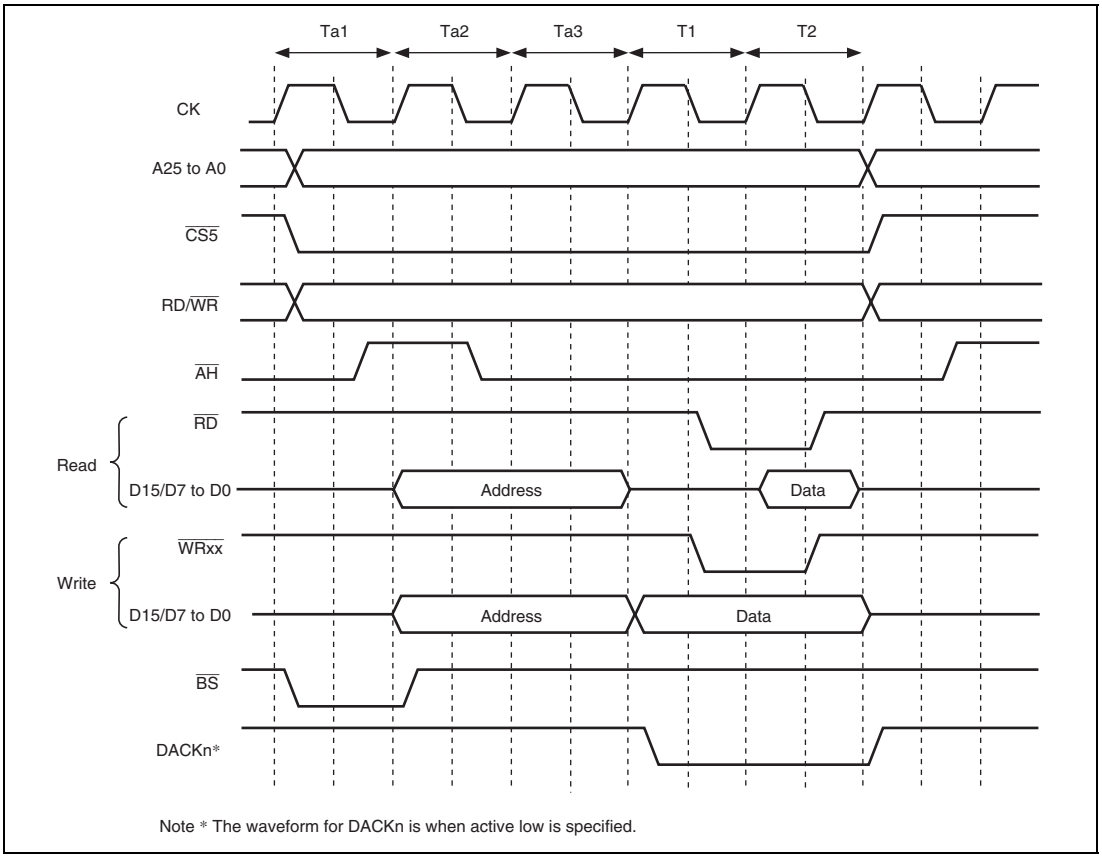


Figure 10.11 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

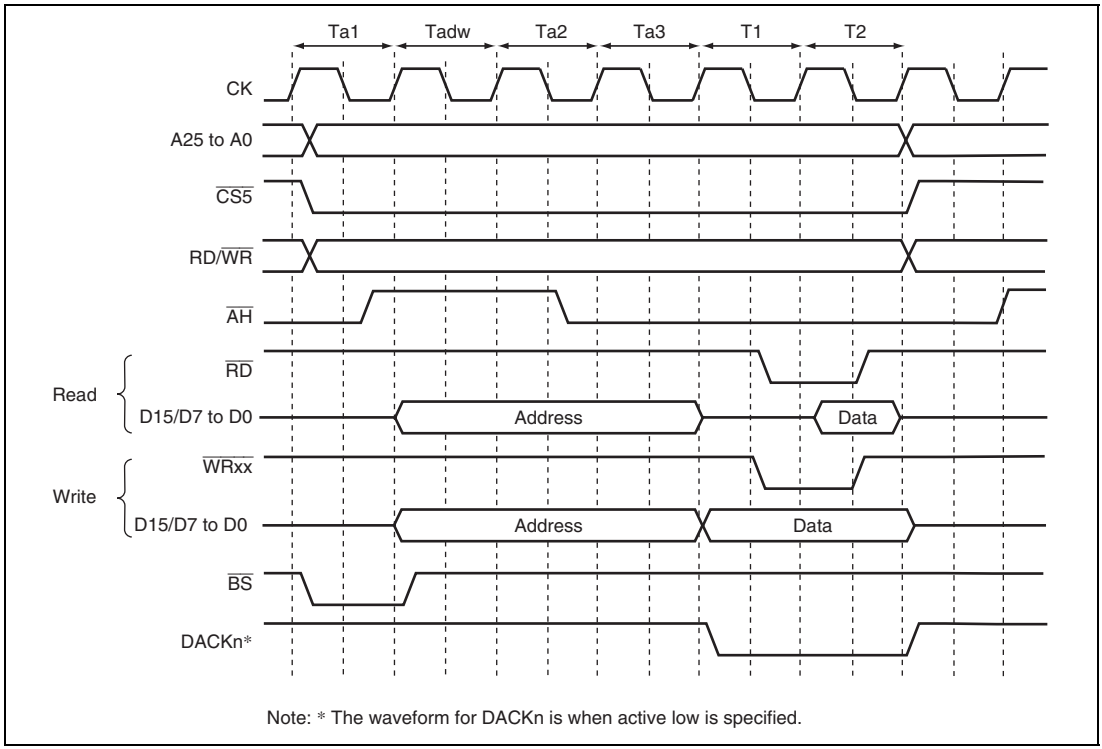


Figure 10.12 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

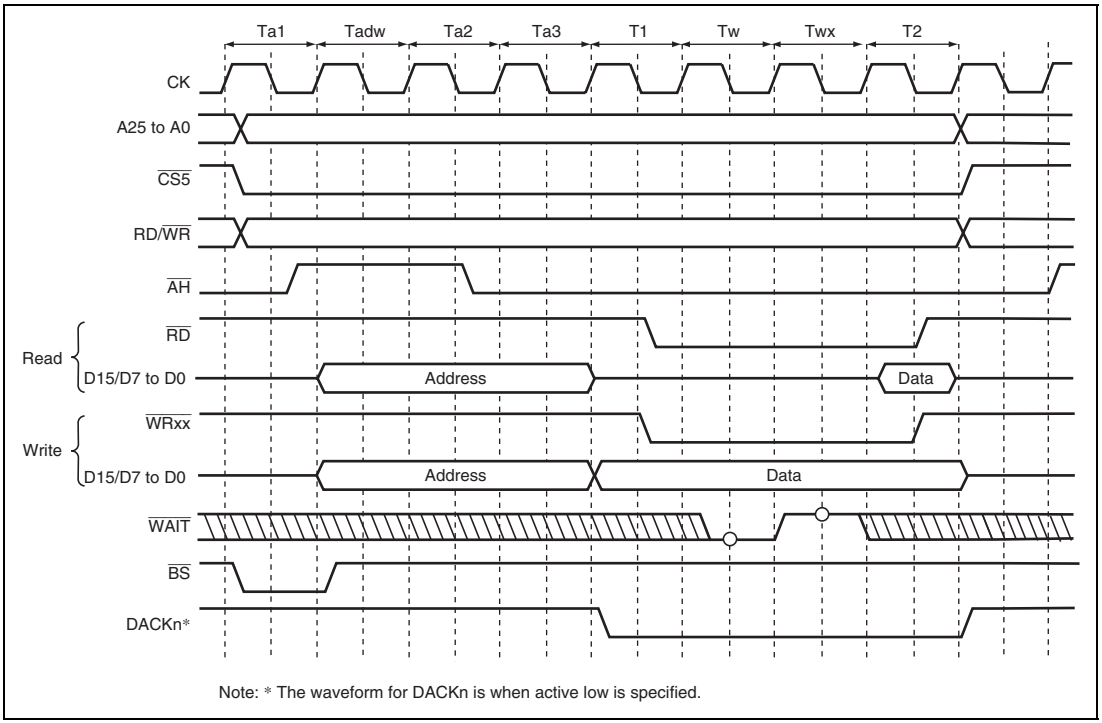


Figure 10.13 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

10.5.6 SDRAM Interface

(1) SDRAM Direct Connection

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.

The control signals for direct connection of SDRAM are $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, $\overline{\text{CASU}}$, RD/$\overline{\text{WR}}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{\text{CS2}}$, and $\overline{\text{CS3}}$. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to 32 bits or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, and specific address signals. These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS, EMRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx and the byte to be accessed, see section 10.5.1, Endian/Access Size and Data Alignment. Figures 10.14 to 10.16 show examples of the connection of the SDRAM with the LSI.

As shown in figure 10.16, two sets of SDRAMs of 32Mbytes or smaller can be connected to the same CS space by using $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$. In this case, a total of 8 banks are assigned to the same CS space: 4 banks specified by $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$, and 4 banks specified by $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$. When accessing the address with A25 = 0, $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$ are asserted. When accessing the address with A25 = 1, $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are asserted.

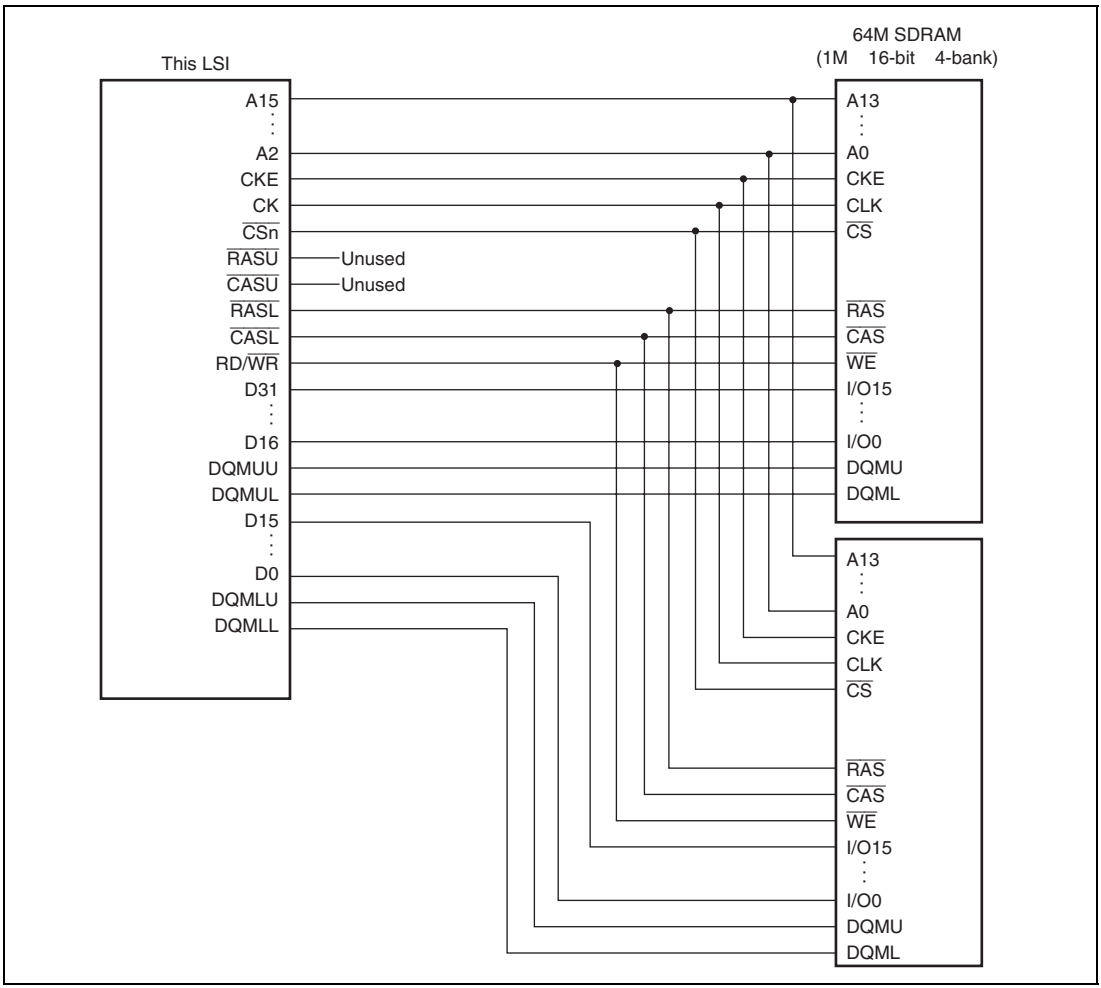


Figure 10.14 Example of 32-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

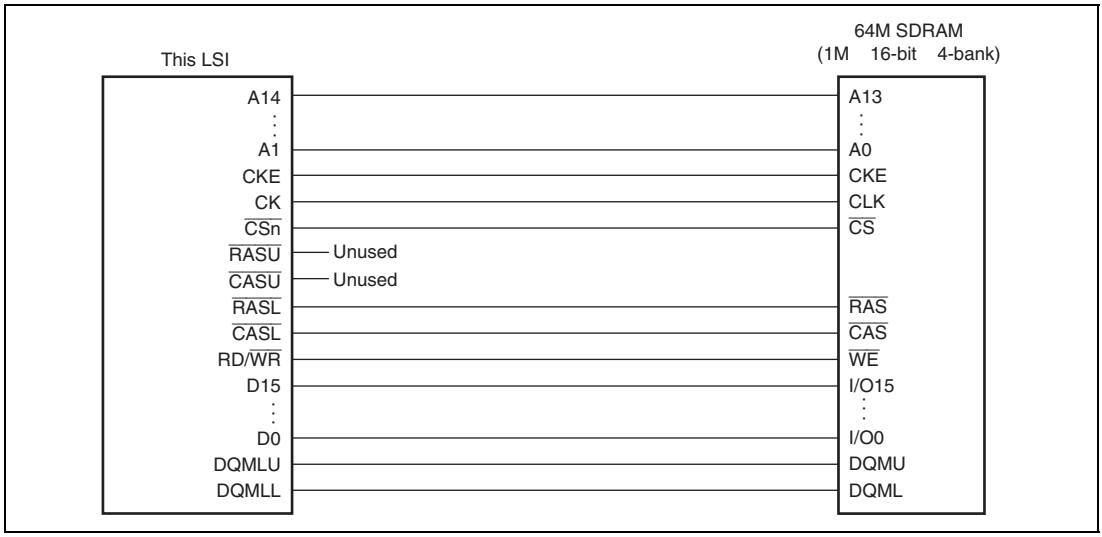


Figure 10.15 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

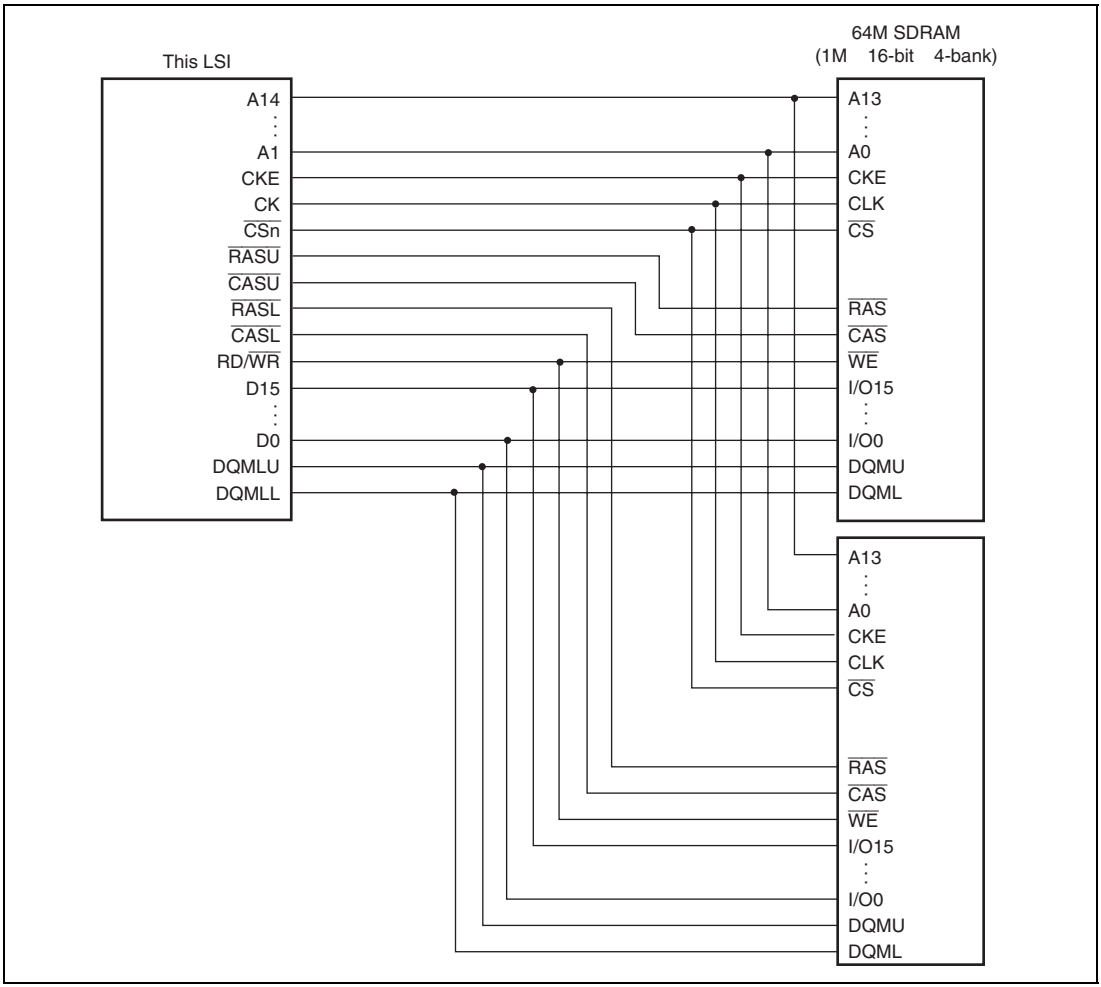


Figure 10.16 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Used)

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR, bits A2ROW[1:0], and A2COL[1:0], A3ROW[1:0], and A3COL[1:0] in SDCR. Tables 10.9 to 10.14 show the relationship between the settings of bits BSZ[1:0], A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A25 to A18 are not multiplexed and the original values of address are always output at these pins.

When the data bus width is set to 16 bits (BSZ[1:0] = B'10), the A0 pin of SDRAM specifies a word address. Therefore, connect the A0 pin of SDRAM to the A1 pin of the LSI; then connect the A1 pin of SDRAM to the A2 pin of the LSI, and so on. When the data bus width is set to 32 bits (BSZ[1:0] = B'11), the A0 pin of SDRAM specifies a longword address. Therefore, connect the A0 pin of SDRAM to the A2 pin of the LSI; then connect the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

Table 10.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**00 (11 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*[2]	A21*[2]	A12 (BA1)	Specifies bank
A12	A20*[2]	A20*[2]	A11 (BA0)	
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (512 Kwords × 32 bits × 4 banks, column 8 bits product): 1				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 10.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**01 (12 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A13 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
128-Mbit product (1 Mword × 32 bits × 4 banks, column 8 bits product): 1				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to access the mode.
2. Bank address specification

Table 10.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**01 (12 Bits)**	**01 (9 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12 (BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords × 32 bits × 4 banks, column 9 bits product): 1 128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 10.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2*3	A25*2*3	A13 (BA1)	Specifies bank
A14	A24*2	A24*2	A12 (BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*1	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 10 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 10.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25*2*3	A25*2*3	A14 (BA1)	Specifies bank
A15	A24*2	A24*2	A13 (BA0)	
A14	A23*2	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*1	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 9 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 10.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*2	A20*2	A11 (BA0)	Specifies bank
A11	A19	L/H*1	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 10.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Specifies bank
A13	A21*2	A21*2	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 10.13 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies bank
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 10.13 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13 (BA1)	Specifies bank
A13	A23*2	A23*2	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 10.14 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 10.14 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2*3	A25*2*3	A14 (BA1)	Specifies bank
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

(3) Burst Read

A burst read occurs in the following cases with this LSI.

- Access size in reading is larger than data bus width.
- 16-byte transfer in DMAC

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 4 times to read 16-byte continuous data from the SDRAM that is connected to a 32-bit data bus. This access is called the burst read with the burst number 4. Table 10.15 shows the relationship between the access size and the number of bursts.

Table 10.15 Relationship between Access Size and Number of Bursts

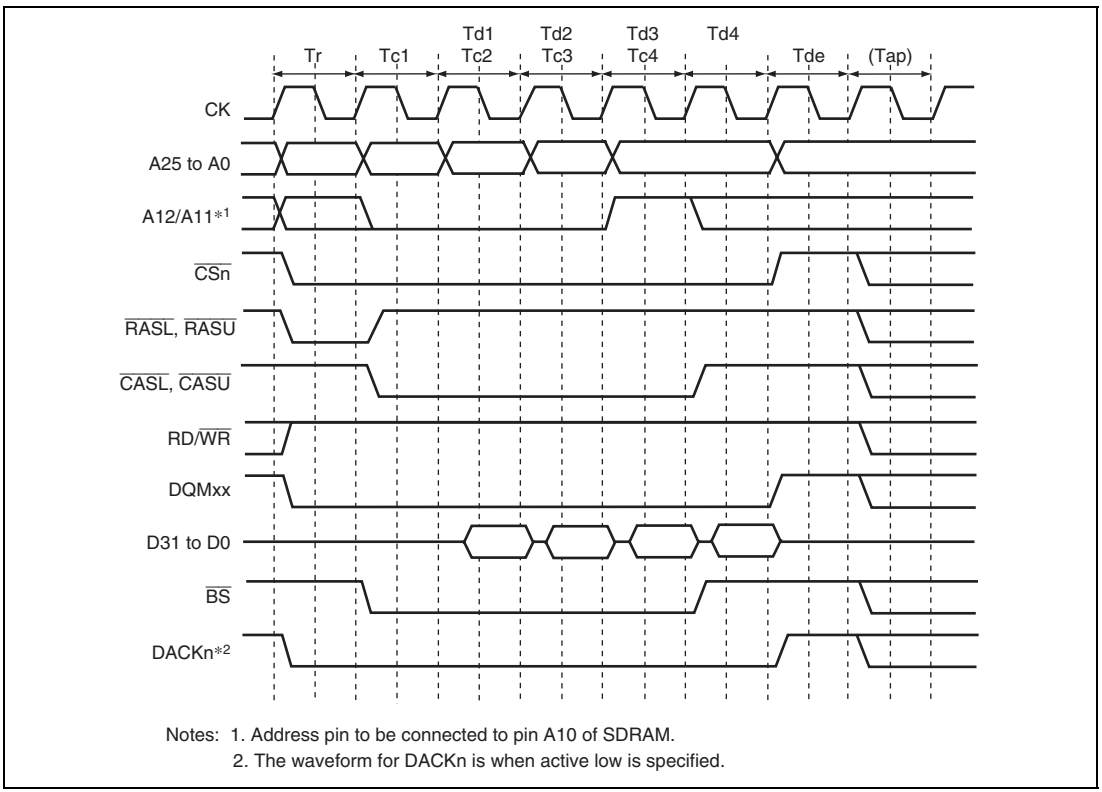
Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4

Figures 10.17 and 10.18 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CK) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.

In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect the SDRAM in variable frequencies. Figure 10.18 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output can be specified using the WTRCD1 and WTRCD0 bits in CS3WCR. If the WTRCD1 and WTRCD0 bits specify one cycles or more, a Trw cycle where the NOT command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles

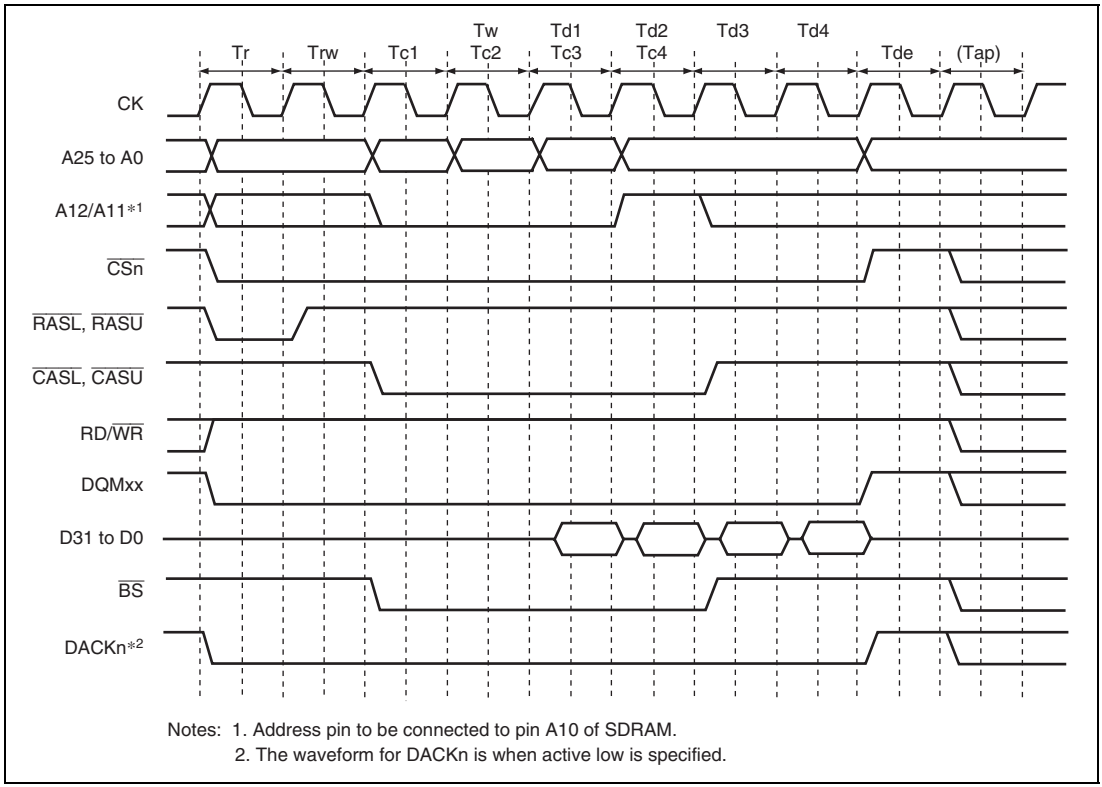
from the Tc1 cycle where the READ command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR and WTRCD0 bit in CS3WCR. The number of cycles from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS latency for the SDRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the SDRAM.

A Tde cycle is an idle cycle required to transfer the read data into this LSI and occurs once for every burst read or every single read.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.17 Burst Read Basic Timing (CAS Latency 1, Auto-Precharge)



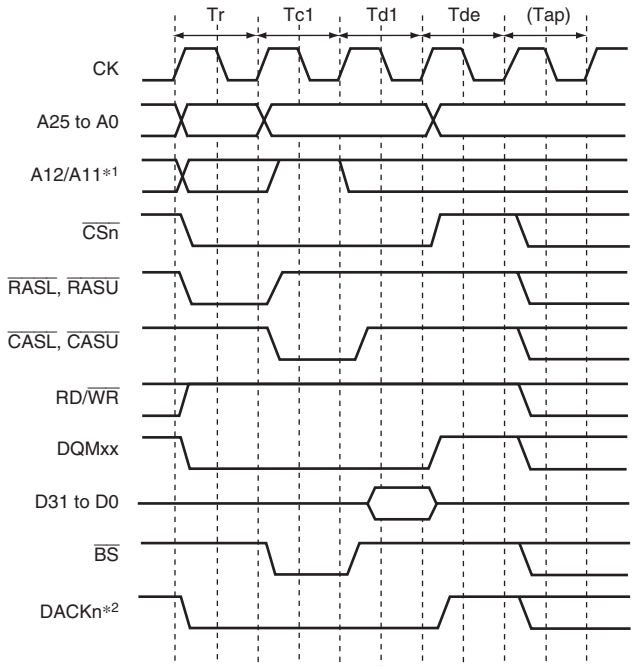
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.18 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD[1:0] = 1 Cycle, Auto-Precharge)

(4) Single Read

A read access ends in one cycle when the data bus width is larger than or equal to the access size. As the SDRAM is set to the burst read with the burst length 1, only the required data is output. A read access that ends in one cycle is called single read.

Figure 10.19 shows the single read basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.19 Basic Timing for Single Read (CAS Latency 1, Auto-Precharge)

(5) Burst Write

A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- 16-byte transfer in DMAC

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 4 times to write 16-byte continuous data to the SDRAM that is connected to a 32-bit data bus. This access is called burst write with the burst number 4.

The relationship between the access size and the number of bursts is shown in table 10.15.

Figure 10.20 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. Between the Trwl and the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.

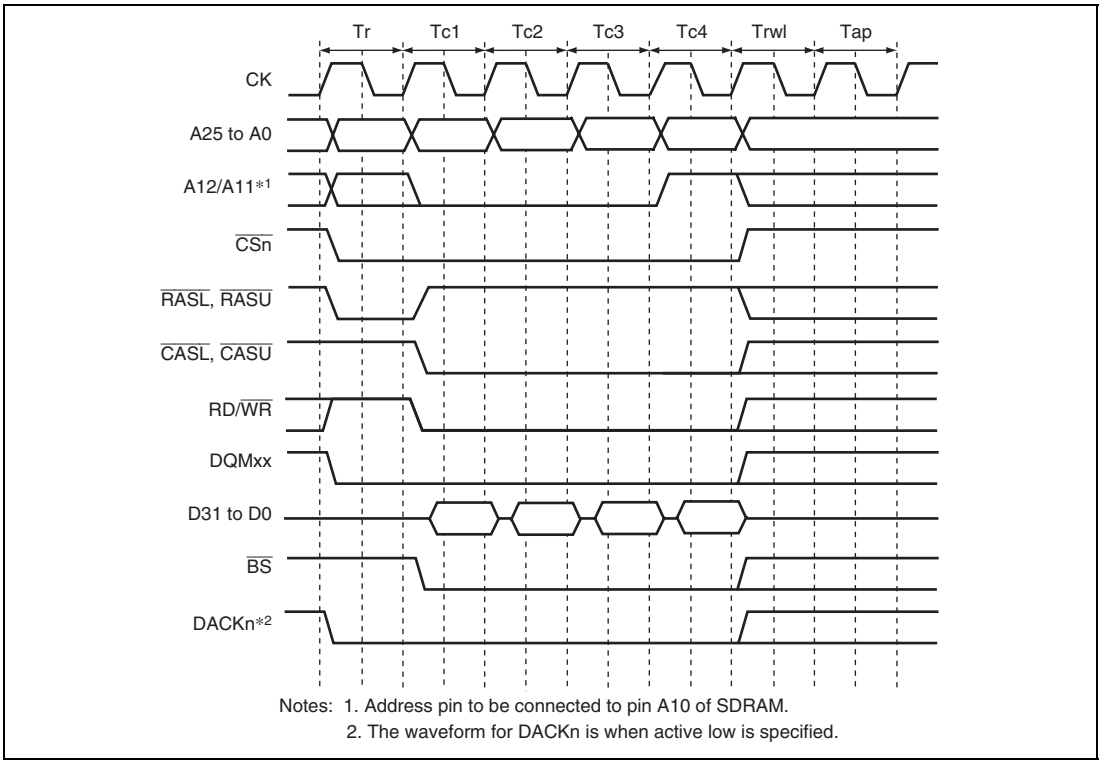
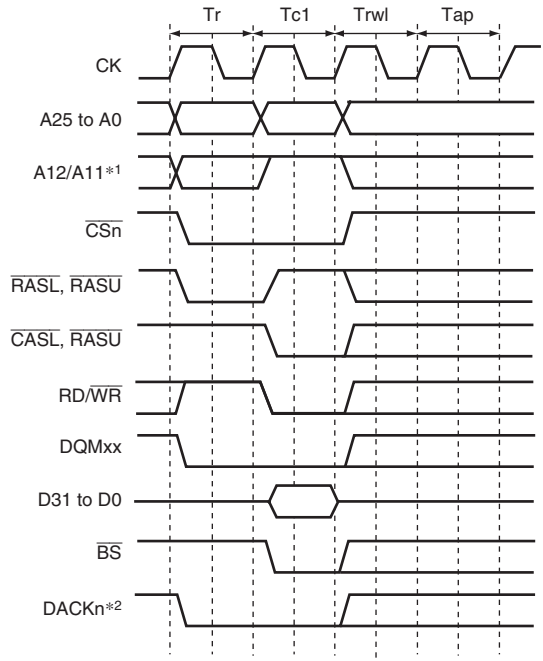


Figure 10.20 Basic Timing for Burst Write (Auto-Precharge)

(6) Single Write

A write access ends in one cycle when the data bus width is larger than or equal to access size. As a single write or burst write with burst length 1 is set in SDRAM, only the required data is output. The write access that ends in one cycle is called single write. Figure 10.21 shows the single write basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.21 Single Write Basic Timing (Auto-Precharge)

(7) Bank Active

The SDRAM bank function can be used to support high-speed access to the same row address. When the BACTV bit in SDCR is 1, access is performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for CS3 space. When CS3 space is set to bank-active mode, CS2 space should be set to normal space or SRAM with byte selection. When CS2 and CS3 spaces are both set to SDRAM, auto-precharge mode must be set.

When the bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As SDRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the WTRP1 and WTPR0 bits in CS3WCR.

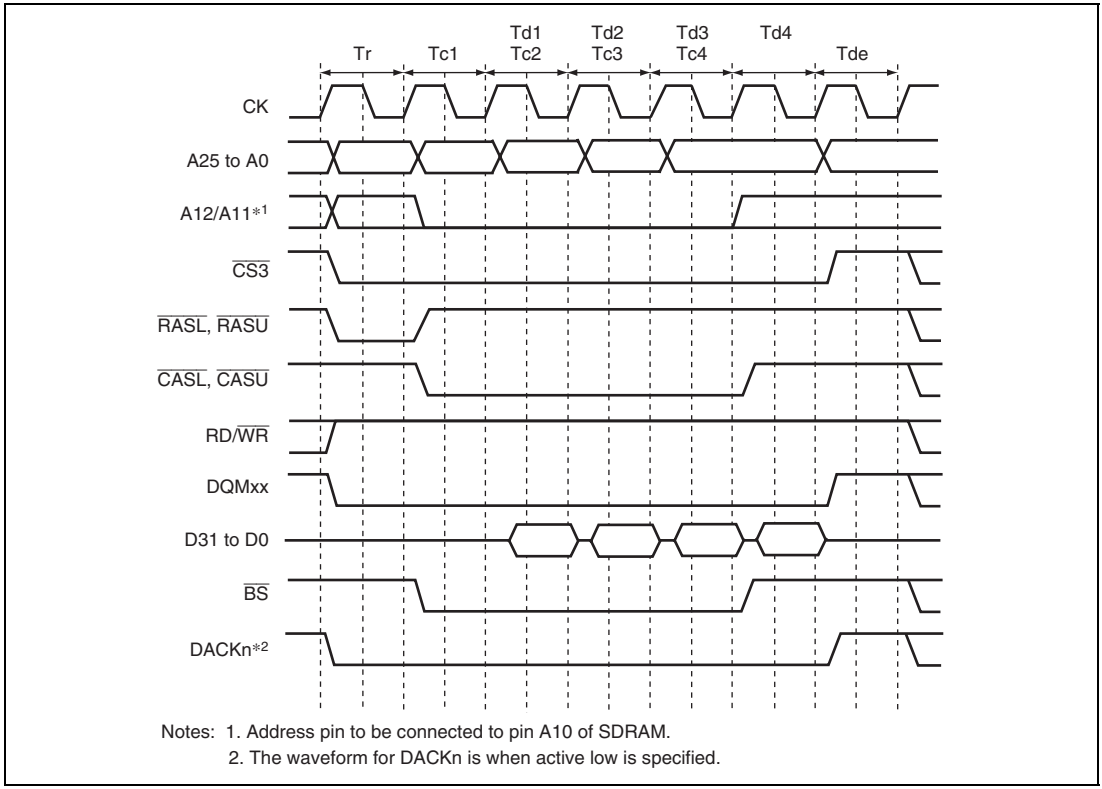
In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 10.22, a burst read cycle for the same row address in figure 10.23, and a burst read cycle for different row addresses in figure 10.24. Similarly, a burst write cycle without auto-precharge is shown in figure 10.25, a burst write cycle for the same row address in figure 10.26, and a burst write cycle for different row addresses in figure 10.27.

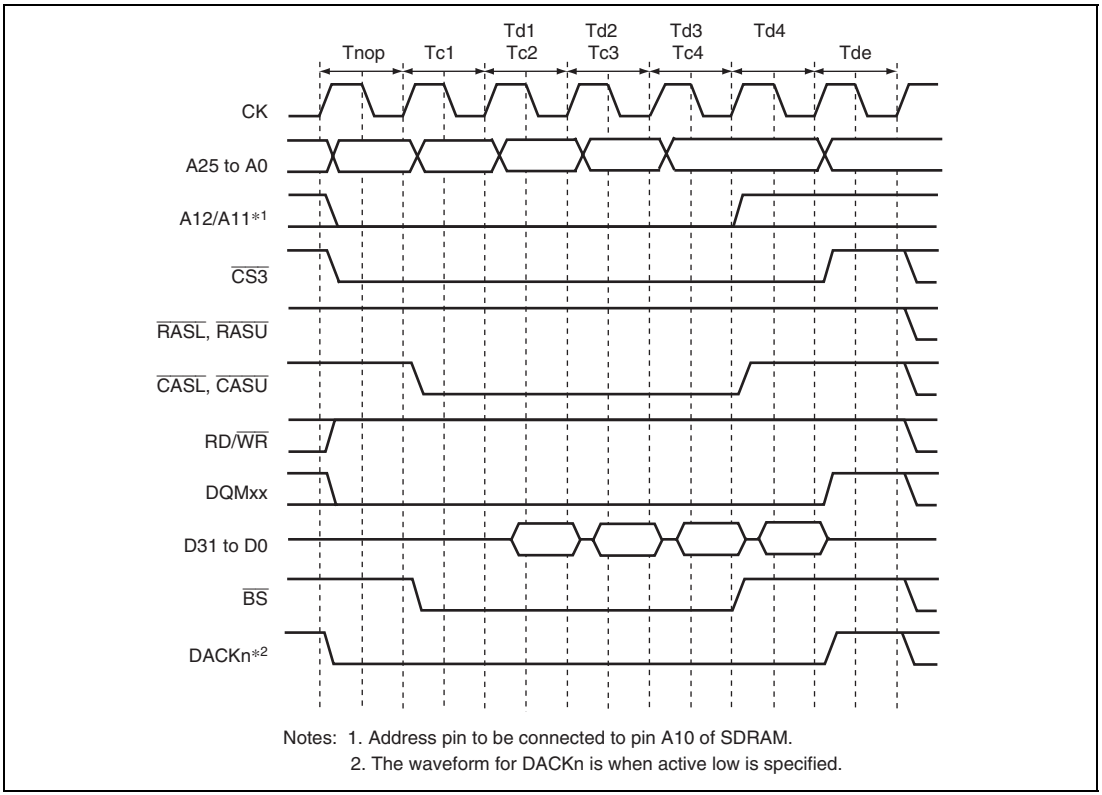
In figure 10.23, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only access cycles to the respective banks in the area 3 space are considered, as long as access cycles to the same row address continue, the operation starts with the cycle in figure 10.22 or 10.25, followed by repetition of the cycle in figure 10.23 or 10.26. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 10.23 or 10.26 is executed instead of that in figure 10.24 or 10.27. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.



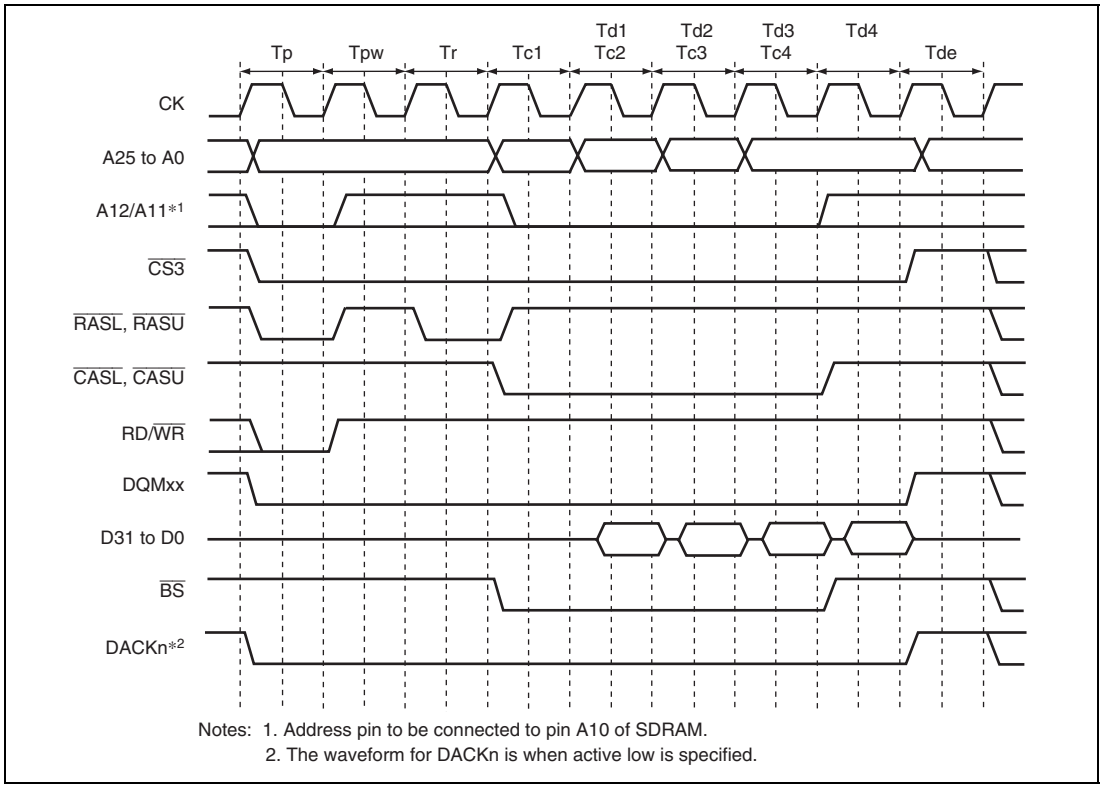
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.22 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)



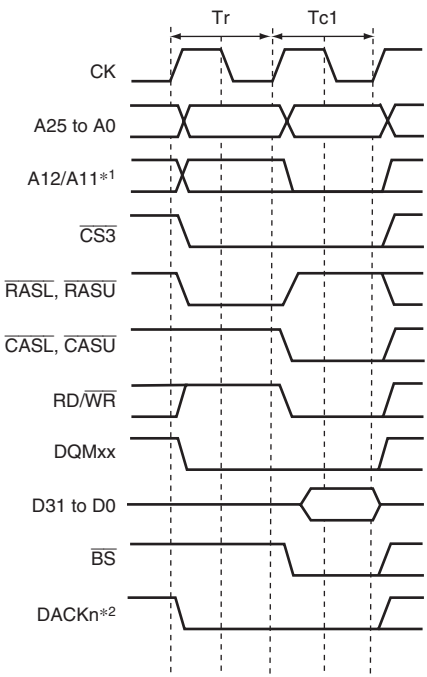
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.23 Burst Read Timing
(Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)



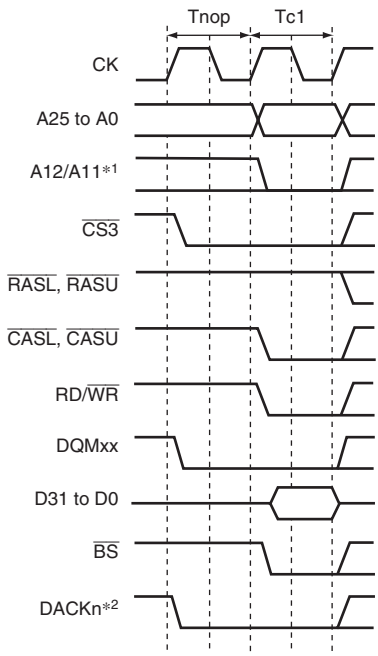
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.24 Burst Read Timing
(Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)



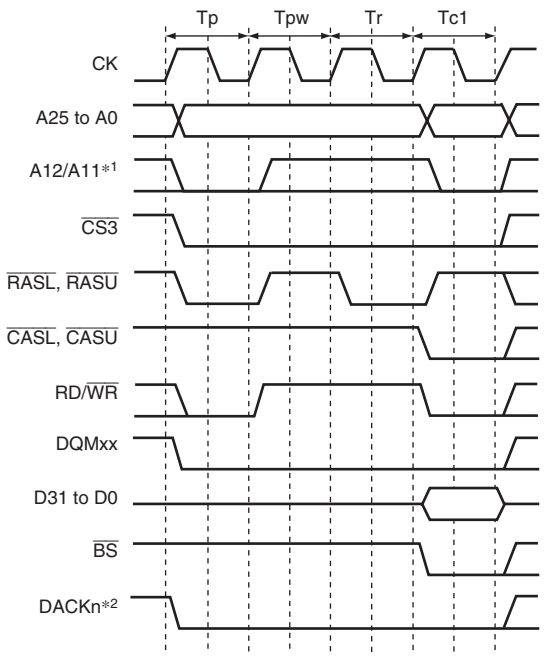
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.25 Single Write Timing (Bank Active, Different Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.26 Single Write Timing (Bank Active, Same Row Addresses in the Same Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.27 Single Write Timing
(Bank Active, Different Row Addresses in the Same Bank)

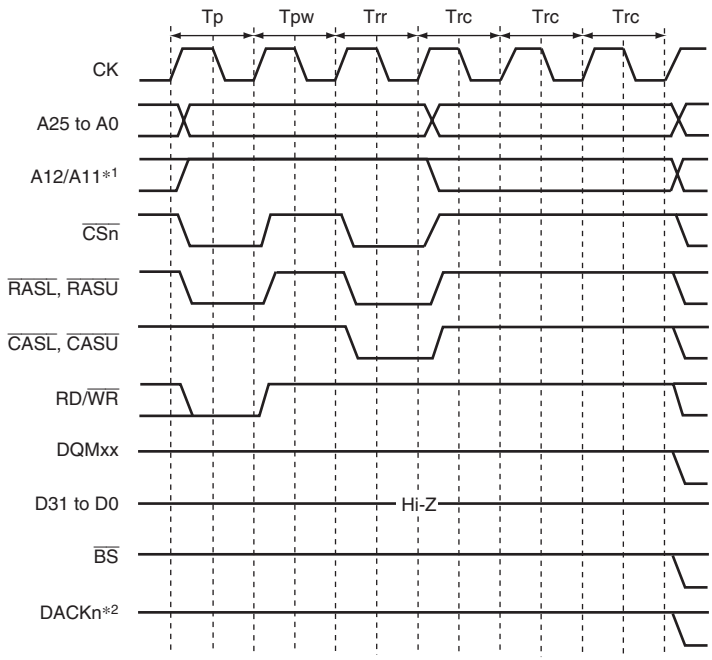
(8) Refreshing

This LSI has a function for controlling SDRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC2 to RRC0 bits in RTCSR. If SDRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

(a) Auto-refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS2 to CKS0 and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC2 to RRC0. At the same time, RTCNT is cleared to zero and the count-up is restarted.

Figure 10.28 shows the auto-refresh cycle timing. After starting, the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to pre-charged state from active state when some bank is being pre-charged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR. A new command is not issued for the duration of the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR after the Trr cycle. The WTRC1 and WTRC0 bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle cycle is inserted between the Tp cycle and Trr cycle when the setting value of the WTRP1 and WTRP0 bits in CS3WCR is longer than or equal to 1 cycle.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.28 Auto-Refresh Timing

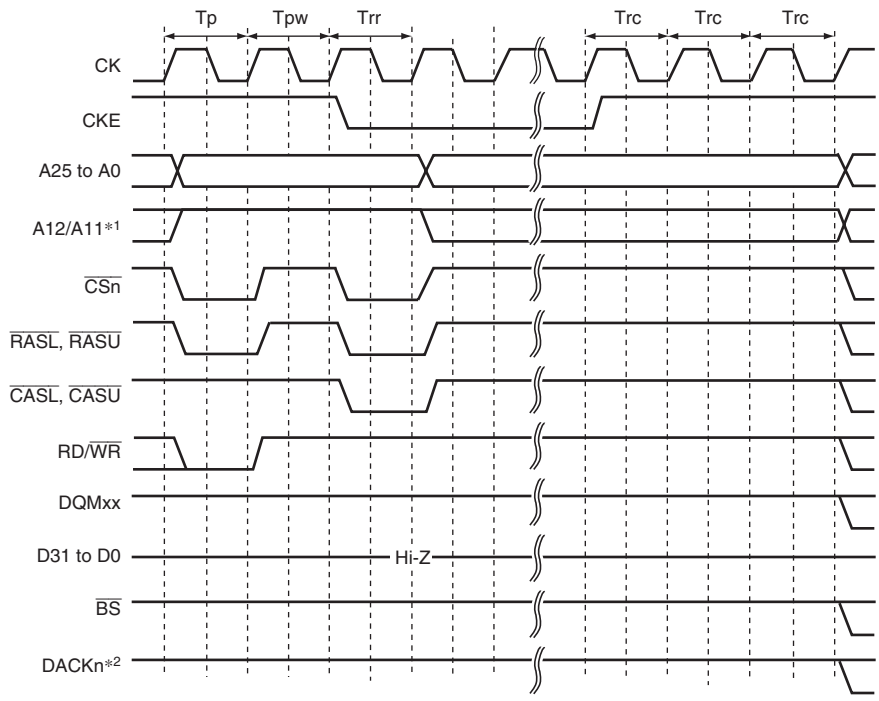
(b) Self-refreshing

Self-refresh mode is a standby mode in which the refresh timing and refresh addresses are generated within the SDRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WSR. SDRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR.

Self-refresh timing is shown in figure 10.29. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode due to an interrupt. Note that the necessary signals such as CKE must be driven even in standby state by setting the HIZCNT bit in CMNCR to 1.

The self-refresh state is not cleared by a manual reset. In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.29 Self-Refresh Timing

(9) Relationship between Refresh Requests and Bus Cycles

If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired. This LSI has the $\overline{\text{REFOUT}}$ pin to request the bus while waiting for refresh execution. For $\overline{\text{REFOUT}}$ pin function selection, see section 22, Pin Function Controller (PFC). This LSI continues to assert $\overline{\text{REFOUT}}$ (low level) until the bus is acquired.

On receiving the asserted $\overline{\text{REFOUT}}$ signal, the external device must negate the $\overline{\text{BREQ}}$ signal and return the bus. If the external bus does not return the bus for a period longer than the specified refresh interval, refresh cannot be executed and the SDRAM contents may be lost.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring.

If a bus mastership is requested during self-refresh, the bus will not be released until the refresh is completed.

(10) Low-Frequency Mode

When the SLOW bit in SDCR is set to 1, output of commands, addresses, and write data, and fetch of read data are performed at a timing suitable for operating SDRAM at a low frequency.

Figure 10.30 shows the access timing in low-frequency mode. In this mode, commands, addresses, and write data are output in synchronization with the falling edge of CK, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CK, which is half a cycle faster than the normal timing. This timing allows the hold time of commands, addresses, write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time of commands, addresses, write data, and read data are not guaranteed. Take the operating frequency and timing design into consideration when making the SLOW bit setting.

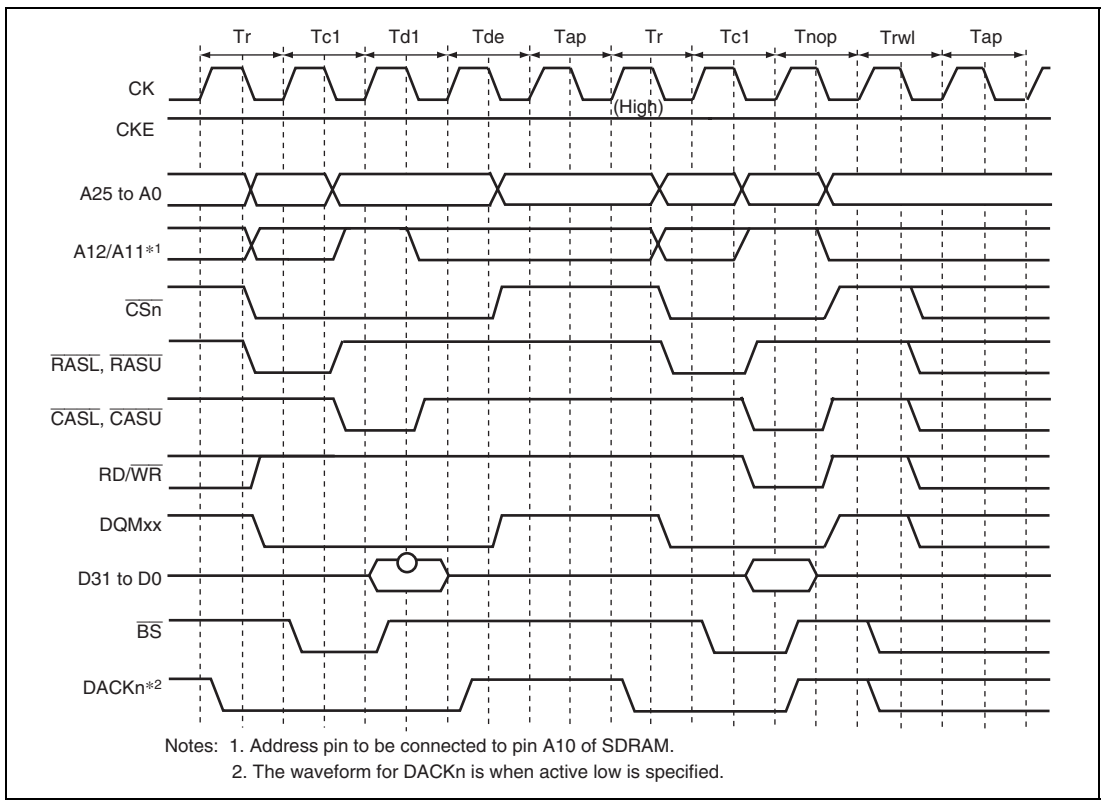


Figure 10.30 Low-Frequency Mode Access Timing

(11) Power-Down Mode

If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note that if an access occurs in power-down mode, a cycle of overhead occurs because a cycle is needed to assert the CKE in order to cancel power-down mode.

Figure 10.31 shows the access timing in power-down mode.

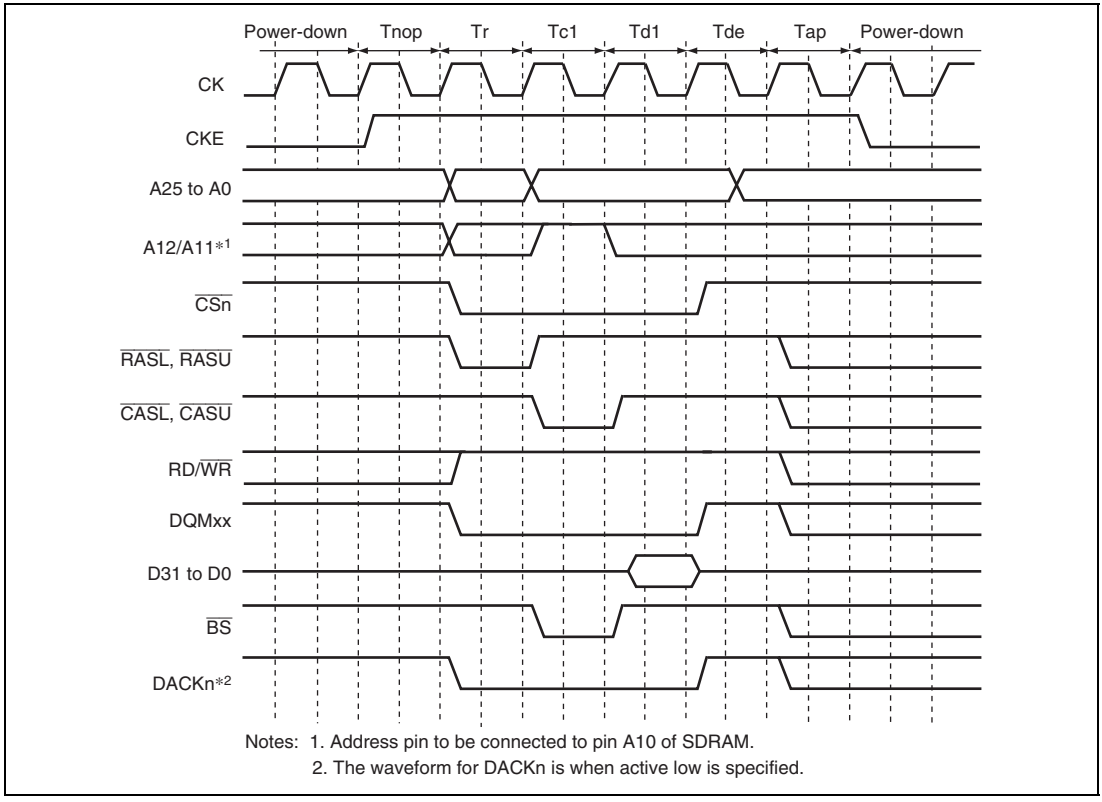


Figure 10.31 Power-Down Mode Access Timing

(12) Power-On Sequence

In order to use SDRAM, mode setting must first be made for SDRAM after the power on and subsequent pause period specified for the used SDRAM. This pause period should be obtained by a power-on reset generating circuit or software.

To perform SDRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the SDRAM mode register. In SDRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{CSn}$, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, $\overline{CASL}$, and RD/$\overline{WR}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the SDRAM mode register by performing a write to address H'FFFC4000 + X for CS2 space SDRAM, and to address H'FFFC5000 + X for CS3 space SDRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 2 to 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written in a byte-size access to the addresses shown in table 10.16. In this time 0 is output at the external address pins of A12 or later.

Table 10.16 Access Address in SDRAM Mode Register Write

- Setting for CS2 Space

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460
32 bits	2	H'FFFC4480	H'0000880
	3	H'FFFC44C0	H'00008C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060
32 bits	2	H'FFFC4080	H'0000080
	3	H'FFFC40C0	H'00000C0

- Setting for CS3 Space

  Burst read/single write (burst length 1):

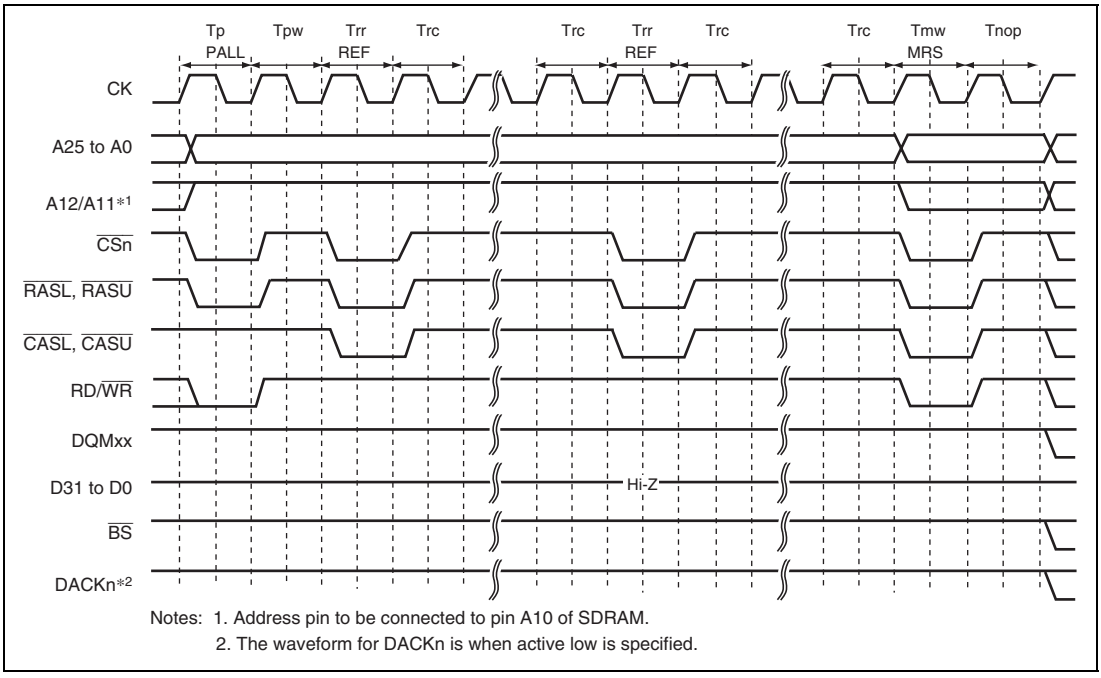
Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460
32 bits	2	H'FFFC5880	H'0000880
	3	H'FFFC58C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060
32 bits	2	H'FFFC5080	H'0000080
	3	H'FFFC50C0	H'00000C0

Mode register setting timing is shown in figure 10.32. A PALL command (all bank pre-charge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the WTRC1 and WTRC0 bits in CS3WCR, are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or more, are inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer to the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer than the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 10.32 SDRAM Mode Write Timing (Based on JEDEC)

(13) Low-Power SDRAM

The low-power SDRAM can be accessed using the same protocol as the normal SDRAM.

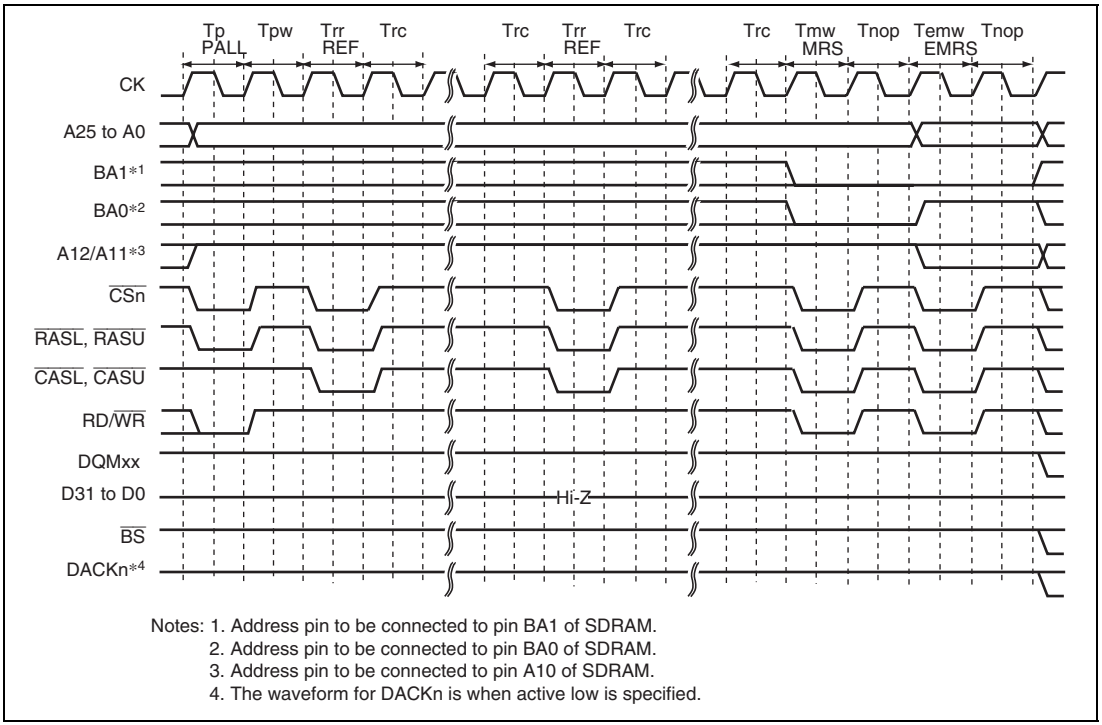
The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which there is data in a work area other than the specific area can be lost without severe repercussions.

The low-power SDRAM supports the extension mode register (EMRS) in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table below. For example, if data H'0YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XX0 and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 10.17 Output Addresses when EMRS Command Is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'FFFC4XX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'FFFC5XX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'FFFC4XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (with refresh)	H'FFFC5XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS2 MRS + EMRS (without refresh)	H'FFFC4XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (without refresh)	H'FFFC5XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY



Notes: 1. Address pin to be connected to pin BA1 of SDRAM.
2. Address pin to be connected to pin BA0 of SDRAM.
3. Address pin to be connected to pin A10 of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 10.33 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit in the SDCR is set to 1 while the DEEP and RFSH bits in the SDCR are set to 1, the low-power SDRAM enters deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel deep power-down mode. Before executing an access after returning from deep power-down mode, the power-up sequence must be re-executed.

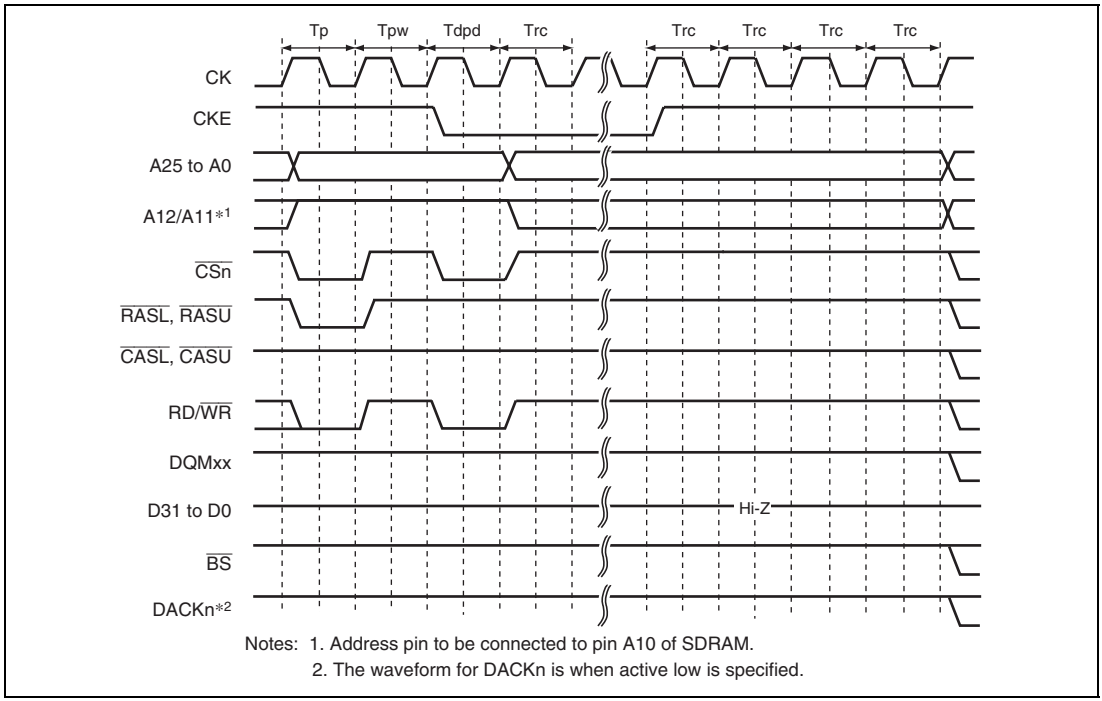


Figure 10.34 Deep Power-Down Mode Transition Timing

10.5.7 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called burst mode or page mode. In a burst ROM (clock asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent access cycles are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent access cycles, addresses are changed at the falling edge of the CK.

For the 1st access cycle, the number of wait cycles specified by the W3 to W0 bits in CSnWCR is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the W1 to W0 bits in CSnWCR is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that does not perform the burst operation in the burst ROM (clock asynchronous) interface, access timing is same as a normal space.

Table 10.18 lists a relationship between bus width, access size, and the number of bursts. Figure 10.35 shows a timing chart.

Table 10.18 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes	00, 01	16	1
		10	4	4
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes	00	8	1
		01	2	4
		10*[1]	4	2
			2, 4, 2	3
32 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	1	1
	16 bytes*[2]	Not affected	4	1

Notes: 1. When the bus width is 16 bits, the access size is 16 bits, and the BST[1:0] bits in CSnWCR are 10, the number of bursts and access count depend on the access start address. At address H'xxx0 or H'xxx8, 4-4 burst access is performed. At address H'xxx4 or H'xxxC, 2-4-2 burst access is performed.

2. Only the DMAC is capable of transfer with 16 bytes as the unit of access. The maximum unit of access for the DTC and CPU is 32 bits.

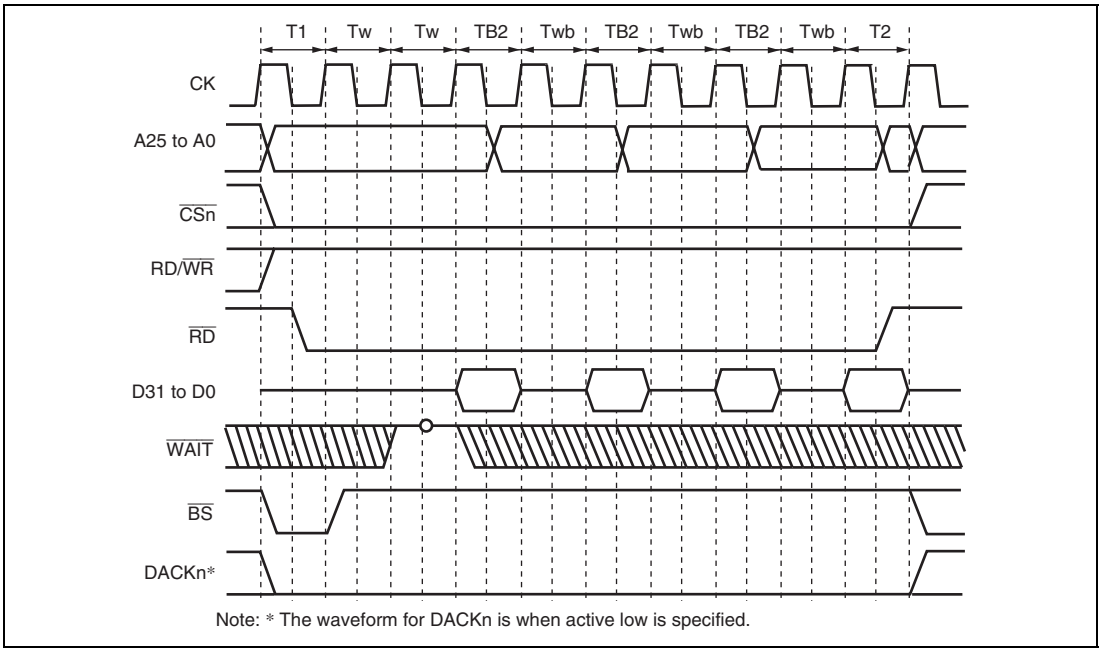


Figure 10.35 Burst ROM Access Timing (Clock Asynchronous)
(Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

10.5.8 SRAM Interface with Byte Selection

The SRAM interface with byte selection is for access to an SRAM which has a byte-selection pin ($\overline{\text{WRxx}}$). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the SRAM interface with byte selection is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WRxx}}$ pin, which is different from that for the normal space interface. The basic access timing is shown in figure 10.36. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WRxx}}$). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WRxx}}$ pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 10.37 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 10.38 shows the access timing when a software wait is specified.

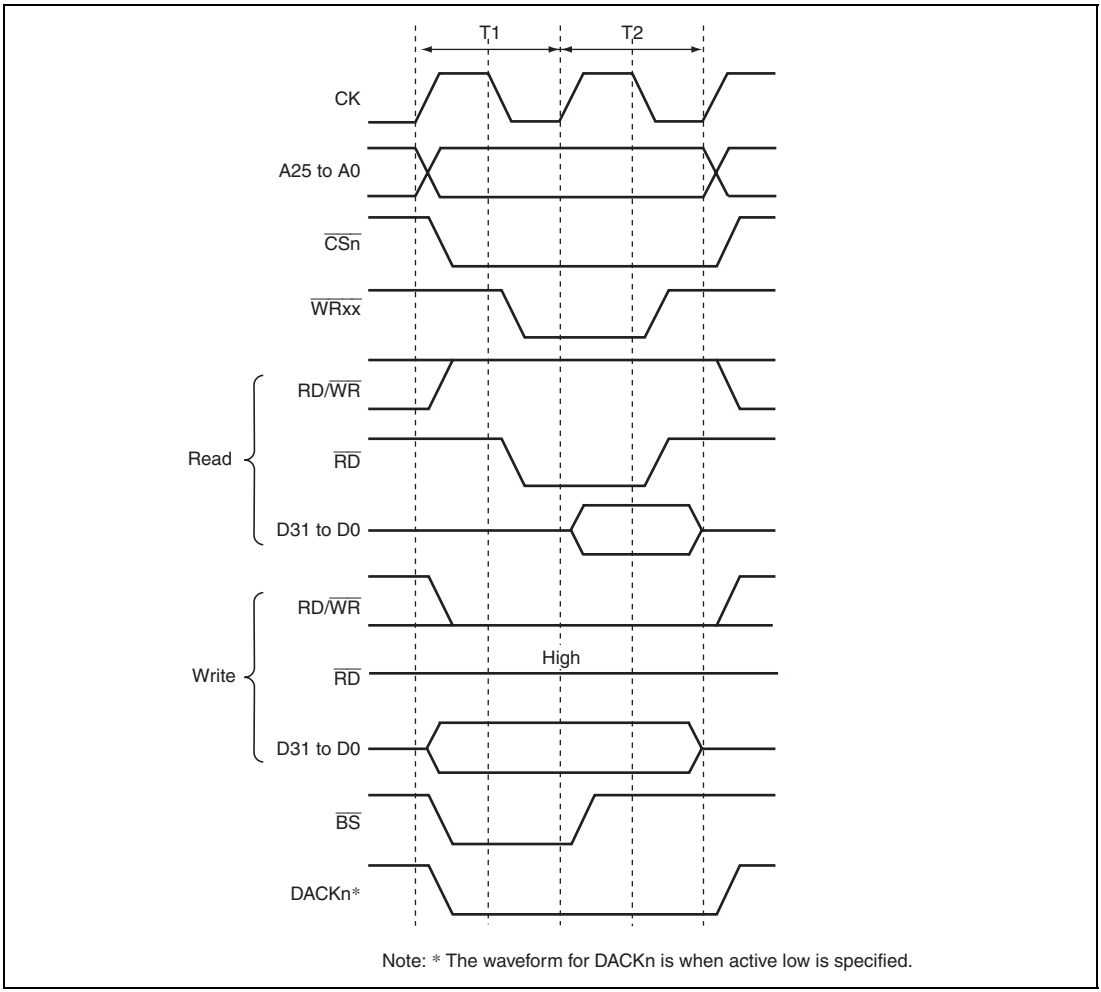


Figure 10.36 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

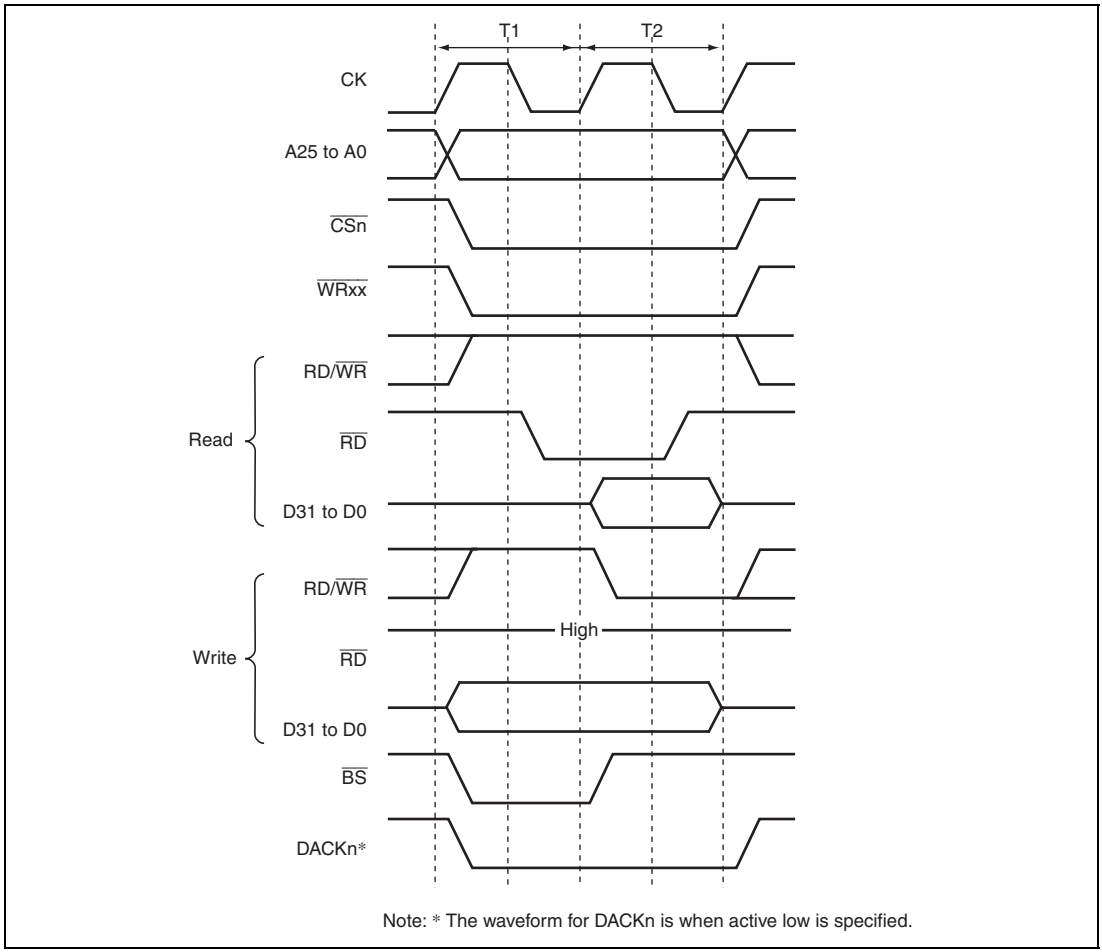


Figure 10.37 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

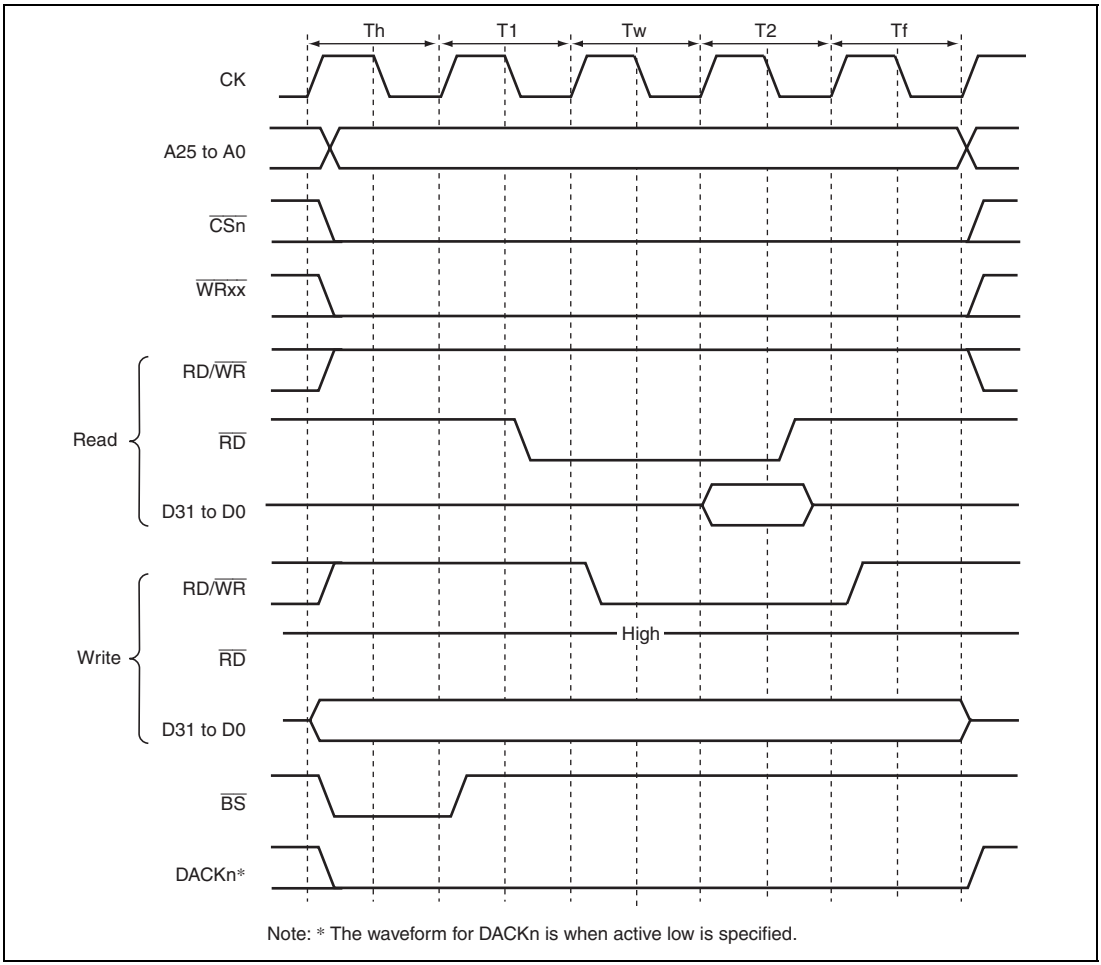


Figure 10.38 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

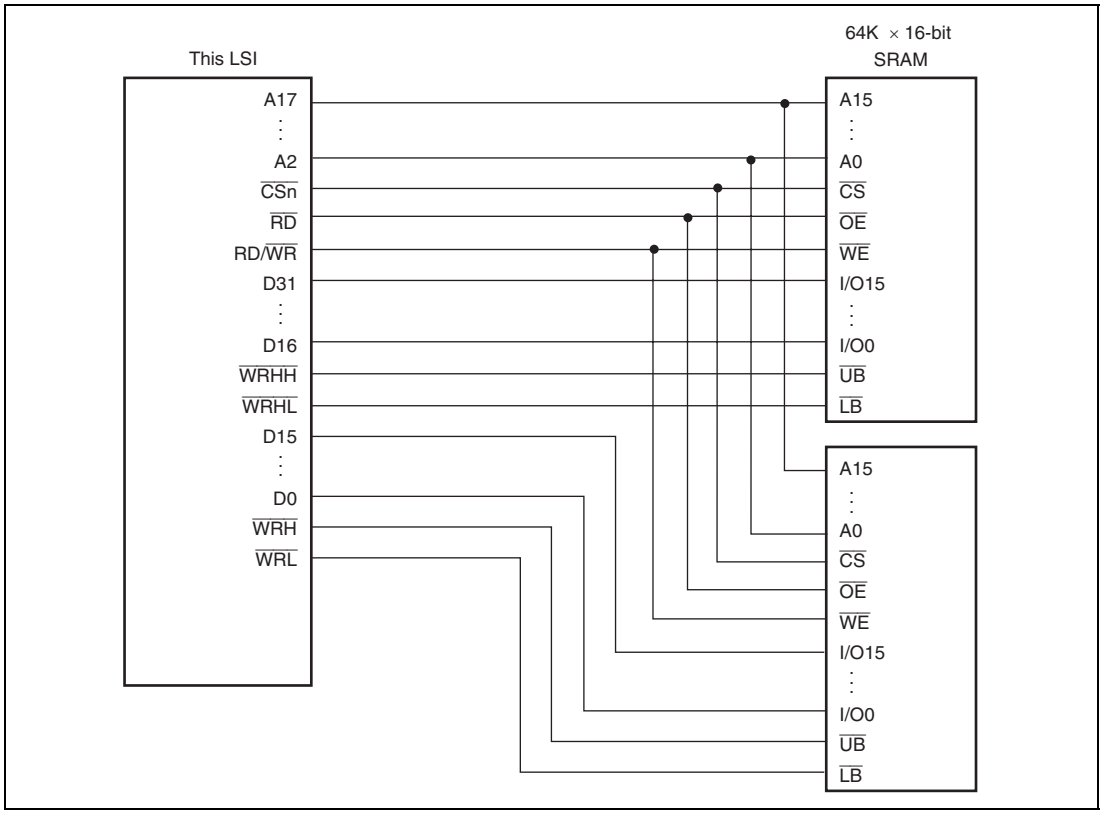


Figure 10.39 Example of Connection with 32-Bit Data-Width SRAM with Byte Selection

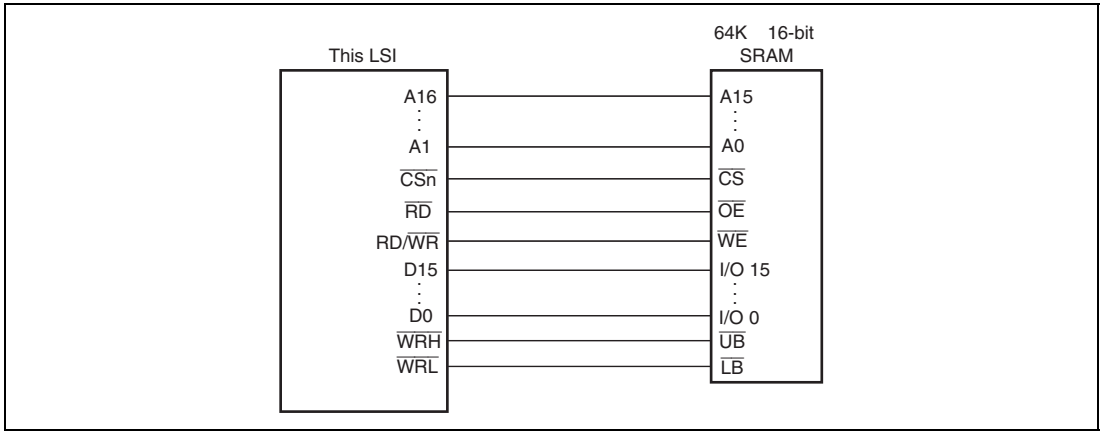


Figure 10.40 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

10.5.9 Burst ROM (Clock Synchronous) Interface

The burst ROM (clock synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for CS0 space.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM (clock synchronous) is accessed, the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. If the bus width is 32 bits, the burst length must be specified as 4. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, using 16-byte read by the DMA is recommended. The burst ROM interface performs write access in the same way as normal space access.

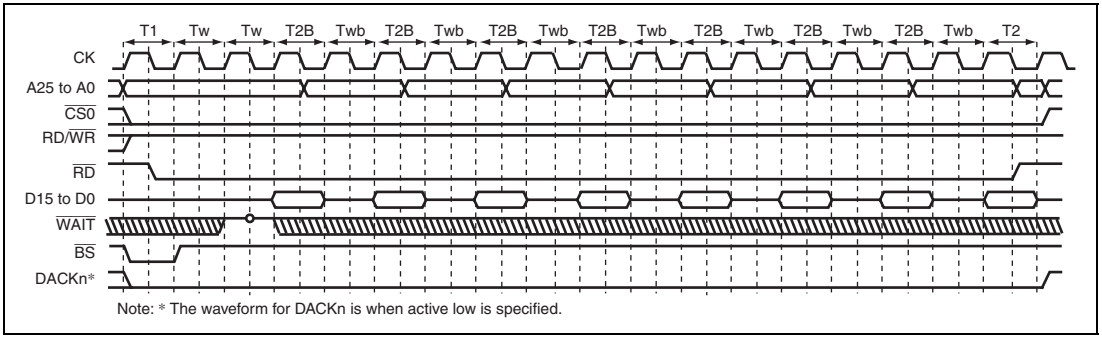


Figure 10.41 Burst ROM Access Timing (Clock Synchronous)
(Burst Length = 8, Wait Cycles Inserted in First Access = 2,
Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

10.5.10 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. A function that avoids data collisions by inserting idle (wait) cycles between continuous access cycles has been newly added.

The number of wait cycles between access cycles can be set by the WM bit in CSnWCR, bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR, and bits DMAIW2 to DMAIW0 and DMAIWA in CMNCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space
6. Data output from an external device caused by DMA single address transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single address transfer is followed by any type of access (DMAIWA = 1)

For the specification of the number of idle cycles between access cycles described above, refer to the description of each register.

Besides the idle cycles between access cycles specified by the registers, idle cycles must be inserted to interface with the internal bus or to obtain the minimum pulse width for a multiplexed pin ($\overline{\text{WRxx}}$). The following gives detailed information about the idle cycles and describes how to estimate the number of idle cycles.

The number of idle cycles on the external bus from $\overline{\text{CSn}}$ negation to $\overline{\text{CSn}}$ or $\overline{\text{CSm}}$ assertion is described below.

There are eight conditions that determine the number of idle cycles on the external bus as shown in table 10.19. The effects of these conditions are shown in figure 10.42.

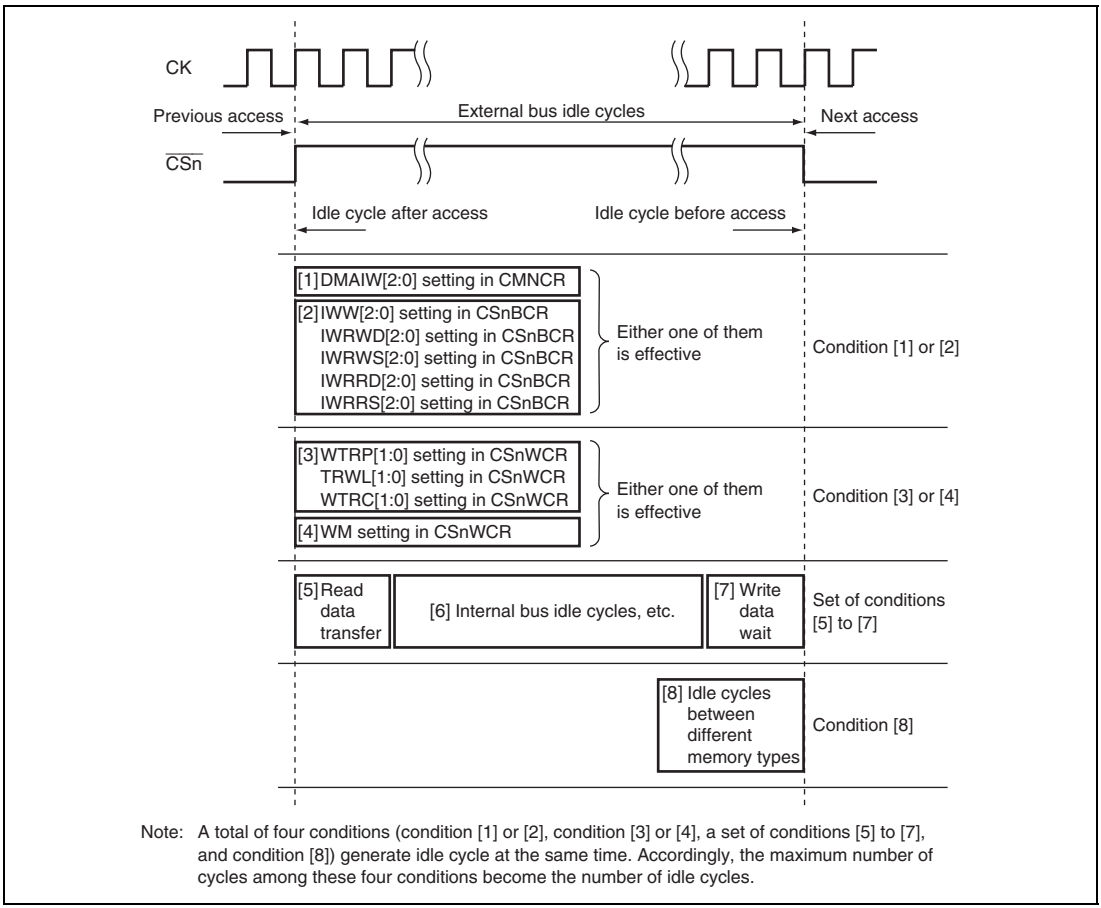
Table 10.19 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
(1)	DMAIW[2:0] in CMNCR	These bits specify the number of idle cycles for DMA single address transfer. This condition is effective only for single address transfer and generates idle cycles after the access is completed.	0 to 12	When 0 is specified for the number of idle cycles, the DACK signal may be asserted continuously. This causes a discrepancy between the number of cycles detected by the device with DACK and the DMAC transfer count, resulting in a malfunction.
(2)	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access other than single address transfer. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition is effective only for access cycles other than single address transfer and generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
(3)	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait cycles and idle cycles between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle cycles after the access is completed	0 to 3	Specify these bits in accordance with the specification of the target SDRAM.
(4)	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	

No.	Condition	Description	Range	Note
(5)	Read data transfer cycle	One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not B'00.	0 or 1*	One idle cycle is always generated after a read cycle with SDRAM interface.
(6)	Internal bus idle cycles, etc.	External bus access requests from the CPU or DMAC/DTC and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the BSC when the access size is larger than the external data bus width.	0 or larger	The number of internal bus idle cycles may not become 0 depending on the Iϕ:Bϕ clock ratio. Tables 10.20 and 10.21 show the relationship between the clock ratio and the minimum number of internal bus idle cycles.
(7)	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles are frequently available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through CSnBCR.
(8)	Idle cycles between different memory types	To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2.5	The number of idle cycles depends on the target memory types. See table 10.22.

Note: * This is the case for consecutive read operations when the data read are stored in separate registers.

In the above conditions, a total of four conditions, that is, condition (1) or (2) (either one is effective), condition (3) or (4) (either one is effective), a set of conditions (5) to (7) (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition (8) are generated at the same time. The maximum number of idle cycles among these four conditions becomes the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition (1) or (2).



Note: A total of four conditions (condition [1] or [2], condition [3] or [4], a set of conditions [5] to [7], and condition [8]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 10.42 Idle Cycle Conditions

Table 10.20 Minimum Number of Idle Cycles on Internal Bus (CPU Operation)

	Clock Ratio (Iϕ:Bϕ)			
CPU Operation	**8:1**	**4:1**	**2:1**	**1:1**
Write → write	1	2	2	3
Write → read	0	0	0	1
Read → write	1	2	2	3
Read → read	0	0	0	1

Table 10.21 Minimum Number of Idle Cycles on Internal Bus (DMAC Operation)

	Transfer Mode	
DMAC Operation	**Dual Address**	**Single Address**
Write → write	0	2
Write → read	0 or 2	0
Read → write	0	0
Read → read	0	2

Notes: 1. The write → write and read → read columns in dual address transfer indicate the cycles in the divided access cycles.

2. For the write → read cycles in dual address transfer, 0 is applicable when different channels are activated successively and 2 is applicable when the same channel is activated successively. When a DMAC activation request is issued from an on-chip peripheral module and the activation is made successively in burst mode, the number of cycle is 0 even when the same channel is activated successively.

3. The write → read and read → write columns in single address transfer indicate the case when different channels are activated successively. The "write" means transfer from a device with DACK to external memory and the "read" means transfer from external memory to a device with DACK.

Table 10.22 Number of Idle Cycles Inserted between Access Cycles to Different Memory Types

	Next Cycle							
Previous Cycle	**SRAM**	**Burst ROM (Asynchronous)**	**MPX-I/O**	**Byte SRAM (BAS = 0)**	**Byte SRAM (BAS = 1)**	**SDRAM**	**SDRAM (Low-Frequency Mode)**	**Burst ROM (Synchronous)**
SRAM	0	0	1	0	1	1	1.5	0
Burst ROM (asynchronous)	0	0	1	0	1	1	1.5	0
MPX-I/O	1	1	0	1	1	1	1.5	1
Byte SRAM (BAS = 0)	0	0	1	0	1	1	1.5	0
Byte SRAM (BAS = 1)	1	1	2	1	0	0	1.5	1
SDRAM	1	1	2	1	0	0	—	1
SDRAM (low-frequency mode)	1.5	1.5	2.5	1.5	0.5	—	1	1.5
Burst ROM (synchronous)	0	0	1	0	1	1	1.5	0

Figure 10.43 shows sample estimation of idle cycles between access cycles. In the actual operation, the idle cycles may become shorter than the estimated value due to the write buffer effect or may become longer due to internal bus idle cycles caused by stalling in the pipeline due to CPU instruction execution or CPU register conflicts. Please consider these errors when estimating the idle cycles.

Sample Estimation of Idle Cycles between Access Cycles

This example estimates the idle cycles for data transfer from the CS1 space to CS2 space by CPU access. Transfer is repeated in the following order: CS1 read → CS1 read → CS2 write → CS2 write → CS1 read → ...

- Conditions
  The bits for setting the idle cycles between access cycles in CS1BCR and CS2BCR are all set to 0.
  In CS1WCR and CS2WCR, the WM bit is set to 1 (external $\overline{\text{WAIT}}$ pin disabled) and the HW[1:0] bits are set to 00 ($\overline{\text{CS}}$ negation is not extended).
  Iϕ:Bϕ is set to 4:1, and no other processing is done during transfer.
  For both the CS1 and CS2 spaces, normal SRAM devices are connected, the bus width is 32 bits, and access size is also 32 bits.

The idle cycles generated under each condition are estimated for each pair of access cycles. In the following table, R indicates a read cycle and W indicates a write cycle.

Condition	R → R	R → W	W → W	W → R	Note
[1] or [2]	0	0	0	0	CSnBCR is set to 0.
[3] or [4]	0	0	0	0	The WM bit is set to 1.
[5]	1	1	0	0	Generated after a read cycle.
[6]	0	2	2	0	See the Iϕ:Bϕ = 4:1 column in table 10.20.
[7]	0	1	0	0	No idle cycle is generated for the second time due to the write buffer effect.
[5] + [6] + [7]	1	4	2	0	
[8]	0	0	0	0	Value for SRAM → SRAM access
Estimated idle cycles	1	4	2	0	Maximum value among conditions [1] or [2], [3] or [4], [5] + [6] + [7], and [8]
Actual idle cycles	1	4	2	1	The estimated value does not match the actual value in the W → R cycles because the internal idle cycles due to condition [6] is estimated as 0 but actually an internal idle cycle is generated due to execution of a loop condition check instruction.

Figure 10.43 Comparison between Estimated Idle Cycles and Actual Value

10.5.11 Bus Arbitration

The bus arbitration of this LSI has the bus mastership in the normal state and releases the bus mastership after receiving a bus request from another device. This LSI has three internal bus masters: CPU, DMAC and DTC. The priority level of bus mastership transferred between these bus masters is:

Bus mastership request from external device (BREQ) > DTC > DMAC > CPU

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CSn}}$ signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. Between the read and write cycles of a TAS instruction
2. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)
3. 16-byte transfer by the DMAC
4. Setting the BLOCK bit in CMNCR to 1

Moreover, by using DPRTY bit in CMNCR, whether the bus mastership request is received or not can be selected during DMAC burst transfer.

The LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the external device has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

With the SDRAM interface, all bank pre-charge commands (PALLs) are issued when active banks exist and the bus is released after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CK. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CK. The bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, CKE, DQMxx, $\overline{\text{WRxx}}$, $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the

high-impedance state at subsequent rising edges of CK. Bus request signals are sampled at the falling edge of CKIO. Note that CKE, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$ can continue to be driven at the previous value even in the bus-released state by setting the HIZCNT bit in CMNCR.

The sequence for reclaiming the bus mastership from an external device is described below. 1.5 cycles after the negation of $\overline{\text{BREQ}}$ is detected at the falling edge of CK, the bus control signals are driven high. The bus acknowledge signal is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CK where address and data signals are driven. Figure 10.44 shows the bus arbitration timing.

When it is necessary to refresh SDRAM while releasing the bus mastership, the bus mastership should be returned using the $\overline{\text{REFOUT}}$ signal. For details on the selection of $\overline{\text{REFOUT}}$, see section 22, Pin Function Controller (PFC). The $\overline{\text{REFOUT}}$ signal is kept asserting at low level until the bus mastership is acquired. The $\overline{\text{BREQ}}$ signal is negated by asserting the $\overline{\text{REFOUT}}$ signal and the bus mastership is returned from the external device. If the bus mastership is not returned for a refreshing period or longer, the contents of SDRAM cannot be guaranteed because a refreshing cannot be executed.

While releasing the bus mastership, the SLEEP instruction (to enter sleep mode or standby mode), as well as a manual reset, cannot be executed until the LSI obtains the bus mastership.

The $\overline{\text{BREQ}}$ input signal is ignored and the $\overline{\text{BACK}}$ output signal is placed in the high impedance state in software standby mode or deep software standby mode. If the bus mastership request is required in this state, the bus mastership must be released by pulling down the $\overline{\text{BACK}}$ pin to enter standby mode.

The bus mastership release ($\overline{\text{BREQ}}$ signal for high level negation) after the bus mastership request ($\overline{\text{BREQ}}$ signal for low level assertion) must be performed after the bus usage permission ($\overline{\text{BACK}}$ signal for low level assertion). If the $\overline{\text{BREQ}}$ signal is negated before the $\overline{\text{BACK}}$ signal is asserted, only one cycle of the $\overline{\text{BACK}}$ signal is asserted depending on the timing of the $\overline{\text{BREQ}}$ signal to be negated and this may cause a bus contention between the external device and the LSI.

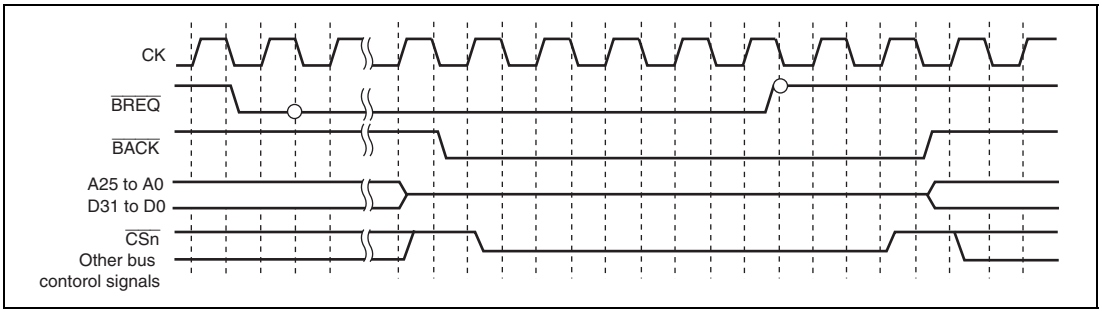


Figure 10.44 Bus Arbitration Timing

10.5.12 Others

(1) Reset

The bus state controller (BSC) can be initialized completely only at power-on reset. At power-on reset, all signals are negated and data output buffers are turned off regardless of the bus cycle state after the internal reset is synchronized with the internal clock. All control registers are initialized. In software standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, only the current bus cycle being executed is completed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle.

(2) Access from the Side of the LSI Internal Bus Master

This LSI bus is divided into three buses: CPU bus, internal bus, and peripheral bus. The CPU is connected to the CPU bus, the other internal bus masters and the bus state controller are connected to the internal bus, and the low-speed peripheral modules and on-chip RAM (for data retention) are connected to the peripheral bus. The on-chip memories other than on-chip RAM (for data retention) are connected to both of the CPU bus and internal bus. Accesses can be performed from the CPU bus to the internal bus, but cannot be performed in the reverse direction.

In read cycles from on-chip peripheral modules, the read cycle is activated via the internal bus and peripheral bus. The read data is transferred to the CPU via the peripheral bus, internal bus, and CPU bus.

In accesses to external spaces or on-chip peripheral modules, the access cycle is activated via the internal bus. Therefore, when the CPU accesses on-chip memory, the DMAC/DTC can be activated without generating bus arbitration.

Since the bus state controller (BSC) incorporates a four-stage write buffer, the BSC can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC. Accordingly, to perform dual address DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next write cycle will not be initiated until the previous write cycle is completed.

Changing the registers in the BSC while the write buffer is operating may disrupt correct write access. Therefore, do not change the registers in the BSC immediately after a write access. If this change becomes necessary, do it after executing a dummy read of the write data.

(3) Accesses to On-Chip Peripheral Module Registers or On-Chip RAM (for Data Retention)

To access an on-chip module register or on-chip RAM (for data retention), two or more peripheral module clock (Pϕ) cycles are required. When the CPU writes data to the internal peripheral registers or on-chip RAM (for data retention), the CPU performs the succeeding instructions without waiting for the completion of writing to registers or on-chip RAM (for data retention).

For example, a case is described here in which the system is transferring to software standby mode for power savings. To make this transition, the SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR register is indispensable to complete writing to the STBY bit.

To reflect the change by the internal peripheral registers or on-chip RAM (for data retention) while performing the succeeding instruction, execute a dummy read from the registers or on-chip RAM (for data retention) after the write instruction to the register or on-chip RAM (for data retention) to which the write is given, and then execute the succeeding instruction.

(4) Number of Access Cycles

Tables 10.23 (1) to (3) show the number of cycles required to access the on-chip memory, on-chip I/O registers, and external devices.

Table 10.23 Number of Access Cycles (1)

		Access Destination					
		On-Chip ROM	On-Chip RAM (High-Speed)	On-Chip RAM (for Data Retention)	External Device*4		
Bus width		32 bits	32 bits	32 bits	8 bits	16 bits	32 bits
Access from CPU	Instruction fetch	1	1	2Pϕ + m1 + m2 + m3	9Bϕ + m1 + m3	5Bϕ + m1 + m3	3Bϕ + m1 + m3
	Data read (longword)	1	1				
	Data read (word)	1	1		5Bϕ + m1 + m3	3Bϕ + m1 + m3	
	Data read (byte)	1	1		3Bϕ + m1 + m3		
	Data write*1 (longword)	—	1	2Pϕ + m1 + m2	9Bϕ + m1	5Bϕ + m1	3Bϕ + m1
	Data write*1 (word)	—	1		5Bϕ + m1	3Bϕ + m1	
	Data write*1 (byte)	—	1		3Bϕ + m1		
Access from modules other than CPU	Data read (longword)	3Bϕ to 3Bϕ + 2Iϕ*2	1Bϕ to 3Bϕ*3	2Pϕ + m2	9Bϕ	5Bϕ	3Bϕ
	Data read (word)				5Bϕ	3Bϕ	
	Data read (byte)				3Bϕ		
	Data write*1 (longword)	—		2Pϕ + m2	9Bϕ	5Bϕ	3Bϕ
	Data write*1 (word)	—			5Bϕ	3Bϕ	
	Data write*1 (byte)	—			3Bϕ		

Notes: The m1, m2, and m3 are determined by the internal clock rate and they vary according to the internal clock state as follows.

			Iϕ Bϕ			
			1:1	**2:1**	**4:1**	**8:1**
Bϕ Pϕ	1:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ	1Bϕ	1Bϕ	1Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	2:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	4:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	8:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ

1. Using the write buffer, the bus master can execute the succeeding processing before the previous write cycle is completed. For details, see section 10.5.12 (2), Access from the Side of the LSI Internal Bus Master.
2. When Iϕ:Bϕ = 1:1 or 1:1/2, the value is 3Bϕ + 2Iϕ.
   When Iϕ:Bϕ = 1:1/4 or 1:1/8, the value is 3Bϕ.
3. When Iϕ:Bϕ = 1:1, the value is 3Bϕ.
   When or 1/2:1/4, the value is 2Bϕ.
   When Iϕ:Bϕ = 1:1/8, the value is 1Bϕ.
4. Indicates the number of access cycles when the normal space is accessed with no wait (no idle cycles, without no wait cycles, and external wait input ignored).

Table 10.23 Number of Access Cycles (2)

		Access Destination			
		On-Chip I/O Register			
		BSC	USB	Flash Control, PLDRL of I/O Port	TIM32C, KEYC, PFDRL of I/O Port
Bus width		32 bits	32 bits	16 bits	16 bits
Access from CPU	Instruction fetch	—	—	—	—
	Data read (longword)	1Bϕ + m1 + m3	3Bϕ + m1 + m3	10Pϕ + m1 + m2 + m3	8Pϕ + m1 + m2 + m3
	Data read (word)			5Pϕ + m1 + m2 + m3	4Pϕ + m1 + m2 + m3
	Data read (byte)				
	Data write* (longword)	3Bϕ + m1	3Bϕ + m1	10Pϕ + m1 + m2	8Pϕ + m1 + m2
	Data write* (word)			5Pϕ + m1 + m2	4Pϕ + m1 + m2
	Data write* (byte)				
Access from modules other than CPU	Data read (longword)	1Bϕ	3Bϕ	10Pϕ + m2	8Pϕ + m2
	Data read (word)			5Pϕ + m2	4Pϕ + m2
	Data read (byte)				
	Data write* (longword)	3Bϕ	3Bϕ	10Pϕ + m2	8Pϕ + m2
	Data write* (word)			5Pϕ + m2	4Pϕ + m2
	Data write* (byte)				

Note: The m1, m2, and m3 are determined by the internal clock rate and they vary according to the internal clock state as follows.

			Iϕ Bϕ			
			1:1	**2:1**	**4:1**	**8:1**
Bϕ Pϕ	1:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ	1Bϕ	1Bϕ	1Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	2:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	4:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	8:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ

* Using the write buffer, the bus master can execute the succeeding processing before the previous write cycle is completed. For details, see section 10.5.12 (2), Access from the Side of the LSI Internal Bus Master.

Table 10.23 Number of Access Cycles (3)

		On-Chip I/O Register			
Access Destination		**PDCRH1/2, PDCRL1/2, PGCRL1, PGPCRL of PFC, DPSTBCR, DPSWCR, DPSIER, SIFR, DPSIECR, RSTSR for Power-Down Mode,**	**IIC3**	**LVDS (for SH72315A only), ADSR of ADC**	**On-Chip I/O Registers for Modules Other Than Those in Table 10.23 (2) and in the Left Columns**
Bus width		16 bits	8 bits	32 bits	16 bits
Access from CPU	Instruction fetch	—	—	—	—
	Data read (longword)	6Pϕ + m1 +m2 + m3	8Pϕ + m1 + m2 + m3	2Pϕ + m1 + m2 + m3	4Pϕ + m1 + m2 + m3
	Data read (word)	3Pϕ + m1 +m2 + m3	4Pϕ + m1 + m2 + m3		2Pϕ + m1 + m2 + m3
	Data read (byte)		2Pϕ + m1 + m2 + m3		
	Data write* (longword)	6Pϕ + m1 + m2	8Pϕ + m1 + m2	2Pϕ + m1 + m2	4Pϕ + m1 + m2
	Data write* (word)	3Pϕ + m1 + m2	4Pϕ + m1 + m2		2Pϕ + m1 + m2
	Data write* (byte)		2Pϕ + m1 + m2		
Access from modules other than CPU	Data read (longword)	6Pϕ + m2	8Pϕ + m2	2Pϕ + m2	4Pϕ + m2
	Data read (word)	3Pϕ + m2	4Pϕ + m2		2Pϕ + m2
	Data read (byte)		2Pϕ + m2		
	Data write* (longword)	6Pϕ + m2	8Pϕ + m2	2Pϕ + m2	4Pϕ + m2
	Data write* (word)	3Pϕ + m2	4Pϕ + m2		2Pϕ + m2
	Data write* (byte)		2Pϕ + m2		

Note: The m1, m2, and m3 are determined by the internal clock rate and they vary according to the internal clock state as follows.

			Iϕ Bϕ			
			1:1	**2:1**	**4:1**	**8:1**
Bϕ Pϕ	1:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ	1Bϕ	1Bϕ	1Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	2:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ	1Bϕ to 2Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	4:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ	1Bϕ to 4Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ
	8:1	m1	2Iϕ	2Iϕ to 3Iϕ	2Iϕ to 5Iϕ	2Iϕ to 9Iϕ
		m2	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ	1Bϕ to 8Bϕ
		m3	2Iϕ	3Iϕ	4Iϕ	4Iϕ

* Using the write buffer, the bus master can execute the succeeding processing before the previous write cycle is completed. For details, see section 10.5.12 (2), Access from the Side of the LSI Internal Bus Master.

Synchronous logic and a layered bus structure have been adopted for this LSI. Data on each bus are input and output in synchronization with rising edges of the corresponding clock signal. The CPU bus, the internal bus, and the peripheral bus are synchronized with the Iϕ, Bϕ, and Pϕ clock, respectively.

Figure 10.45 shows an example of the timing of write access to the peripheral bus when Iϕ:Bϕ:Pϕ = 4:4:1. Data are output to the CPU bus, which is connected to the CPU, in synchronization with Iϕ. When Iϕ:Bϕ = 1:1, transfer of data from the CPU bus to the internal bus takes 2 × Iϕ + 1 × Bϕ. When transfer is performed from the internal bus to the peripheral bus with Bϕ:Pϕ = 4:1, there are 4 cycles of Bϕ to one cycle of 1 × Pϕ, so four transfers to the peripheral bus can proceed in one cycle of Bϕ. Thus, a period of up to 4 × Bϕ may be required before a rising edge of Pϕ, which is the time of transfer from the internal bus to the peripheral bus (a case where this takes 4 × Bϕ is indicated in figure 10.45). When Bϕ: Pϕ = 4:1, transfer of data from the internal bus to the peripheral bus takes a period from 1 × Bϕ to 4 × Bϕ. The relation between the timing of data transfer to the CPU bus and the rising edge of Pϕ depends on the state of program execution. In figure 10.45, the time required for access is 2 × Iϕ + 4 × Bϕ + 2 × Pϕ.

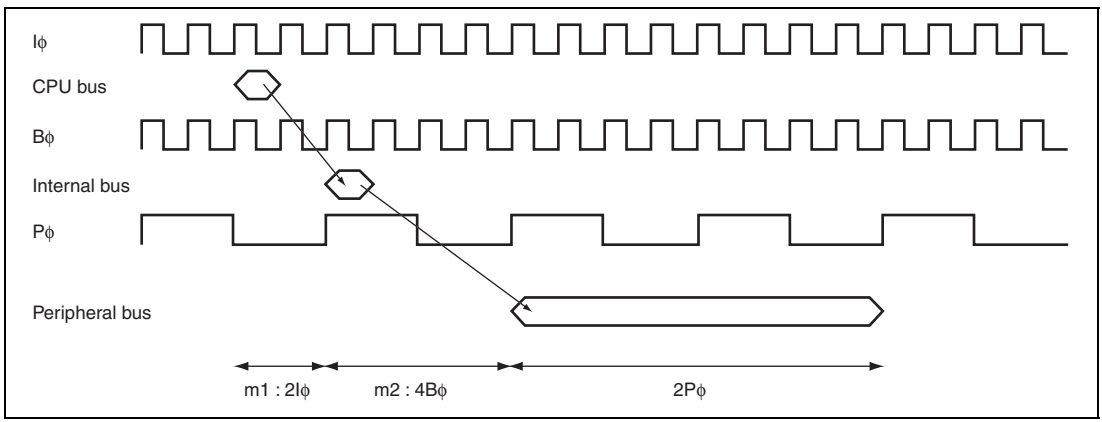


Figure 10.45 Timing of Write Access to On-Chip Peripheral I/O Registers and On-Chip RAM (for Data Retention) When Iϕ;Bϕ:Pϕ = 4:4:1

Figure 10.46 shows an example of timing of read access to the peripheral bus when Iϕ:Bϕ:Pϕ = 4:2:1. Transfer from the CPU bus to the peripheral bus is performed in the same way as for write access. In the case of reading, however, values output onto the peripheral bus must be transferred to the CPU. Although transfers from the peripheral bus to the internal bus and from the internal bus to the CPU bus are performed in synchronization with the rising edge of the respective bus clocks, a period of 3 × Iϕ is actually required because Iϕ ≥ Bϕ≥ Pϕ. In the case shown in the figure 10.46, the time required for access is 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 3 × Iϕ.

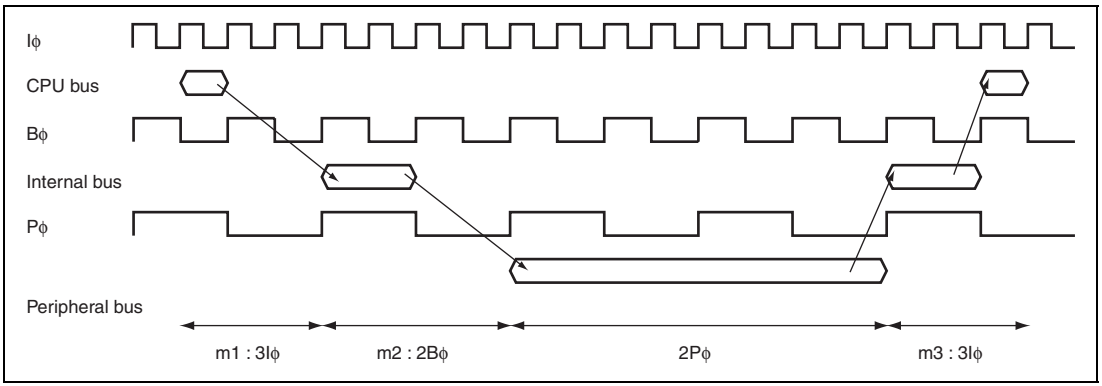


Figure 10.46 Timing of Read Access to On-Chip Peripheral I/O Registers and On-Chip RAM (for Data Retention) When Iϕ:Bϕ:Pϕ = 4:2:1

10.6 Interrupt Source

Table 10.24 gives details on this interrupt source. The compare match interrupt enable bit (CMIE) in the refresh timer control/status register (RTCSR) can be used to enable or disable the interrupt source.

The compare match interrupt (CMI) is generated when the compare match flag (CMF) and compare match interrupt enable bit (CMIE) in RTCSR are set to 1.

Clearing the interrupt flag bit to 0 cancels the interrupt request.

Table 10.24 Interrupt Source

Abbreviation	Interrupt Source	Interrupt Enable Bit	Interrupt Flag
CMI	Compare match interrupt	CMIE	CMF

Section 11 Direct Memory Access Controller (DMAC)

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

11.1 Features

- Number of channels: Four channels (channels 0 to 3)
  CH0 and CH1 channels can only receive external requests.
- 4-Gbyte physical address space
- Transfer data length is selectable: Byte, word (two bytes), longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests
  — External request
  — On-chip peripheral module request
  — Auto request
- Selectable bus modes
  — Cycle steal mode (normal mode and intermittent mode)
  — Burst mode
- Selectable channel priority levels: The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be sent to the CPU on completion of half- or full-data transfer. Through the HE and HIE bits in CHCR, an interrupt is specified to be issued to the CPU when half of the initially specified DMA transfer is completed.
- External request detection: There are following four types of DREQ input detection.
  — Low level detection
  — High level detection
  — Rising edge detection
  — Falling edge detection
- Transfer request acknowledge and transfer end signals: Active levels for DACK and TEND can be set independently.

- Support of reload functions in DMA transfer information registers: DMA transfer using the same information as the current transfer can be repeated automatically without specifying the information again. Modifying the reload registers during DMA transfer enables next DMA transfer to be done using different transfer information. The reload function can be enabled or disabled independently in each channel.

Figure 11.1 shows the block diagram of the DMAC.

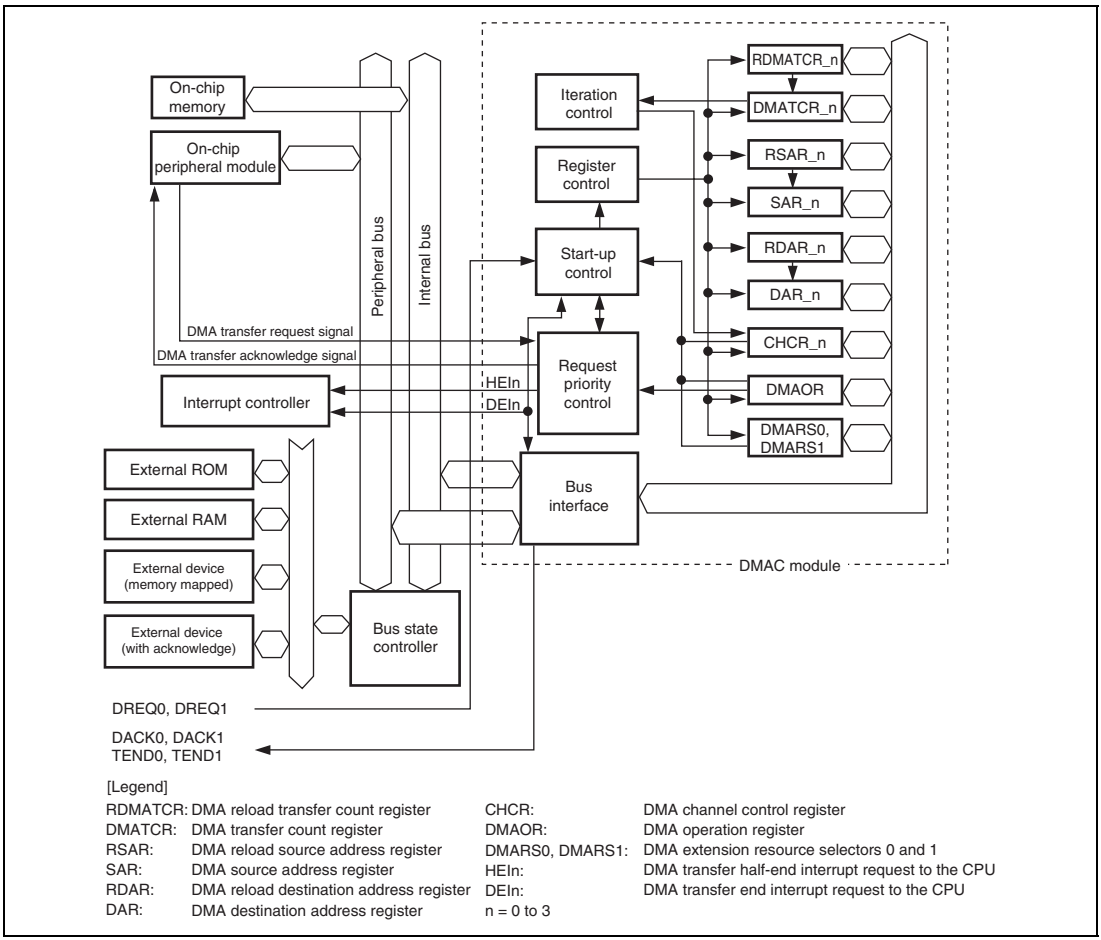


Figure 11.1 Block Diagram of DMAC

11.2 Input/Output Pins

The external pins for DMAC are described below. Table 11.1 lists the configuration of the pins that are connected to external bus. DMAC has pins for two channels (CH0 and CH1) as the external bus use.

Table 11.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	DMA transfer request	DREQ0	I	DMA transfer request input from an external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer request acknowledge output from channel 0 to an external device
	DMA transfer end	TEND0	O	DMA transfer end output for channel 0
1	DMA transfer request	DREQ1	I	DMA transfer request input from an external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer request acknowledge output from channel 1 to an external device
	DMA transfer end	TEND1	O	DMA transfer end output for channel 1

11.3 Register Descriptions

The DMAC has the registers listed in table 11.2. There are four control registers and three reload registers for each channel, and one common control register is used by all channels. In addition, there is one extension resource selector per two channels. Each channel number is expressed in the register names, as in SAR_0 for SAR in channel 0. For the addresses and states of these registers in each processing status, refer to section 34, List of Registers.

Table 11.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA source address register_0	SAR_0	R/W	H'00000000	H'FFFE1000	16, 32
	DMA destination address register_0	DAR_0	R/W	H'00000000	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR_0	R/W	H'00000000	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR_0	R/W	H'00000000	H'FFFE100C	16, 32
	DMA reload source address register_0	RSAR_0	R/W	H'00000000	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR_0	R/W	H'00000000	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR_0	R/W	H'00000000	H'FFFE1108	16, 32
1	DMA source address register_1	SAR_1	R/W	H'00000000	H'FFFE1010	16, 32
	DMA destination address register_1	DAR_1	R/W	H'00000000	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR_1	R/W	H'00000000	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR_1	R/W	H'00000000	H'FFFE101C	16, 32
	DMA reload source address register_1	RSAR_1	R/W	H'00000000	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR_1	R/W	H'00000000	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR_1	R/W	H'00000000	H'FFFE1118	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	DMA source address register_2	SAR_2	R/W	H'00000000	H'FFFE1020	16, 32
	DMA destination address register_2	DAR_2	R/W	H'00000000	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR_2	R/W	H'00000000	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR_2	R/W	H'00000000	H'FFFE102C	16, 32
	DMA reload source address register_2	RSAR_2	R/W	H'00000000	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR_2	R/W	H'00000000	H'FFFE1124	16, 32
	DMA reload transfer count register_2	RDMATCR_2	R/W	H'00000000	H'FFFE1128	16, 32
3	DMA source address register_3	SAR_3	R/W	H'00000000	H'FFFE1030	16, 32
	DMA destination address register_3	DAR_3	R/W	H'00000000	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR_3	R/W	H'00000000	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR_3	R/W	H'00000000	H'FFFE103C	16, 32
	DMA reload source address register_3	RSAR_3	R/W	H'00000000	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR_3	R/W	H'00000000	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR_3	R/W	H'00000000	H'FFFE1138	16, 32
Common	DMA operation register	DMAOR	R/W	H'0000	H'FFFE1200	16
0 and 1	DMA extension resource selector 0	DMARS0	R/W	H'0000	H'FFFE1300	16
2 and 3	DMA extension resource selector 1	DMARS1	R/W	H'0000	H'FFFE1304	16

11.3.1 DMA Source Address Registers (SAR)

The DMA source address registers (SAR) are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address. When the data of an external device with DACK is transferred in single address mode, SAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

SAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.3.2 DMA Destination Address Registers (DAR)

The DMA destination address registers (DAR) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address. When the data of an external device with DACK is transferred in single address mode, DAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

DAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.3.3 DMA Transfer Count Registers (DMATCR)

The DMA transfer count registers (DMATCR) are 32-bit readable/writable registers that specify the number of DMA transfers. The transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of DMATCR are always read as 0, and the write value should always be 0. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

DMATCR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.3.4 DMA Channel Control Registers (CHCR)

The DMA channel control registers (CHCR) are 32-bit readable/writable registers that control DMA transfer mode.

The DO, AM, AL, DL, and DS bits which specify the DREQ and DACK external pin functions and the TL bit which specifies the TEND external pin function can be read and written to in channels 0 and 1, but they are reserved in channels 2 and 3. Before modifying the CHCR setting, clear the DE bit for the corresponding channel.

CHCR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC	-	-	RLD	SARE	DARE	TCRE	-	DO	TL	-	-	HE	HIE	AM	AL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R/(W)*	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Descriptions
31	TC	0	R/W	Transfer Count Mode Specifies whether to transmit data once or for the count specified in DMATCR by one transfer request. Note that when this bit is set to 0, the TB bit must not be set to 1 (burst mode). When the SCI, SCIF, IIC3, RSPI, or RCAN-ET is selected for the transfer request source, this bit (TC) must be set to 0. 0: Transmits data once by one transfer request 1: Transmits data for the count specified in DMATCR by one transfer request
30, 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
28	RLD	0	R/W	Reload Function Enable or Disable Enables or disables the reload function. 0: Disables the reload function 1: Enables the reload function
27	SARE	0	R/W	SAR Reload Enable Enables or disables reloading the RSAR value into SAR. When this bit is 1, the RLD bit should also be 1. 0: Disables reloading the RSAR value into SAR. 1: Enables reloading the RSAR value into SAR.
26	DARE	0	R/W	DAR Reload Enable Enables or disables reloading the RDAR value into DAR. When this bit is 1, the RLD bit should also be 1. 0: Disables reloading the RDAR value into DAR. 1: Enables reloading the RDAR value into DAR.
25	TCRE	0	R/W	RDMATCR Reload Enable Enables or disables reloading the RDMATCR value into DMATCR. When this bit is 1, the RLD bit should also be 1. 0: Disables reloading the RDMATCR value into DMATCR. 1: Enables reloading the RDMATCR value into DMATCR.
24	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR_0 and CHCR_1. This bit is reserved in CHCR_2 and CHCR_3; it is always read as 0 and the write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1

Bit	Bit Name	Initial Value	R/W	Descriptions
22	TL	0	R/W	Transfer End Level Specifies the TEND signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is reserved in CHCR_2 and CHCR_3; it is always read as 0 and the write value should always be 0. 0: Low-active output from TEND 1: High-active output from TEND
21, 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	HE	0	R/(W)*	Half-End Flag This bit is set to 1 when the transfer count reaches half of the DMATCR value that was specified before transfer starts. If DMA transfer ends because of an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR before the transfer count reaches half of the initial DMATCR value, the HE bit is not set to 1. If DMA transfer ends due to an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR after the HE bit is set to 1, the bit remains set to 1. To clear the HE bit, write 0 to it after HE = 1 is read. 0: DMATCR > (DMATCR set before transfer starts)/2 during DMA transfer or after DMA transfer is terminated [Clearing condition] • Writing 0 after reading HE = 1. 1: DMATCR ≤ (DMATCR set before transfer starts)/2

Bit	Bit Name	Initial Value	R/W	Descriptions
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether to issue an interrupt request to the CPU when the transfer count reaches half of the DMATCR value that was specified before transfer starts. When the HIE bit is set to 1, the DMAC requests an interrupt to the CPU when the HE bit becomes 1. 0: Disables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2 1: Enables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2
17	AM	0	R/W	Acknowledge Mode Specifies whether DACK is output in data read cycle or in data write cycle in dual address mode. In single address mode, DACK is always output regardless of the specification by this bit. This bit is valid only in CHCR_0 and CHCR_1. This bit is reserved in CHCR_2 and CHCR_3; it is always read as 0 and the write value should always be 0. 0: DACK output in read cycle (dual address mode) 1: DACK output in write cycle (dual address mode)
16	AL	0	R/W	Acknowledge Level Specifies the DACK (acknowledge) signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is reserved in CHCR_2 and CHCR_3; it is always read as 0 and the write value should always be 0. 0: Low-active output from DACK 1: High-active output from DACK

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	DM[1:0]	00	R/W	Destination Address Mode These bits select whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, DM1 and DM0 bits are ignored when data is transferred to an external device with DACK.) 00: Fixed destination address (Setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer, setting prohibited in 16-byte transfer) 11: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer)

Bit	Bit Name	Initial Value	R/W	Descriptions
13, 12	SM[1:0]	00	R/W	Source Address Mode These bits select whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with DACK.) 00: Fixed source address (Setting prohibited in 16-byte-unit transfer) 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer, setting prohibited in 16-byte-unit transfer) 11: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer)

Bit	Bit Name	Initial Value	R/W	Descriptions
11 to 8	RS[3:0]	0000	R/W	Resource Select These bits specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state when DMA enable bit (DE) is set to 0. 0000: External request, dual address mode 0001: Setting prohibited 0010: External request/single address mode External address space → External device with DACK 0011: External request/single address mode External device with DACK → External address space 0100: Auto request 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: DMA extension resource selector 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited Note: External request specification is valid only in CHCR_0 and CHCR_1. If a request source is selected in channels CHCR_2 and CHCR_3, no operation will be performed.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	DREQ Level
6	DS	0	R/W	DREQ Edge Select These bits specify the sampling method of the DREQ pin input and the sampling level. These bits are valid only in CHCR_0 and CHCR_1. These bits are reserved in CHCR_2 and CHCR_3; they are always read as 0 and the write value should always be 0. If the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, the specification by these bits is ignored. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies bus mode when DMA transfers data. Note that burst mode must not be selected when TC = 0. 0: Cycle steal mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	Transfer Size These bits specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte unit 01: Word unit (two bytes) 10: Longword unit (four bytes) 11: 16-byte unit (four longwords)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when TE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag This bit is set to 1 when DMATCR becomes 0 and DMA transfer ends. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR becomes 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR). To clear the TE bit, write 0 after reading TE = 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been terminated [Clearing condition] • Writing 0 after reading TE = 1 1: DMA transfer ends by the specified count (DMATCR = 0)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto-request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this case, all of the bits TE, NMIF in DMAOR, and AE must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0 as in the case of auto-request mode. Clearing the DE bit to 0 can terminate the DMA transfer. Before modifying the CHCR setting, clear the DE bit for the corresponding channel to 0. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Only 0 can be written to clear the flag after 1 is read.

11.3.5 DMA Reload Source Address Registers (RSAR)

The DMA reload source address registers (RSAR) are 32-bit readable/writable registers.

When the RLD and SARE bits in CHCR are set to 1, the RSAR value is written to the source address register (SAR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RSAR during the current DMA transfer. When the reload function is disabled, RSAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

RSAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.3.6 DMA Reload Destination Address Registers (RDAR)

The DMA reload destination address registers (RDAR) are 32-bit readable/writable registers.

When the RLD and DARE bits in CHCR are set to 1, the RDAR value is written to the destination address register (DAR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RDAR during the current DMA transfer. When the reload function is disabled, RDAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

RDAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.3.7 DMA Reload Transfer Count Registers (RDMATCR)

The DMA reload transfer count registers (RDMATCR) are 32-bit readable/writable registers.

When the RLD and TCRE bits in CHCR are set to 1, the RDMATCR value is written to the transfer count register (DMATCR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RDMATCR during the current DMA transfer. When the reload function is disabled, RDMATCR is ignored.

The upper eight bits of RDMATCR are always read as 0, and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

RDMATCR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.3.8 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register also shows the DMA transfer status.

DMAOR is initialized to H'0000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CMS[1:0]		-	-	PR[1:0]		-	-	-	-	-	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select These bits select either normal mode or intermittent mode in cycle steal mode. It is necessary that the bus modes of all channels be set to cycle steal mode to make intermittent mode valid. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer for every 16 cycles of Bϕ clock. 11: Intermittent mode 64 Executes one DMA transfer for every 64 cycles of Bϕ clock.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PR[1:0]	00	R/W	Priority Mode These bits select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 01: Fixed mode 2: CH0 > CH2 > CH3 > CH1 10: Setting prohibited 11: Round-robin mode
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*	Address Error Flag Indicates whether an address error has occurred by the DMAC. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading 1. 0: No DMAC address error 1: DMAC address error occurred [Clearing condition] • Write 0 after having read this bit as 1.
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading 1. When the NMI is input, the DMA transfer in progress can be done in one transfer unit. Even if the NMI interrupt is input while the DMAC is not in operation, the NMIF bit is set to 1. 0: No NMI interrupt 1: NMI interrupt occurred [Clearing condition] • Write 0 after having read this bit as 1.

Bit	Bit Name	Initial Value	R/W	Description
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfer on all channels. If the DME bit and DE bit in CHCR are set to 1, DMA transfer is enabled. However, transfer is enabled only when the TE bit in CHCR of the transfer corresponding channel, the NMIF bit in DMAOR, and the AE bit are all cleared to 0. Clearing the DME bit to 0 can terminate the DMA transfer on all channels. 0: DMA transfer is disabled on all channels 1: DMA transfer is enabled on all channels

Note: * To clear the flag, only 0 can be written after 1 is read.

If the priority mode bits are modified after a DMA transfer, the channel priority is initialized.

If fixed mode 2 is specified, the channel priority is specified as CH0 > CH2 > CH3 > CH1. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3. If round-robin mode is specified, the transfer end channel is reset.

The DMAC internal operation for an address error is as follows:

- No address error: Read (source to DMAC) → Write (DMAC to destination)
- Address error in source address: Nop → Nop
- Address error in destination address: Read → Nop

Table 11.3 Combinations of Priority Mode Bits

	Transfer End	Priority Mode Bits		Priority Level at the End of Transfer: High			Low
Mode	**CH No.**	**PR[1]**	**PR[0]**	**0**	**1**	**2**	**3**
Mode 0 (fixed mode 1)	Any channel	0	0	CH0	CH1	CH2	CH3
Mode 1 (fixed mode 2)	Any channel	0	1	CH0	CH2	CH3	CH1
Mode 2 (round-robin mode)	CH0	1	1	CH1	CH2	CH3	CH0
	CH1	1	1	CH2	CH3	CH0	CH1
	CH2	1	1	CH3	CH0	CH1	CH2
	CH3	1	1	CH0	CH1	CH2	CH3

11.3.9 DMA Extension Resource Selectors 0 and 1 (DMARS0, DMARS1)

The DMA extension resource selectors (DMARS) are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 is for channels 0 and 1 and DMARS1 is for channels 2 and 3. Table 11.4 shows the specifiable combinations.

DMARS is initialized to H'0000 by a reset and retains the value in software standby mode and module standby mode.

- DMARS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH1 MID[5:0]						CH1 RID[1:0]		CH0 MID[5:0]						CH0 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH3 MID[5:0]						CH3 RID[1:0]		CH2 MID[5:0]						CH2 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer requests from the various modules specify MID and RID as shown in table 11.4.

Table 11.4 DMARS Settings

Peripheral Module		Setting Value for One Channel ({MID, RID})	MID	RID	Function
SCI_0	TXI0	H'81	B'100000	B'01	Transmit
	RXI0	H'82		B'10	Receive
SCI_1	TXI1	H'85	B'100001	B'01	Transmit
	RXI1	H'86		B'10	Receive
SCI_2	TXI2	H'89	B'100010	B'01	Transmit
	RXI2	H'8A		B'10	Receive
SCI_3	TXI3	H'8D	B'100011	B'01	Transmit
	RXI3	H'8E		B'10	Receive
SCIF_4	TXI4	H'91	B'100100	B'01	Transmit
	RXI4	H'92		B'10	Receive
SCIF_5	TXI5	H'95	B'100101	B'01	Transmit
	RXI5	H'96		B'10	Receive
SCIF_6	TXI6	H'99	B'100110	B'01	Transmit
	RXI6	H'9A		B'10	Receive
SCIF_7	TXI7	H'9D	B'100111	B'01	Transmit
	RXI7	H'9E		B'10	Receive
IIC3	IITXI	H'A1	B'101000	B'01	Transmit
	IIRXI	H'A2		B'10	Receive
LVDS (SH72315A only)	LVRXI	H'A9	B'101010	B'01	Receive
A/D converter_0	ADI0	H'B1	B'101100	B'01	—
A/D converter_1	ADI1	H'B2	B'101100	B'10	—

Peripheral Module		Setting Value for One Channel ({MID, RID})	MID	RID	Function
MTU2_3S	TGI3AS	H'D1	B'110100	B'01	—
MTU2_4S	TGI4AS	H'D2	B'110100	B'10	—
MTU2_0	TGI0A	H'E1	B'111000	B'01	—
MTU2_1	TGI1A	H'E2	B'111000	B'10	—
MTU2_2	TGI2A	H'E3	B'111000	B'11	—
MTU2_3	TGI3A	H'ED	B'111011	B'01	—
MTU2_4	TGI4A	H'EE	B'111011	B'10	—
CMT2	CM2I	H'F0	B'111100	B'00	—
	IC0I	H'F4	B'111101	B'00	—
	IC1I	H'F5	B'111101	B'01	—
	OC0I	H'F6	B'111101	B'10	—
	OC1I	H'F7	B'111101	B'11	—
CMT_0	CMI0	H'F9	B'111110	B'01	—
CMT_1	CMI1	H'FA	B'111110	B'10	—
RSPI_0	SPTXI0	H'FC	B'111111	B'00	Transmit
	SPRXI0	H'FD	B'111111	B'01	Receive
RCAN-ET	RM0_0	H'FE	B'111111	B'10	Receive

When MID or RID other than the values listed in table 11.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS[3:0]) in CHCR0 to CHCR3 have been set to B'1000. Otherwise, even if DMARS has been set, the transfer request source is not accepted.

11.4 Operation

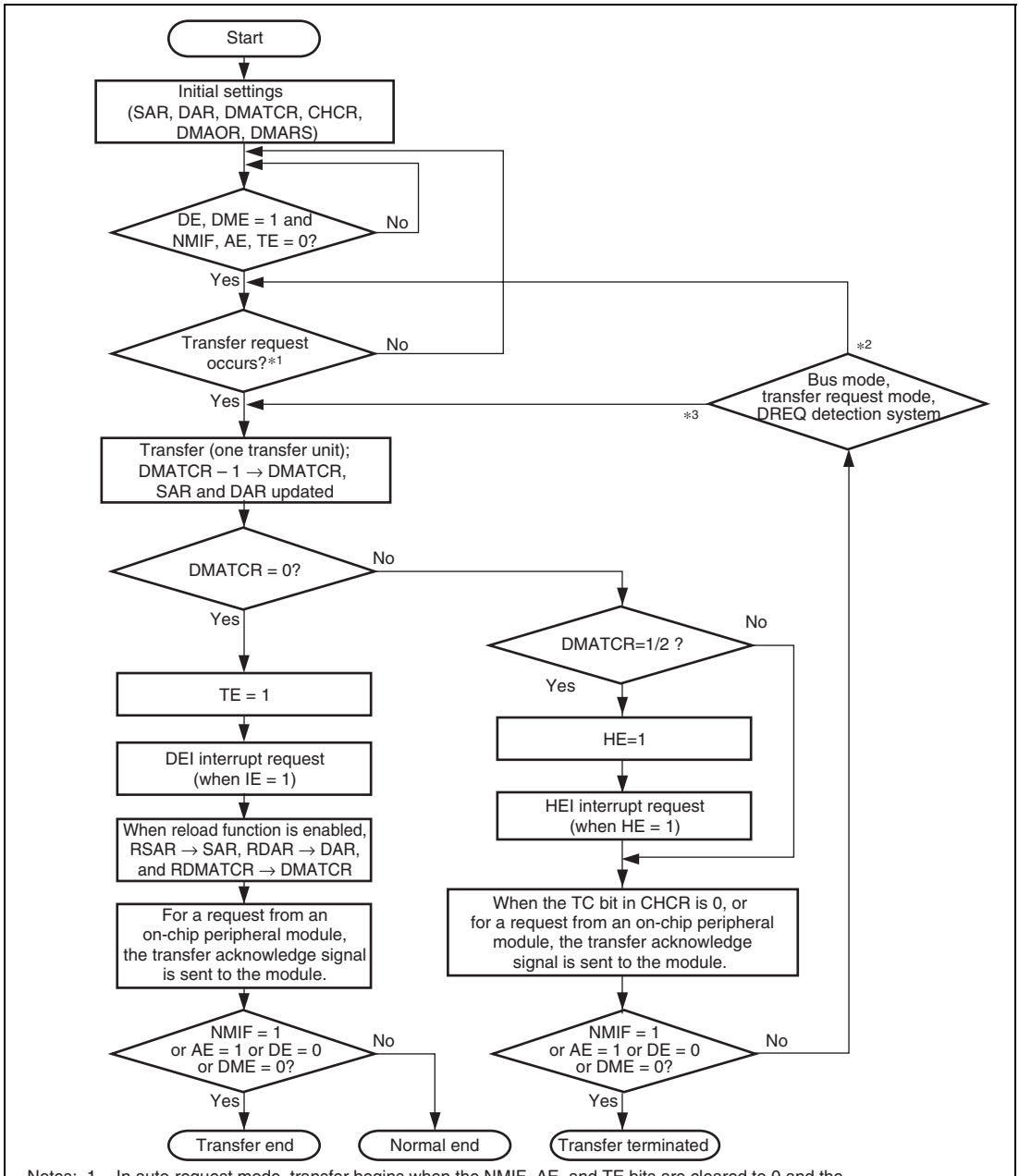
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

11.4.1 Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extension resource selector (DMARS) are set for the target transfer conditions, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request comes and transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS0 and TS1 settings). For an auto request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When half of the specified transfer count is exceeded (when DMATCR reaches half of the initial value), an HEI interrupt is sent to the CPU if the HIE bit in CHCR is set to 1.
4. When transfer has been completed for the specified count (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
5. When an address error in the DMAC or an NMI interrupt is generated, the transfer is terminated. Transfers are also terminated when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0.

Figure 11.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are cleared to 0 and the DE and DME bits are set to 1.
2. DREQ level detection in burst mode (external request) or cycle steal mode.
3. DREQ edge detection in burst mode (external request), or auto request mode in burst mode.

Figure 11.2 DMA Transfer Flowchart

11.4.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated in external devices and on-chip peripheral modules that are neither the transfer source nor destination.

Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The request mode is selected by the RS[3:0] bits in CHCR_0 to CHCR_3 and DMARS0 and DMARS1.

(1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR_0 to CHCR_3 and the DME bit in DMAOR are set to 1, the transfer begins so long as the TE bits in CHCR_0 to CHCR_3, and the AE and NMIF bits in DMAOR are 0.

(2) External Request Mode

In this mode a transfer is performed at the request signals (DREQ0 and DREQ1) of an external device. Choose one of the modes shown in table 11.5 according to the application system. When the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), DMA transfer is performed upon a request at the DREQ input.

Table 11.5 Selecting External Request Modes with the RS Bits

RS[3]	RS[2]	RS[1]	RS[0]	Address Mode	Transfer Source	Transfer Destination
0	0	0	0	Dual address mode	Any	Any
0	0	1	0	Single address mode	External memory, memory-mapped external device	External device with DACK
			1		External device with DACK	External memory, memory-mapped external device

Choose to detect DREQ by either the edge or level of the signal input with the DL and DS bits in CHCR_0 and CHCR_1 as shown in table 11.6. The source of the transfer request does not have to be the data transfer source or destination.

Table 11.6 Selecting External Request Detection with DL and DS Bits

CHCR		
DL bit	**DS bit**	**Detection of External Request**
0	0	Low level detection
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When DREQ is accepted, the DREQ pin enters the request accept disabled state (non-sensitive period). After issuing acknowledge DACK signal for the accepted DREQ, the DREQ pin again enters the request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to detect the next DREQ after outputting DACK.

Overrun 0: Transfer is terminated after the same number of transfer has been performed as requests.

Overrun 1: Transfer is terminated after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 11.7 Selecting External Request Detection with DO Bit

CHCR	
DO bit	**External Request**
0	Overrun 0
1	Overrun 1

(3) On-Chip Peripheral Module Request

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

DMA transfer request signals from on-chip peripheral modules to the DMAC include transmit data empty and receive data full requests from the SCI and SCIF, transmit buffer empty and receive buffer full requests from the RSPI, data frame receive requests from the RCAN-ET (RM0_0 only), receive data full requests from the LVDS (SH72315A only), A/D conversion end request from the A/D converter, transmit and receive requests from the IIC3, and compare match request from the CMT, CMT2, MTU2, and MTU2S.

When a transfer request signal is sent in on-chip peripheral module request mode while DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, and NMIF = 0), DMA transfer is performed.

When the transmit data empty from the SCI or SCIF is selected, specify the corresponding SCI or SCIF transmit data register as the transfer destination. Likewise, when the receive data full from the SCI or SCIF is selected, specify the corresponding SCI or SCIF receive data register as the transfer source. When the transmit buffer empty from the RSPI is selected, specify the RSPI transmit buffer (SPTX) as the transfer source. Likewise, when the receive buffer full is selected, specify the RSPI receive buffer (SPRX) as the transfer destination. When the data frame reception from the RCAN-ET (RM0_0 only) is selected, specify the CONTROL0H to CONTROL1L as the transfer source.

When the receive data full from the LVDS (SH72315A only) is selected, specify the receive data register as the transfer source. When a transfer request is made by the A/D converter, the transfer source must be the A/D data register (ADDR). When the IIC3 transmission is selected as the transfer request, the transfer destination must be ICDRT; when the IIC3 reception is selected as the transfer request, the transfer source must be ICDRR. Any address can be specified for data transfer source and destination when a transfer request is sent from the CMT, CMT2, MTU2, or MTU2S.

Table 11.8 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer	DMA Transfer	Transfer	Transfer	Bus
RS[3:0]	MID	RID	Request Source	Request Signal	Source	Destination	Mode
1000	100000	01	SCI_0 transmit	TXI0 (transmit data empty)	Any	SCTDR0	Cycle steal
		10	SCI_0 receive	RXI0 (receive data full)	SCRDR0	Any	
	100001	01	SCI_1 transmit	TXI1 (transmit data empty)	Any	SCTDR1	
		10	SCI_1 receive	RXI1 (receive data full)	SCRDR1	Any	
	100010	01	SCI_2 transmit	TXI2 (transmit data empty)	Any	SCTDR2	
		10	SCI_2 receive	RXI2 (receive data full)	SCRDR2	Any	
	100011	01	SCI_3 transmit	TXI3 (transmit data empty)	Any	SCTDR3	
		10	SCI_3 receive	RXI3 (receive data full)	SCRDR3	Any	
	100100	01	SCIF_4 transmit	TXI4 (transmit FIFO data empty)	Any	SCFTDR4	
		10	SCIF_4 receive	RXI4 (receive FIFO data full)	SCFRDR4	Any	
	100101	01	SCIF_5 transmit	TXI5 (transmit FIFO data empty)	Any	SCFTDR5	
		10	SCIF_5 receive	RXI5 (receive FIFO data full)	SCFRDR5	Any	
	100110	01	SCIF_6 transmit	TXI6 (transmit FIFO data empty)	Any	SCFTDR6	
		10	SCIF_6 receive	RXI6 (receive FIFO data full)	SCFRDR6	Any	
	100111	01	SCIF_7 transmit	TXI7 (transmit FIFO data empty)	Any	SCFTDR7	
		10	SCIF_7 receive	RXI7 (receive FIFO data full)	SCFRDR7	Any	
	101000	01	IIC3 transmit	IITXI (transmit data empty)	Any	ICDRT	
		10	IIC3 receive	IIRXI (receive data full)	ICDRR	Any	
	101010	01	LVDS (SH72315A only)	LVRXI (receive data full)	LVFRDR	Any	Cycle steal or burst
	101100	01	A/D converter_0	ADI0 (A/D conversion end)	ADDR0 to ADDR7, ADSDR	Any	Cycle steal
	101100	10	A/D converter_1	ADI1 (A/D conversion end)	ADDR8 to ADDR15, ADSDR	Any	

CHCR	DMARS		DMA Transfer	DMA Transfer	Transfer	Transfer	Bus
RS[3:0]	MID	RID	Request Source	Request Signal	Source	Destination	Mode
1000	110100	01	MTU2S_3	TGIA_3S	Any	Any	Cycle steal or burst
	110100	10	MTU2S_4	TGIA_4S	Any	Any	
	111000	01	MTU2_0	TGI0A	Any	Any	
	111000	11	MTU2_1	TGI1A	Any	Any	
	111000	11	MTU2_2	TGI2A	Any	Any	
	111011	01	MTU2_3	TGI3A	Any	Any	
	111011	10	MTU2_4	TGI4A	Any	Any	
	111100	00	CMT2	CM2I (compare match)	Any	Any	
	111101	00		IC0I (input capture)	Any	Any	
	111101	01		IC1I (input capture)	Any	Any	
	111101	10		OC0I (output compare)	Any	Any	
	111101	11		OC1I (output compare)	Any	Any	
	111110	01	CMT_0	CMI0 (compare match)	Any	Any	
	111110	10	CMT_1	CMI1 (compare match)	Any	Any	
	111111	00	RSPI_0	SPTXI0	Any	SPTX	
	111111	01		SPRXI0	SPRX	Any	
	111111	10	RCAN-ET	RM0_0	CONTROL0H to CONTROL0L*	Any	

Note: Read the message control fields 0 (CONTROL0H) to 1 (CONTROL1H) in mailbox 0, for example, by using transfer count mode.

11.4.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Three modes (fixed mode 1, fixed mode 2, and round-robin mode) are selected using the PR1 and PR0 bits in DMAOR.

(1) Fixed Mode

In fixed modes, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

Fixed mode 1: CH0 > CH1 > CH2 > CH3

Fixed mode 2: CH0 > CH2 > CH3 > CH1

These are selected by the PR1 and PR0 bits in the DMA operation register (DMAOR).

(2) Round-Robin Mode

Each time one unit of word, byte, longword, or 16 bytes is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished is rotated to the lowest of the priority order among the round-robin channels. The round-robin mode operation is shown in figure 11.3. The priority in round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after a reset.

When round-robin mode has been specified, do not concurrently specify cycle steal mode and burst mode as the bus modes of any two or more channels.

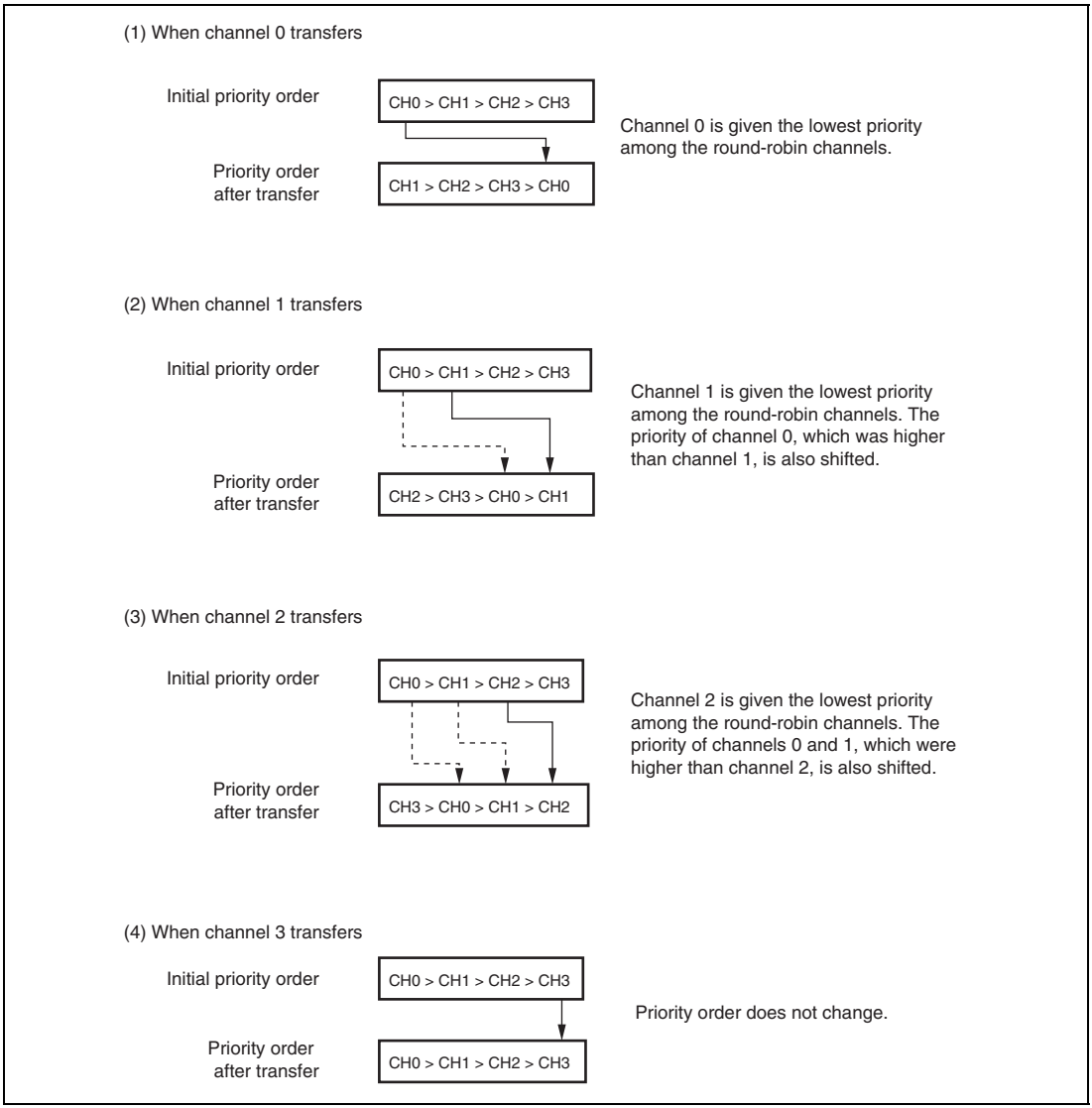


Figure 11.3 Round-Robin Mode

Figure 11.4 shows how the priority order changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 is given the lowest priority among the round-robin channels.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 is given the lowest priority among the round-robin channels.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 are lowered in priority so that channel 3 is given the lowest priority among the round-robin channels.

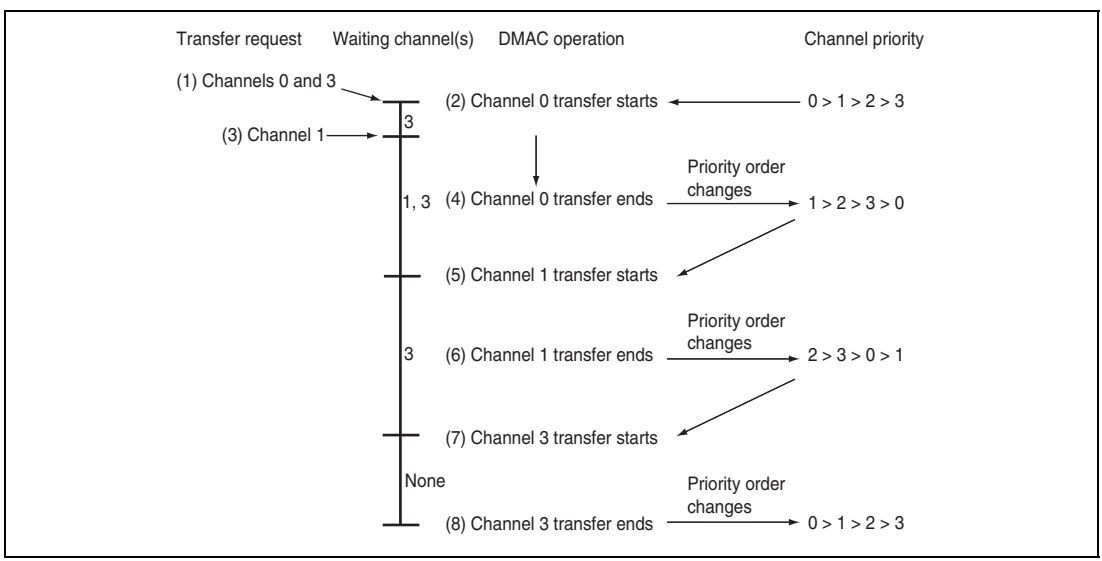


Figure 11.4 Changes in Channel Priority in Round-Robin Mode

11.4.4 DMA Transfer Types

DMA transfer has two types: single address mode transfer and dual address mode transfer. They depend on the number of bus cycles of access to the transfer source and destination. A data transfer timing depends on the bus mode, which is cycle steal mode or burst mode. The DMAC supports the transfers shown in table 11.9.

Table 11.9 Supported DMA Transfers

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**On-Chip Memory**
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
On-chip memory	Not available	Dual	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is available only for on-chip peripheral modules that support longword access.

(1) Address Modes

(a) Dual Address Mode

In dual address mode, both the transfer source and destination are accessed (selected) by an address. The transfer source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 11.5, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a data write cycle.

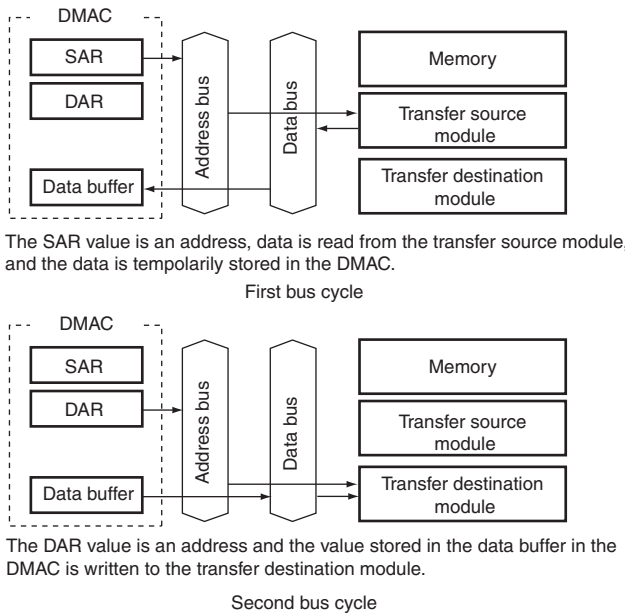


Figure 11.5 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. The AM bit in the channel control register (CHCR) can specify whether the DACK is output in read cycle or write cycle.

Figure 11.6 shows an example of DMA transfer timing in dual address mode.

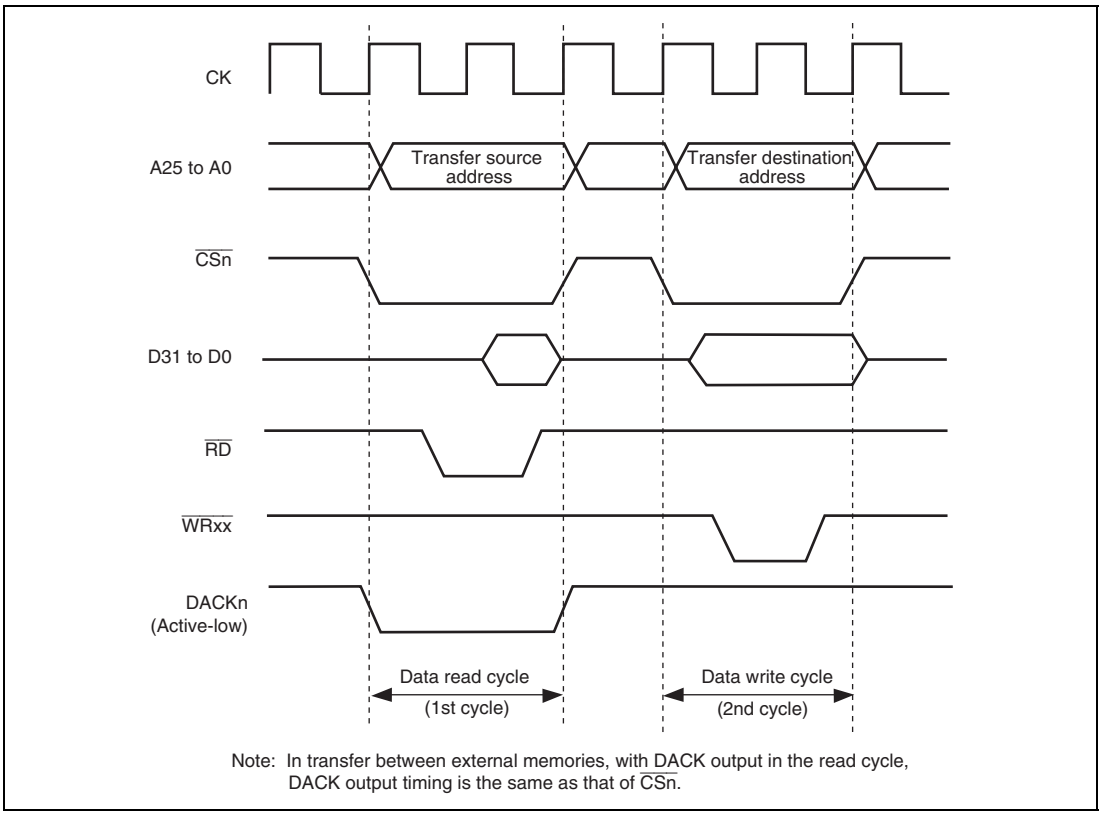


Figure 11.6 Example of DMA Transfer Timing in Dual Mode (Transfer Source: Normal Memory, Transfer Destination: Normal Memory)

(b) Single Address Mode

In single address mode, both the transfer source and destination are external devices, either of them is accessed (selected) by the DACK signal, and the other device is accessed by an address. In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK shown in figure 11.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

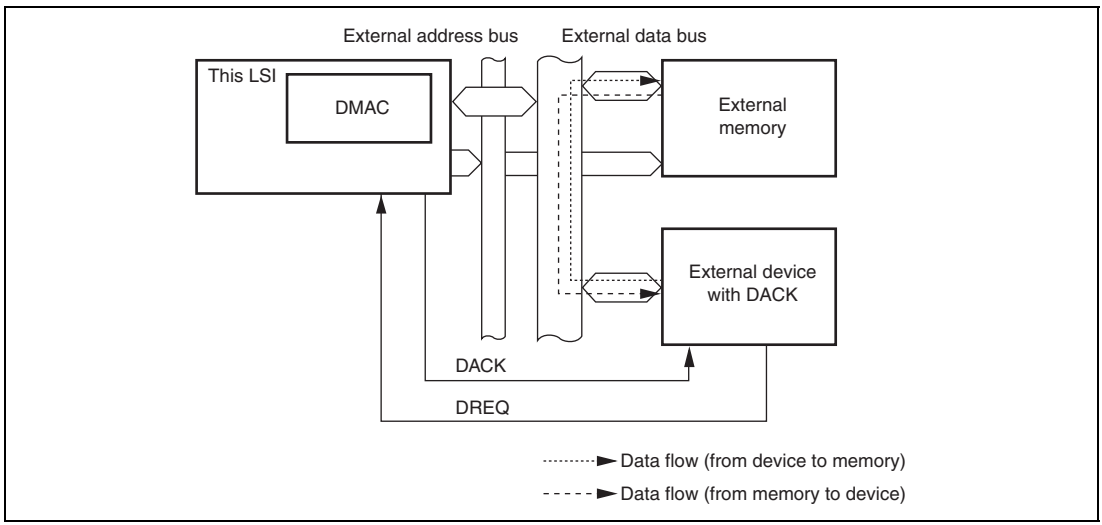


Figure 11.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an external device with DACK and a memory-mapped external device, and (2) transfer between an external device with DACK and external memory. In both cases, only the external request signal (DREQ) is used for transfer requests.

Figure 11.8 shows an example of DMA transfer timing in single address mode.

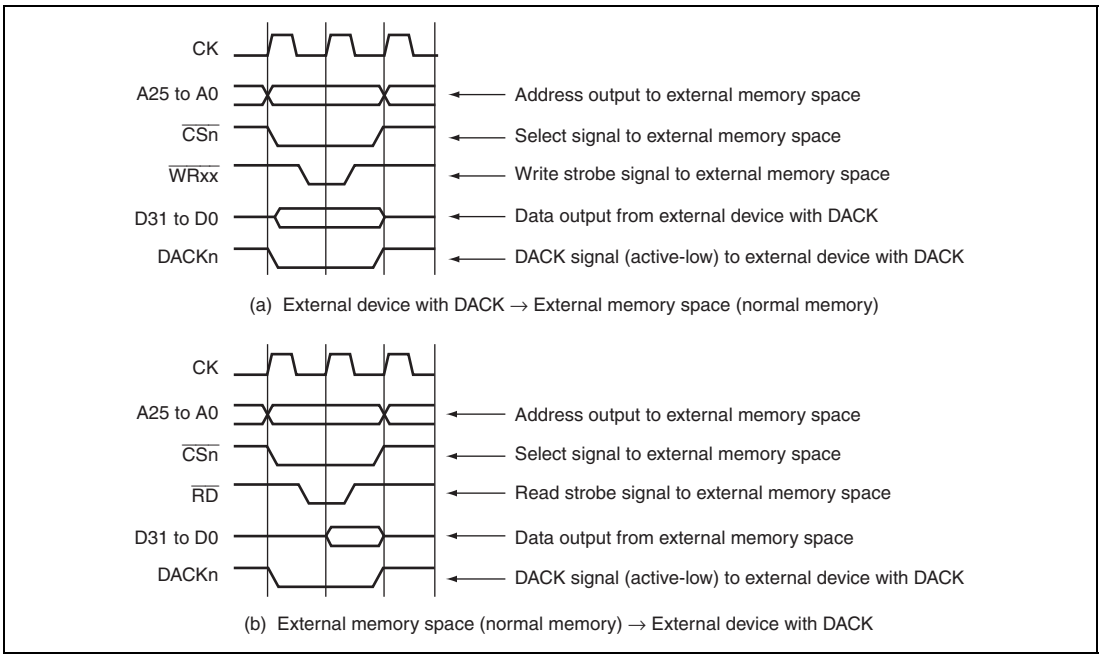


Figure 11.8 Example of DMA Transfer Timing in Single Address Mode

(2) Bus Modes

There are two bus modes; cycle steal and burst. Select the mode by the TB bits in the channel control registers (CHCR).

(a) Cycle Steal Mode

- Normal mode

  In normal mode of cycle steal, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another transfer request occurs, the bus mastership is obtained from another bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to another bus master. This is repeated until the transfer end conditions are satisfied.

  The cycle-steal normal mode can be used for any transfer section; transfer request source, transfer source, and transfer destination.

  Figure 11.9 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are:

  — Dual address mode

  — DREQ low level detection

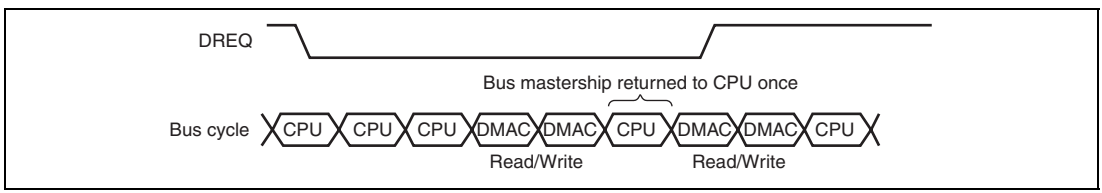


Figure 11.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- Intermittent Mode 16 and Intermittent Mode 64

  In intermittent mode of cycle steal, DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16 bytes) is completed. If the next transfer request occurs after that, DMAC obtains the bus mastership from other bus master after waiting for 16 or 64 cycles of Bϕ clock. DMAC then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than normal mode of cycle steal.

  The cycle-steal intermittent mode can be used for any transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle steal mode in all channels.

Figure 11.10 shows an example of DMA transfer timing in cycle-steal intermittent mode. Transfer conditions shown in the figure are:

— Dual address mode

— DREQ low level detection

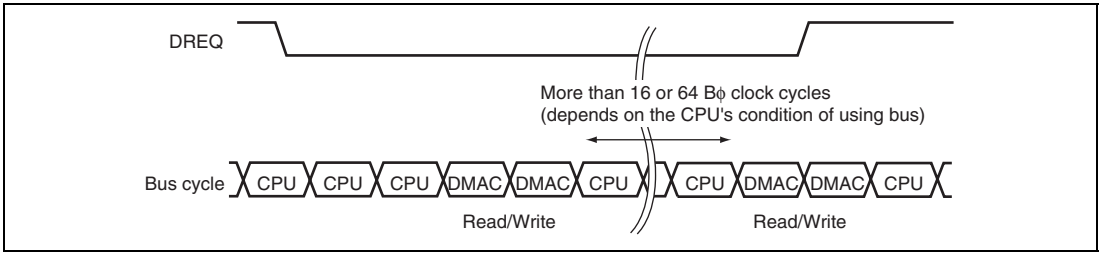


Figure 11.10 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(b) Burst Mode

In burst mode, once the DMAC obtains the bus mastership, it does not release the bus mastership and continues to perform transfer until the transfer end condition is satisfied. In external request mode with low level detection of the DREQ pin, however, when the DREQ pin is driven high, the bus mastership is passed to another bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

Figure 11.11 shows DMA transfer timing in burst mode.

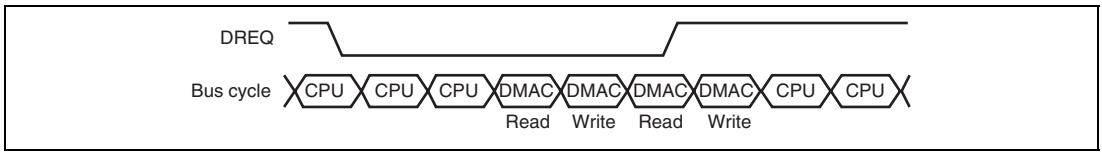


Figure 11.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 11.10 shows the relationship between request modes and bus modes by DMA transfer category.

Table 11.10 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32/128	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1
	External memory and external memory	All*4	B/C	8/16/32/128	0 to 3*3
	External memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 3*3
	Memory-mapped external device and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 3*3
	External memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 3*3
	Memory-mapped external device and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 3*3
	On-chip peripheral module and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 3*3
	On-chip memory and on-chip memory	All*4	B/C	8/16/32/128	0 to 3*3
	On-chip memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 3*3
	On-chip memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 3*3
	On-chip memory and external memory	All*4	B/C	8/16/32/128	0 to 3*3
Single	External device with DACK and external memory	External	B/C	8/16/32/128	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1

[Legend]

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. However, along with the exception of CMT, CMT2, MTU2, and MTU2S as the transfer request source, the requesting module must be designated as the transfer source or the transfer destination.

2. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.

3. If the transfer request is an external request, channels 0 and 1 are only available.

4. External requests, auto requests, and on-chip peripheral module requests are all available. In the case of on-chip peripheral module requests, however, the CMT CMT2, MTU2, and MTU2S are only available.

5. Only cycle steal except for the CMT, CMT2, MTU2, MTU2S, and LVDS (SH72315A only) as the transfer request source.

(4) Bus Mode and Channel Priority

In priority fixed mode (CH0 > CH1), when channel 1 is transferring data in burst mode and a request arrives for transfer on channel 0, which has higher-priority, the data transfer on channel 0 will begin immediately. In this case, if the transfer on channel 0 is also in burst mode, the transfer on channel 1 will only resume on completion of the transfer on channel 0.

When channel 0 is in cycle steal mode, one transfer-unit of data on this channel, which has the higher priority, is transferred. Data is then transferred continuously to channel 1 without releasing the bus. The bus mastership will then switch between the two in this order: channel 0, channel 1, channel 0, channel 1, etc. That is, the CPU cycle after the data transfer in cycle steal mode is replaced with a burst-mode transfer cycle (priority execution of burst-mode cycle). An example of this is shown in figure 11.12.

When multiple channels are in burst mode, data transfer on the channel that has the highest priority is given precedence. When DMA transfer is being performed on multiple channels, the bus mastership is not released to another bus-master device until all of the competing burst-mode transfers have been completed.

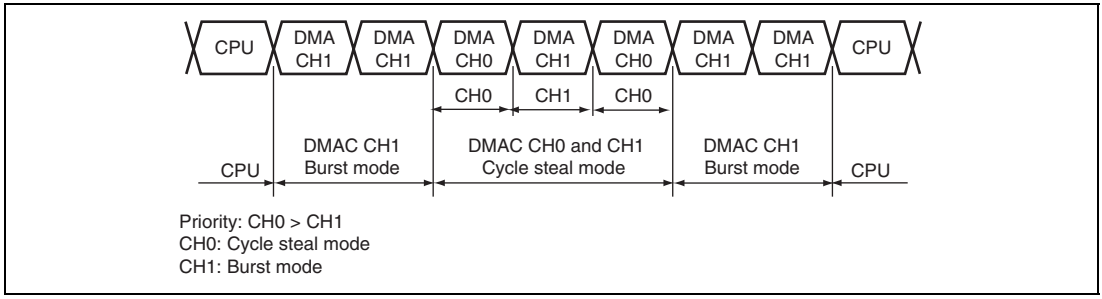


Figure 11.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 11.3. Note that channels in cycle steal and burst modes must not be mixed.

11.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing

(1) Number of Bus Cycles

When the DMAC is the bus master, the number of bus cycles is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 10, Bus State Controller (BSC).

(2) DREQ Pin Sampling Timing

Figures 11.13 to 11.16 show the DREQ input sampling timings in each bus mode.

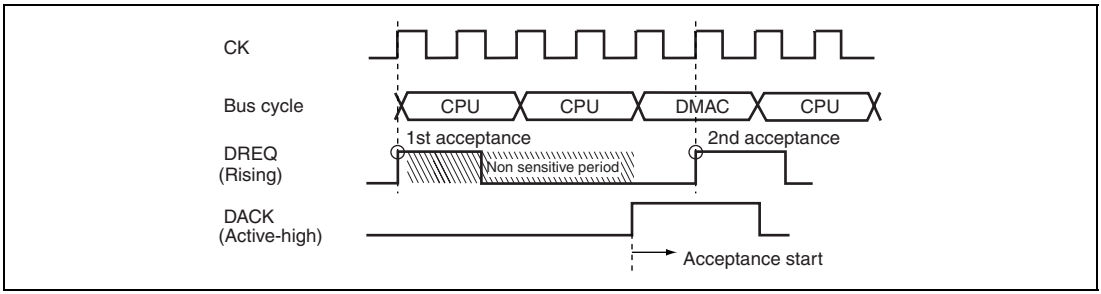


Figure 11.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

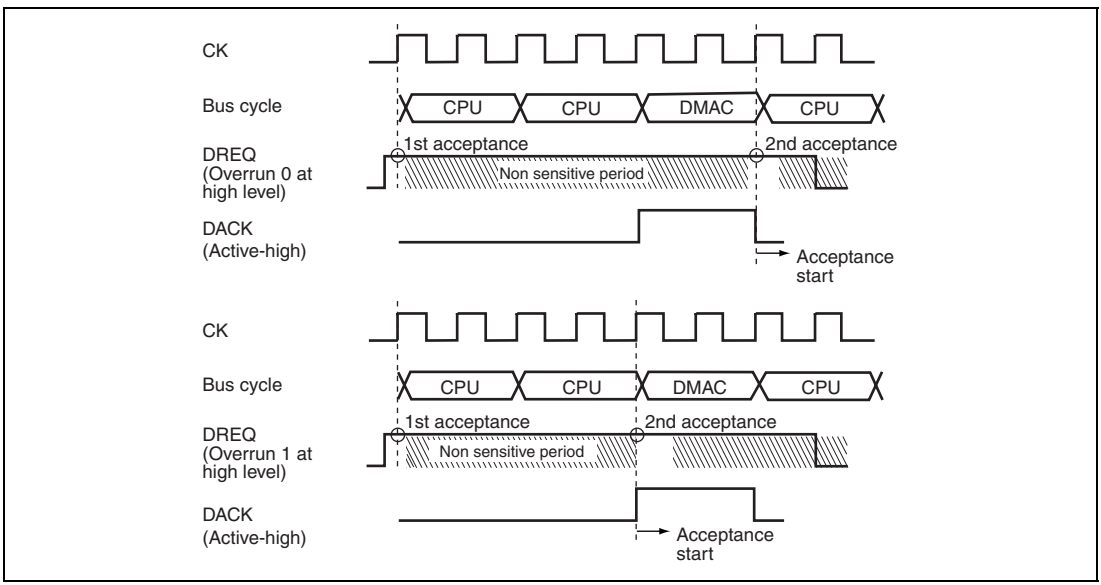


Figure 11.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

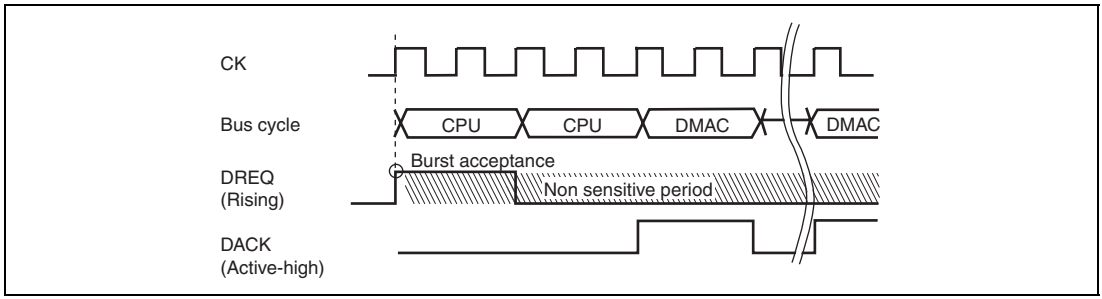


Figure 11.15 Example of DREQ Input Detection in Burst Mode Edge Detection

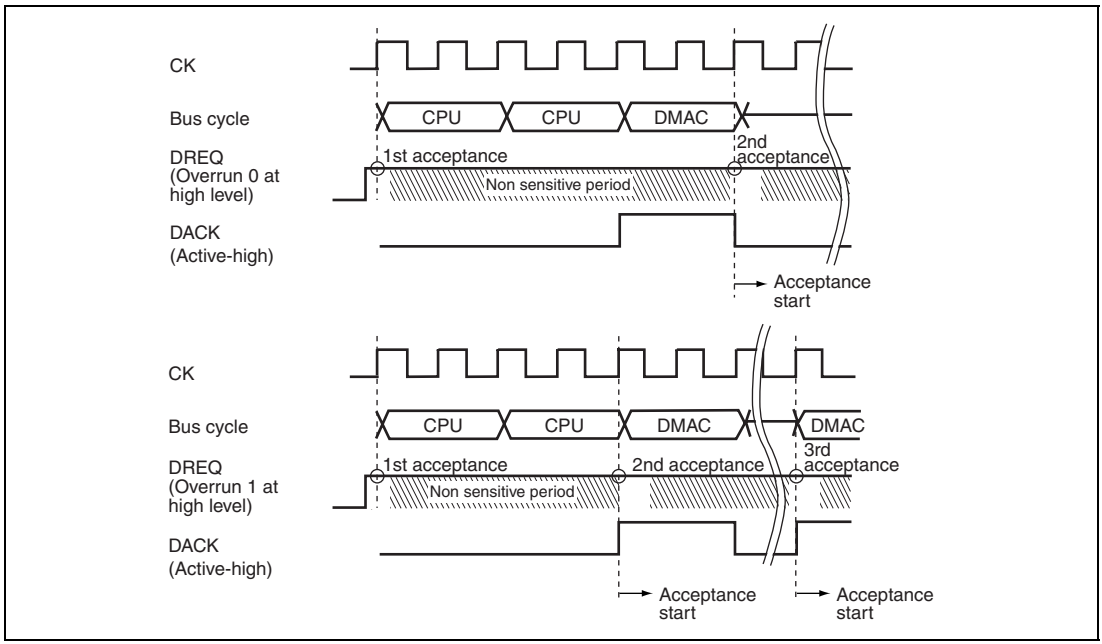


Figure 11.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 11.17 shows the TEND output timing.

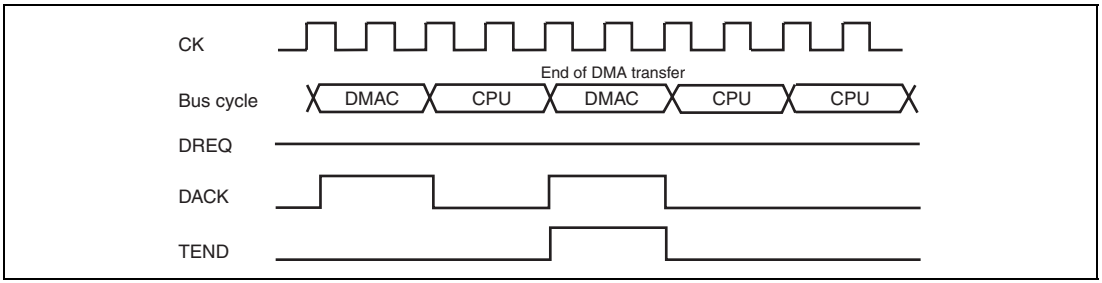
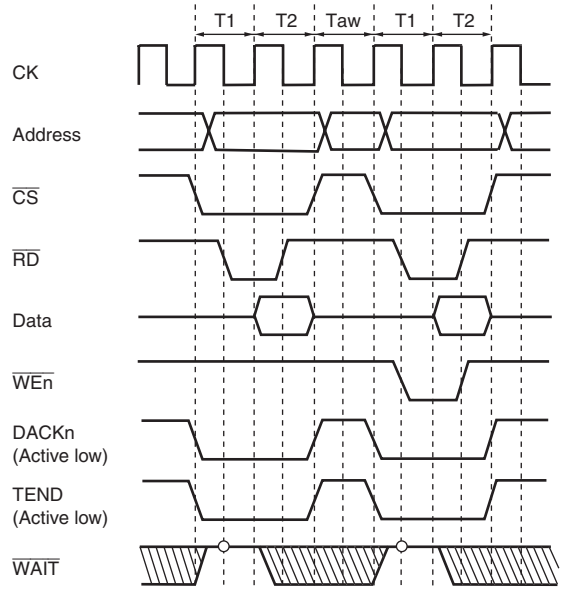


Figure 11.17 Example of DMA Transfer End Signal Timing (Cycle Steal Mode Level Detection)

The unit of the DMA transfer is divided into multiple bus cycles when 16-byte transfer is performed for an 8-bit or 16-bit external device, when longword access is performed for an 8-bit or 16-bit external device, or when word access is performed for an 8-bit external device. When a setting is made so that the DMA transfer size is divided into multiple bus cycles and the $\overline{CS}$ signal is negated between bus cycles, note that DACK and TEND are divided like the $\overline{CS}$ signal for data alignment. Also, if the DREQ detection is set to level-detection mode (DS bit in CHCR = 0), the DREQ sampling may not be detected correctly with divided DACK, and one extra overrun may occur at maximum.

Use a setting that does not divide DACK or specify a transfer size smaller than the external device bus width if DACK is divided. Figure 11.18 shows this example.



Note: TEND is asserted for the last unit of DMA transfer. If a transfer unit is divided into multiple bus cycles and the CS is negated between the bus cycles, TEND is also divided.

Figure 11.18 BSC Normal Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

11.5 Interrupt Sources

11.5.1 Interrupt Sources and Priority Order

The interrupt sources of the DMAC are the data transfer end interrupt (TEI) and data transfer half-end interrupt (HEI) for each channel. Table 11.11 lists the interrupt sources and their order of priority.

The IE and HIE bits in the DMA channel control registers (CHCRs) enable or disable the respective interrupt sources. Furthermore, the interrupt requests are independently conveyed to the interrupt controller.

A data-transfer end interrupt (TEI) is generated when, the transfer end flag and the transfer end interrupt enable (IE) bit in the DMA channel control register (CHCR) are set to 1. A data-transfer half end interrupt (HEI) is generated when the half-end flag and the half-end interrupt enable (HIE) bit in the DMA channel control register (CHCR) are set to 1. Clearing the interrupt flag bit to 0 cancels the interrupt request.

Priority among the channels is adjustable by the interrupt controller. The order of priority for interrupts of a given channel is fixed. For details, refer to section 7, Interrupt Controller (INTC).

Table 11.11 Interrupt Sources

Channel	Interrupt Source	Interrupt Enable Bit	Interrupt Flag	Priority
0	Data transfer end interrupt (TEI_0)	IE	TE	High
	Data transfer half end interrupt (HEI_0)	HIE	HE	
1	Data transfer end interrupt (TEI_1)	IE	TE	
	Data transfer half end interrupt (HEI_1)	HIE	HE	
2	Data transfer end interrupt (TEI_2)	IE	TE	
	Data transfer half end interrupt (HEI_2)	HIE	HE	
3	Data transfer end interrupt (TEI_3)	IE	TE	
	Data transfer half end interrupt (HEI_3)	HIE	HE	Low

11.6 Usage Notes

11.6.1 Setting of the Half-End Flag and the Half-End Interrupt

Since the following points for caution apply in cases where reference to the state of the half-end flag in the CHCR register or the half-end interrupt is used in conjunction with the reload function, please take care on these points.

Ensure that the reloaded number of transfers (the value set in RDMATCR) is always the same as the number of transfers that was initially set (the value set in DMATCR).

If the initial setting in DMATCR and the value for the second and later transfers in RDMATCR are different, the timing with which the half-end flag is set may be faster than half the number of transfers, or the half-end flag might not be set at all. The same considerations apply to the half-end interrupt.

11.6.2 Timing of DACK and TEND Outputs

When the external memory is MPX-I/O, assertion of the DACK output has the same timing as the data cycle. For details, see the respective figures under section 10.5.5, MPX-I/O Interface, in section 10, Bus State Controller (BSC). When the memory is other than the MPX-I/O, the DACK output is asserted with the same timing as the corresponding CS signal.

The TEND output does not depend on the type of memory and is always asserted with the same timing as the corresponding CS signal.

11.6.3 Setting Module Standby Mode

The DMAC operation can be enabled or disabled by the standby control register. The DMAC is disabled in the initial state. Register accesses are enabled by canceling the module standby mode.

Software standby or module standby mode must not be set during DMAC operation. Before making a transition to software standby or module standby mode, clear the DE bits for all the channels. For details, see section 32, Power-Down Modes.

11.6.4 Accessing DMAC/DTC Registers by DMAC

The DMAC must not access DMAC/DTC registers. The DTC must not access the DMAC registers.

11.6.5 Using SCI or SCIF as DMAC Activation Source

When the DMAC is activated by the SCI TXI interrupt, the SCI TEND flag must not be used as the transfer end flag.

When the DMAC is activated by the SCIF TXI interrupt, the SCIF TEND flag must not be used as the transfer end flag.

11.6.6 Setting CHCR

Before setting CHCR, clear the DE bit of the corresponding channel.

11.6.7 Activating Multiple Channels

One internal request should not be used for multiple channels.

11.6.8 Inputting Transfer Request

Input a transfer request after the DMAC setting is completed.

11.6.9 NMI Interrupt and DMAC Activation

If arbitration occurs between the NMI interrupt and DMAC activation, the NMI interrupt takes priority. At this time, the NMIF bit is set to 1 and the DMAC is not activated.

11.6.10 Using External Request Mode

When at least one channel is activated by an external request, one of the following four conditions should be satisfied:

1. All the channels are used in cycle steal mode.
2. All the channels are used in burst mode and the following three conditions are all satisfied:
   — Channel priority order is set to fixed mode 1 or 2.
   — Dual address mode is set for all the channels.
   — The combination of transfer source and destination addresses is one of the following:
      A. Transfer source address: External address space
         Transfer destination address: External address space
      B. Transfer source address: External address space
         Transfer destination address: Internal address space
      C. Transfer source address: Internal address space
         Transfer destination address: Internal address space
3. All the channels are used in both the cycle steal mode and burst mode, and the following three conditions are all satisfied:
   — Channel priority order is set to fixed mode 1 or 2.
   — Dual address mode is set for all the channels.
   — The transfer source and destination addresses are one of the following combinations:
      A. Transfer source address: External address space
         Transfer destination address: External address space
      B. Transfer source address: External address space
         Transfer destination address: Internal address space
      C. Transfer source address: Internal address space
         Transfer destination address: Internal address space
4. Only one channel is used.

In a case other than the above, the DACKn or TENDn pin may indicate an incorrect transfer channel and the DMA transfer may not be performed until a power-on reset is provided. In addition, if this error status is generated in burst mode, the CPU cannot fetch instructions, resulting in the system stop.

11.6.11 Using On-Chip Peripheral Module Request Mode or Auto Request Mode

When at least one channel is activated by an on-chip peripheral module request or an auto request and the DACKn or TENDn pin is used, one of the following conditions should be satisfied:

1. All the channels are used in cycle steal mode.
2. All the channels are used in burst mode and the following three conditions are all satisfied:
   - Channel priority order is set to fixed mode 1 or 2.
   - Dual address mode is set for all the channels.
   - The combination of transfer source and destination addresses is one of the following:
     A. Transfer source address: External address space
        Transfer destination address: External address space
     B. Transfer source address: External address space
        Transfer destination address: Internal address space
     C. Transfer source address: Internal address space
        Transfer destination address: Internal address space
3. All the channels are used in both the cycle steal mode and burst mode, and the following three conditions are all satisfied:
   - Channel priority order is set to fixed mode 1 or 2.
   - Dual address mode is set for all the channels.
   - The transfer source and destination addresses are one of the following combinations:
     A. Transfer source address: External address space
        Transfer destination address: External address space
     B. Transfer source address: External address space
        Transfer destination address: Internal address space
     C. Transfer source address: Internal address space
        Transfer destination address: Internal address space
4. Only one channel is used.

In a case other than the above, the DACKn or TENDn pin may indicate an incorrect transfer channel.

Section 12 Multi-Function Timer Pulse Unit 2 (MTU2)

This LSI has an on-chip multi-function timer pulse unit 2 (MTU2) that comprises six 16-bit timer channels.

12.1 Features

- Maximum 16 pulse input/output lines and three pulse input lines
- Selection of eight counter input clocks for each channel (four clocks for channel 5)
- The following operations can be set for channels 0 to 4:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture is possible
  - Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 12-phase PWM output is possible in combination with synchronous operation.
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- Dead time compensation counter available in channel 5
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

Table 12.1 MTU2 Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		Pφ/1 Pφ/4 Pφ/16 Pφ/64 TCLKA TCLKB TCLKC TCLKD	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/1024 TCLKA TCLKB TCLKC	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 Pφ/1024 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 Pφ/1024 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64
General registers (TGR)		TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRU_5 TGRV_5 TGRW_5
General registers/ buffer registers		TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4	—
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	Input pins TIC5U TIC5V TIC5W
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	√	√	√	√	√	—
	1 output	√	√	√	√	√	—
	Toggle output	√	√	√	√	√	—
Input capture function		√	√	√	√	√	√
Synchronous operation		√	√	√	√	√	—
PWM mode 1		√	√	√	√	√	—
PWM mode 2		√	√	√	—	—	—
Complementary PWM mode		—	—	—	√	√	—
Reset PWM mode		—	—	—	√	√	—
AC synchronous motor drive mode		√	—	—	√	√	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Phase counting mode	—	√	√	—	—	—
Buffer operation	√	v	—	√	√	—
Dead time compensation counter function	—	—	—	—	—	√
DMAC activation	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture and TCNT overflow or underflow	—
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture or TCNT overflow or underflow	TGR compare match or input capture
A/D converter start trigger	TGRA_0 compare match or input capture TGRE_0 compare match	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture TCNT_4 underflow (trough) in complement ary PWM mode	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Interrupt sources	7 sources	4 sources	4 sources	5 sources	5 sources	3 sources
	• Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	• Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	• Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	• Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	• Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow	• Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA_4 and TCNT_4 • A/D converter start request at a match between TADCORB_4 and TCNT_4	—
Interrupt skipping function	v	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts	—

[Legend]

√: Possible

—: Not possible

Figure 12.1 shows a block diagram of the MTU2.

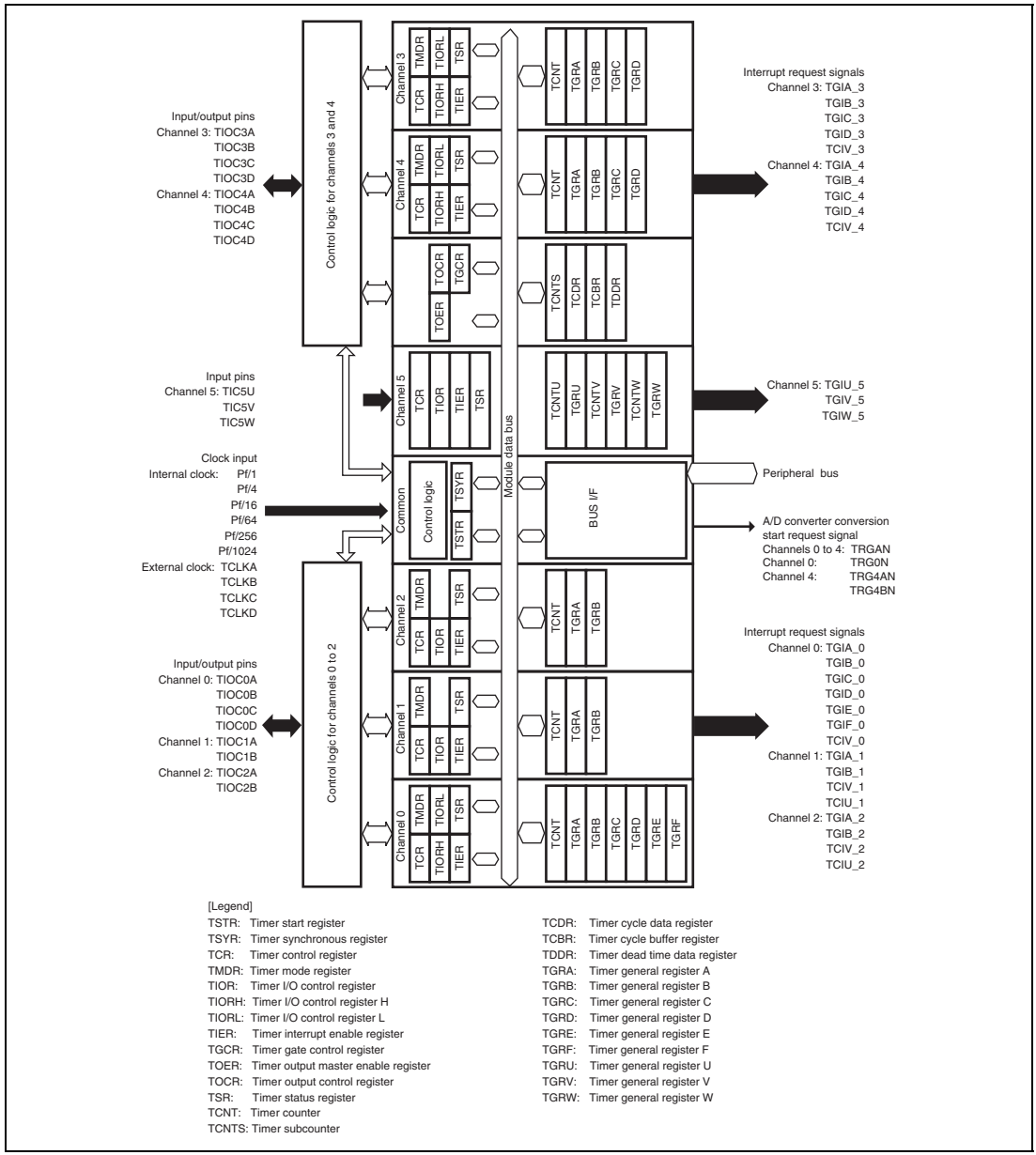


Figure 12.1 Block Diagram of MTU2

12.2 Input/Output Pins

Table 12.2 Pin Configuration

Channel	Pin Name	I/O	Function
Common	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOC0A	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOC0B	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOC0C	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOC0D	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOC1A	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOC1B	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOC2A	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOC2B	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOC3A	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOC3B	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOC3C	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOC3D	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOC4A	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOC4B	I/O	TGRB_4 input capture input/output compare output/PWM output pin
	TIOC4C	I/O	TGRC_4 input capture input/output compare output/PWM output pin
	TIOC4D	I/O	TGRD_4 input capture input/output compare output/PWM output pin
5	TIC5U	Input	TGRU_5 input capture input/external pulse input pin
	TIC5V	Input	TGRV_5 input capture input/external pulse input pin
	TIC5W	Input	TGRW_5 input capture input/external pulse input pin

12.3 Register Descriptions

The MTU2 has the following registers. For details on register addresses and register states during each process, refer to section 34, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

Table 12.3 Register Descriptions

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200	8, 16, 32
Timer control register_4	TCR_4	R/W	H'00	H'FFFE4201	8
Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202	8, 16
Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE4203	8
Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204	8, 16, 32
Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE4206	8, 16
Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207	8
Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208	8, 16
Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE4209	8
Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A	8
Timer gate control register	TGCR	R/W	H'80	H'FFFE420D	8
Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E	8, 16
Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F	8
Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210	16, 32
Timer counter_4	TCNT_4	R/W	H'0000	H'FFFE4212	16
Timer cycle data register	TCDR	R/W	H'FFFF	H'FFFE4214	16, 32
Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216	16
Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16, 32
Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FFFE421C	16, 32

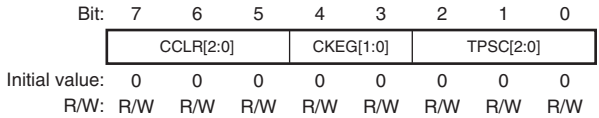
Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FFFE421E	16
Timer subcounter	TCNTS	R	H'0000	H'FFFE4220	16, 32
Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE4222	16
Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16, 32
Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FFFE4228	16, 32
Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	16
Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C	8, 16
Timer status register_4	TSR_4	R/W	H'C0	H'FFFE422D	8
Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE4230	8, 16
Timer interrupt skipping counter	TITCNT	R	H'00	H'FFFE4231	8
Timer buffer transfer set register	TBTER	R/W	H'00	H'FFFE4232	8
Timer dead time enable register	TDER	R/W	H'01	H'FFFE4234	8
Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236	8
Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238	8, 16
Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FFFE4239	8
Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FFFE4240	16
Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FFFE4244	16, 32
Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FFFE4246	16
Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE4248	16, 32
Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16
Timer waveform control register	TWCR	R/W	H'00	H'FFFE4260	8
Timer start register	TSTR	R/W	H'00	H'FFFE4280	8, 16
Timer synchronous register	TSYR	R/W	H'00	H'FFFE4281	8
Timer counter synchronous start register	TCSYSTR	R/W	H'00	H'FFFE4282	8

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284	8
Timer control register_0	TCR_0	R/W	H'00	H'FFFE4300	8, 16, 32
Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE4301	8
Timer I/O control registerH_0	TIORH_0	R/W	H'00	H'FFFE4302	8, 16
Timer I/O control registerL_0	TIORL_0	R/W	H'00	H'FFFE4303	8
Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE4304	8, 16, 32
Timer status register_0	TSR_0	R/W	H'C0	H'FFFE4305	8
Timer counter_0	TCNT_0	R/W	H'0000	H'FFFE4306	16
Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FFFE4308	16, 32
Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FFFE430A	16
Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FFFE430C	16, 32
Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FFFE430E	16
Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FFFE4320	16, 32
Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FFFE4322	16
Timer interrupt enable register2_0	TIER2_0	R/W	H'00	H'FFFE4324	8, 16
Timer status register2_0	TSR2_0	R/W	H'C0	H'FFFE4325	8
Timer buffer operation transfer mode register_0	TBTM_0	R/W	H'00	H'FFFE4326	8
Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380	8, 16
Timer mode register_1	TMDR_1	R/W	H'00	H'FFFE4381	8
Timer I/O control register_1	TIOR_1	R/W	H'00	H'FFFE4382	8
Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FFFE4384	8, 16, 32
Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385	8
Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE4386	16
Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16, 32

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FFFE438A	16
Timer input capture control register	TICCR	R/W	H'00	H'FFFE4390	8
Timer control register_2	TCR_2	R/W	H'00	H'FFFE4000	8, 16
Timer mode register_2	TMDR_2	R/W	H'00	H'FFFE4001	8
Timer I/O control register_2	TIOR_2	R/W	H'00	H'FFFE4002	8
Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FFFE4004	8, 16, 32
Timer status register_2	TSR_2	R/W	H'C0	H'FFFE4005	8
Timer counter_2	TCNT_2	R/W	H'0000	H'FFFE4006	16
Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FFFE4008	16, 32
Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16
Timer counter U_5	TCNTU_5	R/W	H'0000	H'FFFE4080	16, 32
Timer general register U_5	TGRU_5	R/W	H'FFFF	H'FFFE4082	16
Timer control register U_5	TCRU_5	R/W	H'00	H'FFFE4084	8
Timer I/O control register U_5	TIORU_5	R/W	H'00	H'FFFE4086	8
Timer counter V_5	TCNTV_5	R/W	H'0000	H'FFFE4090	16, 32
Timer general register V_5	TGRV_5	R/W	H'FFFF	H'FFFE4092	16
Timer control register V_5	TCRV_5	R/W	H'00	H'FFFE4094	8
Timer I/O control register V_5	TIORV_5	R/W	H'00	H'FFFE4096	8
Timer counter W_5	TCNTW_5	R/W	H'0000	H'FFFE40A0	16, 32
Timer general register W_5	TGRW_5	R/W	H'FFFF	H'FFFE40A2	16
Timer control register W_5	TCRW_5	R/W	H'00	H'FFFE40A4	8
Timer I/O control register W_5	TIORW_5	R/W	H'00	H'FFFE40A6	8
Timer status register_5	TSR_5	R/W	H'00	H'FFFE40B0	8
Timer interrupt enable register_5	TIER_5	R/W	H'00	H'FFFE40B2	8
Timer start register_5	TSTR_5	R/W	H'00	H'FFFE40B4	8
Timer compare match clear register	TCNTCMPCLR	R/W	H'00	H'FFFE40B6	8

12.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The MTU2 has a total of eight TCR registers, one each for channels 0 to 4 and three (TCRU_5, TCRV_5, and TCRW_5) for channel 5. TCR register settings should be conducted only when TCNT operation is stopped.



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See tables 12.4 and 12.5 for details.
4, 3	CKEG[1:0]	00	R/W	Clock Edge 0 and 1 These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. MPϕ/4 both edges = MPϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is MPϕ/4 or slower. When MPϕ/1 or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation compiles with the initial value. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges
2 to 0	TPSC[2:0]	000	R/W	Time Prescaler 0 to 2 These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 12.6 to 12.10 for details.

[Legend]

x: Don't care

Table 12.4 CCLR0 to CCLR2 (Channels 0, 3, and 4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3, 4	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Table 12.5 CCLR0 to CCLR2 (Channels 1 and 2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

Table 12.6 TPSC0 to TPSC2 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 12.7 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 12.8 TPSC0 to TPSC2 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 12.9 TPSC0 to TPSC2 (Channels 3 and 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3, 4	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	Internal clock: counts on Pϕ/256
			1	Internal clock: counts on Pϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

Table 12.10 TPSC1 and TPSC0 (Channel 5)

Channel	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	Internal clock: counts on Pϕ/1
		1	Internal clock: counts on Pϕ/4
	1	0	Internal clock: counts on Pϕ/16
		1	Internal clock: counts on Pϕ/64

Note: Bits 7 to 2 are reserved in channel 5. These bits are always read as 0. The write value should always be 0.

12.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. The MTU2 has five TMDR registers, one each for channels 0 to 4. TMDR register settings should be changed only when TCNT operation is stopped.

Bit:	7	6	5	4	3	2	1	0
	-	BFE	BFB	BFA	MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	BFE	0	R/W	Buffer Operation E Specifies whether TGRE_0 and TGRF_0 are to operate in the normal way or to be used together for buffer operation. TGRF compare match is generated when TGRF is used as the buffer register. In channels 1 to 4, this bit is reserved. It is always read as 0 and the write value should always be 0. 0: TGRE_0 and TGRF_0 operate normally 1: TGRE_0 and TGRF_0 used together for buffer operation

Bit	Bit Name	Initial Value	R/W	Description
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated in a mode other than complementary PWM. TGRD compare match is generated in complementary PWM mode. When compare match occurs during the Tb period in complementary PWM mode, TGFD is set. Therefore, set the TGIED bit in the timer interrupt enable register 3/4 (TIER_3/4) to 0. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated in a mode other than complementary PWM. TGRC compare match is generated when in complementary PWM mode. When compare match for channel 4 occurs during the Tb period in complementary PWM mode, TGFC is set. Therefore, set the TGIEC bit in the timer interrupt enable register 4 (TIER_4) to 0. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation
3 to 0	MD[3:0]	0000	R/W	Modes 0 to 3 These bits are used to set the timer operating mode. See table 12.11 for details.

Table 12.11 Setting of Operation Mode by Bits MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Setting prohibited
		1	0	PWM mode 1
			1	PWM mode 2*[1]
	1	0	0	Phase counting mode 1*[2]
			1	Phase counting mode 2*[2]
		1	0	Phase counting mode 3*[2]
			1	Phase counting mode 4*[2]
1	0	0	0	Reset synchronous PWM mode*[3]
			1	Setting prohibited
		1	X	Setting prohibited
	1	0	0	Setting prohibited
			1	Complementary PWM mode 1 (transmit at crest)*[3]
		1	0	Complementary PWM mode 2 (transmit at trough)*[3]
			1	Complementary PWM mode 2 (transmit at crest and trough)*[3]

[Legend]

X: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

3. Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, and 2.

12.3.3 Timer I/O Control Register (TIOR)

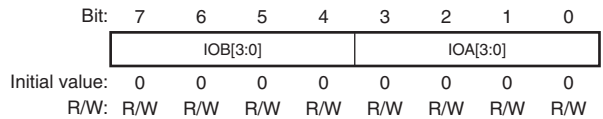
The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU2 has a total of eleven TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2, and three (TIORU_5, TIORV_5, and TIORW_5) for channel 5.

TIOR should be set while TMDR is set in normal operation, PWM mode, or phase counting mode.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4



Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 12.12 TIOR_1: Table 12.14 TIOR_2: Table 12.15 TIORH_3: Table 12.16 TIORH_4: Table 12.18
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify the function of TGRA. See the following tables. TIORH_0: Table 12.20 TIOR_1: Table 12.22 TIOR_2: Table 12.23 TIORH_3: Table 12.24 TIORH_4: Table 12.26

- TIORL_0, TIORL_3, TIORL_4

Bit:	7	6	5	4	3	2	1	0
	IOD[3:0]				IOC[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOD[3:0]	0000	R/W	I/O Control D0 to D3 Specify the function of TGRD. See the following tables. TIORL_0: Table 12.13 TIORL_3: Table 12.17 TIORL_4: Table 12.19
3 to 0	IOC[3:0]	0000	R/W	I/O Control C0 to C3 Specify the function of TGRC. See the following tables. TIORL_0: Table 12.21 TIORL_3: Table 12.25 TIORL_4: Table 12.27

- TIORU_5, TIORV_5, TIORW_5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	IOC[4:0]				
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	IOC[4:0]	00000	R/W	I/O Control C0 to C4 Specify the function of TGRU_5, TGRV_5, and TGRW_5. For details, see table 12.28.

Table 12.12 TIORH_0 (Channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOC0B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.13 TIORL_0 (Channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOC0D Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.14 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOC1B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of TGRC_0 compare match/input capture

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.15 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOC2B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.16 TIORH_3 (Channel 3)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOC3B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.17 TIORL_3 (Channel 3)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOC3D Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.18 TIORH_4 (Channel 4)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOC4B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.19 TIORL_4 (Channel 4)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_4 Function**	**TIOC4D Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_4 is set to 1 and TGRD_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.20 TIORH_0 (Channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOC0A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.21 TIORL_0 (Channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOC0C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.22 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOC1A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.23 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOC2A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.24 TIORH_3 (Channel 3)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOC3A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.25 TIORL_3 (Channel 3)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOC3C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.26 TIORH_4 (Channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOC4A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 12.27 TIORL_4 (Channel 4)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_4 Function**	**TIOC4C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.28 TIORU_5, TIORV_5, and TIORW_5 (Channel 5)

					Description	
Bit 4 IOC4	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRU_5, TGRV_5, and TGRW_5 Function	TIC5U, TIC5V, and TIC5W Pin Function
0	0	0	0	0	Compare match register	Compare match
				1		Setting prohibited
			1	X		Setting prohibited
		1	X	X		Setting prohibited
	1	X	X	X		Setting prohibited
1	0	0	0	0	Input capture register	Setting prohibited
				1		Input capture at rising edge
			1	0		Input capture at falling edge
				1		Input capture at both edges
		1	X	X		Setting prohibited
	1	0	0	0		Setting prohibited
				1		Measurement of low pulse width of external input signal
						Capture at trough in complementary PWM mode
			1	0		Measurement of low pulse width of external input signal
						Capture at crest in complementary PWM mode
				1		Measurement of low pulse width of external input signal
						Capture at crest and trough in complementary PWM mode
		1	0	0		Setting prohibited
				1		Measurement of high pulse width of external input signal
						Capture at trough in complementary PWM mode
			1	0		Measurement of high pulse width of external input signal
						Capture at crest in complementary PWM mode
				1		Measurement of high pulse width of external input signal
						Capture at crest and trough in complementary PWM mode

[Legend]

X: Don't care

12.3.4 Timer Compare Match Clear Register (TCNTCMPCLR)

TCNTCMPCLR is an 8-bit readable/writable register that specifies requests to clear TCNTU_5, TCNTV_5, and TCNTW_5. The MTU2 has one TCNTCMPCLR in channel 5.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMPCLR5U	CMPCLR5V	CMPCLR5W
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CMPCLR5U	0	R/W	TCNT Compare Clear 5U Enables or disables requests to clear TCNTU_5 at TGRU_5 compare match or input capture. 0: Disables TCNTU_5 to be cleared to H'0000 at TCNTU_5 and TGRU_5 compare match or input capture 1: Enables TCNTU_5 to be cleared to H'0000 at TCNTU_5 and TGRU_5 compare match or input capture
1	CMPCLR5V	0	R/W	TCNT Compare Clear 5V Enables or disables requests to clear TCNTV_5 at TGRV_5 compare match or input capture. 0: Disables TCNTV_5 to be cleared to H'0000 at TCNTV_5 and TGRV_5 compare match or input capture 1: Enables TCNTV_5 to be cleared to H'0000 at TCNTV_5 and TGRV_5 compare match or input capture

Bit	Bit Name	Initial Value	R/W	Description
0	CMPCLR5W	0	R/W	TCNT Compare Clear 5W Enables or disables requests to clear TCNTW_5 at TGRW_5 compare match or input capture. 0: Disables TCNTW_5 to be cleared to H'0000 at TCNTW_5 and TGRW_5 compare match or input capture 1: Enables TCNTW_5 to be cleared to H'0000 at TCNTW_5 and TGRW_5 compare match or input capture

12.3.5 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. The MTU2 has seven TIER registers, two for channel 0 and one each for channels 1 to 5.

- TIER_0, TIER_1, TIER_2, TIER_3, TIER_4

Bit:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Converter Start Request Enable Enables or disables generation of A/D converter start requests by TGRA input capture/compare match. 0: A/D converter start request generation disabled 1: A/D converter start request generation enabled

Bit	Bit Name	Initial Value	R/W	Description
6	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by TCNT_4 underflow (trough) in complementary PWM mode. In channels 0 to 3, bit 6 is reserved. It is always read as 0 and the write value should always be 0. 0: A/D converter start request generation by TCNT_4 underflow (trough) disabled 1: A/D converter start request generation by TCNT_4 underflow (trough) enabled
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial Value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

- TIER2_0

Bit:	7	6	5	4	3	2	1	0
	TTGE2	-	-	-	-	-	TGIEF	TGIEE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by compare match between TCNT_0 and TGRE_0. 0: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 disabled 1: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 enabled
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGIEF	0	R/W	TGR Interrupt Enable F Enables or disables interrupt requests by compare match between TCNT_0 and TGRF_0. 0: Interrupt requests (TGIF) by TGFE bit disabled 1: Interrupt requests (TGIF) by TGFE bit enabled
0	TGIEE	0	R/W	TGR Interrupt Enable E Enables or disables interrupt requests by compare match between TCNT_0 and TGRE_0. 0: Interrupt requests (TGIE) by TGEE bit disabled 1: Interrupt requests (TGIE) by TGEE bit enabled

- TIER_5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TGIE5U	TGIE5V	TGIE5W
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TGIE5U	0	R/W	TGR Interrupt Enable 5U Enables or disables TGIU_5 interrupt requests by the CMFU5 bit when the CMFU5 bit in TSR_5 is set to 1. 0: TGIU_5 interrupt requests disabled 1: TGIU_5 interrupt requests enabled
1	TGIE5V	0	R/W	TGR Interrupt Enable 5V Enables or disables TGIV_5 interrupt requests by the CMFV5 bit when the CMFV5 bit in TSR_5 is set to 1. 0: TGIV_5 interrupt requests disabled 1: TGIV_5 interrupt requests enabled
0	TGIE5W	0	R/W	TGR Interrupt Enable 5W Enables or disables TGIW_5 interrupt requests by the CMFW5 bit when the CMFW5 bit in TSR_5 is set to 1. 0: TGIW_5 interrupt requests disabled 1: TGIW_5 interrupt requests enabled

12.3.6 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU2 has seven TSR registers, two for channel 0 and one each for channels 1 to 5.

- TSR_0, TSR_1, TSR_2, TSR_3, TSR_4

Bit:	7	6	5	4	3	2	1	0
	TCFD	-	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1 to 4. In channel 0, bit 7 is reserved. It is always read as 1 and the write value should always be 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	TCFU	0	R/(W)*1	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1*2 [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF)

Bit	Bit Name	Initial Value	R/W	Description
4	TCFV	0	R/(W)*1	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1*2 [Setting condition] • When the TCNT value overflows (changes from H'FFFF to H'0000) In channel 4, when the TCNT_4 value underflows (changes from H'0001 to H'0000) in complementary PWM mode, this flag is also set.
3	TGFD	0	R/(W)*1	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1*2 • When DTC is activated by TGID interrupt, and the DISEL bit of MRB in DTC is cleared to 0. [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register

Bit	Bit Name	Initial Value	R/W	Description
2	TGFC	0	R/(W)*1	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When DTC is activated by TGIC interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to TGFC after reading TGFC = 1*2 [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register
1	TGFB	0	R/(W)*1	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing. [Clearing condition] • When DTC is activated by TGIB interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to TGFB after reading TGFB = 1*2 [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register

Bit	Bit Name	Initial Value	R/W	Description
0	TGFA	0	R/(W)*1	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing. [Clearing conditions] • When DMAC is activated by TGIA interrupt. • When DTC is activated by TGIA interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to TGFA after reading TGFA = 1*2 [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. After reading 1, when the next flag set is generated before writing 0, the flag will not be cleared by writing 0. Read 1 again and write 0 in this case.

- TSR2_0

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TGFF	TGFE
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGFF	0	R/(W)*1	Compare Match Flag F Status flag that indicates the occurrence of compare match between TCNT_0 and TGRF_0. [Clearing condition] • When 0 is written to TGFF after reading TGFF = 1*2 [Setting condition] • When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register
0	TGFE	0	R/(W)*1	Compare Match Flag E Status flag that indicates the occurrence of compare match between TCNT_0 and TGRE_0. [Clearing condition] • When 0 is written to TGFE after reading TGFE = 1*2 [Setting condition] • When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. After reading 1 when the next flag set is generated before writing 0, the flag will not be cleared by writing 0. Read 1 again and write 0 in this case.

- TSR_5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMFU5	CMFV5	CMFW5
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CMFU5	0	R/(W)*1	Compare Match/Input Capture Flag U5 Status flag that indicates the occurrence of TGRU_5 input capture or compare match. [Clearing condition] • When DTC is activated by TGIU_5 interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to CMFU5 after reading CMFU5 = 1 [Setting conditions] • When TCNTU_5 = TGRU_5 and TGRU_5 is functioning as output compare register • When TCNTU_5 value is transferred to TGRU_5 by input capture signal and TGRU_5 is functioning as input capture register • When TCNTU_5 value is transferred to TGRU_5 and TGRU_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control registers U_5, V_5, and W_5 (TIORU_5, TIORV_5, and TIORW_5).*2

Bit	Bit Name	Initial Value	R/W	Description
1	CMFV5	0	R/(W)*1	Compare Match/Input Capture Flag V5 Status flag that indicates the occurrence of TGRV_5 input capture or compare match. [Clearing condition] • When DTC is activated by TGIV_5 interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to CMFV5 after reading CMFV5 = 1 [Setting conditions] • When TCNTV_5 = TGRV_5 and TGRV_5 is functioning as output compare register • When TCNTV_5 value is transferred to TGRV_5 by input capture signal and TGRV_5 is functioning as input capture register • When TCNTV_5 value is transferred to TGRV_5 and TGRV_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control registers U_5, V_5, and W_5 (TIORU_5, TIORV_5, and TIORW_5).*2

Bit	Bit Name	Initial Value	R/W	Description
0	CMFW5	0	R/(W)*1	Compare Match/Input Capture Flag W5 Status flag that indicates the occurrence of TGRW_5 input capture or compare match. Only 0 can be written to clear this flag. [Clearing condition] • When DTC is activated by TGIW_5 interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to CMFW5 after reading CMFW5 = 1 [Setting conditions] • When TCNTW_5 = TGRW_5 and TGRW_5 is functioning as output compare register • When TCNTW_5 value is transferred to TGRW_5 by input capture signal and TGRW_5 is functioning as input capture register • When TCNTW_5 value is transferred to TGRW_5 and TGRW_5 is functioning as a register for measuring the pulse width of the external input signal. *2

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Timing for transfer is set by the IOC bit in the timer I/O control register U_5/V_5/W_5 (TIORU_5/V_5/W_5).

12.3.7 Timer Buffer Operation Transfer Mode Register (TBTM)

The TBTM registers are 8-bit readable/writable registers that specify the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU2 has three TBTM registers, one each for channels 0, 3, and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TTSE	TTSB	TTSA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TTSE	0	R/W	Timing Select E Specifies the timing for transferring data from TGRF_0 to TGRE_0 when they are used together for buffer operation. In channels 3 and 4, bit 2 is reserved. It is always read as 0 and the write value should always be 0. When channel 0 is used in a mode other than PWM mode, do not set this bit to 1. 0: When compare match E occurs in channel 0 1: When TCNT_0 is cleared
1	TTSB	0	R/W	Timing Select B Specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When the channel is used in a mode other than PWM mode, do not set this bit to 1. 0: When compare match B occurs in each channel 1: When TCNT is cleared in each channel
0	TTSA	0	R/W	Timing Select A Specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When the channel is used in a mode other than PWM mode, do not set this bit to 1. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

12.3.8 Timer Input Capture Control Register (TICCR)

TICCR is an 8-bit readable/writable register that specifies input capture conditions when TCNT_1 and TCNT_2 are cascaded. The MTU2 has one TICCR in channel 1.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	I2BE	I2AE	I1BE	I1AE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	I2BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2B pin in the TGRB_1 input capture conditions. 0: Does not include the TIOC2B pin in the TGRB_1 input capture conditions 1: Includes the TIOC2B pin in the TGRB_1 input capture conditions
2	I2AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2A pin in the TGRA_1 input capture conditions. 0: Does not include the TIOC2A pin in the TGRA_1 input capture conditions 1: Includes the TIOC2A pin in the TGRA_1 input capture conditions
1	I1BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1B pin in the TGRB_2 input capture conditions. 0: Does not include the TIOC1B pin in the TGRB_2 input capture conditions 1: Includes the TIOC1B pin in the TGRB_2 input capture conditions

Bit	Bit Name	Initial Value	R/W	Description
0	I1AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1A pin in the TGRA_2 input capture conditions. 0: Does not include the TIOC1A pin in the TGRA_2 input capture conditions 1: Includes the TIOC1A pin in the TGRA_2 input capture conditions

12.3.9 Timer Synchronous Clear Register (TSYCR)

TSYCR is an 8-bit readable/writable register that specifies conditions for clearing TCNT_3 and TCNT_4 in the MTU2S in synchronization with the MTU2. The MTU2S has one TSYCR in channel 3 but the MTU2 has no TSYCR.

Bit:	7	6	5	4	3	2	1	0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CE0A	0	R/W	Clear Enable 0A Enables or disables counter clearing when the TGFA flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag in TSR_0 1: Enables counter clearing by the TGFA flag in TSR_0
6	CE0B	0	R/W	Clear Enable 0B Enables or disables counter clearing when the TGFB flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag in TSR_0 1: Enables counter clearing by the TGFB flag in TSR_0

Bit	Bit Name	Initial Value	R/W	Description
5	CE0C	0	R/W	Clear Enable 0C Enables or disables counter clearing when the TGFC flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFC flag in TSR_0 1: Enables counter clearing by the TGFC flag in TSR_0
4	CE0D	0	R/W	Clear Enable 0D Enables or disables counter clearing when the TGFD flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFD flag in TSR_0 1: Enables counter clearing by the TGFD flag in TSR_0
3	CE1A	0	R/W	Clear Enable 1A Enables or disables counter clearing when the TGFA flag of TSR_1 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag in TSR_1 1: Enables counter clearing by the TGFA flag in TSR_1
2	CE1B	0	R/W	Clear Enable 1B Enables or disables counter clearing when the TGFB flag of TSR_1 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag in TSR_1 1: Enables counter clearing by the TGFB flag in TSR_1
1	CE2A	0	R/W	Clear Enable 2A Enables or disables counter clearing when the TGFA flag of TSR_2 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag in TSR_2 1: Enables counter clearing by the TGFA flag in TSR_2
0	CE2B	0	R/W	Clear Enable 2B Enables or disables counter clearing when the TGFB flag of TSR_2 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag in TSR_2 1: Enables counter clearing by the TGFB flag in TSR_2

12.3.10 Timer A/D Converter Start Request Control Register (TADCR)

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation. The MTU2 has one TADCR in channel 4.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BF[1:0]		-	-	-	-	-	-	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Initial value:	0	0	0	0	0	0	0	0	0	0*	0	0*	0*	0*	0*	0*
R/W:	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BF[1:0]	00	R/W	TADCOBRA_4/TADCOBRB_4 Transfer Timing Select Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADCORA_4 and TADCORB_4. For details, see table 12.29.
13 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	UT4AE	0	R/W	Up-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 up-count operation
6	DT4AE	0*	R/W	Down-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 down-count operation

Bit	Bit Name	Initial Value	R/W	Description
5	UT4BE	0	R/W	Up-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 up-count operation
4	DT4BE	0*	R/W	Down-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 down-count operation
3	ITA3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping
2	ITA4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping
1	ITB3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping

Bit	Bit Name	Initial Value	R/W	Description
0	ITB4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping

Notes: 1. TADCR must not be accessed in eight bits; it should always be accessed in 16 bits.
2. When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
* Do not set to 1 when complementary PWM mode is not selected.

Table 12.29 Setting of Transfer Timing by Bits BF1 and BF0

Bit 7	Bit 6	
BF1	**BF0**	**Description**
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the TCNT_4 count.*[1]
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the TCNT_4 count.*[2]
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the TCNT_4 count.*[2]

Notes: 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the TCNT_4 count is reached in complementary PWM mode, when compare match occurs between TCNT_3 and TGRA_3 in reset-synchronized PWM mode, or when compare match occurs between TCNT_4 and TGRA_4 in PWM mode 1 or normal operation mode.
2. These settings are prohibited when complementary PWM mode is not selected.

12.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4)

TADCORA_4 and TADCORB_4 are 16-bit readable/writable registers. When the TCNT_4 count reaches the value in TADCORA_4 or TADCORB_4, a corresponding A/D converter start request will be issued.

TADCORA_4 and TADCORB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

12.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the crest or trough of the TCNT_4 count is reached, these register values are transferred to TADCORA_4 and TADCORB_4, respectively.

TADCOBRA_4 and TADCOBRB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

12.3.13 Timer Counter (TCNT)

The TCNT counters are 16-bit readable/writable counters. The MTU2 has eight TCNT counters, one each for channels 0 to 4 and three (TCNTU_5, TCNTV_5, and TCNTW_5) for channel 5.

The TCNT counters are initialized to H'0000 by a reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

12.3.14 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has 21 TGR registers, six for channel 0, two each for channels 1 and 2, four each for channels 3 and 4, and three for channel 5.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

TGRE_0 and TGRF_0 function as compare registers. When the TCNT_0 count matches the TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

TGRU_5, TGRV_5, and TGRW_5 function as compare match, input capture, or external pulse width measurement registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits.
TGR registers are initialized to H'FFFF.

12.3.15 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT for channels 0 to 4.

TSTR_5 is an 8-bit readable/writable register that selects operation/stoppage of TCNTU_5, TCNTV_5, and TCNTW_5 for channel 5.

When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

- TSTR

Bit:	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3
6	CST3	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_4 and TCNT_3 count operation is stopped 1: TCNT_4 and TCNT_3 performs count operation
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	CST2	0	R/W	Counter Start 2 to 0
1	CST1	0	R/W	These bits select operation or stoppage for TCNT.
0	CST0	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation

- TSTR_5

Bit :	7	6	5	4	3	2	1	0
	-	-	-	-	-	CSTU5	CSTV5	CSTW5
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CSTU5	0	R/W	Counter Start U5 Selects operation or stoppage for TCNTU_5. 0: TCNTU_5 count operation is stopped 1: TCNTU_5 performs count operation
1	CSTV5	0	R/W	Counter Start V5 Selects operation or stoppage for TCNTV_5. 0: TCNTV_5 count operation is stopped 1: TCNTV_5 performs count operation
0	CSTW5	0	R/W	Counter Start W5 Selects operation or stoppage for TCNTW_5. 0: TCNTW_5 count operation is stopped 1: TCNTW_5 performs count operation

12.3.16 Timer Synchronous Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	-	-	-	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SYNC4	0	R/W	Timer Synchronous operation 4 and 3
6	SYNC3	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR. 0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNT_4 and TCNT_3 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	SYNC2	0	R/W	Timer Synchronous operation 2 to 0
1	SYNC1	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels.
0	SYNC0	0	R/W	When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR. 0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

12.3.17 Timer Counter Synchronous Start Register (TCSYSTR)

TCSYSTR is an 8-bit readable/writable register that specifies synchronous start of the MTU2 and MTU2S counters. Note that the MTU2S does not have TCSYSTR.

Bit:	7	6	5	4	3	2	1	0
	SCH0	SCH1	SCH2	SCH3	SCH4	-	SCH3S	SCH4S
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*

Note: * Only 1 can be written to set the register.

Bit	Bit Name	Initial Value	R/W	Description
7	SCH0	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_0 in the MTU2. 0: Does not specify synchronous start for TCNT_0 in the MTU2 1: Specifies synchronous start for TCNT_0 in the MTU2 [Clearing condition] • When 1 is set to the CST0 bit of TSTR in MTU2 while SCH0 = 1
6	SCH1	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_1 in the MTU2. 0: Does not specify synchronous start for TCNT_1 in the MTU2 1: Specifies synchronous start for TCNT_1 in the MTU2 [Clearing condition] • When 1 is set to the CST1 bit of TSTR in MTU2 while SCH1 = 1

Bit	Bit Name	Initial Value	R/W	Description
5	SCH2	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_2 in the MTU2. 0: Does not specify synchronous start for TCNT_2 in the MTU2 1: Specifies synchronous start for TCNT_2 in the MTU2 [Clearing condition] • When 1 is set to the CST2 bit of TSTR in MTU2 while SCH2 = 1
4	SCH3	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_3 in the MTU2. 0: Does not specify synchronous start for TCNT_3 in the MTU2 1: Specifies synchronous start for TCNT_3 in the MTU2 [Clearing condition] • When 1 is set to the CST3 bit of TSTR in MTU2 while SCH3 = 1
3	SCH4	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4 in the MTU2. 0: Does not specify synchronous start for TCNT_4 in the MTU2 1: Specifies synchronous start for TCNT_4 in the MTU2 [Clearing condition] • When 1 is set to the CST4 bit of TSTR in MTU2 while SCH4 = 1
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	SCH3S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_3S in the MTU2S. 0: Does not specify synchronous start for TCNT_3S in the MTU2S 1: Specifies synchronous start for TCNT_3S in the MTU2S [Clearing condition] • When 1 is set to the CST3 bit of TSTRS in MTU2S while SCH3S = 1
0	SCH4S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4S in the MTU2S. 0: Does not specify synchronous start for TCNT_4S in the MTU2S 1: Specifies synchronous start for TCNT_4S in the MTU2S [Clearing condition] • When 1 is set to the CST4 bit of TSTRS in MTU2S while SCH4S = 1

Note: Only 1 can be written to set the register.

12.3.18 Timer Read/Write Enable Register (TRWER)

TRWER is an 8-bit readable/writable register that enables or disables access to the registers and counters which have write-protection capability against accidental modification in channels 3 and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RWE
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RWE	1	R/W	Read/Write Enable Enables or disables access to the registers which have write-protection capability against accidental modification. 0: Disables read/write access to the registers 1: Enables read/write access to the registers [Clearing condition] • When 0 is written to the RWE bit after reading RWE = 1

- Registers and counters having write-protection capability against accidental modification
  22 registers: TCR_3, TCR_4, TMDR_3, TMDR_4, TIORH_3, TIORH_4, TIORL_3, TIORL_4, TIER_3, TIER_4, TGRA_3, TGRA_4, TGRB_3, TGRB_4, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT4.

12.3.19 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4. The CST3 and CST4 bits in TSTR should be 0 for writing 1 to TOER bits.
If a 1 is to be written to the TOER register, do so while the CST3 and CST4 bits in the TSTR register are 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	OE4D	0	R/W	Master Enable TIOC4D This bit enables/disables the TIOC4D pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

Bit	Bit Name	Initial Value	R/W	Description
0	OE3B	0	R/W	Master Enable TIOC3B This bit enables/disables the TIOC3B pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

Note: * The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 12.3.20, Timer Output Control Register 1 (TOCR1), and section 12.3.21, Timer Output Control Register 2 (TOCR2). Set these bits to 1 to enable MTU2 output in other than complementary PWM or reset-synchronized PWM mode. When these bits are set to 0, low level is output.

12.3.20 Timer Output Control Register 1 (TOCR1)

TOCR1 is an 8-bit readable/writable register that enables/disables PWM synchronized toggle output in complementary PWM mode/reset synchronized PWM mode, and controls output level inversion of PWM output.

Bit:	7	6	5	4	3	2	1	0
	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*	R/W	R/W	R/W

Note: * This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to the bit.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PSYE	0	R/W	PWM Synchronous Output Enable This bit selects the enable/disable of toggle output synchronized with the PWM period. 0: Toggle output is disabled 1: Toggle output is enabled
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
3	TOCL	0	R/(W)*	TOC Register Write Protection*[1] This bit selects the enable/disable of write access to the TOCS, OLSN, and OLSP bits in TOCR1. 0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled
2	TOCS	0	R/W	TOC Select This bit selects either the TOCR1 or TOCR2 setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode. 0: TOCR1 setting is selected 1: TOCR2 setting is selected
1	OLSN	0	R/W	Output Level Select N*[2]*[3] This bit selects the negative phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 12.30.
0	OLSP	0	R/W	Output Level Select P*[2]*[3] This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 12.31.

Notes: 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
2. Clearing the TOCS0 bit to 0 makes this bit setting valid.
3. The inverse-phase output is the exact inverse of the positive-phase output unless dead time is generated. When no dead time is generated, only the OLSP setting is valid.

Table 12.30 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to active level after elapse of the dead time after count start.

Table 12.31 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Figure 12.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

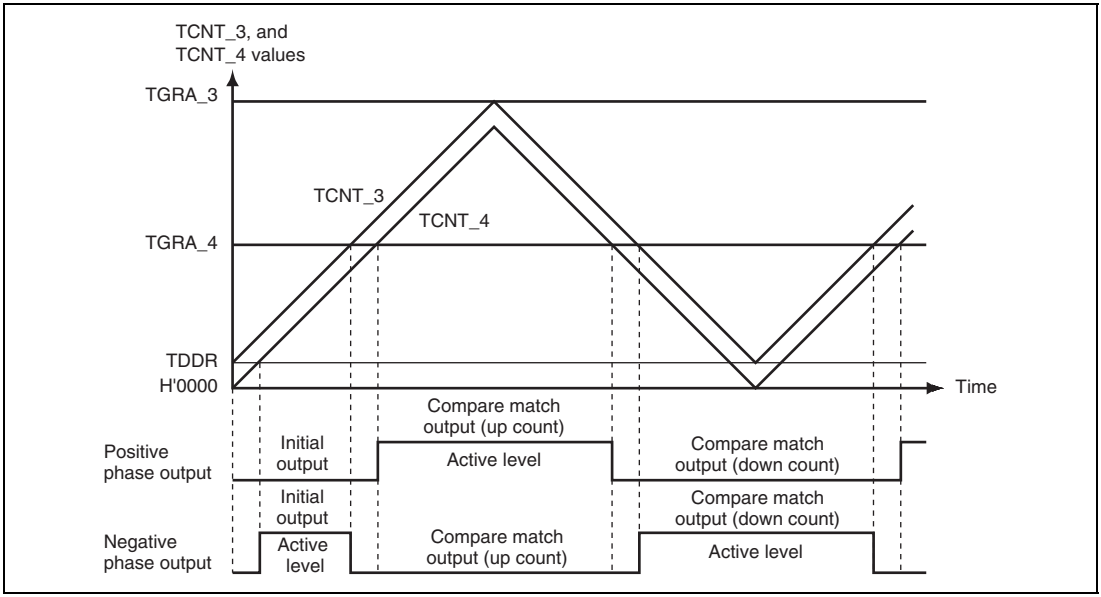


Figure 12.2 Complementary PWM Mode Output Level Example

12.3.21 Timer Output Control Register 2 (TOCR2)

TOCR2 is an 8-bit readable/writable register that controls output level inversion of PWM output in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	BF[1:0]	00	R/W	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBR to TOCR2. For details, see table 12.32.
5	OLS3N	0	R/W	Output Level Select 3N*1*2 This bit selects the output level on TIOC4D in reset-synchronized PWM mode/complementary PWM mode. See table 12.33.
4	OLS3P	0	R/W	Output Level Select 3P*1*2 This bit selects the output level on TIOC4B in reset-synchronized PWM mode/complementary PWM mode. See table 12.34.
3	OLS2N	0	R/W	Output Level Select 2N*1*2 This bit selects the output level on TIOC4C in reset-synchronized PWM mode/complementary PWM mode. See table 12.35.
2	OLS2P	0	R/W	Output Level Select 2P*1*2 This bit selects the output level on TIOC4A in reset-synchronized PWM mode/complementary PWM mode. See table 12.36.
1	OLS1N	0	R/W	Output Level Select 1N*1*2 This bit selects the output level on TIOC3D in reset-synchronized PWM mode/complementary PWM mode. See table 12.37.

Bit	Bit Name	Initial value	R/W	Description
0	OLS1P	0	R/W	Output Level Select 1P*1*2 This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 12.38.

Notes: 1. Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid.
2. The inverse-phase output is the exact inverse of the positive-phase output unless dead time is generated. When no dead time is generated, only the OLSiP setting is valid (i = 1, 2, 3).

Table 12.32 Setting of Bits BF1 and BF0

Bit 7	Bit 6	Description	
BF1	**BF0**	**Complementary PWM Mode**	**Reset-Synchronized PWM Mode**
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the TCNT_4 count.	Transfers data from the buffer register (TOLBR) to TOCR2 when TCNT_3/TCNT_4 is cleared
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the TCNT_4 count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the TCNT_4 count.	Setting prohibited

Table 12.33 TIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	**Initial Output**	**Active Level**	**Up Count**	**Down Count**
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 12.34 TIOC4B Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 12.35 TIOC4C Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 12.36 TIOC4A Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 12.37 TIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The negative phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 12.38 TIOC3B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

12.3.22 Timer Output Level Buffer Register (TOLBR)

TOLBR is an 8-bit readable/writable register that functions as a buffer for TOCR2 and specifies the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	OLS3N	0	R/W	Specifies the buffer value to be transferred to the OLS3N bit in TOCR2.
4	OLS3P	0	R/W	Specifies the buffer value to be transferred to the OLS3P bit in TOCR2.
3	OLS2N	0	R/W	Specifies the buffer value to be transferred to the OLS2N bit in TOCR2.
2	OLS2P	0	R/W	Specifies the buffer value to be transferred to the OLS2P bit in TOCR2.
1	OLS1N	0	R/W	Specifies the buffer value to be transferred to the OLS1N bit in TOCR2.
0	OLS1P	0	R/W	Specifies the buffer value to be transferred to the OLS1P bit in TOCR2.

Figure 12.3 shows an example of the PWM output level setting procedure in buffer operation.

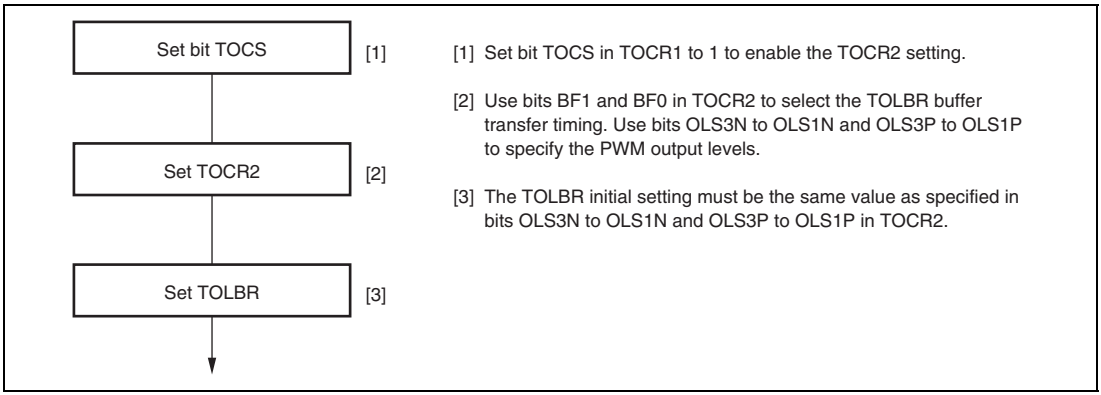


Figure 12.3 PWM Output Level Setting Procedure in Buffer Operation

12.3.23 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary for brushless DC motor control in reset-synchronized PWM mode/complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	BDC	N	P	FB*	WF	VF	UF
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
6	BDC	0	R/W	Brushless DC Motor This bit selects whether to make the functions of this register (TGCR) effective or ineffective. 0: Ordinary output 1: Functions of this register are made effective

Bit	Bit Name	Initial value	R/W	Description
5	N	0	R/W	Negative Phase Output (N) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the reverse pins (TIOC3D, TIOC4C, and TIOC4D) are output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
4	P	0	R/W	Positive Phase Output (P) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the positive pin (TIOC3B, TIOC4A, and TIOC4B) are output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
3	FB*	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the output of the positive/negative phase is carried out automatically with the MTU2/channel 0 TGRA, TGRB, TGRC input capture signals or by writing 0 or 1 to bits 2 to 0 in TGCR. 0: Output switching is external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal) 1: Output switching is carried out by software (setting values of UF, VF, and WF in TGCR).
2	WF	0	R/W	Output Phase Switch 2 to 0 These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2 to 0 is a substitute for external input. See table 12.39.
1	VF	0	R/W	
0	UF	0	R/W	

Note: * Do not set the FB bit to 0 when the BDC bit in MTU2S has been set to 1.

Table 12.39 Output level Select Function

			Function					
Bit 2	Bit 1	Bit 0	TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
		1	ON	OFF	OFF	OFF	OFF	ON
	1	0	OFF	ON	OFF	ON	OFF	OFF
		1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
		1	ON	OFF	OFF	OFF	ON	OFF
	1	0	OFF	OFF	ON	ON	OFF	OFF
		1	OFF	OFF	OFF	OFF	OFF	OFF

12.3.24 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

The initial value of TCNTS is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

12.3.25 Timer Dead Time Data Register (TDDR)

TDDR is a 16-bit register, used only in complementary PWM mode that specifies the TCNT_3 and TCNT_4 counter offset values. In complementary PWM mode, when the TCNT_3 and TCNT_4 counters are cleared and then restarted, the TDDR register value is loaded into the TCNT_3 counter and the count operation starts.

The initial value of TDDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.26 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment).

The initial value of TCDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.27 Timer Cycle Buffer Register (TCBR)

TCBR is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing set in the TMDR register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.28 Timer Interrupt Skipping Set Register (TITCR)

TITCR is an 8-bit readable/writable register that enables or disables interrupt skipping and specifies the interrupt skipping count. The MTU2 has one TITCR.

Bit:	7	6	5	4	3	2	1	0
	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	T3AEN	0	R/W	T3AEN Enables or disables TGIA_3 interrupt skipping. 0: TGIA_3 interrupt skipping disabled 1: TGIA_3 interrupt skipping enabled
6 to 4	3ACOR[2:0]	000	R/W	These bits specify the TGIA_3 interrupt skipping count within the range from 0 to 7.* For details, see table 12.40.
3	T4VEN	0	R/W	T4VEN Enables or disables TCIV_4 interrupt skipping. 0: TCIV_4 interrupt skipping disabled 1: TCIV_4 interrupt skipping enabled

Bit	Bit Name	Initial value	R/W	Description
2 to 0	4VCOR[2:0]	000	R/W	These bits specify the TCIV_4 interrupt skipping count within the range from 0 to 7.* For details, see table 12.41.

Note: * When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter (TICNT).

Table 12.40 Setting of Interrupt Skipping Count by Bits 3ACOR2 to 3ACOR0

Bit 6	Bit 5	Bit 4	
3ACOR2	3ACOR1	3ACOR0	Description
0	0	0	Does not skip TGIA_3 interrupts.
0	0	1	Sets the TGIA_3 interrupt skipping count to 1.
0	1	0	Sets the TGIA_3 interrupt skipping count to 2.
0	1	1	Sets the TGIA_3 interrupt skipping count to 3.
1	0	0	Sets the TGIA_3 interrupt skipping count to 4.
1	0	1	Sets the TGIA_3 interrupt skipping count to 5.
1	1	0	Sets the TGIA_3 interrupt skipping count to 6.
1	1	1	Sets the TGIA_3 interrupt skipping count to 7.

Table 12.41 Setting of Interrupt Skipping Count by Bits 4VCOR2 to 4VCOR0

Bit 2	Bit 1	Bit 0	
4VCOR2	4VCOR1	4VCOR0	Description
0	0	0	Does not skip TCIV_4 interrupts.
0	0	1	Sets the TCIV_4 interrupt skipping count to 1.
0	1	0	Sets the TCIV_4 interrupt skipping count to 2.
0	1	1	Sets the TCIV_4 interrupt skipping count to 3.
1	0	0	Sets the TCIV_4 interrupt skipping count to 4.
1	0	1	Sets the TCIV_4 interrupt skipping count to 5.
1	1	0	Sets the TCIV_4 interrupt skipping count to 6.
1	1	1	Sets the TCIV_4 interrupt skipping count to 7.

12.3.29 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable/writable counter. The MTU2 has one TITCNT. TITCNT retains its value even after stopping the count operation of TCNT_3 and TCNT_4.

Bit:	7	6	5	4	3	2	1	0
	-	3ACNT[2:0]			-	4VCNT[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 4	3ACNT[2:0]	000	R	TGIA_3 Interrupt Counter While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA_3 interrupt occurs. [Clearing conditions] • When the 3ACNT2 to 3ACNT0 value in TITCNT matches the 3ACOR2 to 3ACOR0 value in TITCR • When the T3AEN bit in TITCR is cleared to 0 • When the 3ACOR2 to 3ACOR0 bits in TITCR are cleared to 0
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	4VCNT[2:0]	000	R	TCIV_4 Interrupt Counter While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV_4 interrupt occurs. [Clearing conditions] • When the 4VCNT2 to 4VCNT0 value in TITCNT matches the 4VCOR2 to 4VCOR2 value in TITCR • When the T4VEN bit in TITCR is cleared to 0 • When the 4VCOR2 to 4VCOR2 bits in TITCR are cleared to 0

Note: To clear the TITCNT, clear the bits T3AEN and T4VEN in TITCR to 0.

12.3.30 Timer Buffer Transfer Set Register (TBTER)

TBTER is an 8-bit readable/writable register that enables or disables transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specifies whether to link the transfer with interrupt skipping operation. The MTU2 has one TBTER.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	BTE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	BTE[1:0]	00	R/W	These bits enable or disable transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. For details, see table 12.42.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Table 12.42 Setting of Bits BTE1 and BTE0

Bit 1	Bit 0	
BTE1	**BTE0**	**Description**
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*2
1	1	Setting prohibited

Note: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 12.4.8, Complementary PWM Mode.

2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

12.3.31 Timer Dead Time Enable Register (TDER)

TDER is an 8-bit readable/writable register that controls dead time generation in complementary PWM mode. The MTU2 has one TDER in channel 3. TDER must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TDER
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/(W)

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDER	1	R/(W)	Dead Time Enable Specifies whether to generate dead time. 0: Does not generate dead time 1: Generates dead time* [Clearing condition] • When 0 is written to TDER after reading TDER = 1

Note: * TDDR must be set to 1 or a larger value.

12.3.32 Timer Waveform Control Register (TWCR)

TWCR is an 8-bit readable/writable register that controls the waveform when synchronous counter clearing occurs in TCNT_3 and TCNT_4 in complementary PWM mode and specifies whether to clear the counters at TGRA_3 compare match. The CCE bit and WRE bit in TWCR must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	CCE	-	-	-	-	-	SCC	WRE
Initial value:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R/(W)	R/(W)

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
7	CCE	0*	R/(W)	Compare Match Clear Enable Specifies whether to clear counters at TGRA_3 compare match in complementary PWM mode. 0: Does not clear counters at TGRA_3 compare match 1: Clears counters at TGRA_3 compare match [Setting condition] • When 1 is written to CCE after reading CCE = 0
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	SCC	0	R/(W)	Synchronous Clearing Control Specifies whether to clear TCNT_3 and TCNT_4 in the MTU2S when synchronous counter clearing between the MTU2 and MTU2S occurs in complementary PWM mode. When using this control, place the MTU2S in complementary PWM mode. When modifying the SCC bit while the counters are operating, do not modify the CCE or WRE bits. Counter clearing synchronized with the MTU2 is disabled by the SCC bit setting only when synchronous clearing occurs outside the Tb interval at the trough. When synchronous clearing occurs in the Tb interval at the trough including the period immediately after TCNT_3 and TCNT_4 start operation, TCNT_3 and TCNT_4 in the MTU2S are cleared. For the Tb interval at the trough in complementary PWM mode, see figure 12.40. In the MTU2, this bit is reserved. It is always read as 0 and the write value should always be 0. 0: Enables clearing of TCNT_3 and TCNT_4 in the MTU2S by MTU2-MTU2S synchronous clearing operation 1: Disables clearing of TCNT_3 and TCNT_4 in the MTU2S by MTU2-MTU2S synchronous clearing operation [Setting condition] • When 1 is written to SCC after reading SCC = 0

Bit	Bit Name	Initial Value	R/W	Description
0	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 12.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Note: * Do not set to 1 when complementary PWM mode is not selected.

12.3.33 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer registers (TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

12.4 Operation

12.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, cycle counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always select MTU2 external pins set function using the pin function controller (PFC).

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in TSTR_5 is set to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 12.4 shows an example of the count operation setting procedure.

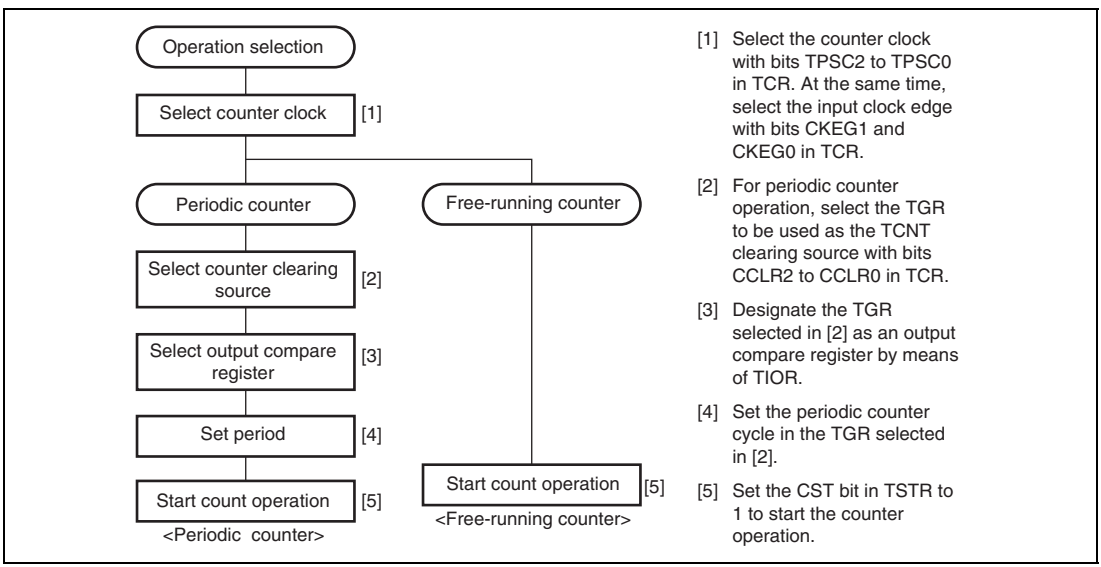


Figure 12.4 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation:

Immediately after a reset, the MTU2's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the MTU2 requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 12.5 illustrates free-running counter operation.

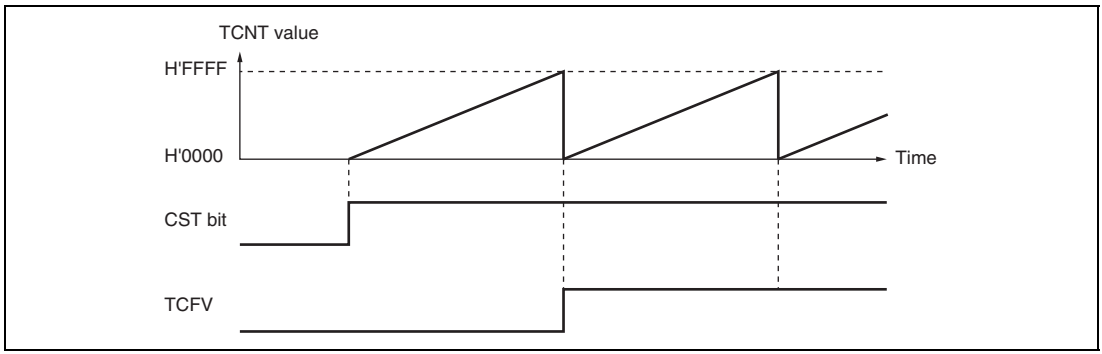


Figure 12.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 to CCLR2 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU2 requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 12.6 illustrates periodic counter operation.

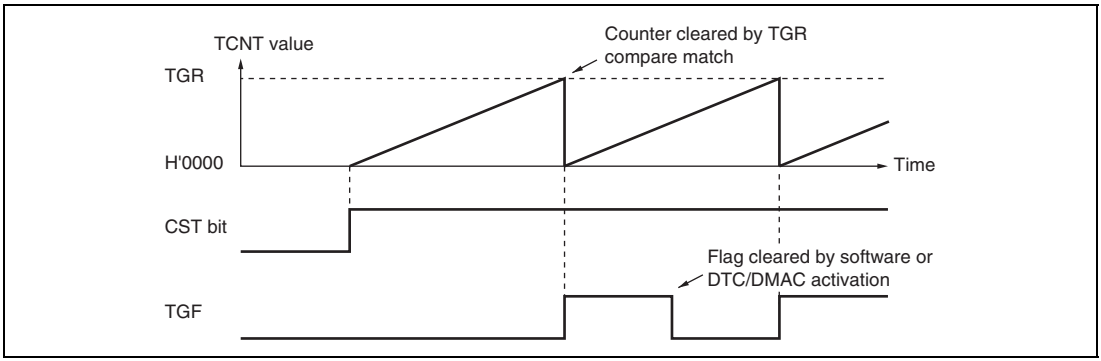


Figure 12.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 12.7 shows an example of the setting procedure for waveform output by compare match.

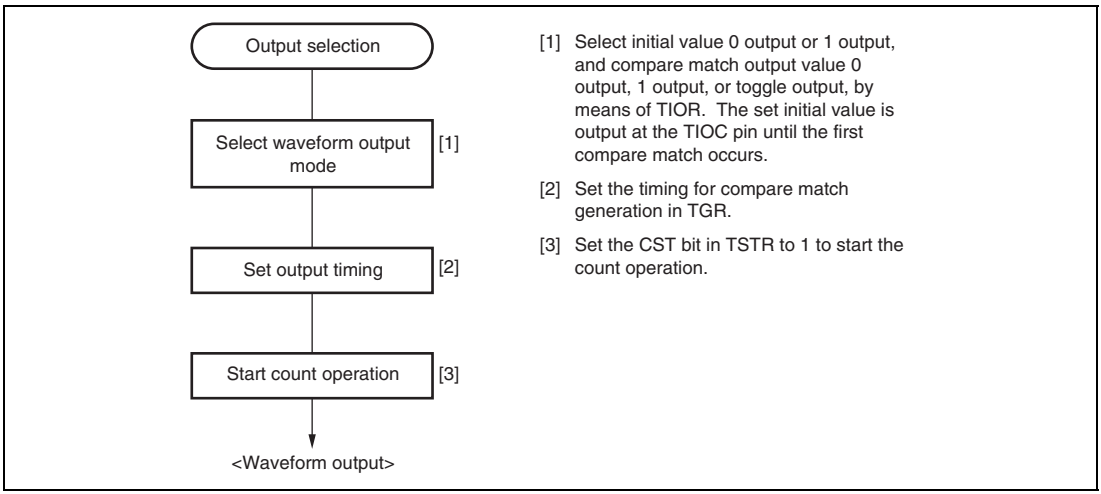


Figure 12.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 12.8 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

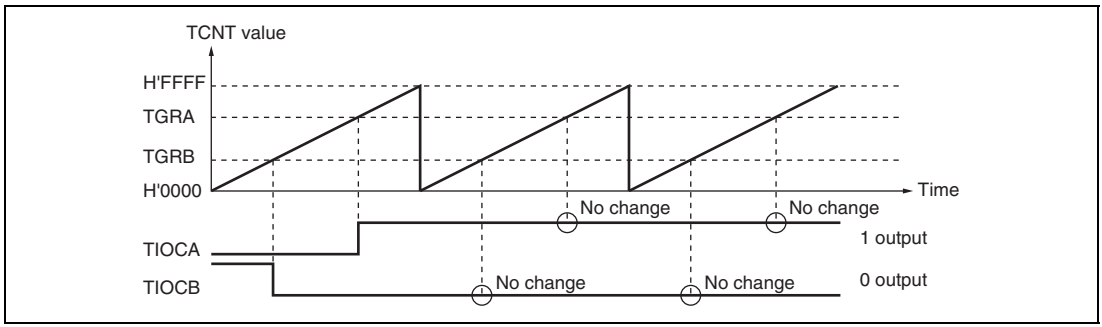


Figure 12.8 Example of 0 Output/1 Output Operation

Figure 12.9 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

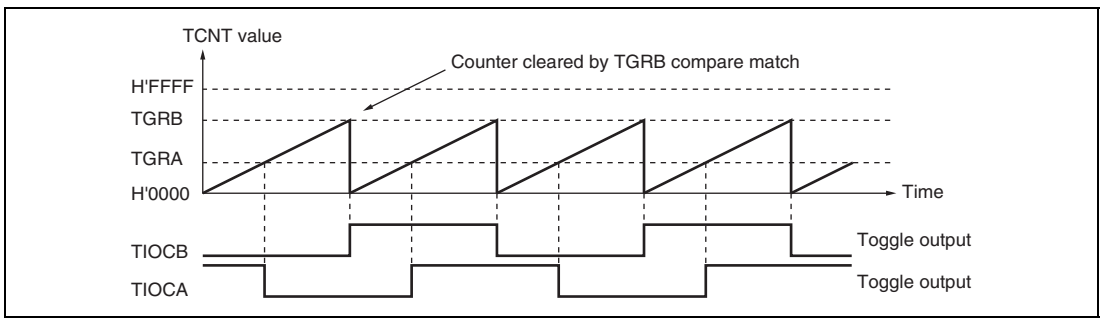


Figure 12.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, Pϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Pϕ/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 12.10 shows an example of the input capture operation setting procedure.

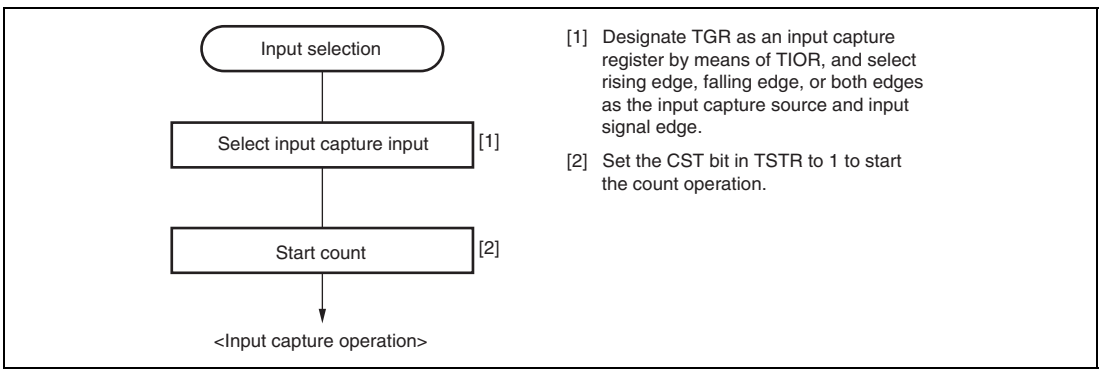


Figure 12.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 12.11 shows an example of input capture operation.

In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, the falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

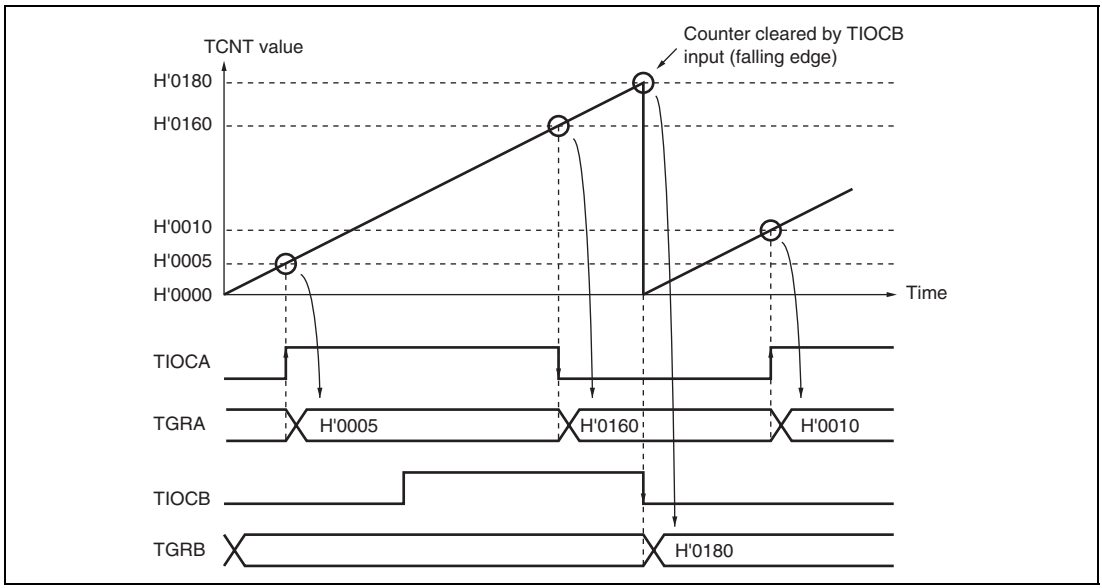


Figure 12.11 Example of Input Capture Operation

12.4.2 Synchronous Operation

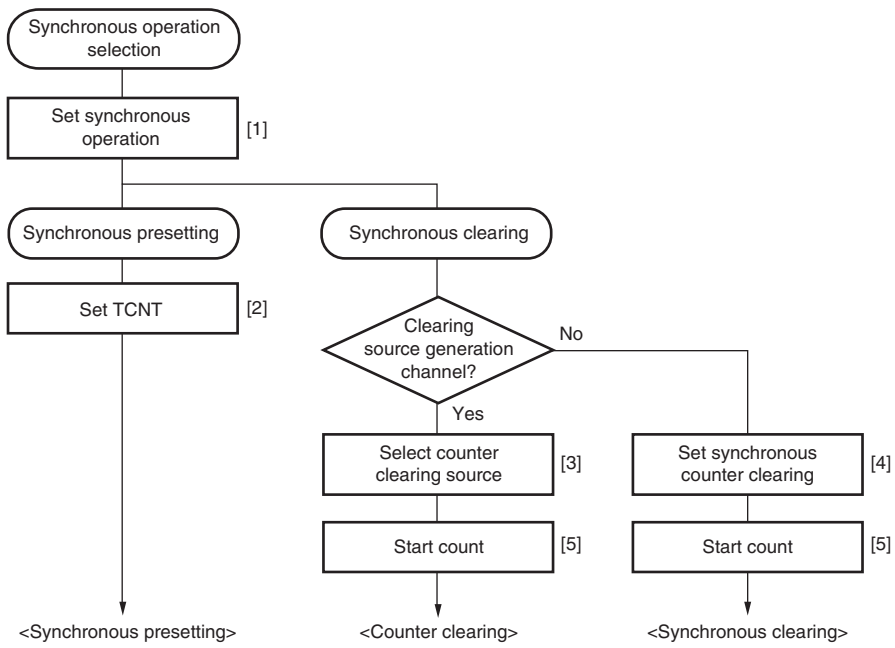
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 4 can all be designated for synchronous operation. Channel 5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 12.12 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 12.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 12.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 12.4.5, PWM Modes.

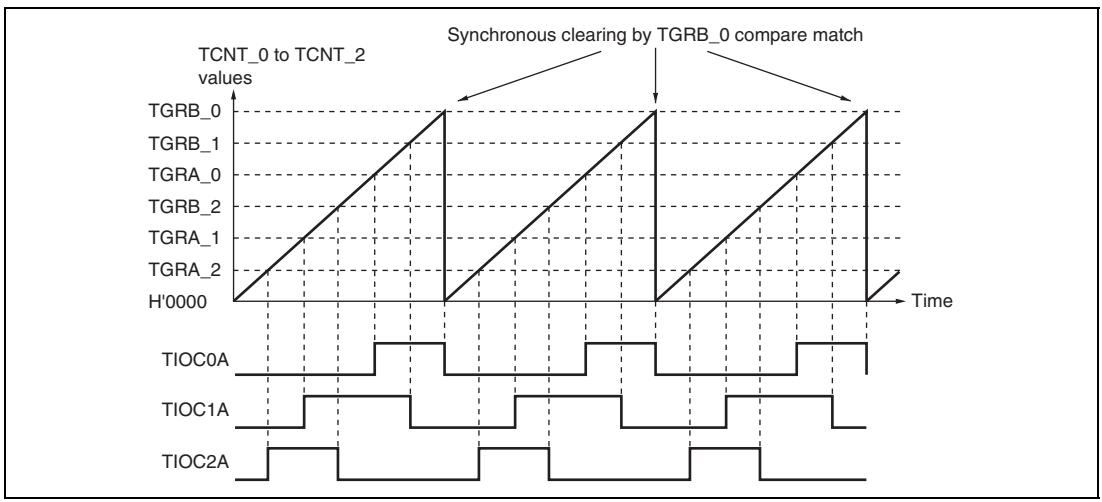


Figure 12.13 Example of Synchronous Operation

12.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4 enables TGRC and TGRD to be used as buffer registers. In channel 0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: TGRE_0 cannot be designated as an input capture register and can only operate as a compare match register.

Table 12.43 shows the register combinations used in buffer operation.

Table 12.43 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 12.14.

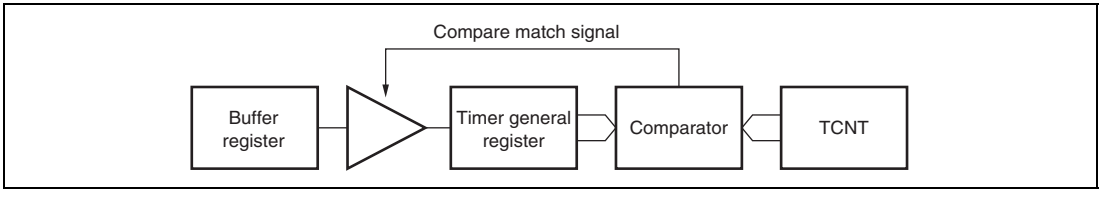


Figure 12.14 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 12.15.

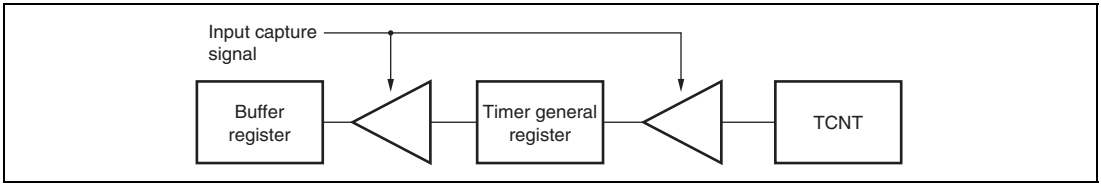


Figure 12.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 12.16 shows an example of the buffer operation setting procedure.

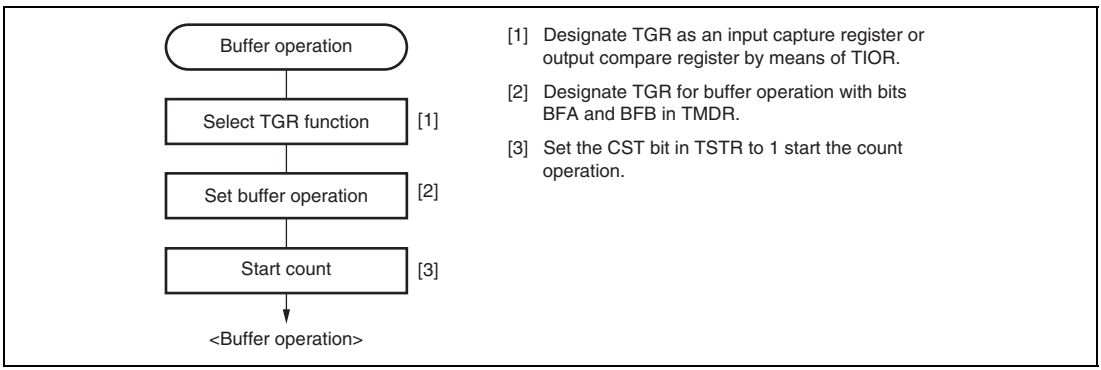


Figure 12.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 12.17 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. In this example, the TTSA bit in TBTM is cleared to 0.

As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time that compare match A occurs.

For details of PWM modes, see section 12.4.5, PWM Modes.

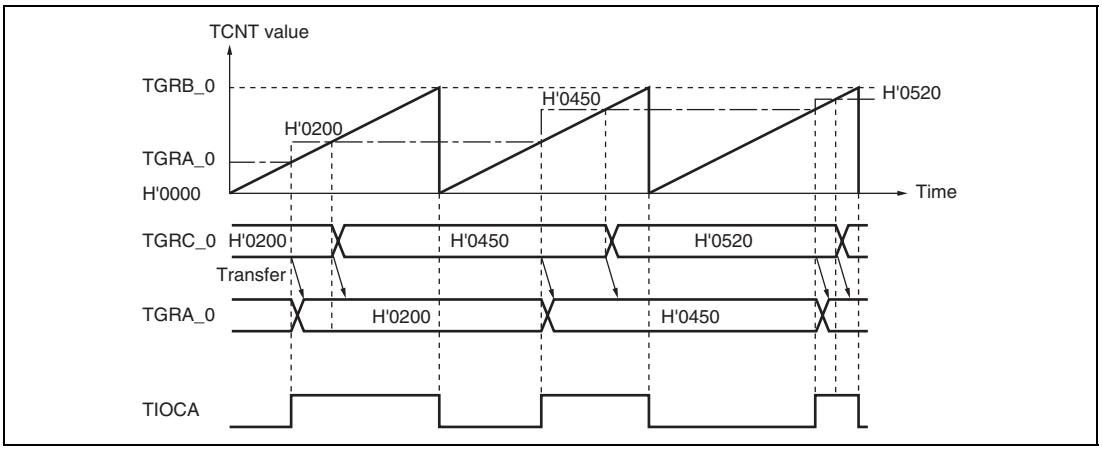


Figure 12.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 12.18 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

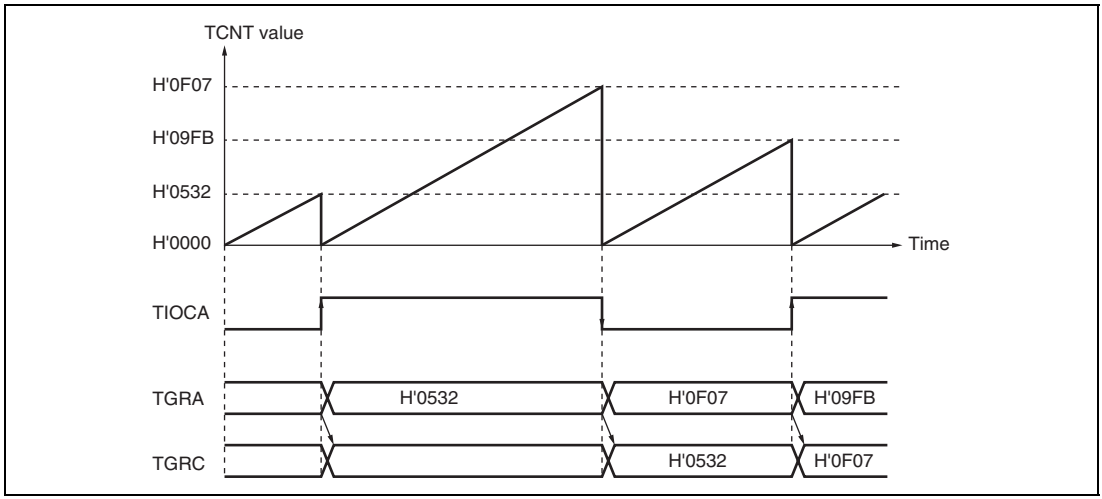


Figure 12.18 Example of Buffer Operation (2)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting the buffer operation transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (H'FFFF to H'0000)
- When H'0000 is written to TCNT during counting
- When TCNT is cleared to H'0000 under the condition specified in the CCLR2 to CCLR0 bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 12.19 shows an operation example in which PWM mode 1 is designated for channel 0 and buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this example are TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. The TTSA bit in TBTM_0 is set to 1.

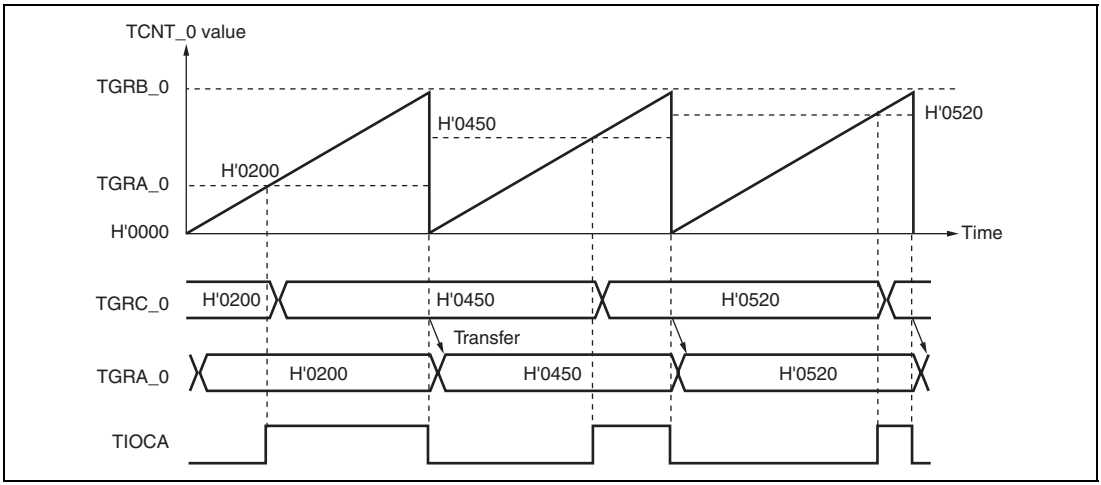


Figure 12.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

12.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 counter clock upon overflow/underflow of TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase counting mode.

Table 12.44 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is invalid and the counters operate independently in phase counting mode.

Table 12.44 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). Edge detection as the condition for input capture is the detection of edges in the signal produced by taking the logical OR of the signals on the main and additional pins. For details, refer to (4),

Cascaded Operation Example (c). For input capture in cascade connection, refer to section 12.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 12.45 shows the TICCR setting and input capture input pins.

Table 12.45 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pins
Input capture from TCNT_1 to TGRA_1	I2AE bit = 0 (initial value)	TIOC1A
	I2AE bit = 1	TIOC1A, TIOC2A
Input capture from TCNT_1 to TGRB_1	I2BE bit = 0 (initial value)	TIOC1B
	I2BE bit = 1	TIOC1B, TIOC2B
Input capture from TCNT_2 to TGRA_2	I1AE bit = 0 (initial value)	TIOC2A
	I1AE bit = 1	TIOC2A, TIOC1A
Input capture from TCNT_2 to TGRB_2	I1BE bit = 0 (initial value)	TIOC2B
	I1BE bit = 1	TIOC2B, TIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 12.20 shows an example of the setting procedure for cascaded operation.

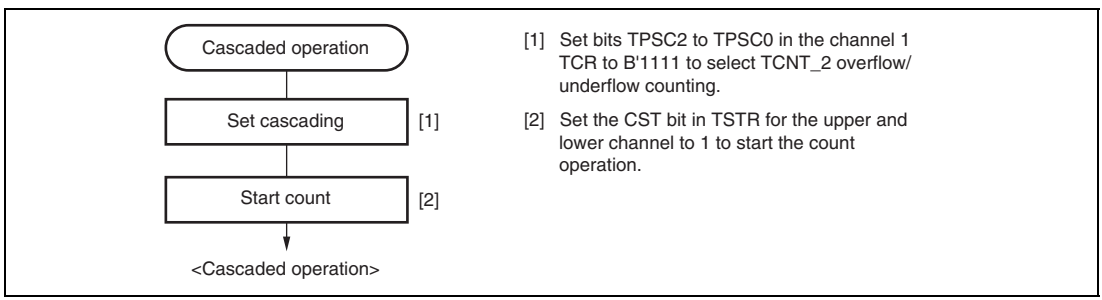


Figure 12.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 12.21 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

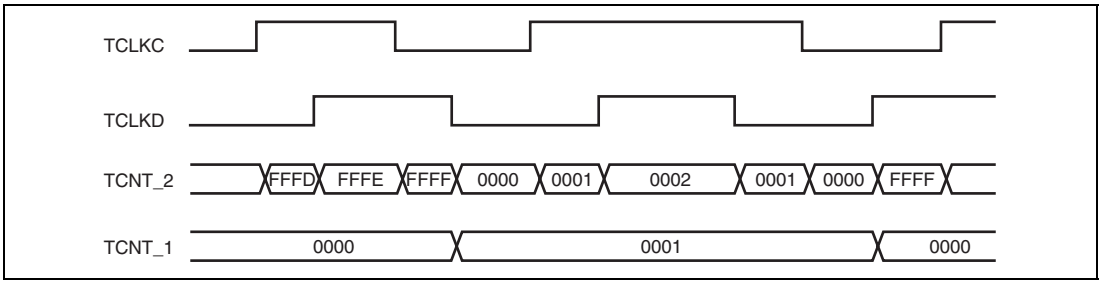


Figure 12.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 12.22 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected the TIOC1A rising edge for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both TIOC1A and TIOC2A is used for the TGRA_1 input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edge is used.

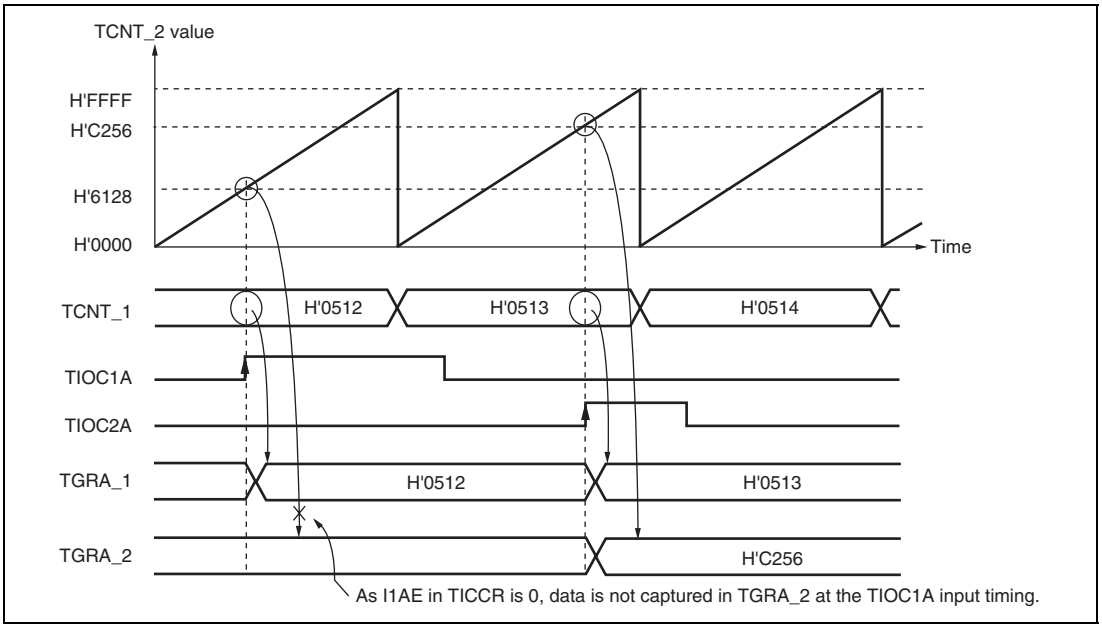
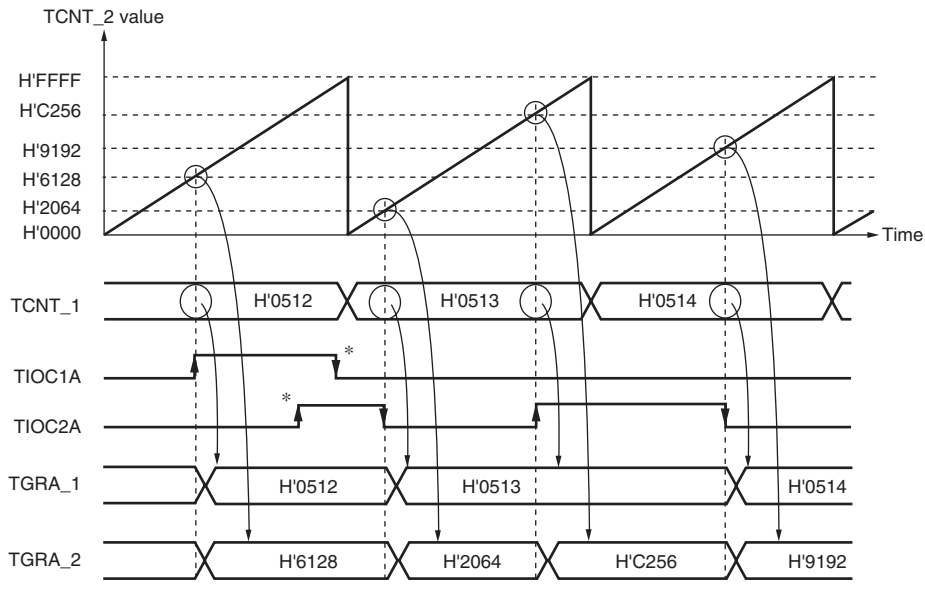


Figure 12.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 12.23 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the TIOC2A and TIOC1A pins in the TGRA_1 and TGRA_2 input capture conditions, respectively. In this example, the IOA0 to IOA3 bits in both TIOR_1 and TIOR_2 have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of TIOC1A and TIOC2A input is used for the TGRA_1 and TGRA_2 input capture conditions.



Note: * When either of the input pins is at the high level, an edge on the other input pin does not act as an input capture condition.

Figure 12.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 12.24 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected TGRA_0 compare match or input capture occurrence for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, as TIOR_1 has selected TGRA_0 compare match or input capture occurrence for the input capture timing, the TIOC2A edge is not used for TGRA_1 input capture condition although the I2AE bit in TICCR has been set to 1.

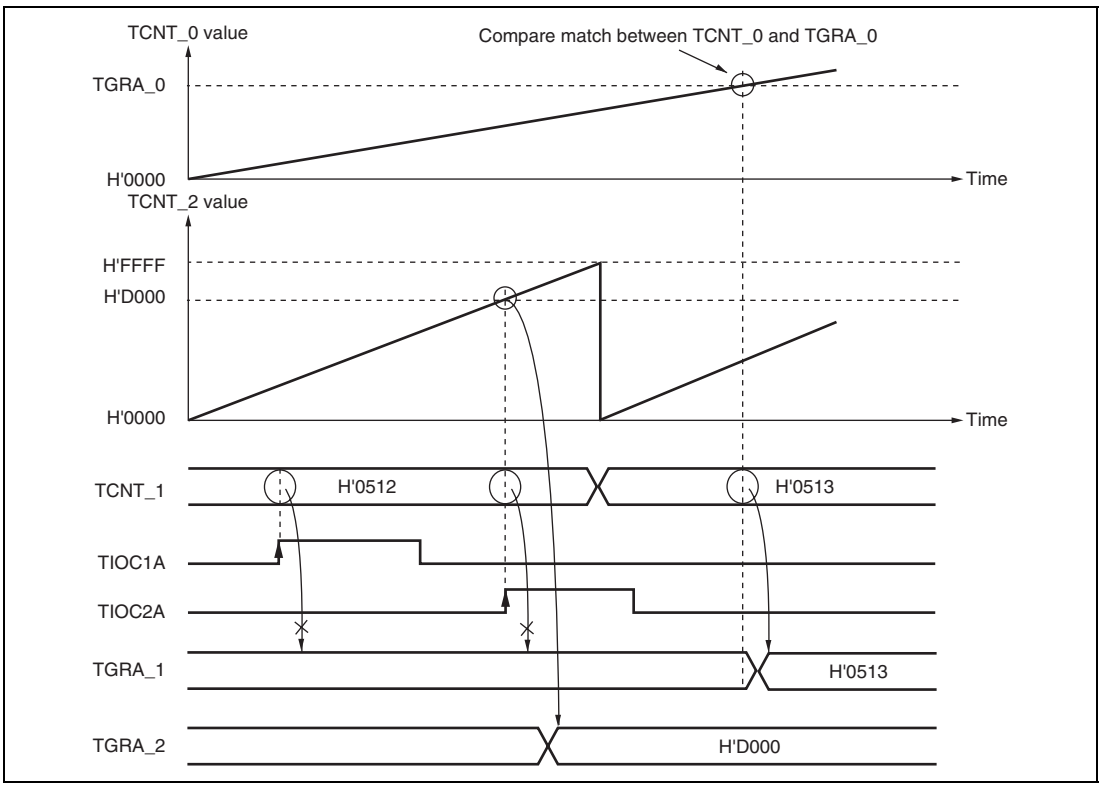


Figure 12.24 Cascaded Operation Example (d)

12.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by the cycle register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 8-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 12.46.

Table 12.46 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOC0A	TIOC0A
	TGRB_0		TIOC0B
	TGRC_0	TIOC0C	TIOC0C
	TGRD_0		TIOC0D
1	TGRA_1	TIOC1A	TIOC1A
	TGRB_1		TIOC1B
2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	Cannot be set
	TGRB_3		Cannot be set
	TGRC_3	TIOC3C	Cannot be set
	TGRD_3		Cannot be set
4	TGRA_4	TIOC4A	Cannot be set
	TGRB_4		Cannot be set
	TGRC_4	TIOC4C	Cannot be set
	TGRD_4		Cannot be set

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

(1) Example of PWM Mode Setting Procedure

Figure 12.25 shows an example of the PWM mode setting procedure.

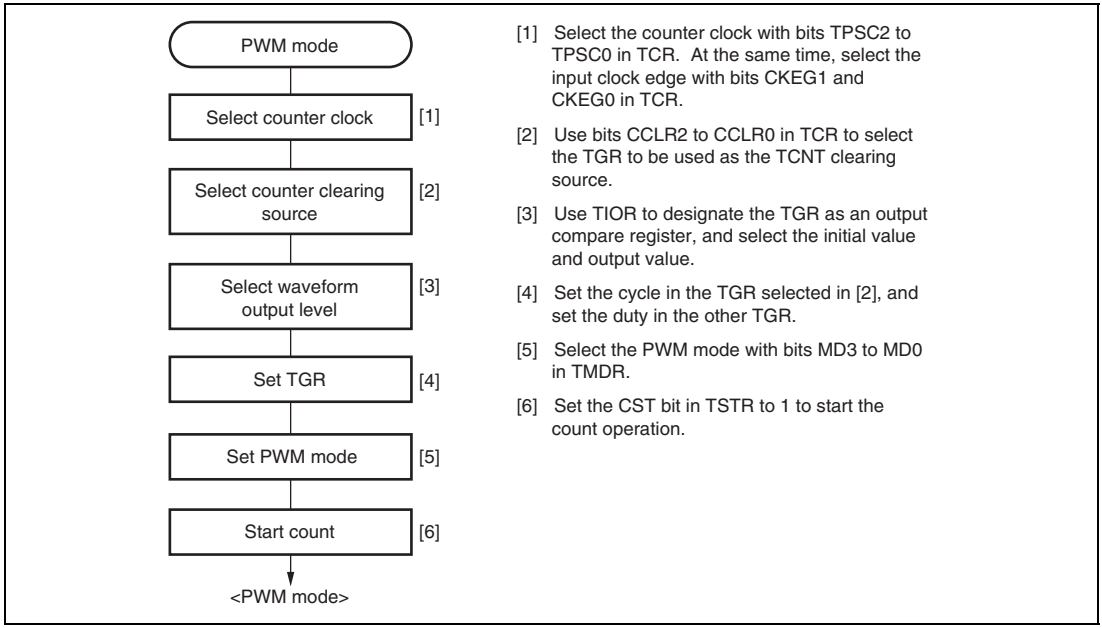


Figure 12.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 12.26 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty levels.

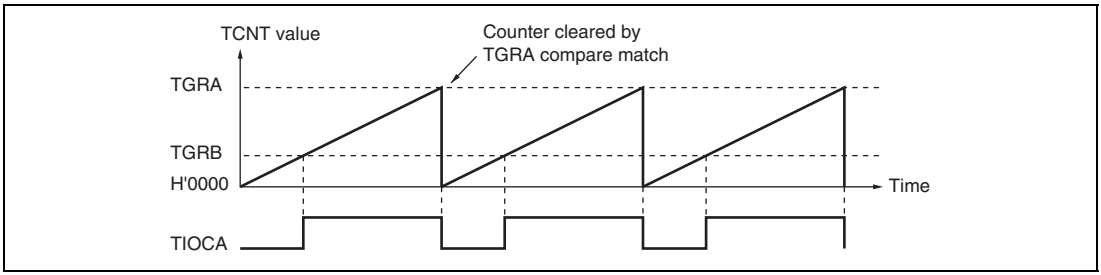


Figure 12.26 Example of PWM Mode Operation (1)

Figure 12.27 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

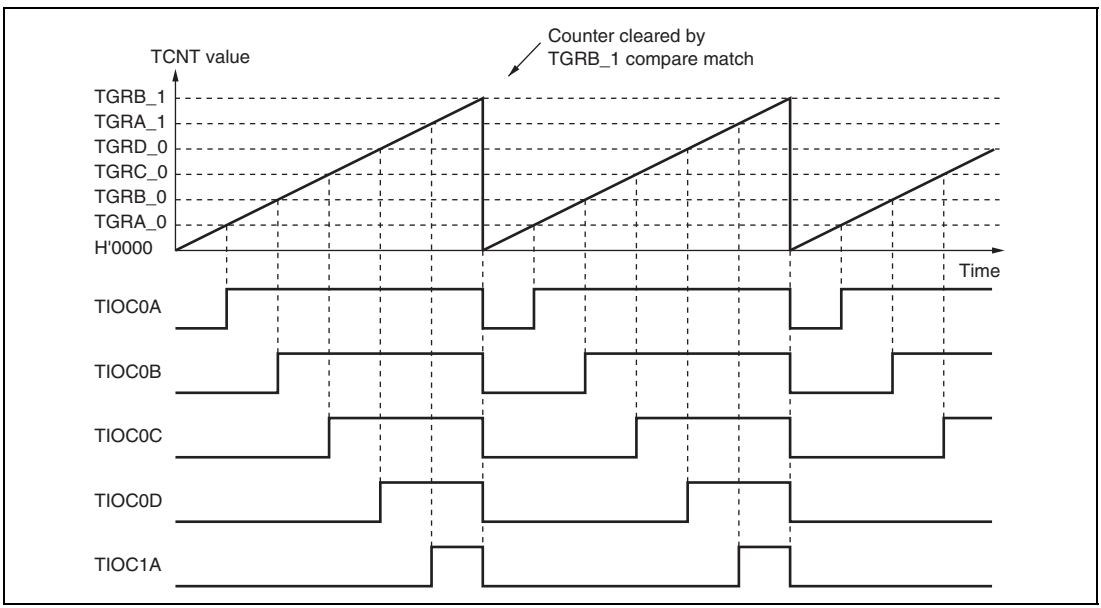


Figure 12.27 Example of PWM Mode Operation (2)

Figure 12.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

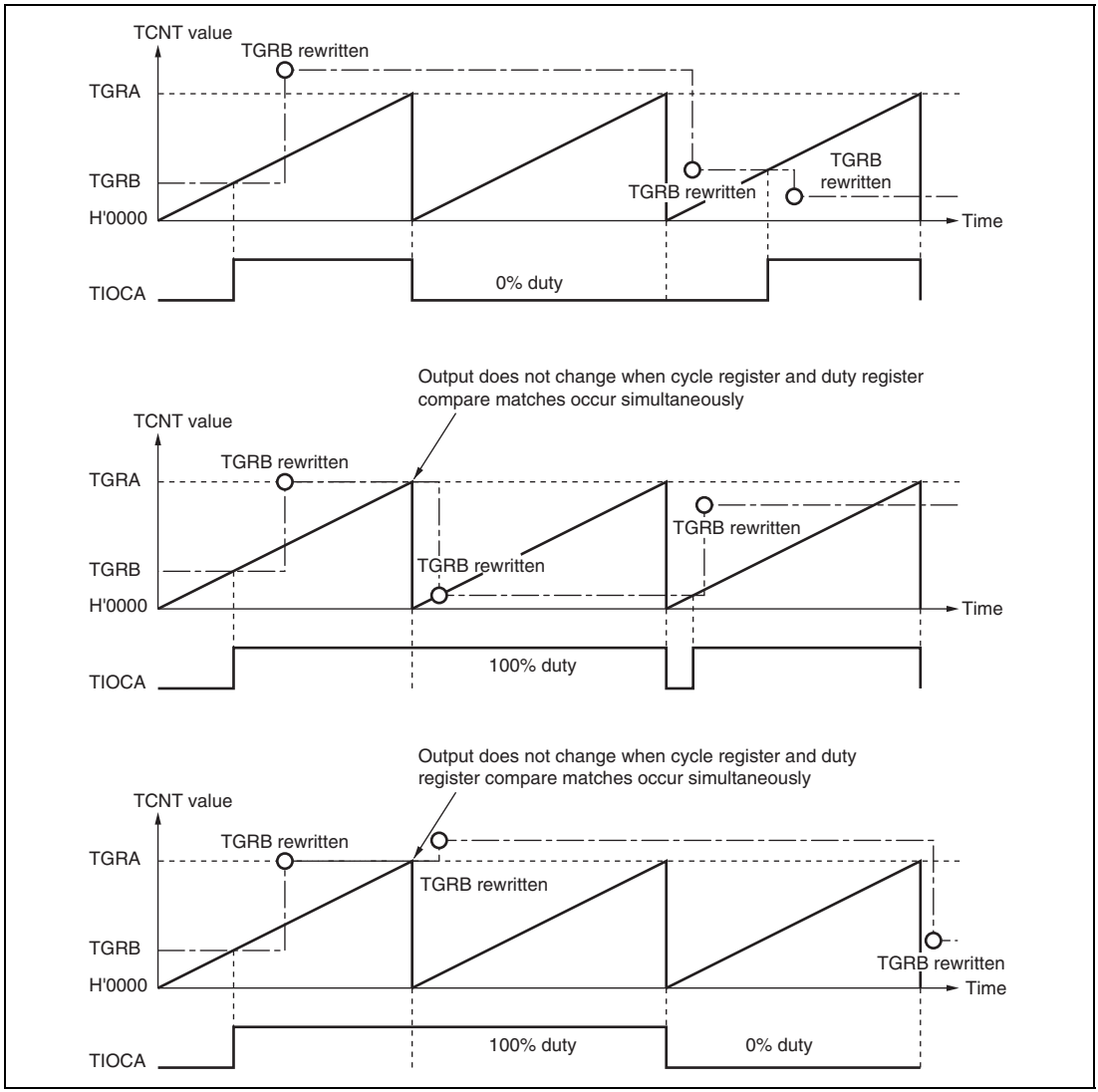


Figure 12.28 Example of PWM Mode Operation (3)

12.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 in TCR, and of TIOR, TIER, and TGR, are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 12.47 shows the correspondence between external clock pins and channels.

Table 12.47 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 12.29 shows an example of the phase counting mode setting procedure.

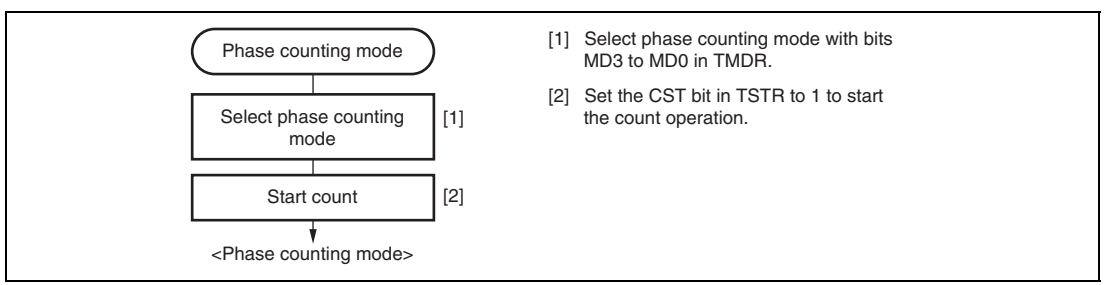


Figure 12.29 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes according to the count conditions.

(a) Phase counting mode 1

Figure 12.30 shows an example of phase counting mode 1 operation, and table 12.48 summarizes the TCNT up/down-count conditions.

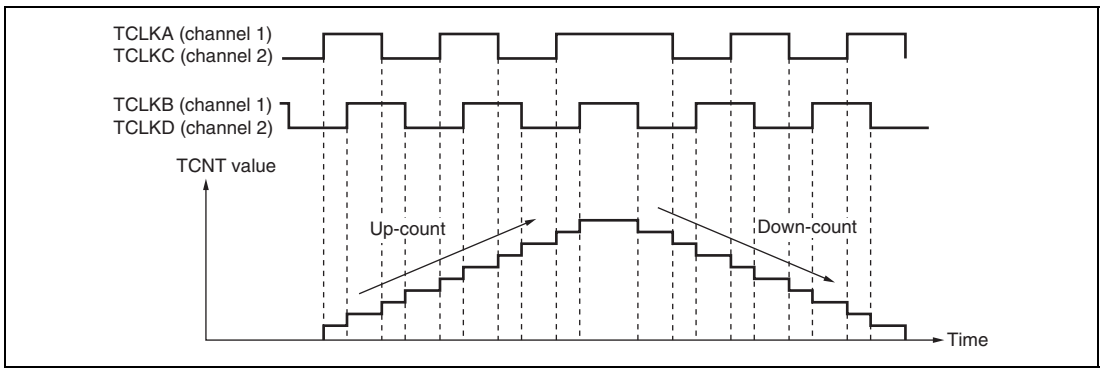


Figure 12.30 Example of Phase Counting Mode 1 Operation

Table 12.48 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]
↑: Rising edge
↓: Falling edge

(b) Phase counting mode 2

Figure 12.31 shows an example of phase counting mode 2 operation, and table 12.49 summarizes the TCNT up/down-count conditions.

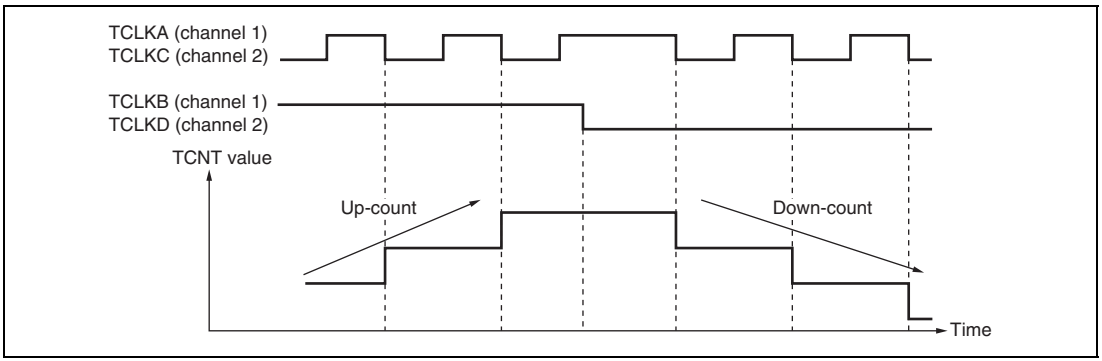


Figure 12.31 Example of Phase Counting Mode 2 Operation

Table 12.49 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌜	Don't care
Low level	⌞	Don't care
⌜	Low level	Don't care
⌞	High level	Up-count
High level	⌞	Don't care
Low level	⌜	Don't care
⌜	High level	Don't care
⌞	Low level	Down-count

[Legend]

⌜: Rising edge

⌞: Falling edge

(c) Phase counting mode 3

Figure 12.32 shows an example of phase counting mode 3 operation, and table 12.50 summarizes the TCNT up/down-count conditions.

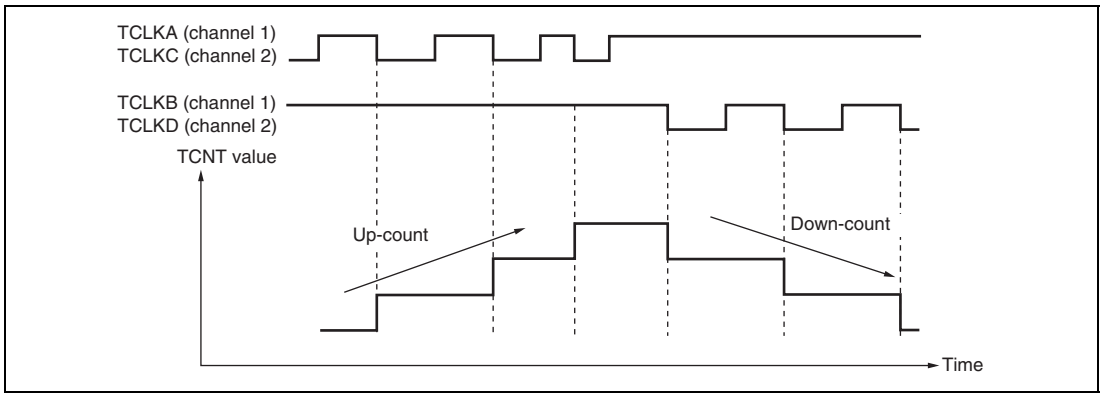


Figure 12.32 Example of Phase Counting Mode 3 Operation

Table 12.50 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]
↑: Rising edge
↓: Falling edge

(d) Phase counting mode 4

Figure 12.33 shows an example of phase counting mode 4 operation, and table 12.51 summarizes the TCNT up/down-count conditions.

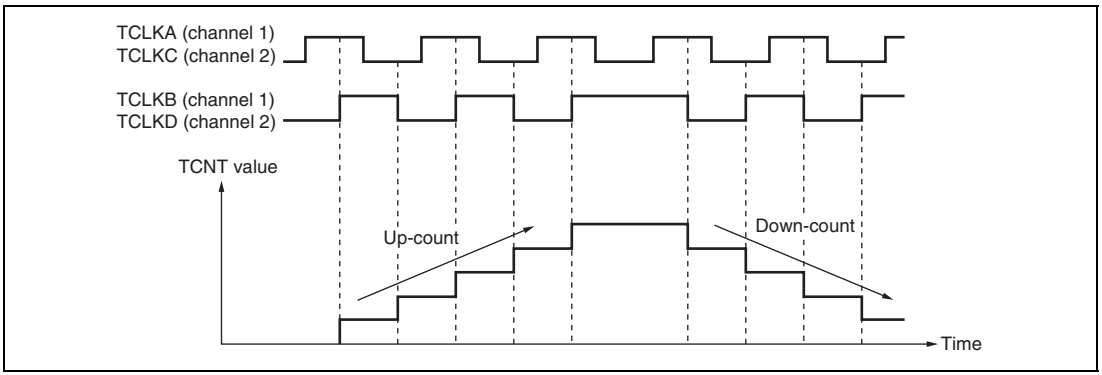


Figure 12.33 Example of Phase Counting Mode 4 Operation

Table 12.51 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑: Rising edge
↓: Falling edge

(3) Phase Counting Mode Application Example

Figure 12.34 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

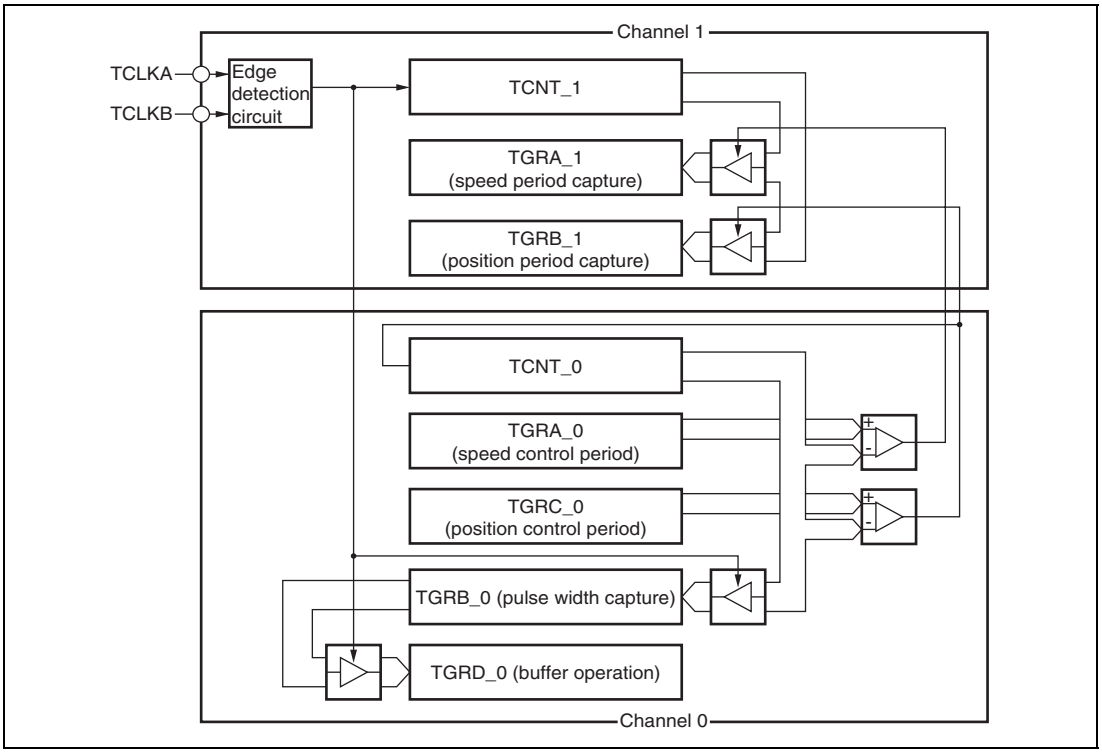


Figure 12.34 Phase Counting Mode Application Example

12.4.7 Reset-Synchronized PWM Mode

In reset-synchronized PWM mode, three-phase output of positive and negative PWM waveforms that share a common wave transition point can be obtained by combining channels 3 and 4.

When set for reset-synchronized PWM mode, the TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, and TIOC4D pins function as PWM output pins and TCNT3 functions as an upcounter.

Table 12.52 shows the PWM output pins used. Table 12.53 shows the settings of the registers.

Table 12.52 Output Pins for Reset-Synchronized PWM Mode

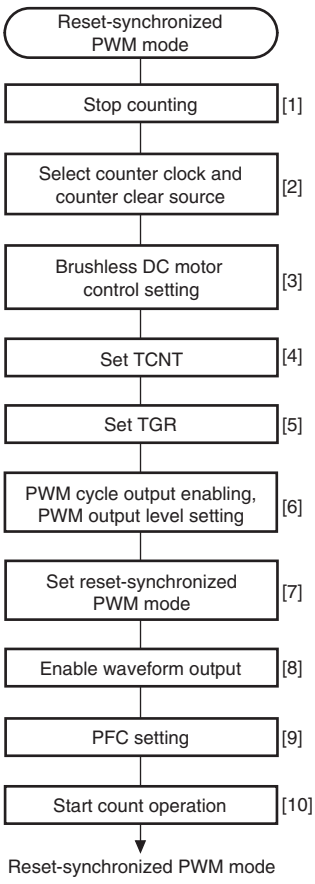
Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 12.53 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGRB_4	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins

(1) Procedure for Selecting the Reset-Synchronized PWM Mode

Figure 12.35 shows an example of procedure for selecting reset-synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in the TSTR to 0 to halt the counting of TCNT. The reset-synchronized PWM mode must be set up while TCNT_3 and TCNT_4 are halted.

[2] Set bits TPSC2-TPSC0 and CKEG1 and CKEG0 in the TCR_3 to select the counter clock and clock edge for channel 3. Set bits CCLR2-CCLR0 in the TCR_3 to select TGRA compare-match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Reset TCNT_3 and TCNT_4 to H'0000.

[5] TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.
$X \leq$ TGRA_3 (X: set value).

[6] Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 12.3.

[7] Set bits MD3-MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. Do not set to TMDR_4.

[8] Set the enabling/disabling of the PWM waveform output pin in TOER.

[9] Set the port control register and the port I/O register.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

Figure 12.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Reset-Synchronized PWM Mode Operation

Figure 12.36 shows an example of operation in reset-synchronized PWM mode. TCNT_3 and TCNT_4 operate as upcounters. The counter is cleared when a TCNT_3 and TGRA_3 compare-match occurs, and then begins incrementing from H'0000. The PWM output pin output toggles with each occurrence of a TGRB_3, TGRA_4, TGRB_4 compare-match, and upon counter clears.

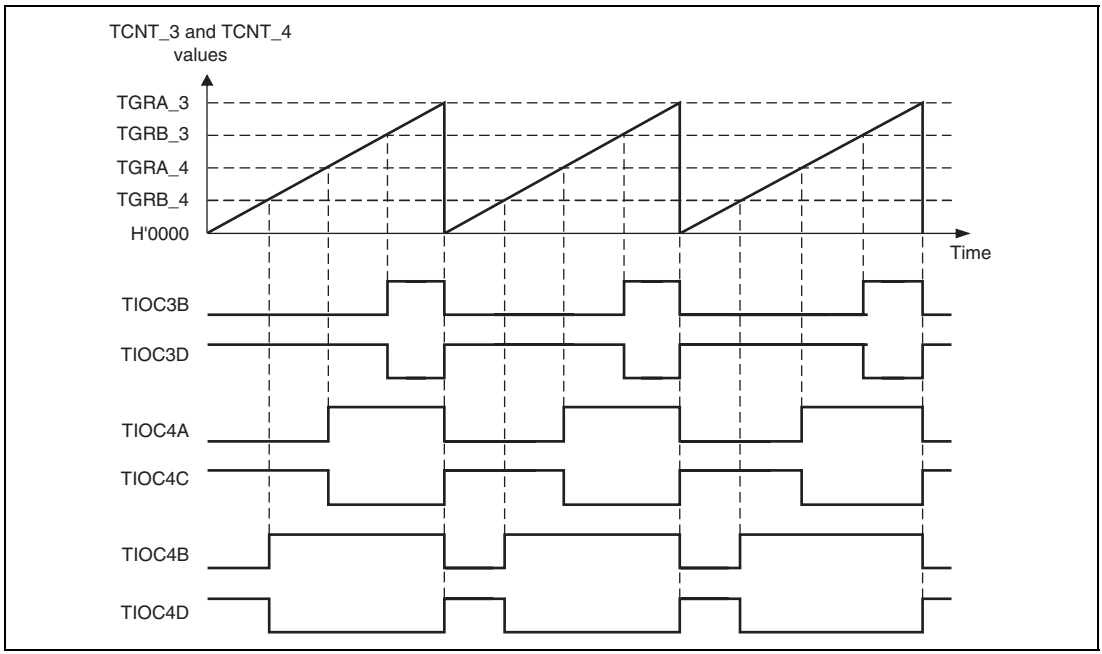


Figure 12.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

12.4.8 Complementary PWM Mode

In complementary PWM mode, three-phase output of non-overlapping positive and negative PWM waveforms can be obtained by combining channels 3 and 4. PWM waveforms without non-overlapping interval are also available.

In complementary PWM mode, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins function as PWM output pins, the TIOC3A pin can be set for toggle output synchronized with the PWM period. TCNT_3 and TCNT_4 function as up/down counters.

Table 12.54 shows the PWM output pins used. Table 12.55 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 12.54 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O port)
	TIOC3B	PWM output pin 1
	TIOC3C	I/O port*
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
4	TIOC4A	PWM output pin 2
	TIOC4B	PWM output pin 3
	TIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	TIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note: * Avoid setting the TIOC3C pin as a timer I/O pin in complementary PWM mode.

Table 12.55 Register Settings for Complementary PWM Mode

Channel	Counter/Register	Description	Read/Write from CPU
3	TCNT_3	Start of up-count from value set in dead time register	Maskable by TRWER setting*
	TGRA_3	Set TCNT_3 upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*
	TGRB_3	PWM output 1 compare register	Maskable by TRWER setting*
	TGRC_3	TGRA_3 buffer register	Always readable/writable
	TGRD_3	PWM output 1/TGRB_3 buffer register	Always readable/writable
4	TCNT_4	Up-count start, initialized to H'0000	Maskable by TRWER setting*
	TGRA_4	PWM output 2 compare register	Maskable by TRWER setting*
	TGRB_4	PWM output 3 compare register	Maskable by TRWER setting*
	TGRC_4	PWM output 2/TGRA_4 buffer register	Always readable/writable
	TGRD_4	PWM output 3/TGRB_4 buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set TCNT_4 and TCNT_3 offset value (dead time value)	Maskable by TRWER setting*
Timer cycle data register (TCDR)		Set TCNT_4 upper limit value (1/2 carrier cycle)	Maskable by TRWER setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGRB_3 temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/TGRA_4 temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/TGRB_4 temporary register	Not readable/writable

Note: * Access can be enabled or disabled according to the setting of bit 0 (RWE) in TRWER (timer read/write enable register).

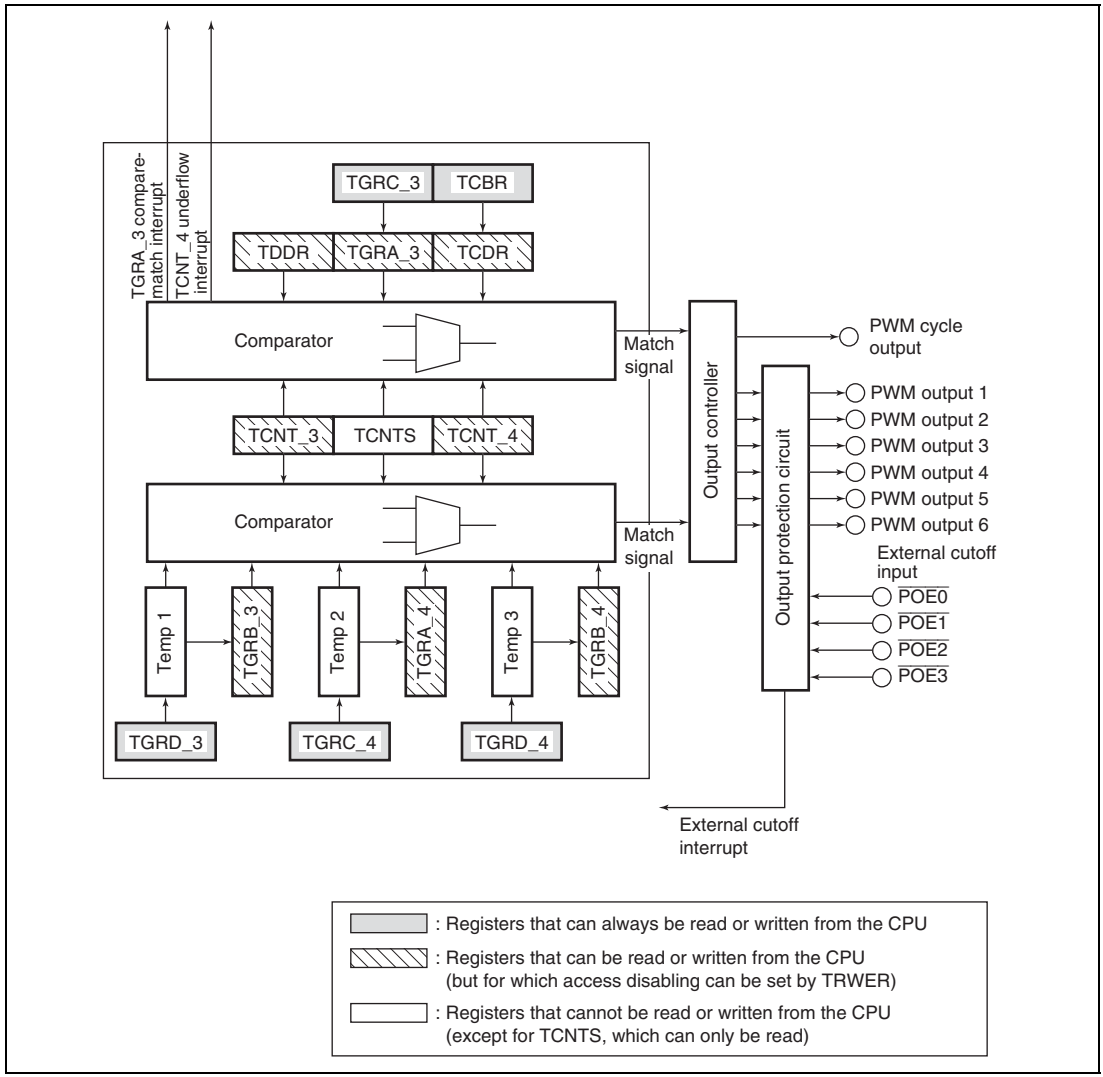
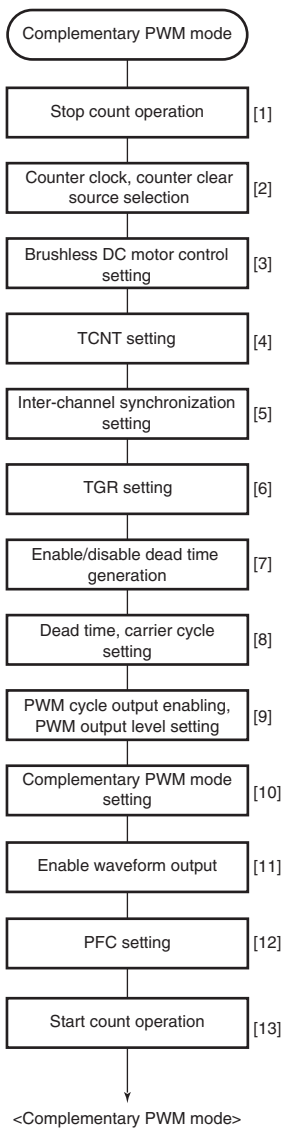


Figure 12.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 12.38.



[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0, and halt timer counter (TCNT) operation. Perform complementary PWM mode setting when TCNT_3 and TCNT_4 are stopped.

[2] Set the same counter clock and clock edge for channels 3 and 4 with bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2 to CCLR0 to set synchronous clearing only when restarting by a synchronous clear from another channel during complementary PWM mode operation.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in TCNT_3. Set TCNT_4 to H'0000.

[5] Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).

[6] Set the output PWM duty in the duty registers (TGRB_3, TGRA_4, TGRB_4) and buffer registers (TGRD_3, TGRC_4, TGRD_4). Set the same initial value in each corresponding TGR.

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[8] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the timer cycle data register (TCDR) and timer cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in TGRA_3 and TGRC_3.

[9] Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 12.3.

[10] Select complementary PWM mode in timer mode register 3 (TMDR_3). Do not set in TMDR_4.

[11] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[12] Set the port control register and the port I/O register.

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

Figure 12.38 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 12.39 illustrates counter operation in complementary PWM mode, and figure 12.40 shows an example of complementary PWM mode operation.

(a) Counter Operation

In complementary PWM mode, three counters—TCNT_3, TCNT_4, and TCNTS—perform up/down-count operations.

TCNT_3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switches to down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

TCNT_4 is initialized to H'0000.

When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It need not be initialized.

When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGRA_3, it is cleared to H'0000.

When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGRA_3.

TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

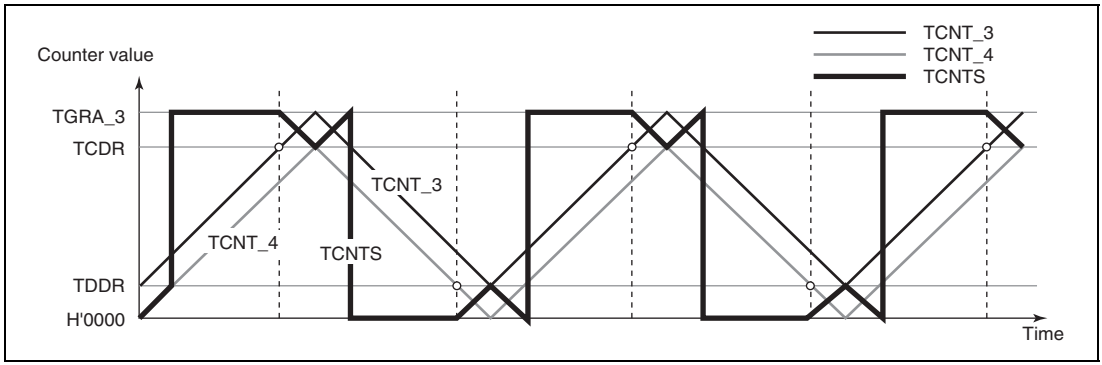


Figure 12.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 12.40 shows an example of complementary PWM mode operation.

The registers which are constantly compared with the counters to perform PWM output are TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches TGRA_3 when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3 to MD0 in the timer mode register (TMDR). Figure 12.40 shows an example in which the mode is selected in which the change is made in the trough.

In the Tb interval (Tb2 in figure 12.40) in which data transfer to the temporary register is not performed, the temporary register has the same function as the compare register, and is compared with the counter. In this interval, therefore, there are two compare match registers for one-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters—TCNT_3, TCNT_4, and TCNTS—and two registers—compare register and temporary register—are compared, and PWM output controlled accordingly.

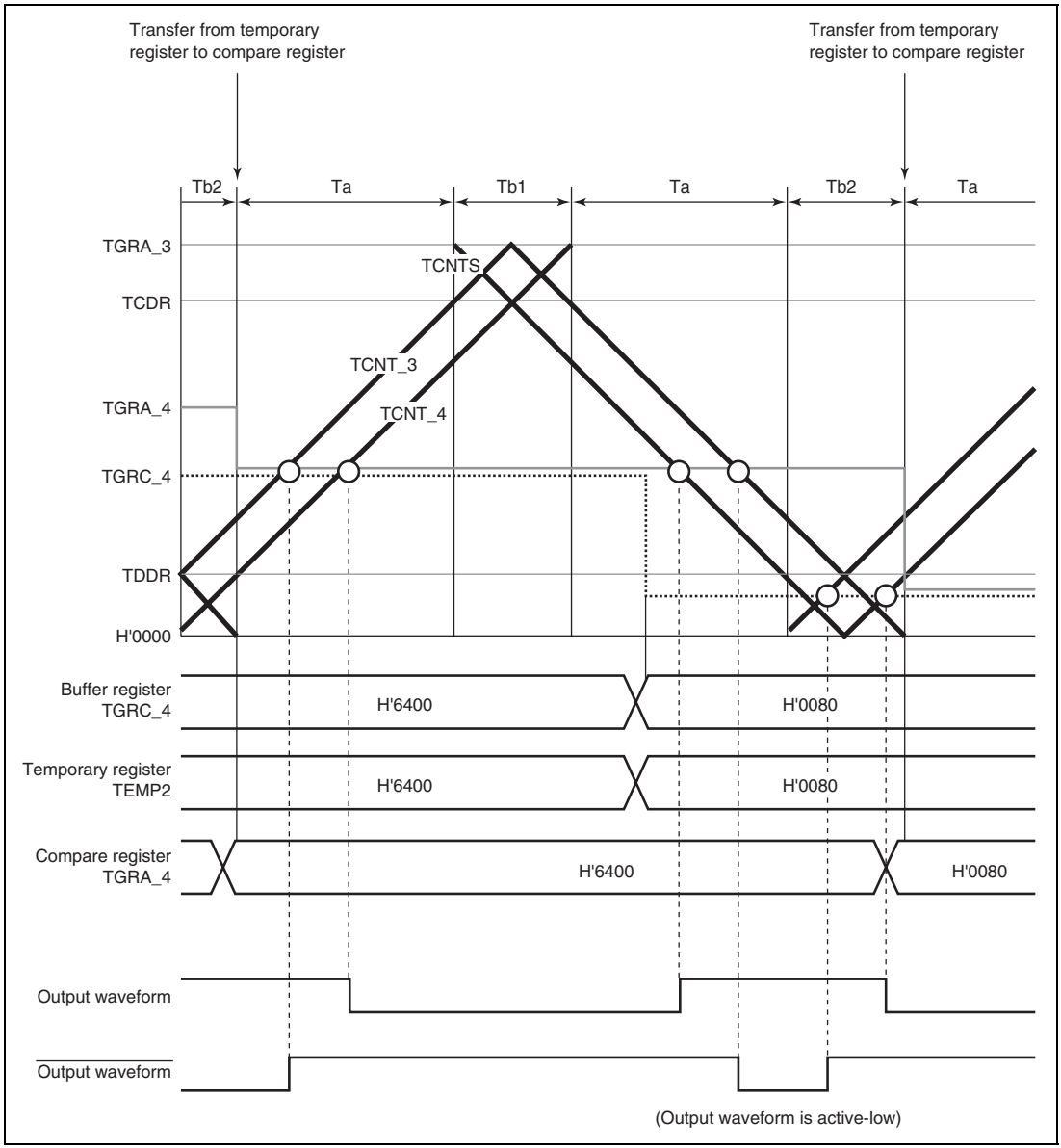


Figure 12.40 Example of Complementary PWM Mode Operation

(c) Initialization

In complementary PWM mode, there are six registers that must be initialized. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

Before setting complementary PWM mode with bits MD3 to MD0 in the timer mode register (TMDR), the following initial register values must be set.

TGRC_3 operates as the buffer register for TGRA_3, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.

Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TGRD_4.

The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

Set TCNT_4 to H'0000 before setting complementary PWM mode.

Table 12.56 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM carrier cycle
TGRD_3, TGRC_4, TGRD_4	Initial PWM duty value for each phase
TCNT_4	H'0000

Note: The TGRC_3 set value must be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC_3 must be set to 1/2 the PWM carrier cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).

The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the TCNT_3 counter start value, and creates non-overlap between TCNT_3 and TCNT_4. Complementary PWM mode should be cleared before changing the contents of TDDR.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after reading TDER = 1.

TGRA_3 and TGRC_3 should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 12.41 shows an example of operation without dead time.

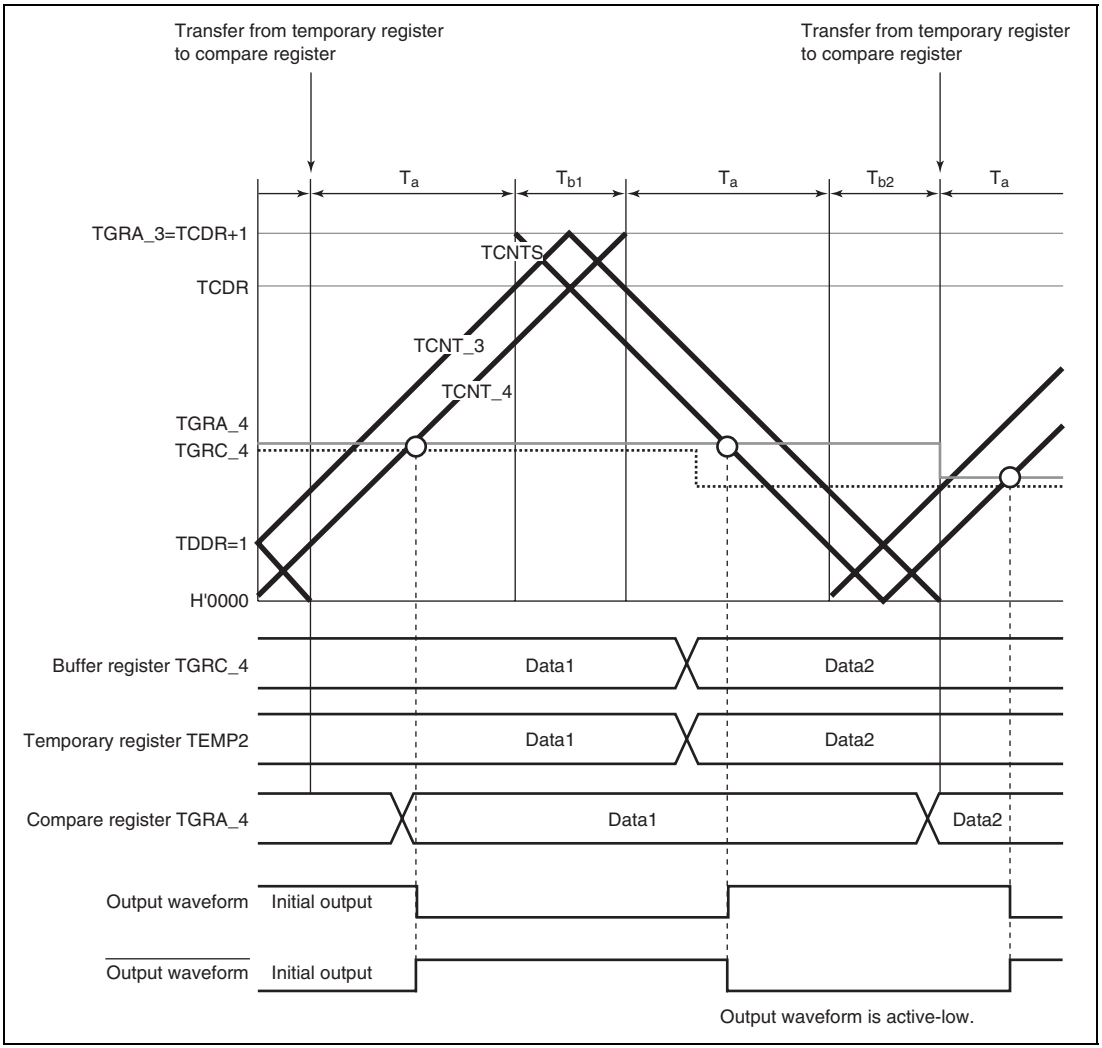


Figure 12.41 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA_3, in which the TCNT_3 upper limit value is set, and TCDR, in which the TCNT_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: TGRA_3 set value = TCDR set value + TDDR set value
Without dead time: TGRA_3 set value = TCDR set value + 1

The TGRA_3 and TCDR settings are made by setting the values in buffer registers TGRC_3 and TCBR. The values set in TGRC_3 and TCBR are transferred simultaneously to TGRA_3 and TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in the timer mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 12.42 illustrates the operation when the PWM cycle is updated at the crest.

See the following section, Register Data Updating, for the method of updating the data in each buffer register.

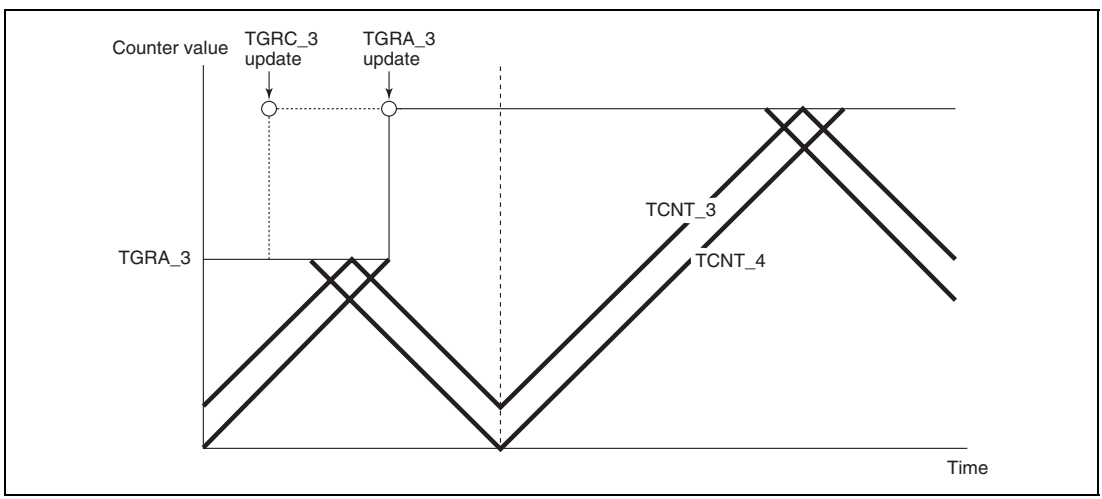


Figure 12.42 Example of PWM Cycle Updating

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD3 to MD0 in the timer mode register (TMDR). Figure 12.43 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

When rewriting buffer register data, a write to TGRD_4 must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

A write to TGRD_4 must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGRD_4 data. In this case, the data written to TGRD_4 should be the same as the data prior to the write operation.

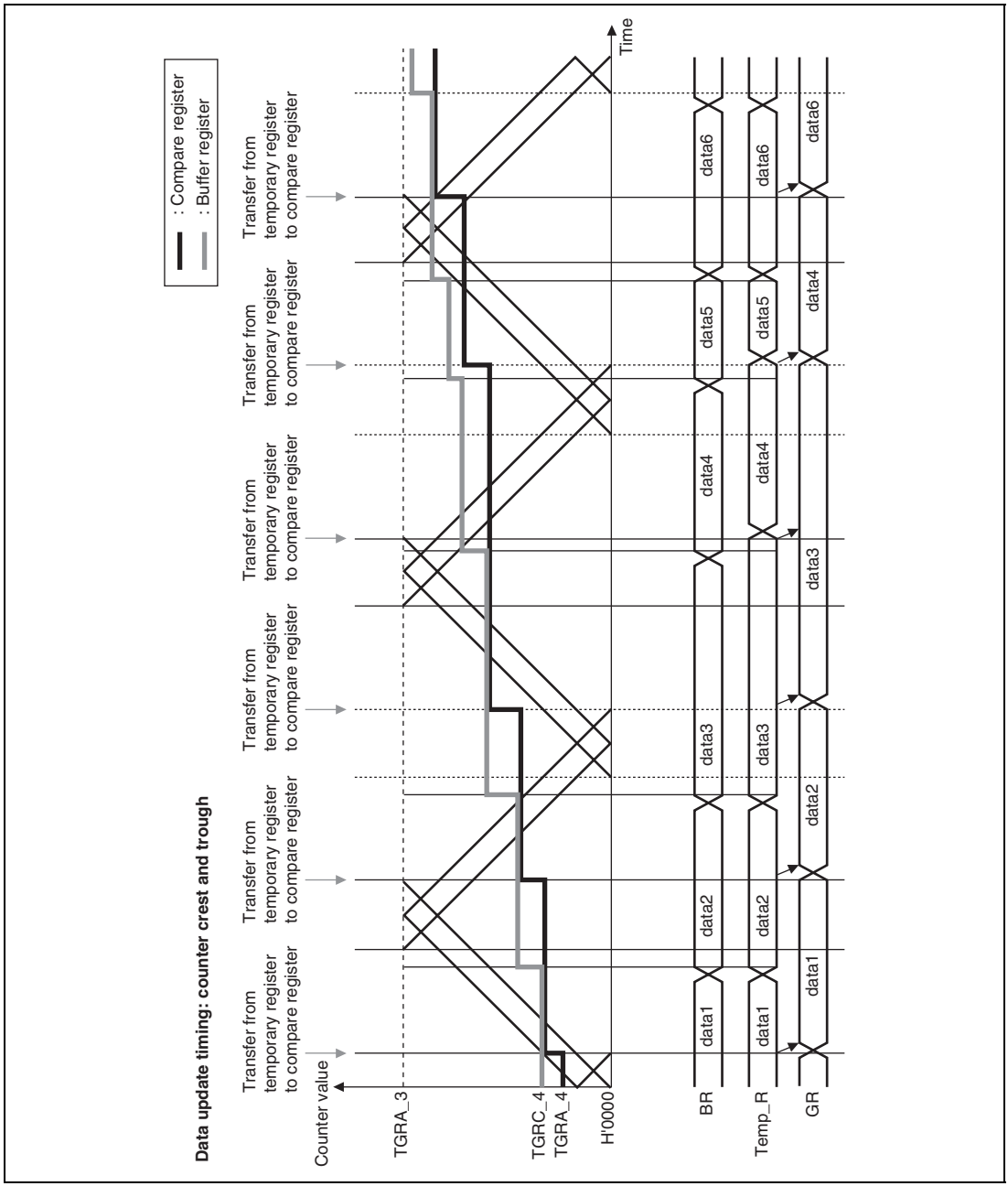


Figure 12.43 Example of Data Update in Complementary PWM Mode

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2).

This initial output is the PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT_4 exceeds the value set in the dead time register (TDDR). Figure 12.44 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in figure 12.45.

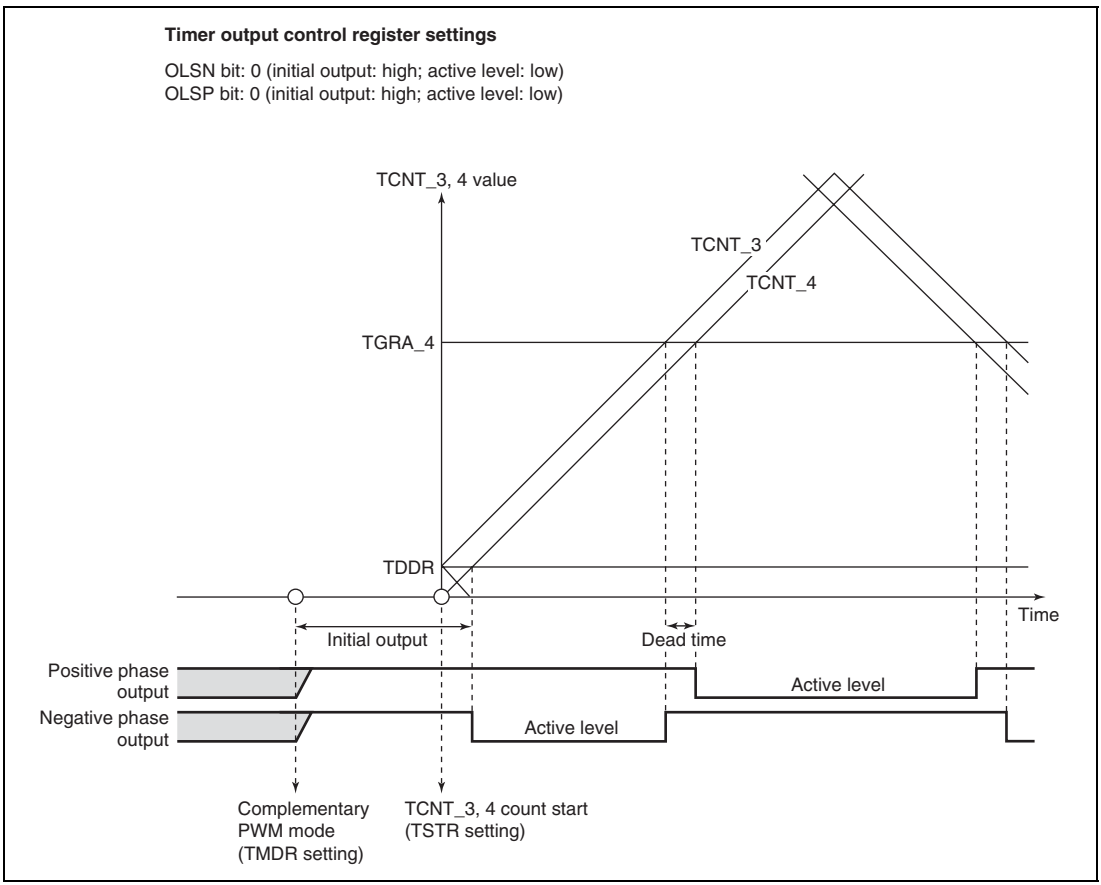


Figure 12.44 Example of Initial Output in Complementary PWM Mode (1)

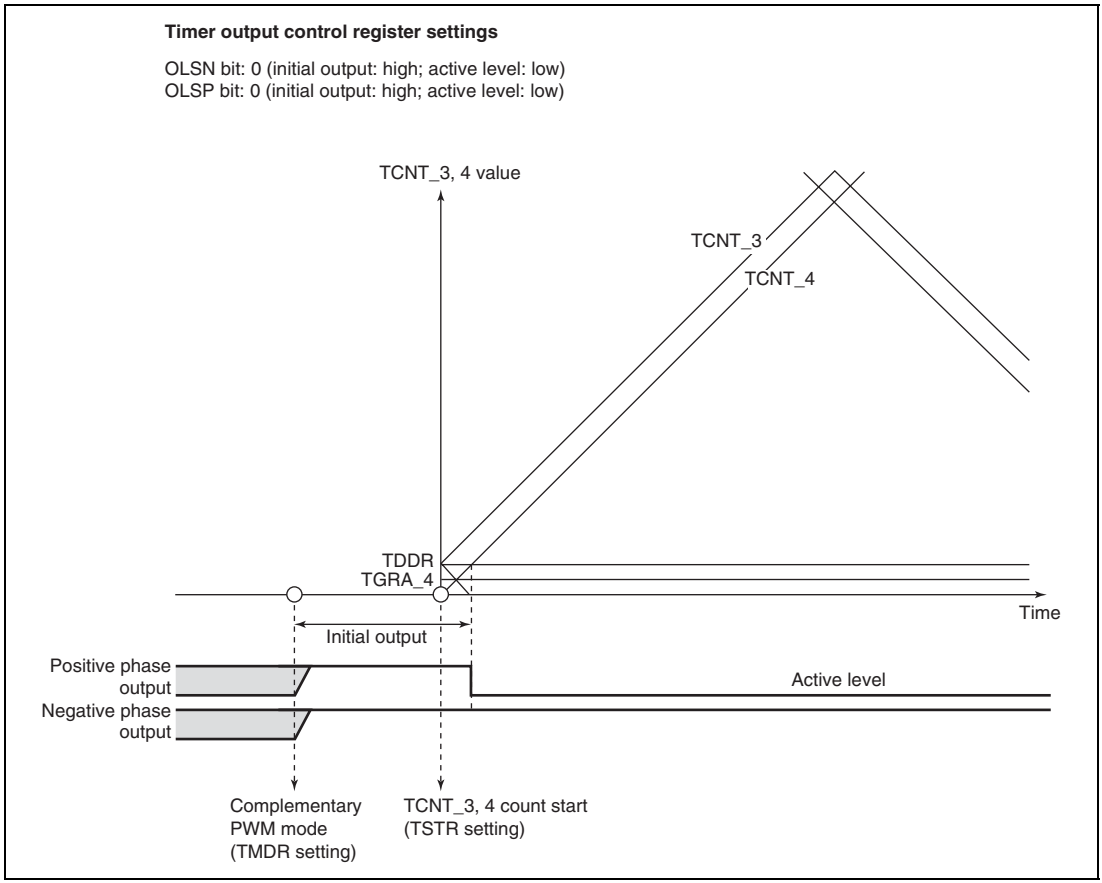


Figure 12.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Complementary PWM Mode PWM Output Generation Method

In complementary PWM mode, 3-phase output is performed of PWM waveforms with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and compare register. While TCNTS is counting, compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of on and off compare-match occurrence may vary, but the compare-match that turns off each phase takes precedence to secure the dead time and ensure that the positive phase and negative phase on times do not overlap. Figures 12.46 to 12.48 show examples of waveform generation in complementary PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating with a delay of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that turns off the negative phase has the highest priority, and compare-matches occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in figure 12.46.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase is not being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in figure 12.47, compare-match **b** is ignored, and the negative phase is turned on by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

Similarly, in the example in figure 12.48, compare-match **a'** with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns off the positive phase, are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.

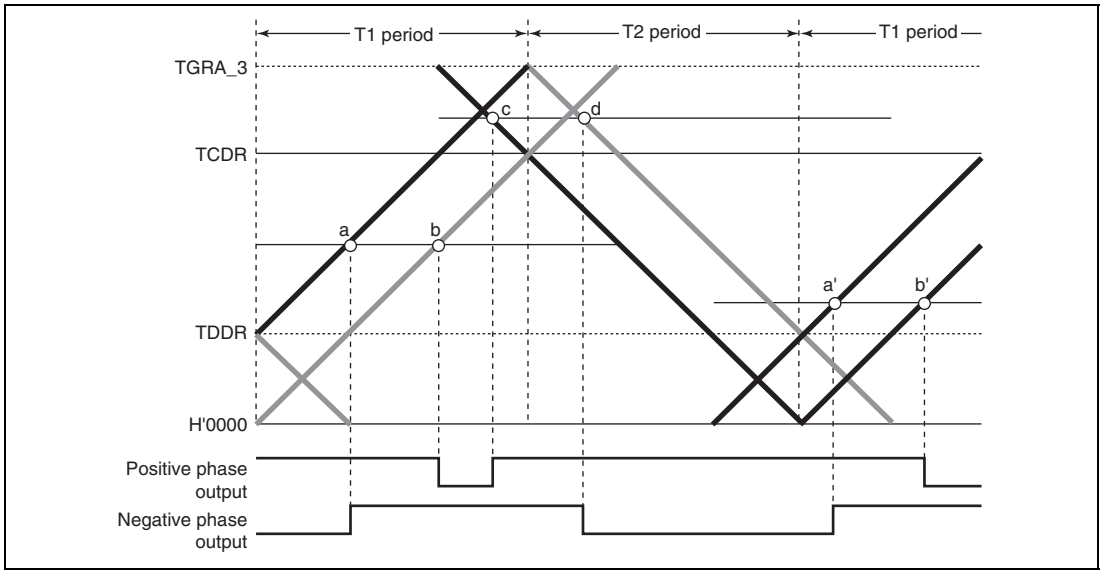


Figure 12.46 Example of Complementary PWM Mode Waveform Output (1)

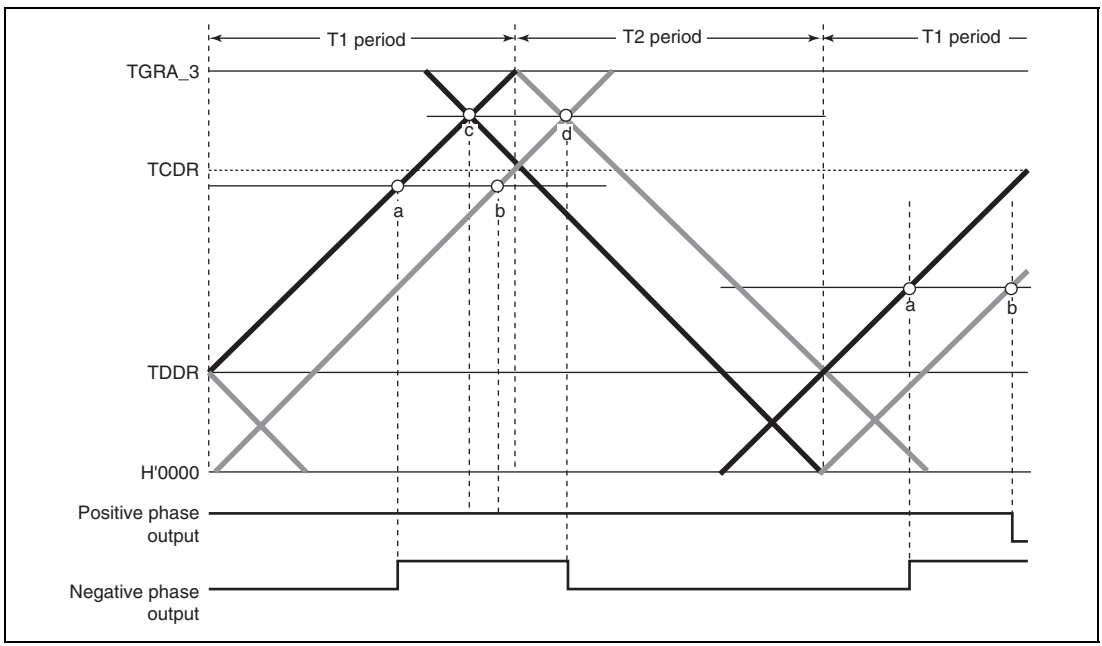


Figure 12.47 Example of Complementary PWM Mode Waveform Output (2)

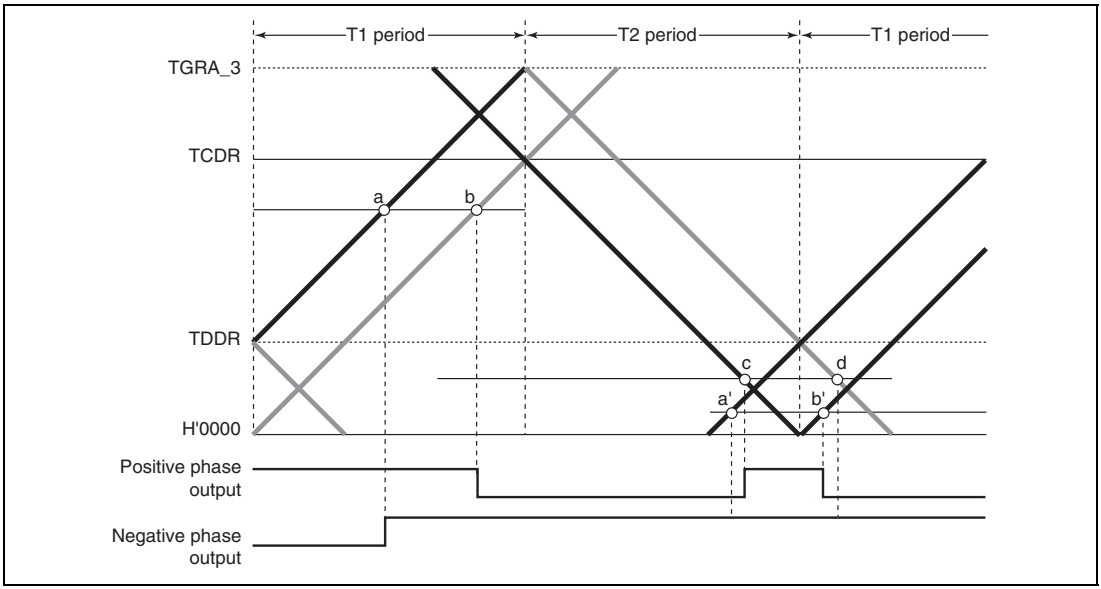


Figure 12.48 Example of Complementary PWM Mode Waveform Output (3)

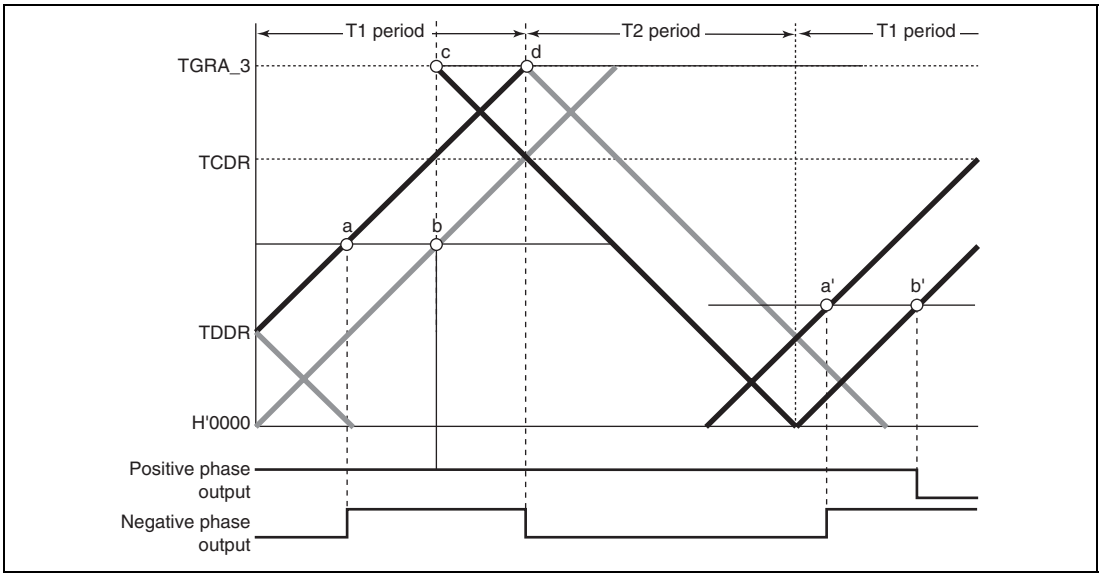


Figure 12.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

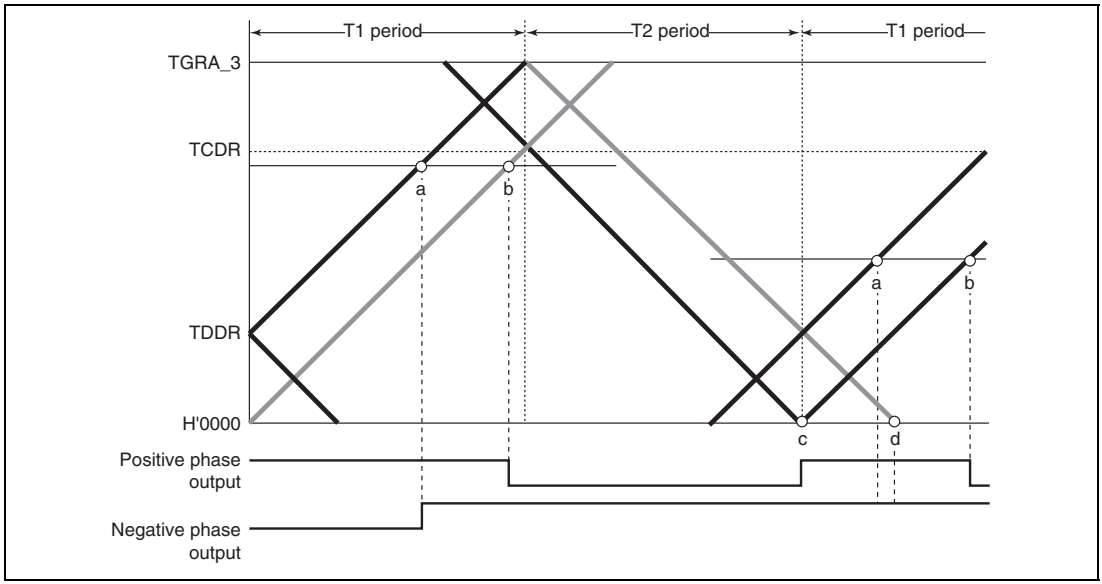


Figure 12.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

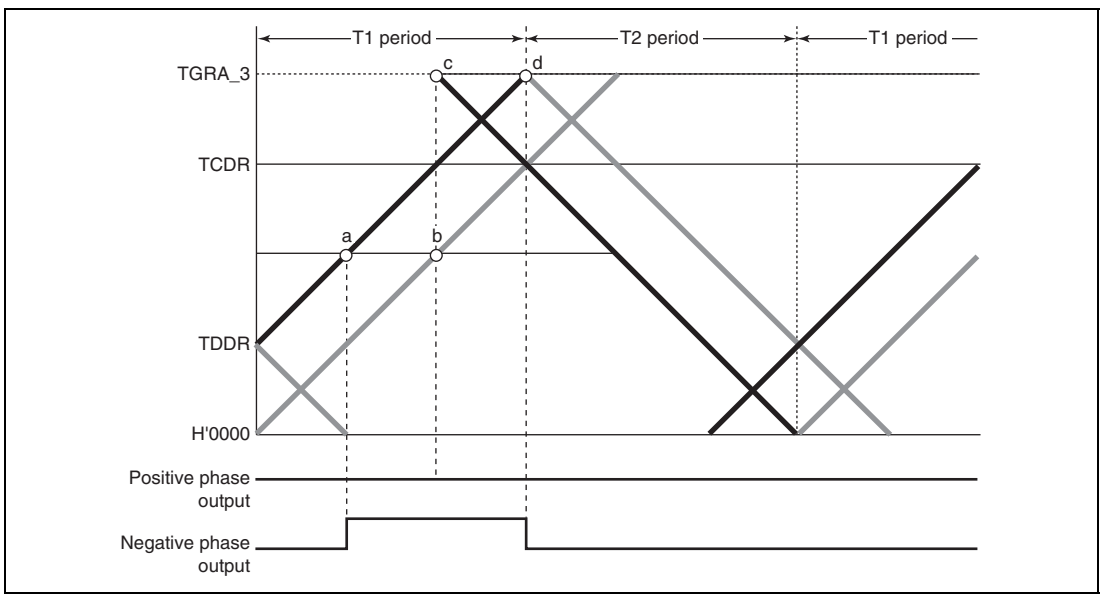


Figure 12.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

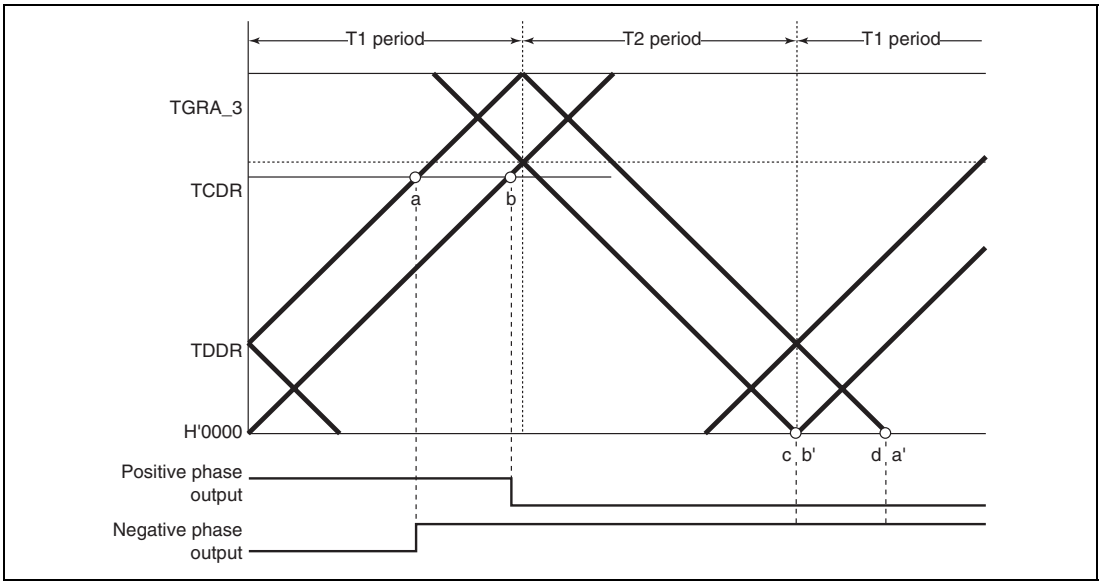


Figure 12.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

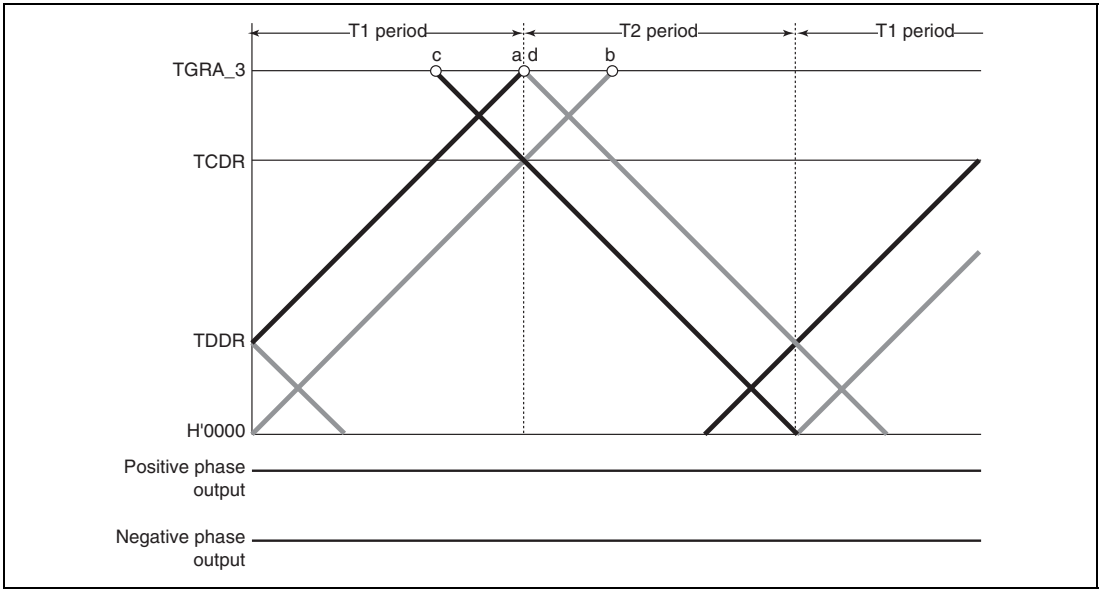


Figure 12.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(k) Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 12.49 to 12.53 show output examples.

100% duty output is performed when the compare register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the compare register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.

On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output can be performed in synchronization with the PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in figure 12.54.

This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

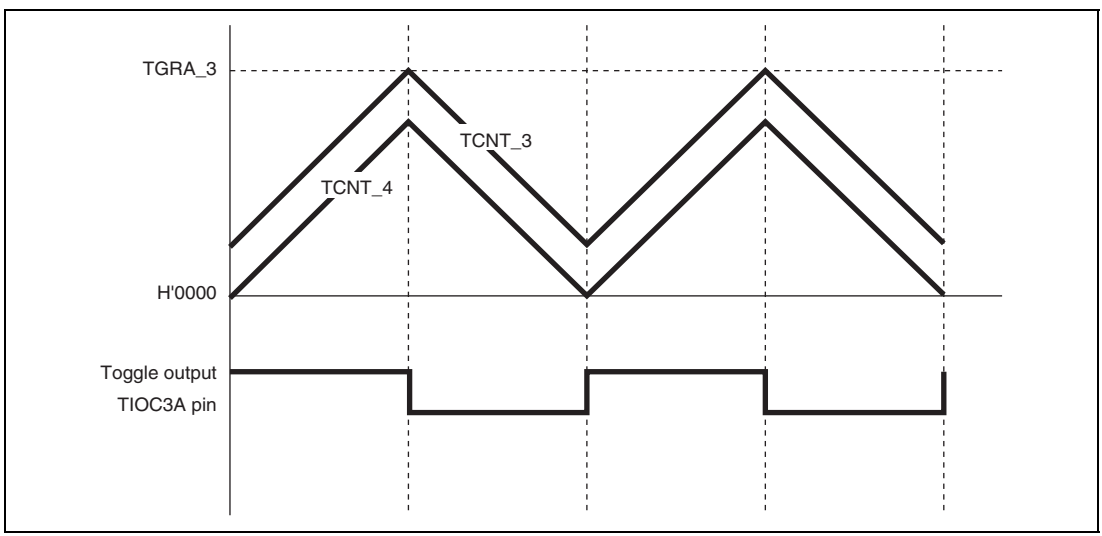


Figure 12.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchronous register (TSYR), and selecting synchronous clearing with bits CCLR2 to CCLR0 in the timer control register (TCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by another channel.

Figure 12.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

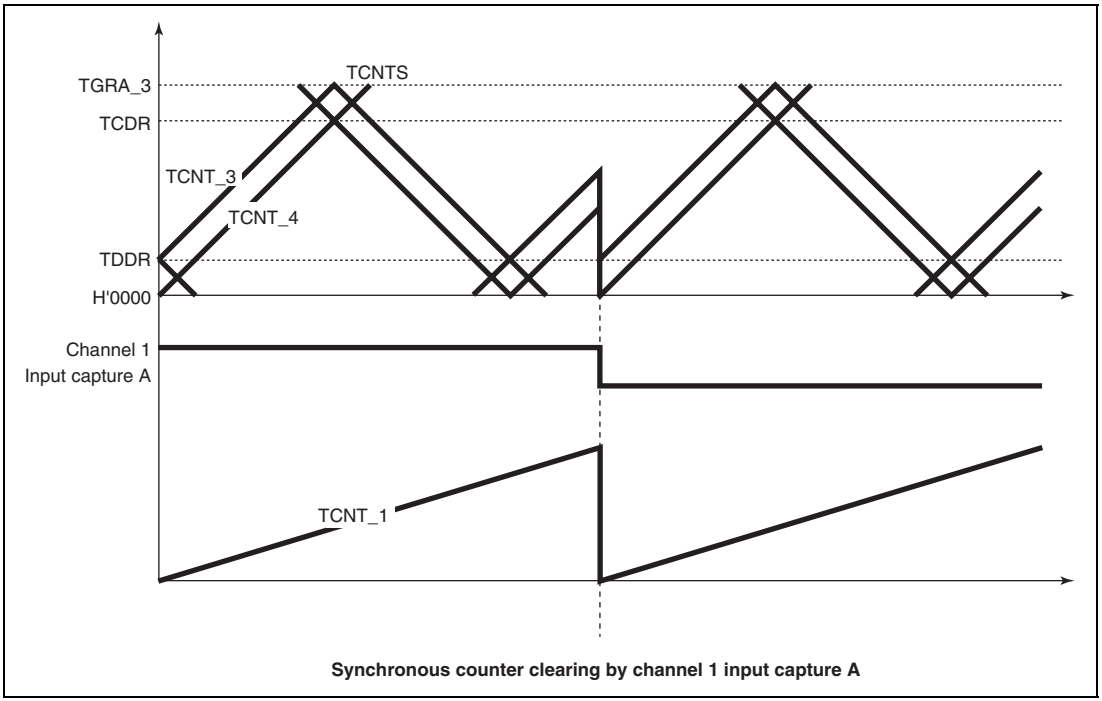


Figure 12.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in figure 12.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLS bits in TOCR is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in figure 12.56) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both the MTU2 and MTU2S. In the MTU2, synchronous clearing generated in channels 0 to 2 in the MTU2 can cause counter clearing in complementary PWM mode; in the MTU2S, compare match or input capture flag setting in channels 0 to 2 in the MTU2 can cause counter clearing.

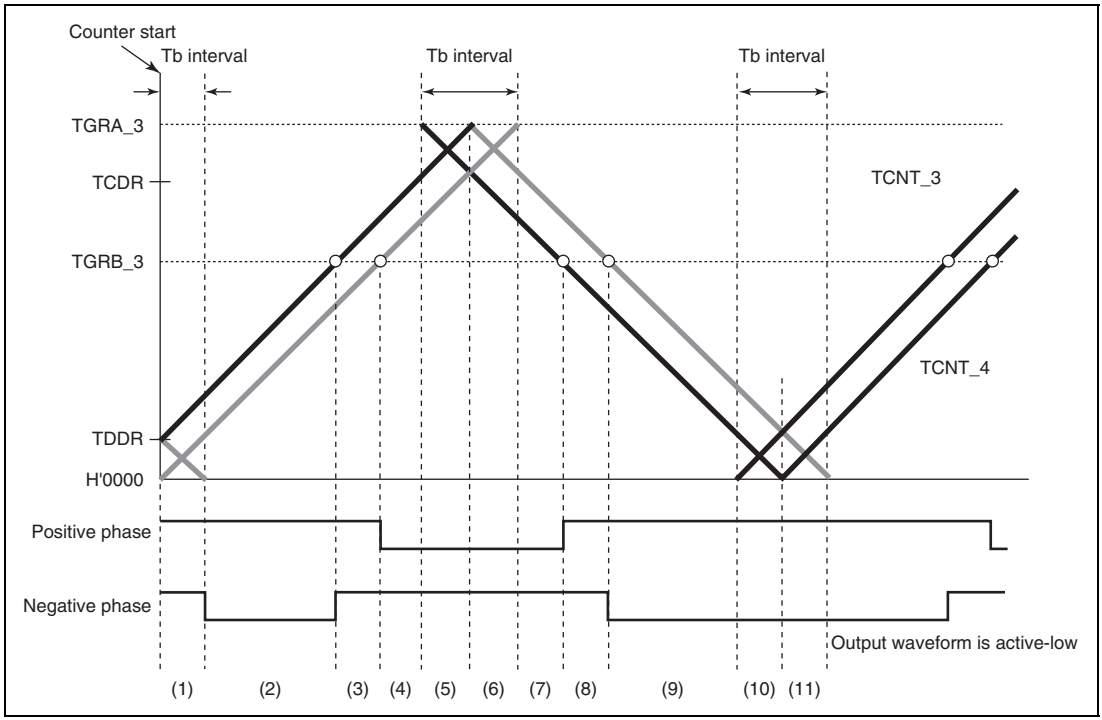


Figure 12.56 Timing for Synchronous Counter Clearing

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in figure 12.57.

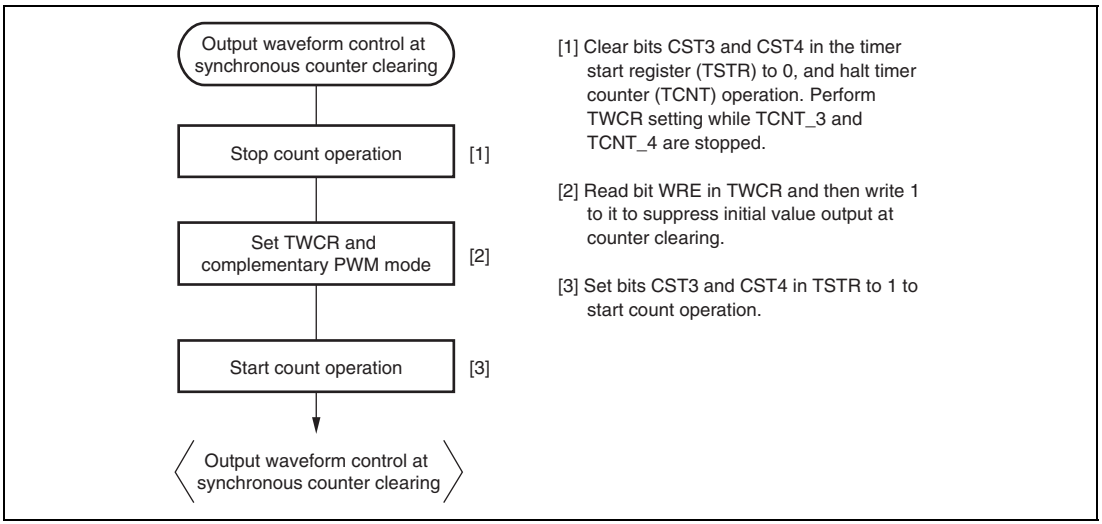


Figure 12.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 12.58 to 12.61 show examples of output waveform control in which the MTU2 operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in figures 12.58 to 12.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 12.56, respectively.

  In the MTU2S, these examples are equivalent to the cases when the MTU2S operates in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is cleared to 0 and the WRE bit is set to 1 in TWCR.

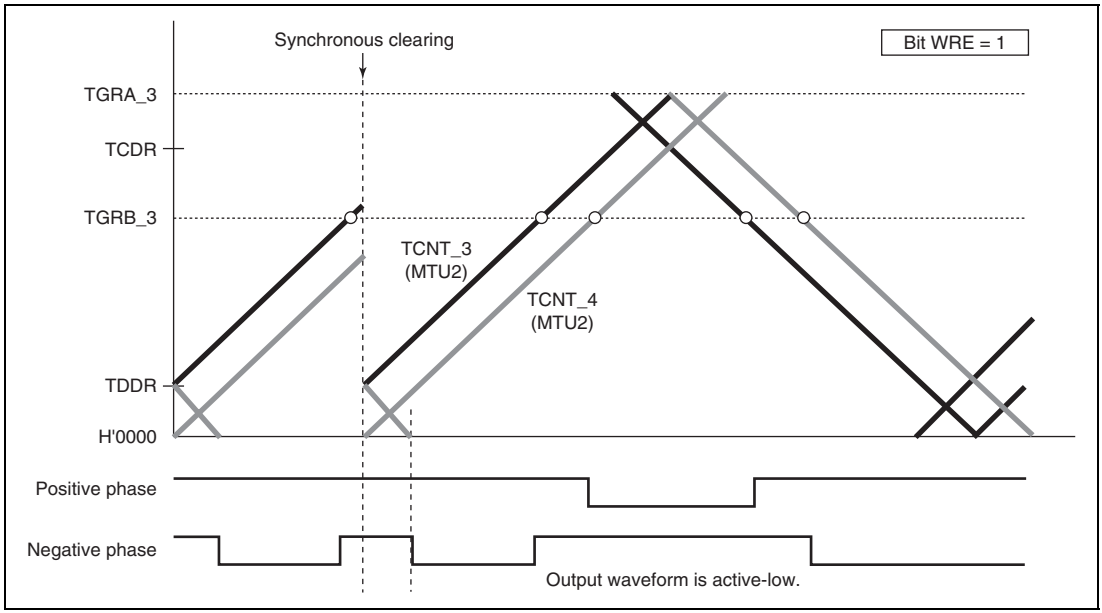


Figure 12.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 12.56; Bit WRE in TWCR of MTU2 is 1)

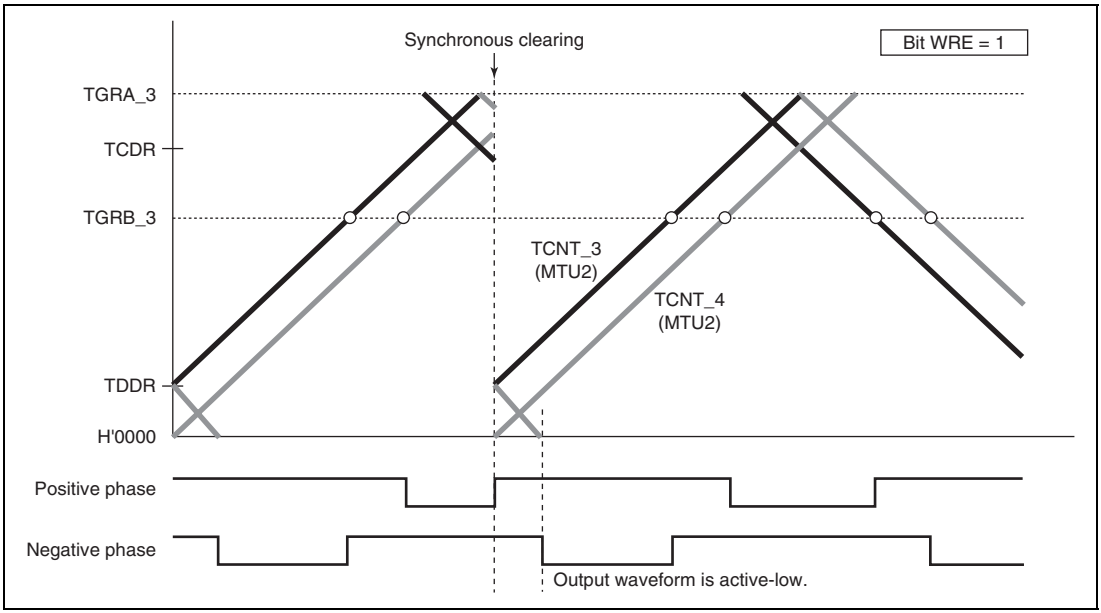


Figure 12.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 12.56; Bit WRE in TWCR of MTU2 is 1)

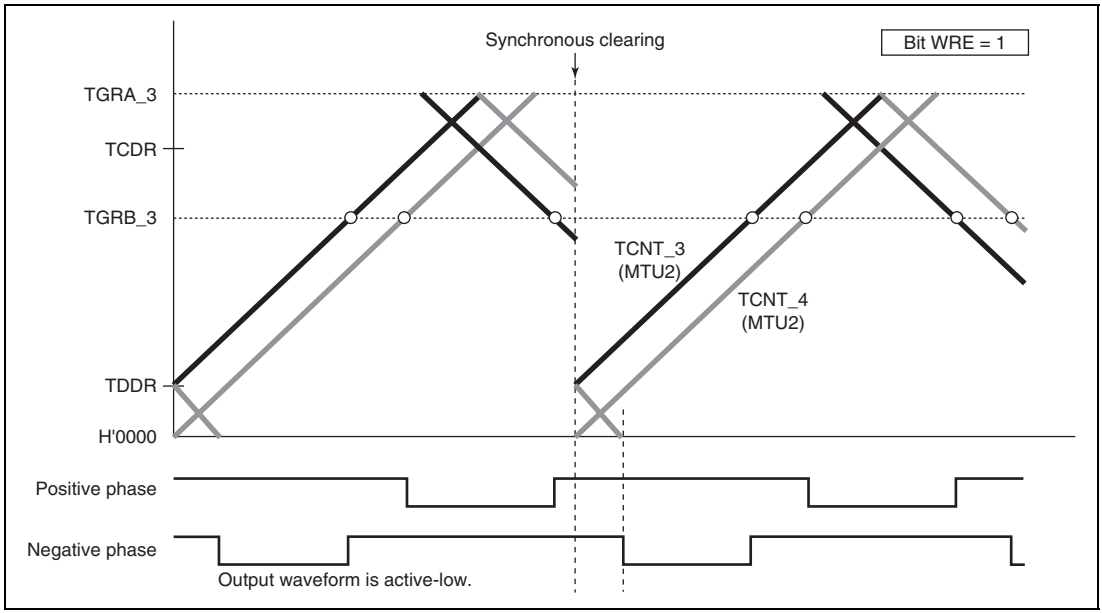


Figure 12.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 12.56; Bit WRE in TWCR is 1)

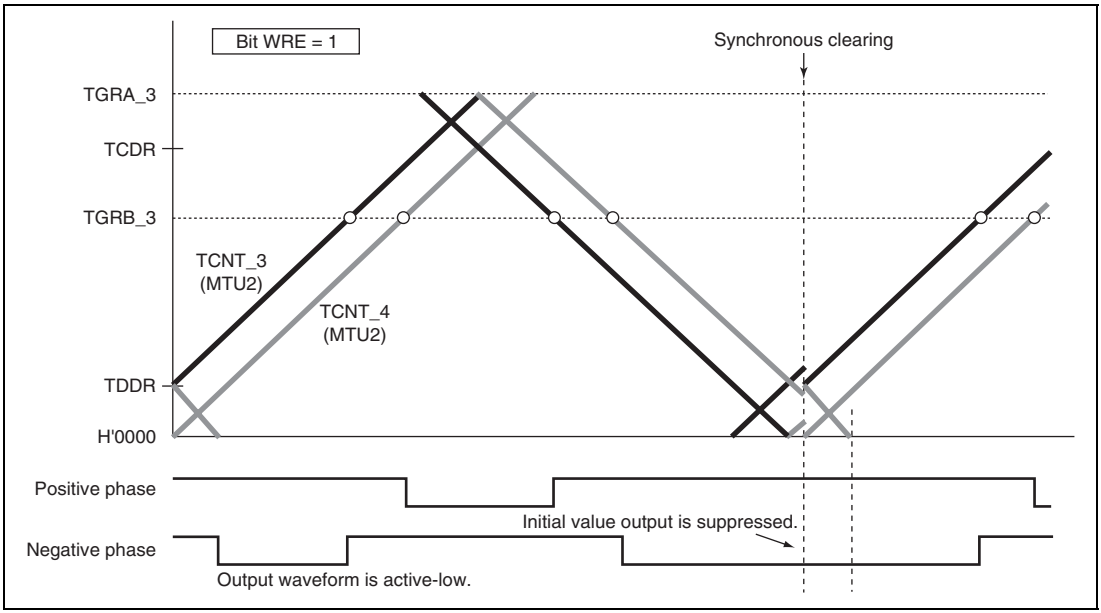


Figure 12.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 12.56; Bit WRE in TWCR is 1)

(o) Suppressing MTU2-MTU2S Synchronous Counter Clearing

In the MTU2S, setting the SCC bit in TWCR to 1 suppresses synchronous counter clearing caused by the MTU2.

Synchronous counter clearing is suppressed only within the interval shown in figure 12.62. When using this function, the MTU2S should be set to complementary PWM mode.

For details of synchronous clearing caused by the MTU2, refer to the description about MTU2S counter clearing caused by MTU2 flag setting source (MTU2-MTU2S synchronous counter clearing) in section 12.4.10, MTU2-MTU2S Synchronous Operation.

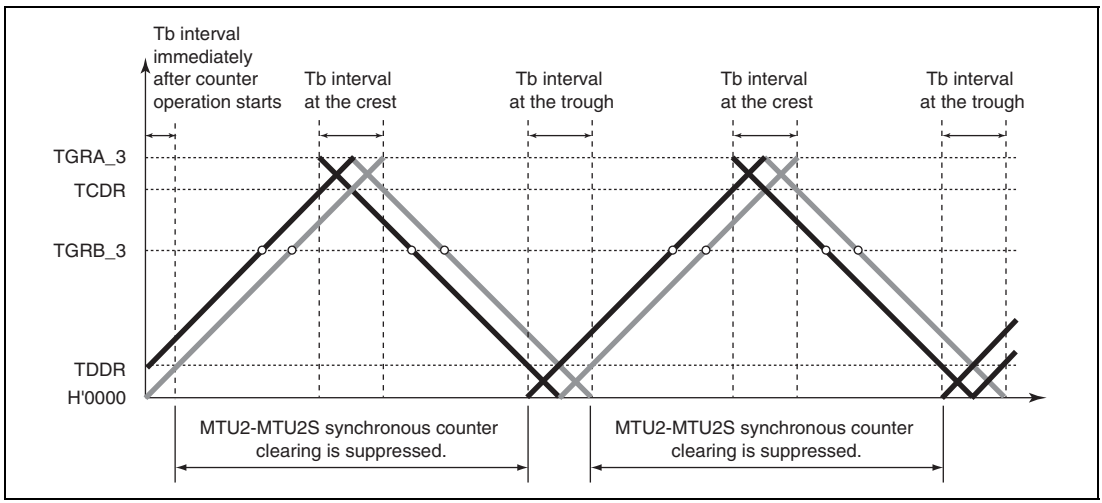


Figure 12.62 MTU2-MTU2S Synchronous Clearing-Suppressed Interval Specified by SCC Bit in TWCR

- Example of Procedure for Suppressing MTU2-MTU2S Synchronous Counter Clearing

  An example of the procedure for suppressing MTU2-MTU2S synchronous counter clearing is shown in figure 12.63.

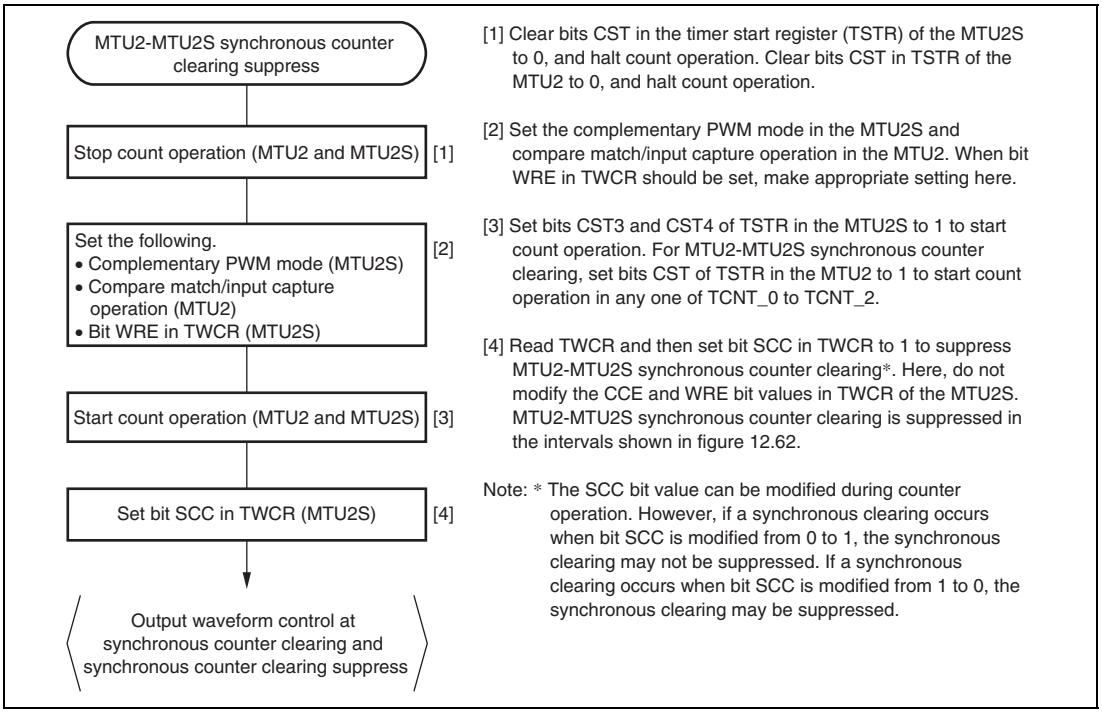


Figure 12.63 Example of Procedure for Suppressing MTU2-MTU2S Synchronous Counter Clearing

- Examples of Suppression of MTU2-MTU2S Synchronous Counter Clearing

  Figures 12.64 to 12.67 show examples of operation in which the MTU2S operates in complementary PWM mode and MTU2-MTU2S synchronous counter clearing is suppressed by setting the SCC bit in TWCR in the MTU2S to 1. In the examples shown in figures 12.64 to 12.67, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 12.56, respectively.

  In these examples, the WRE bit in TWCR of the MTU2S is set to 1.

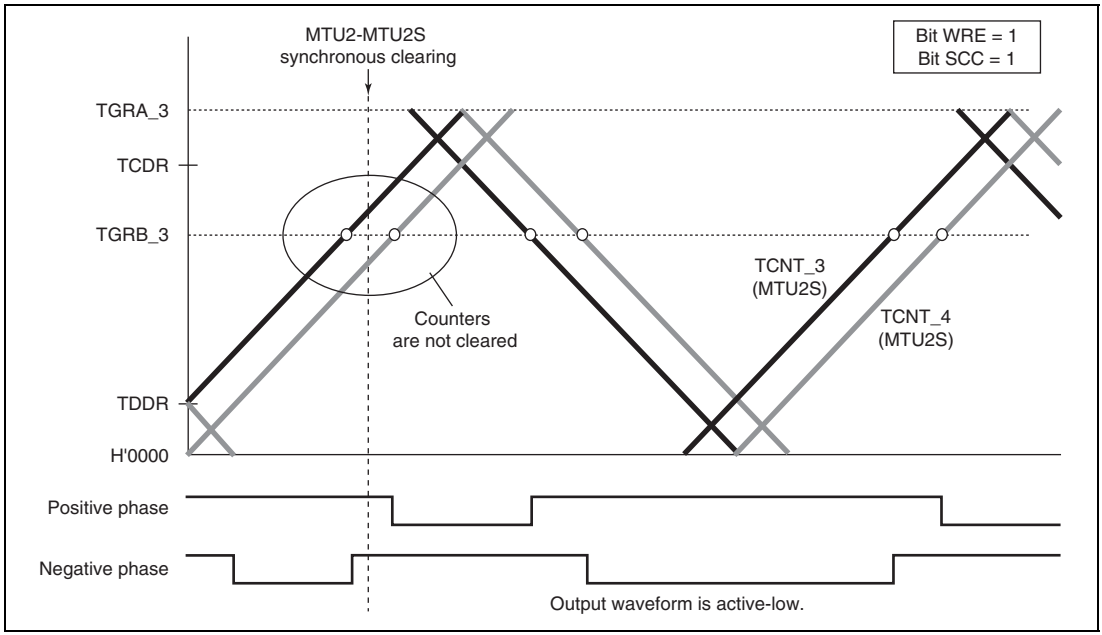


Figure 12.64 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 12.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

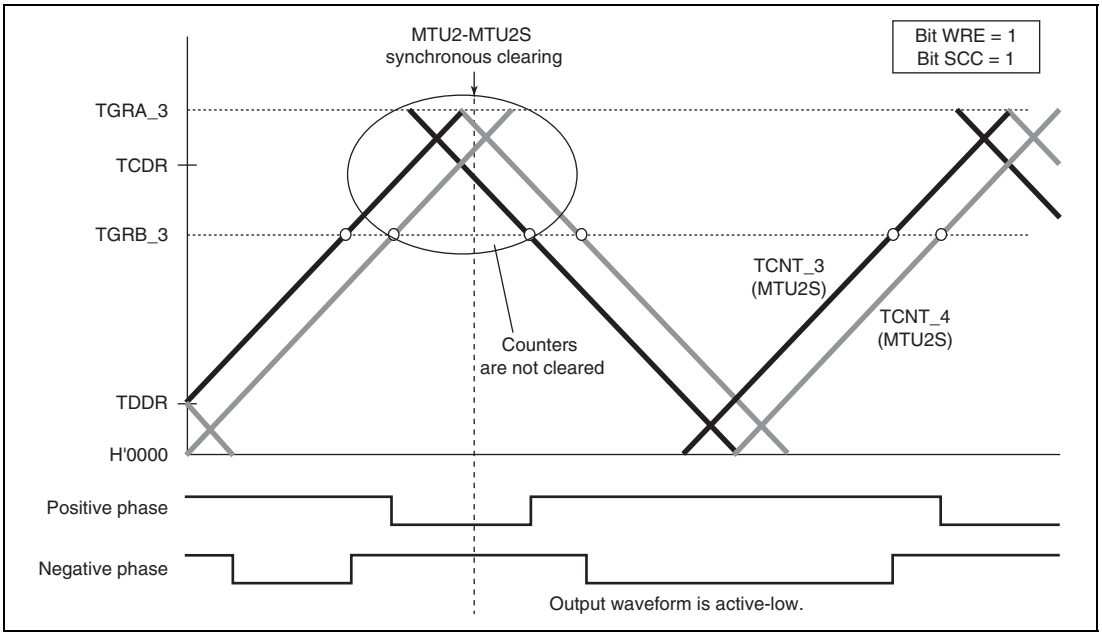


Figure 12.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 12.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

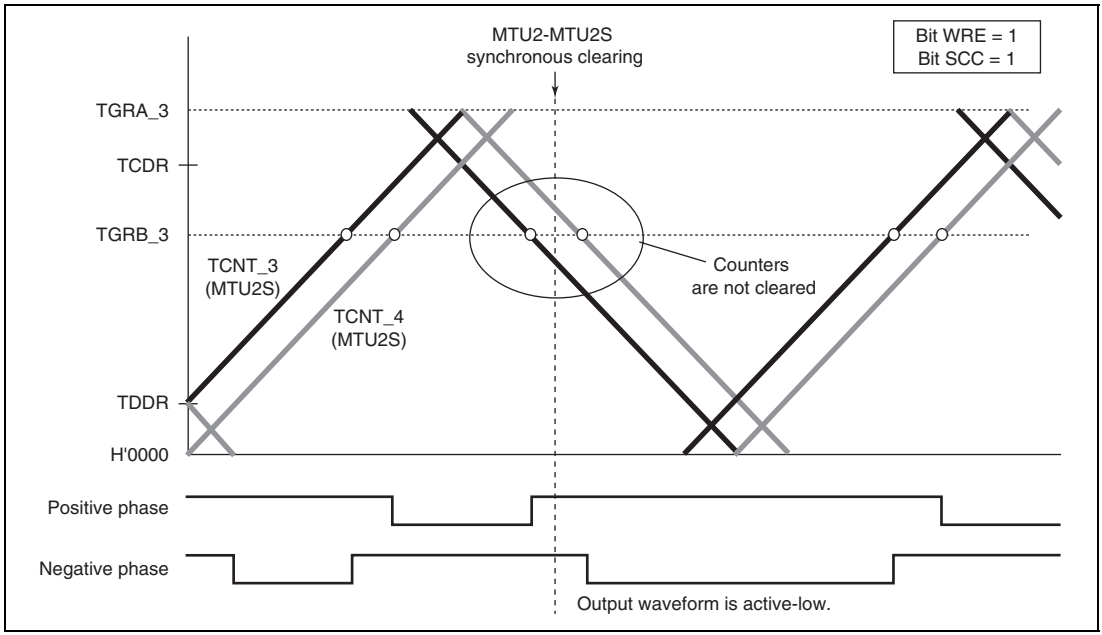


Figure 12.66 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 12.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

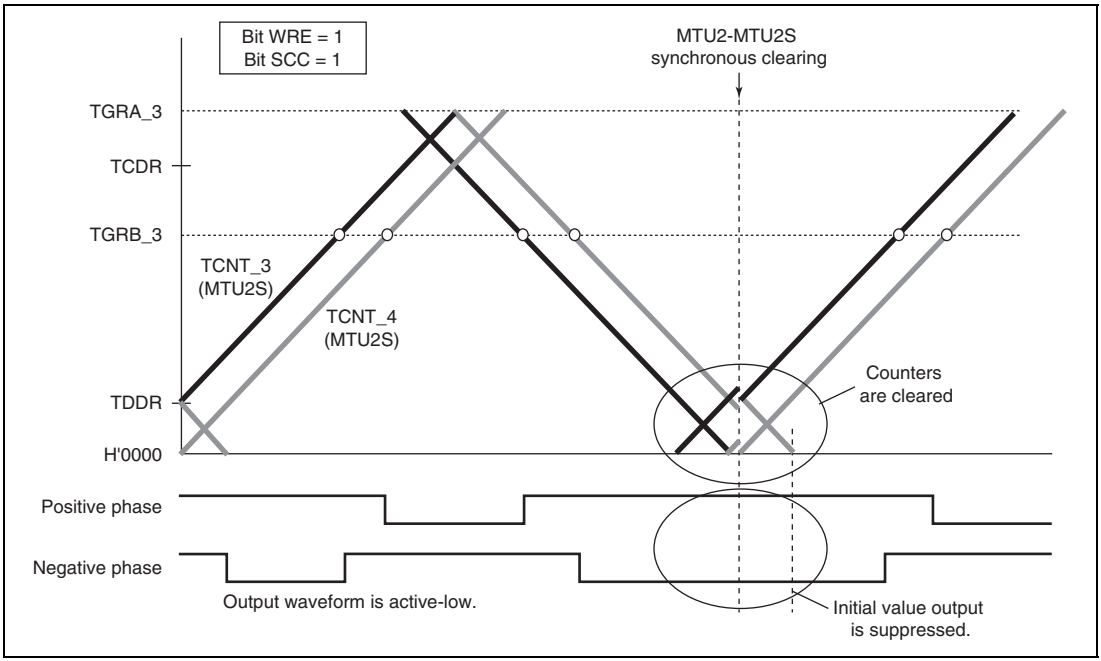


Figure 12.67 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 12.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

(p) Counter Clearing by TGRA_3 Compare Match

In complementary PWM mode, by setting the CCE bit in the timer waveform control register (TWCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by TGRA_3 compare match.

Figure 12.68 illustrates an operation example.

Notes: 1. Use this function only in complementary PWM mode 1 (transfer at crest)
2. Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1 or the CE0A, CE0B, CE0C, CE0D, CE1A, CE1B, CE1C, and CE1D bits in the timer synchronous clear register (TSYCR) to 1).
3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

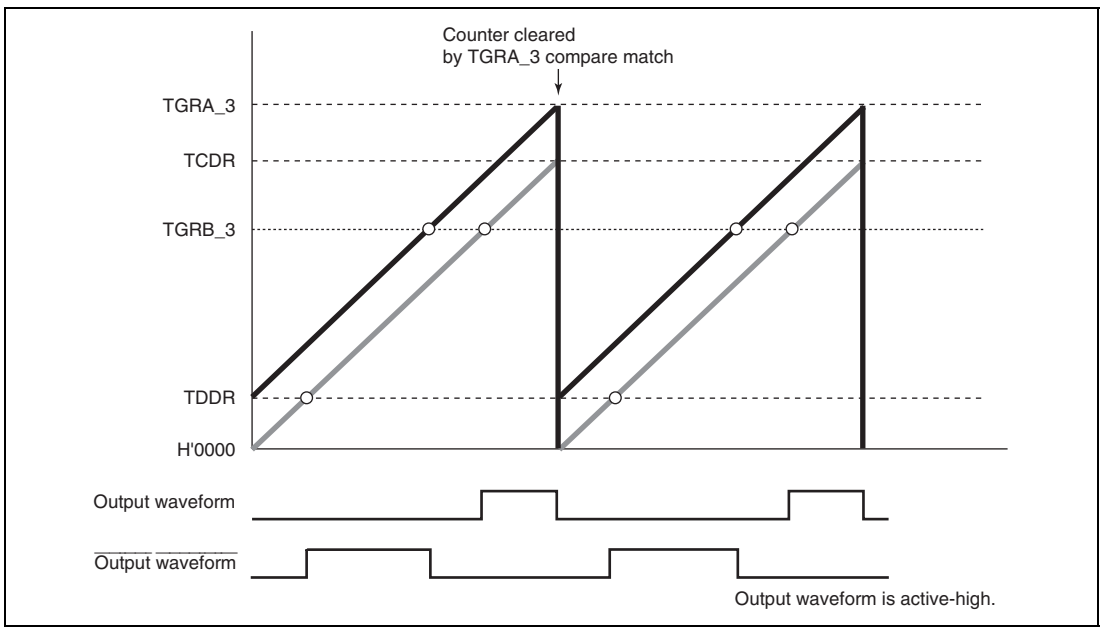


Figure 12.68 Example of Counter Clearing Operation by TGRA_3 Compare Match

(q) Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output

In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 12.69 to 12.72 show examples of brushless DC motor drive waveforms created using TGCR.

When output phase switching for a 3-phase brushless DC motor is performed by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with PFC). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.

The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.

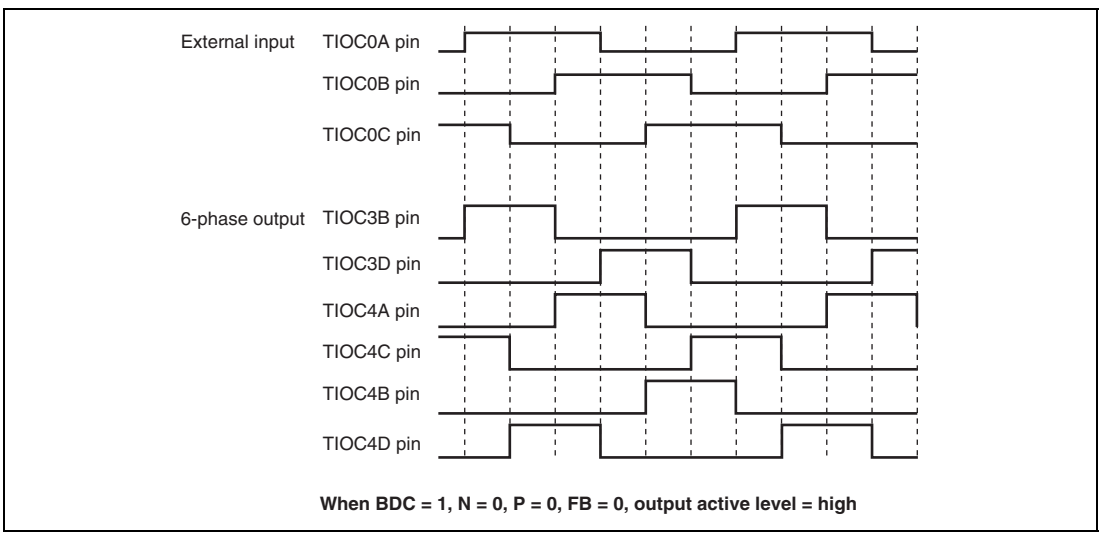


Figure 12.69 Example of Output Phase Switching by External Input (1)

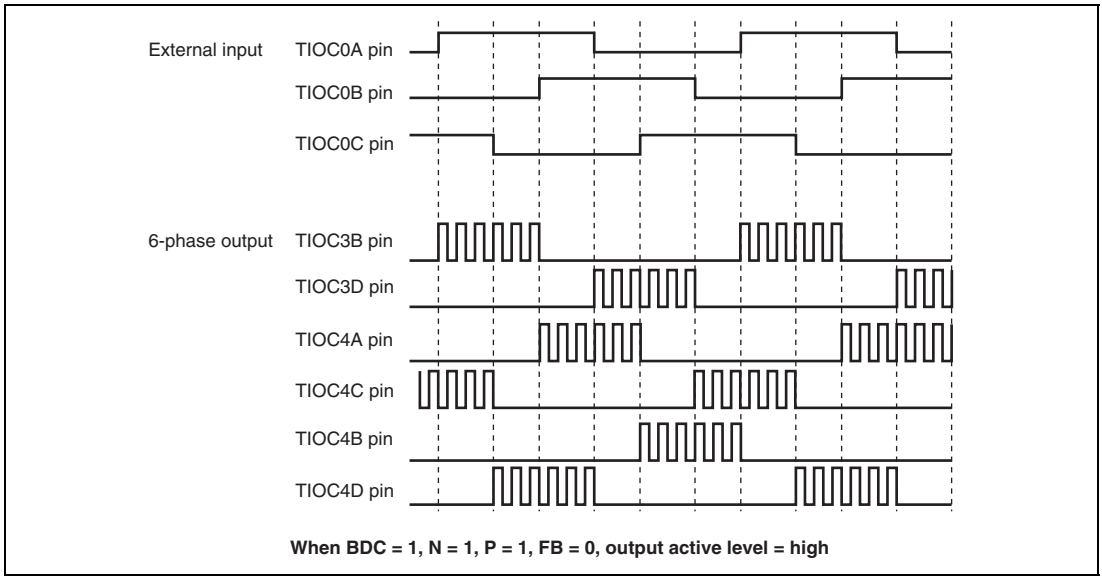


Figure 12.70 Example of Output Phase Switching by External Input (2)

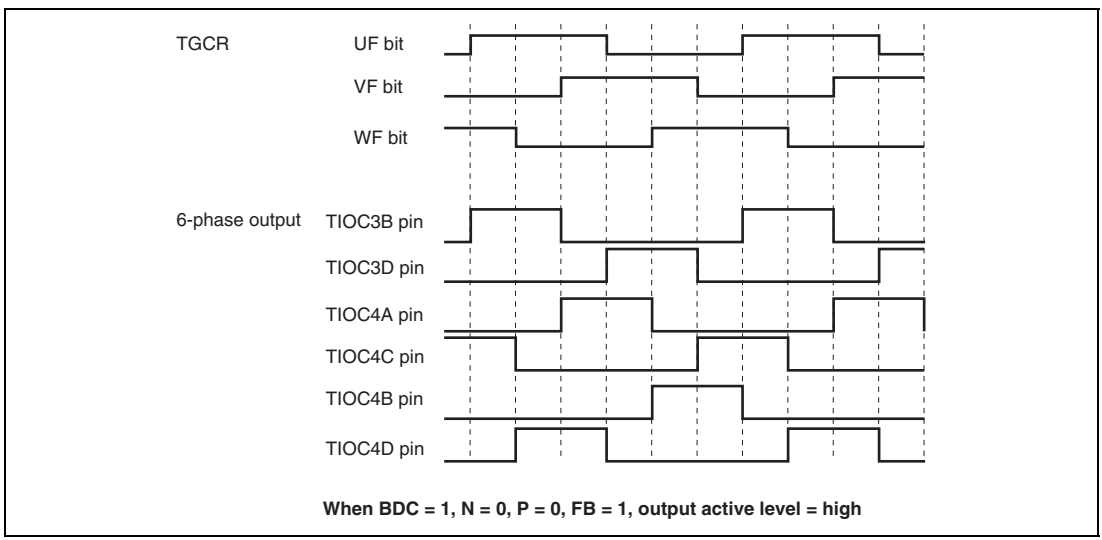


Figure 12.71 Example of Output Phase Switching by Means of UF, VF, and WF Bit Settings (1)

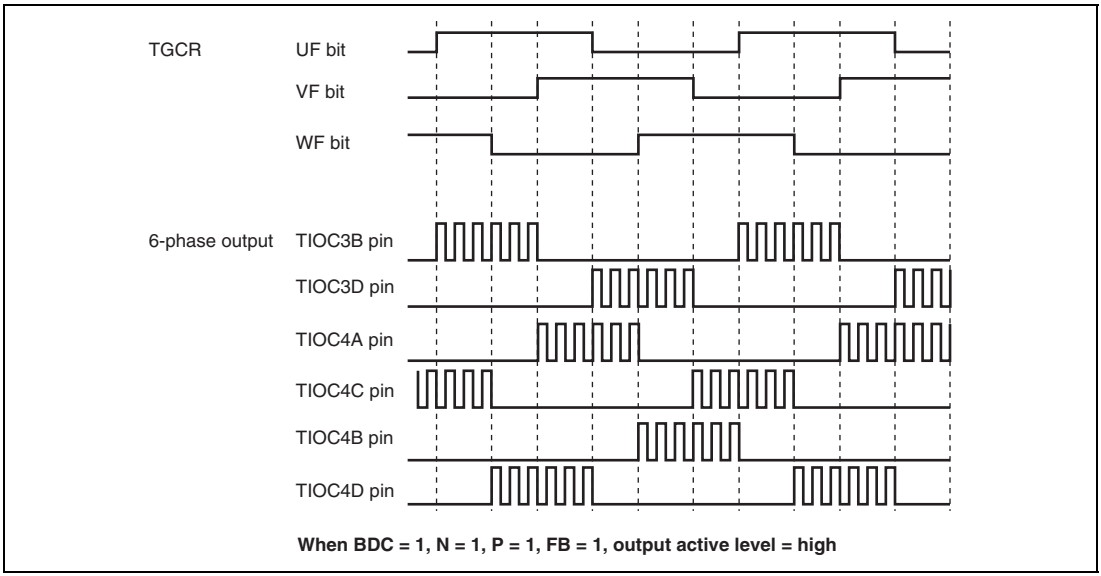


Figure 12.72 Example of Output Phase Switching by Means of UF, VF, and WF Bit Settings (2)

(r) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using a TGRA_3 compare-match, TCNT_4 underflow (trough), or compare-match on a channel other than channels 3 and 4.

When start requests using a TGRA_3 compare-match are specified, A/D conversion can be started at the crest of the TCNT_3 count.

A/D converter start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER). To issue an A/D converter start request at a TCNT_4 underflow (trough), set the TTGE2 bit in TIER_4 to 1.

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA_3 (at the crest) and TCIV_4 (at the trough) in channels 3 and 4 can be skipped up to seven times by making settings in the timer interrupt skipping set register (TITCR).

Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 12.4.9, A/D Converter Start Request Delaying Function.

The setting of the timer interrupt skipping setting register (TITCR) must be done while the TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of TIER_3 and TIER_4 along with under the conditions in which TGFA_3 and TCFV_4 flag settings by compare match never occur. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 12.73 shows an example of the interrupt skipping operation setting procedure. Figure 12.74 shows the periods during which interrupt skipping count can be changed.

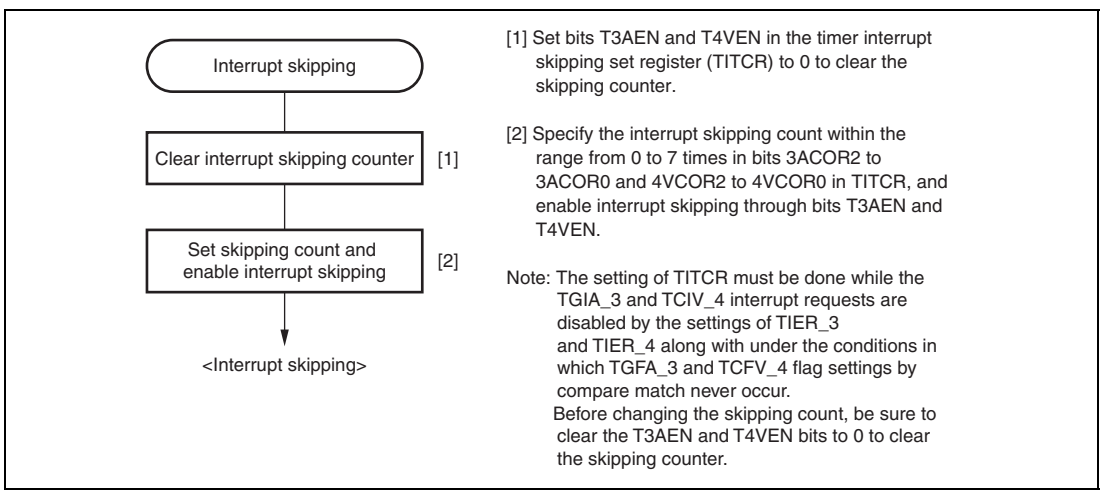


Figure 12.73 Example of Interrupt Skipping Operation Setting Procedure

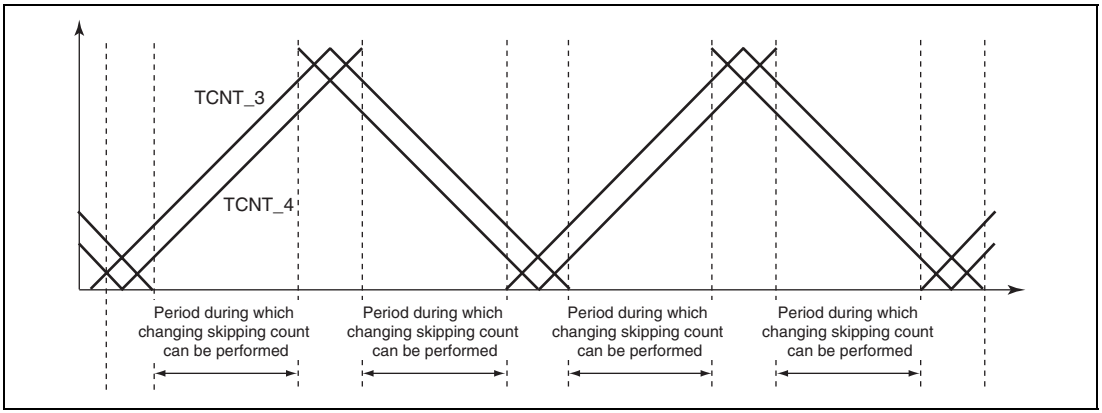


Figure 12.74 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 12.75 shows an example of TGIA_3 interrupt skipping in which the interrupt skipping count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

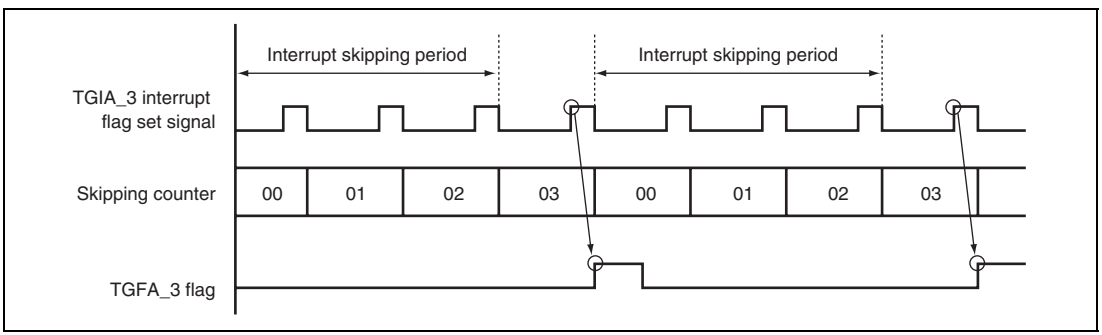


Figure 12.75 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

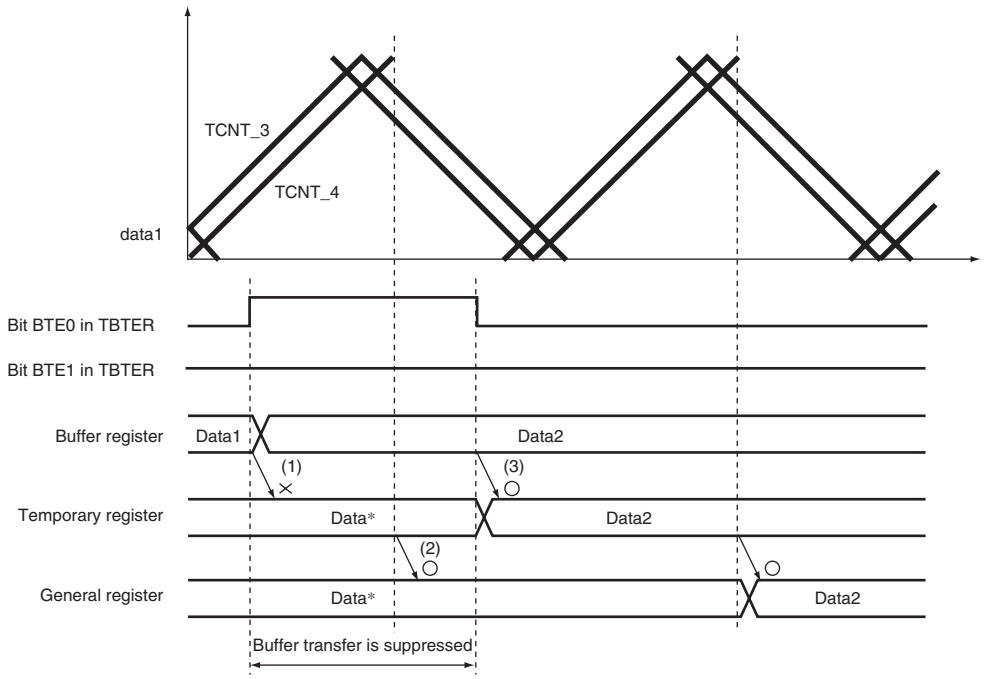
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

Figure 12.76 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 12.77 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. Depending on the timing of interrupt generation and writing to the buffer register, the timing of transfer from the buffer register to the temporary register and from the temporary register to the general register is one of two types.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 12.78 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

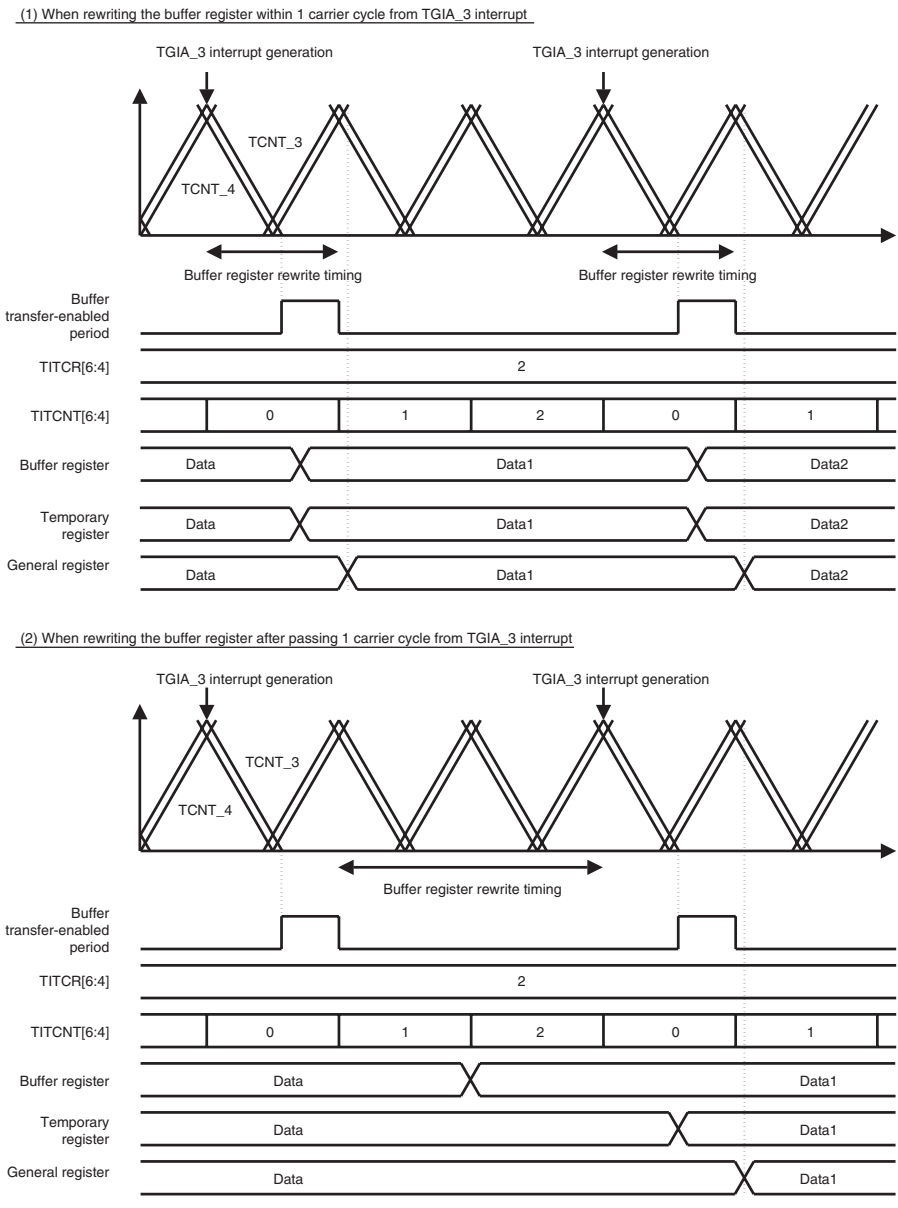


[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

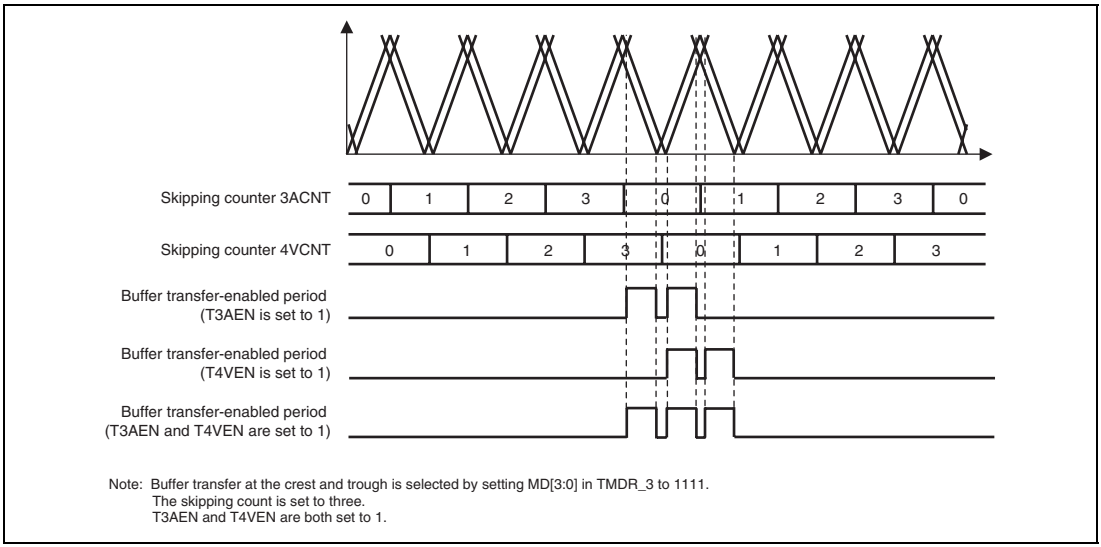
Note: * When buffer transfer at the crest is selected.

Figure 12.76 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



Note: Buffer transfer at the crest is selected by setting MD[3:0] in TMDR_3 to 1101.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and 0, respectively.

Figure 12.77 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)



Note: Buffer transfer at the crest and trough is selected by setting MD[3:0] in TMDR_3 to 1111.
The skipping count is set to three.
T3AEN and T4VEN are both set to 1.

Figure 12.78 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Function

Complementary PWM mode output has the following protection functions.

(a) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be rewritten at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWER). The applicable registers are some (21 in total) of the registers in channels 3 and 4 shown in the following:

- TCR_3 and TCR_4, TMDR_3 and TMDR_4, TIORH_3 and TIORH_4, TIORL_3 and TIORL_4, TIER_3 and TIER_4, TCNT_3 and TCNT_4, TGRA_3 and TGRA_4, TGRB_3 and TGRB_4, TOER, TOCR, TGCR, TCDR, and TDDR.

This function enables miswriting due to CPU runaway to be prevented by disabling CPU access to the mode registers, control registers, and counters. When the applicable registers are read in the access-disabled state, undefined values are returned. Writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

See section 14, Port Output Enable 2 (POE2), for details.

(c) Halting of PWM Output by Oscillation Stop

The 6-phase PWM output pins can detect the clock stop and set the output pin automatically to the high-impedance state. However, the pin state is not guaranteed when the clock starts oscillation again.

See section 5.7, Oscillation Stop Detection, for details.

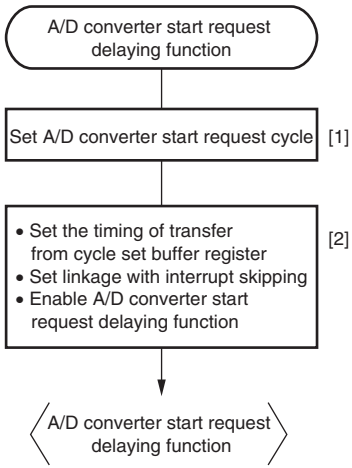
12.4.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in channel 4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4), and timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4).

The A/D converter start request delaying function compares TCNT_4 with TADCORA_4 or TADCORB_4, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

- Example of Procedure for Specifying A/D Converter Start Request Delaying Function
  Figure 12.79 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle buffer register (TADCOBRA_4 or TADCOBRB_4) and timer A/D converter start request cycle register (TADCORA_4 or TADCORB_4). (The same initial value must be specified in the cycle buffer register and cycle register.)

[2] Use bits BF1 and BF2 in the timer A/D converter start request control register (TADCR) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits TU4AE, DT4AE, UT4BE, and DT4BE to enable A/D conversion start requests (TRG4AN or TRG4BN).

Notes: 1. Perform TADCR setting while TCNT_4 is stopped.
2. Do not set BF1 to 1 when complementary PWM mode is not selected.
3. Do not set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, or DT4BE to 1 when complementary PWM mode is not selected.

Figure 12.79 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 12.80 shows a basic example of A/D converter request signal (TRG4AN) operation when the trough of TCNT_4 is specified for the buffer transfer timing and an A/D converter start request signal is output during TCNT_4 down-counting.

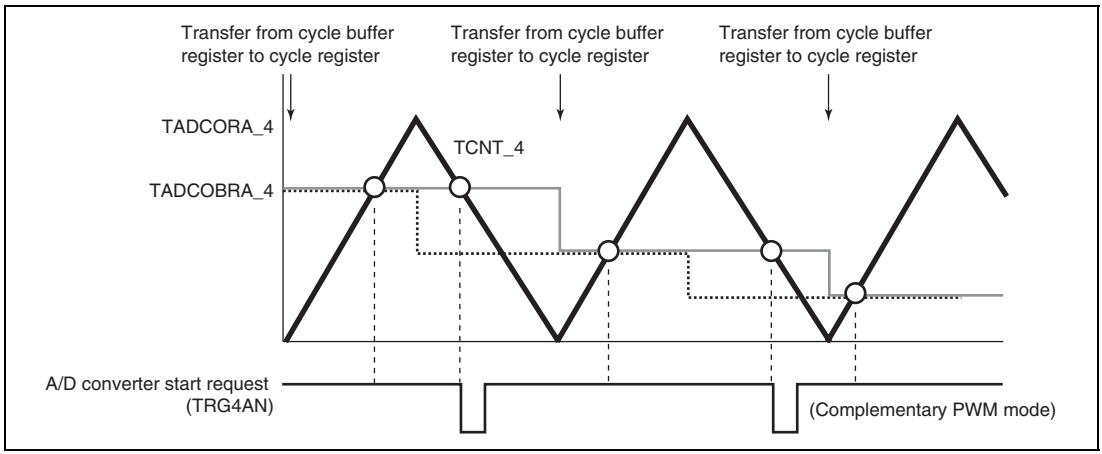


Figure 12.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- Buffer Transfer

  The data in the timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4) is updated by writing data to the timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF1 and BF0 bits in the timer A/D converter start request control register (TADCR_4).

- A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

  A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).

  Figure 12.81 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

  Figure 12.82 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and A/D converter start requests are linked with interrupt skipping.

Note: This function must be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

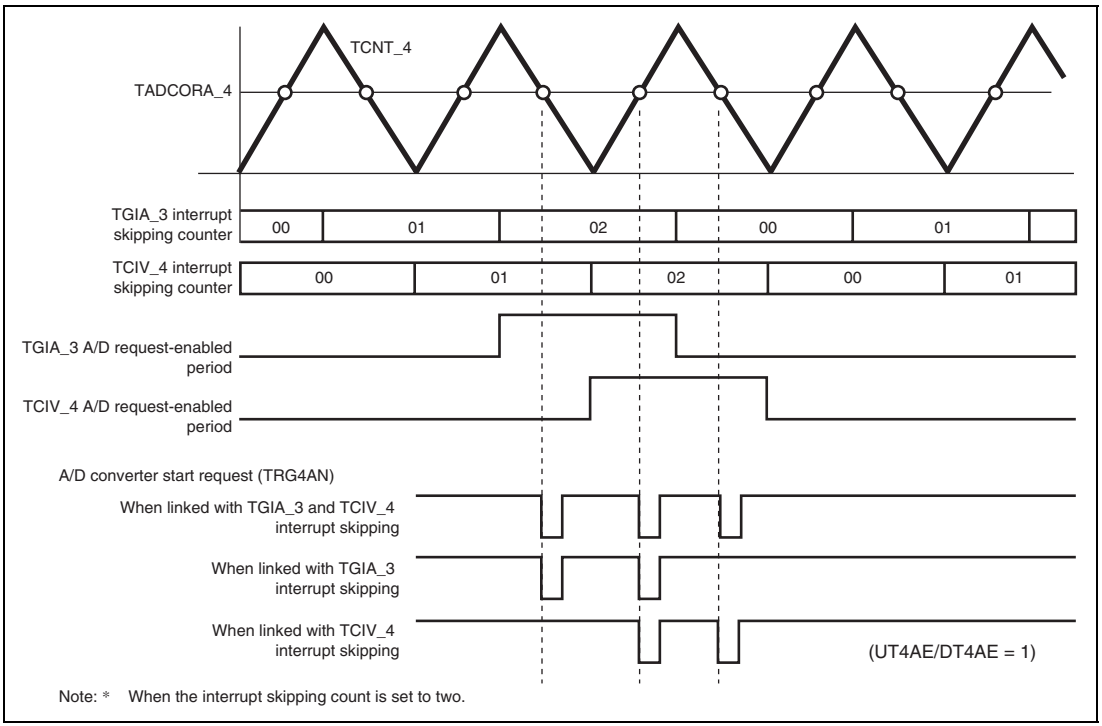


Figure 12.81 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

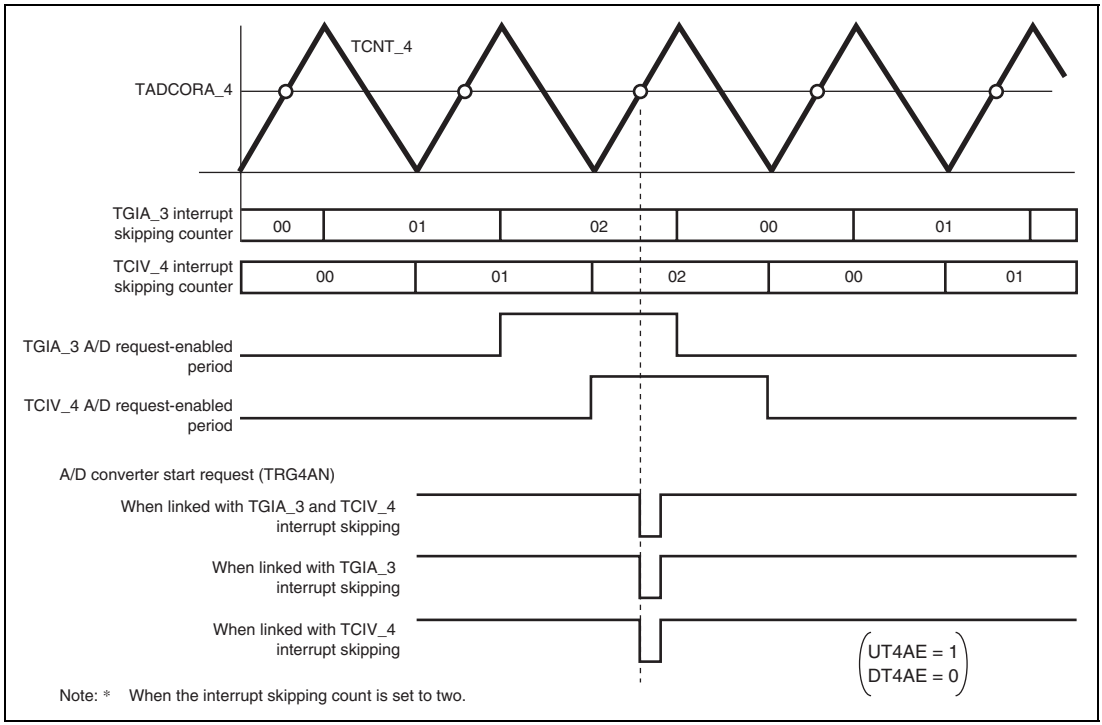


Figure 12.82 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

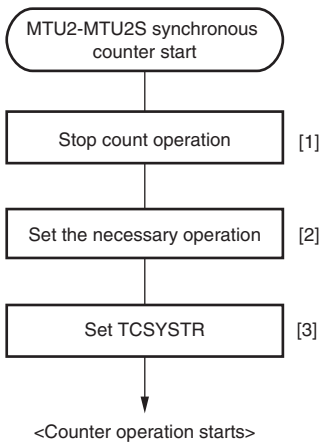
12.4.10 MTU2-MTU2S Synchronous Operation

(1) MTU2-MTU2S Synchronous Counter Start

The counters in the MTU2 and MTU2S which operate at different clock systems can be started synchronously by making the TCSYSTR settings in the MTU2.

(a) Example of MTU2-MTU2S Synchronous Counter Start Setting Procedure

Figure 12.83 shows an example of synchronous counter start setting procedure.



[1] Use TSTR registers of the MTU2 and MTU2S and halt the counters used for synchronous start operation.

[2] Specify necessary operation with appropriate registers such as TCR and TMDR.

[3] In TCSYSTR of the MTU2, set the bits corresponding to the counters to be started synchronously to 1. The TSTRs are automatically set appropriately and the counters start synchronously.

Notes: 1. Even if a bit in TCSYSTR corresponding to an operating counter is cleared to 0, the counter will not stop. To stop the counter, clear the corresponding bit in TSTR to 0 directly.

2. To start channels 3 and 4 in reset-synchronized PWM mode or complementary PWM mode, make appropriate settings in TCYSTR according to the TSTR setting for the respective mode. For details, refer to section 12.4.7, Reset-Synchronized PWM Mode, and section 12.4.8, Complementary PWM Mode.

Figure 12.83 Example of Synchronous Counter Start Setting Procedure

(b) Examples of Synchronous Counter Start Operation

Figures 12.84 (1) to (3) show examples of synchronous counter start operation when the clock frequency ratios between the MTU2 and MTU2S are 1:1, 1:2, and 1:4, respectively. In these examples, the count clock is set to Pϕ/1.

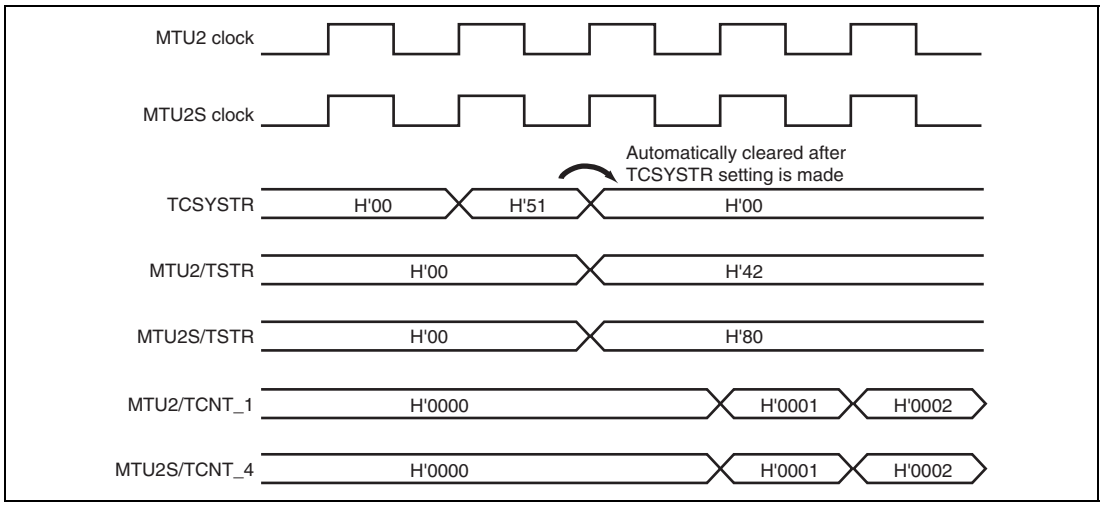


Figure 12.84 (1) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:1)

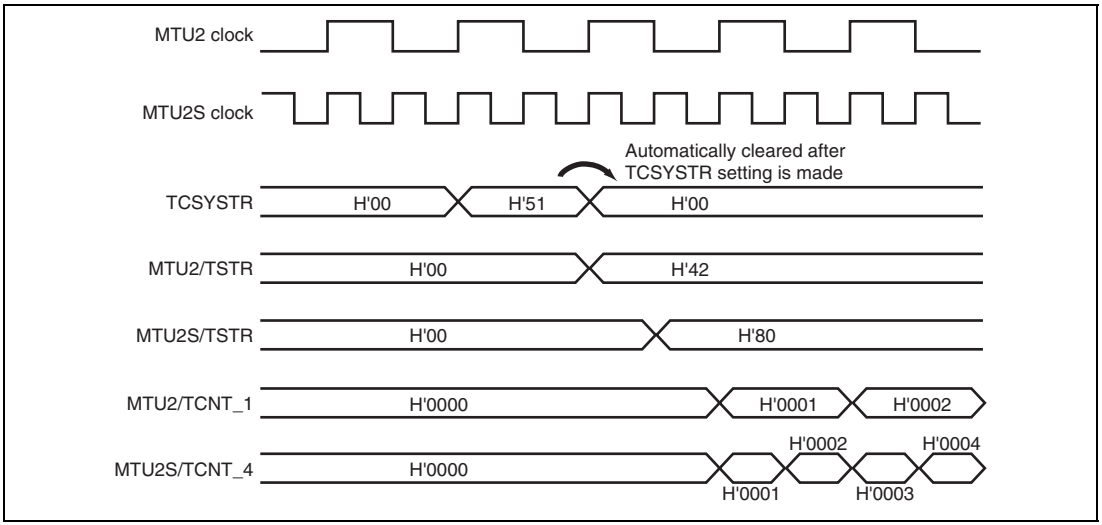


Figure 12.84 (2) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:2)

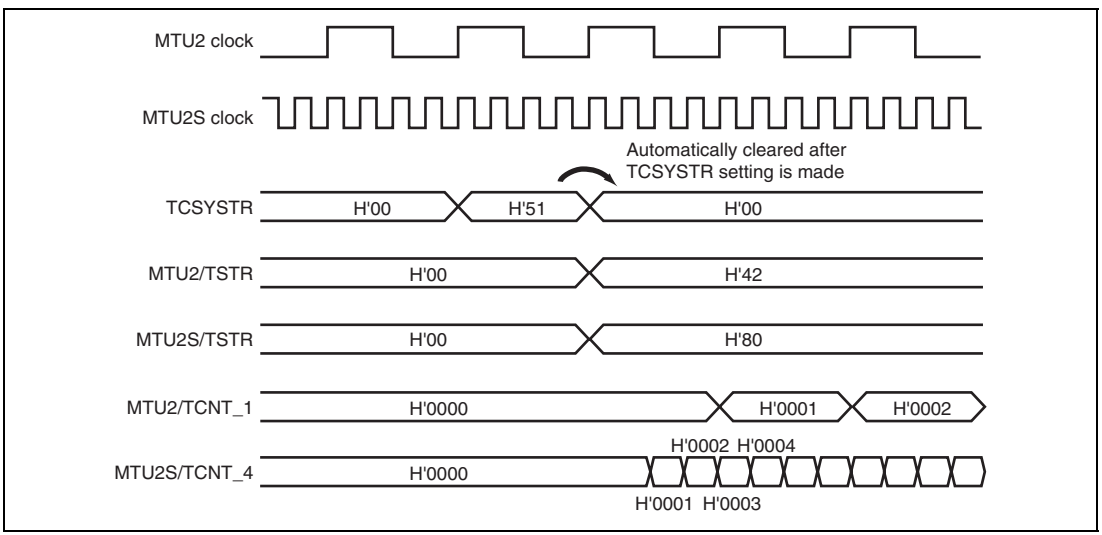


Figure 12.84 (3) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:4)

(2) MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (MTU2-MTU2S Synchronous Counter Clearing)

The MTU2S counters can be cleared by sources for setting the flags in TSR_0 to TSR_2 in the MTU2 through the TSYCR_3 settings in the MTU2S.

(a) Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

Figure 12.85 shows an example of procedure for specifying MTU2S counter clearing by MTU2 flag setting source.

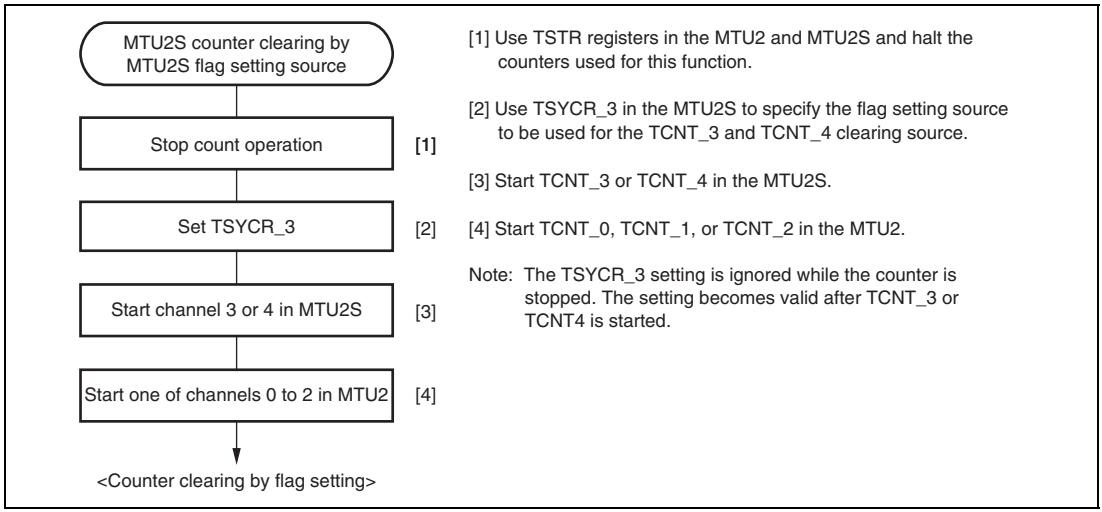


Figure 12.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

(b) Examples of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source

Figures 12.86 (1) and 12.86 (2) show examples of MTS2S counter clearing caused by MTU2 flag setting source.

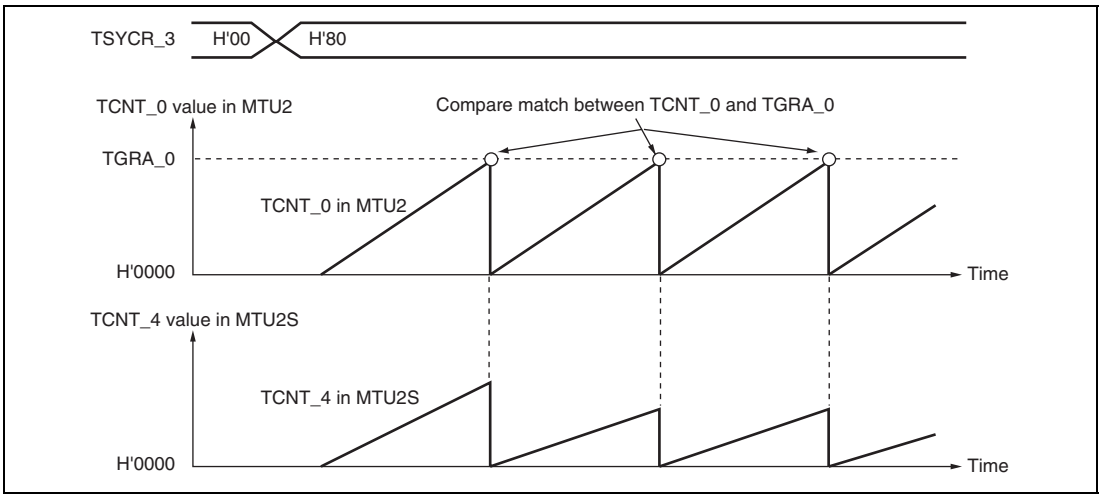


Figure 12.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)

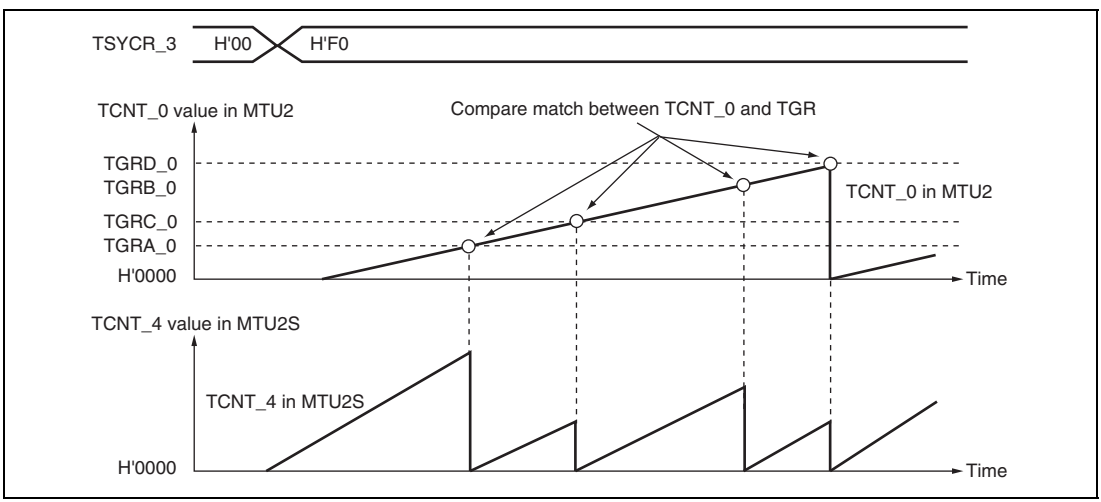


Figure 12.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)

12.4.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in channel 5.

(1) Example of External Pulse Width Measurement Setting Procedure

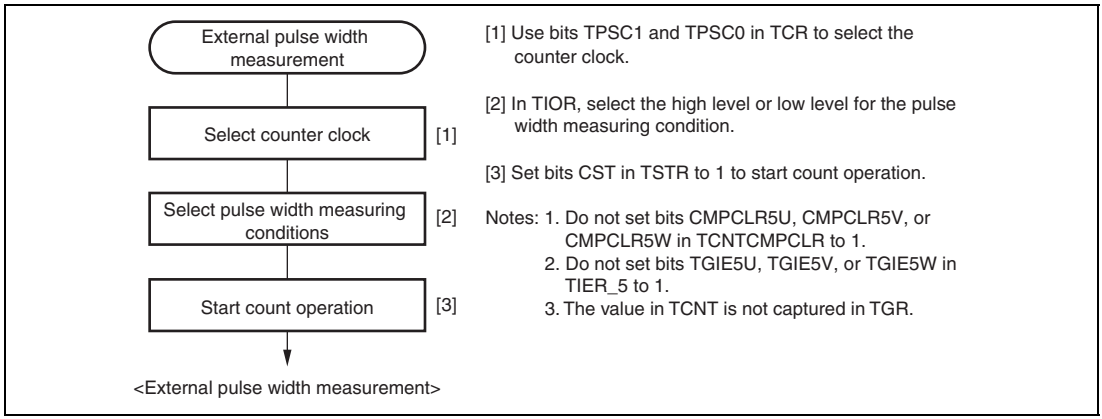


Figure 12.87 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

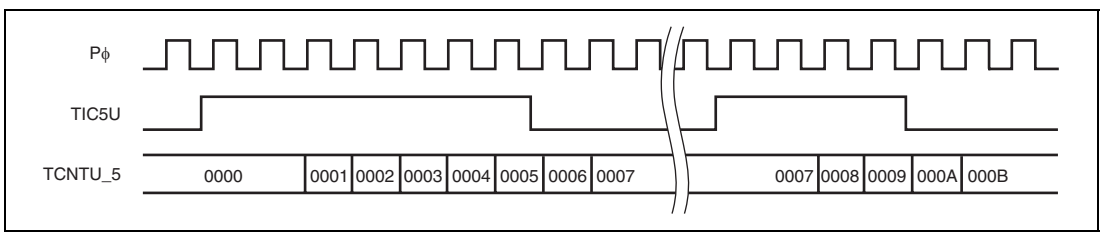


Figure 12.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)

12.4.12 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function while the complementary PWM is in operation.

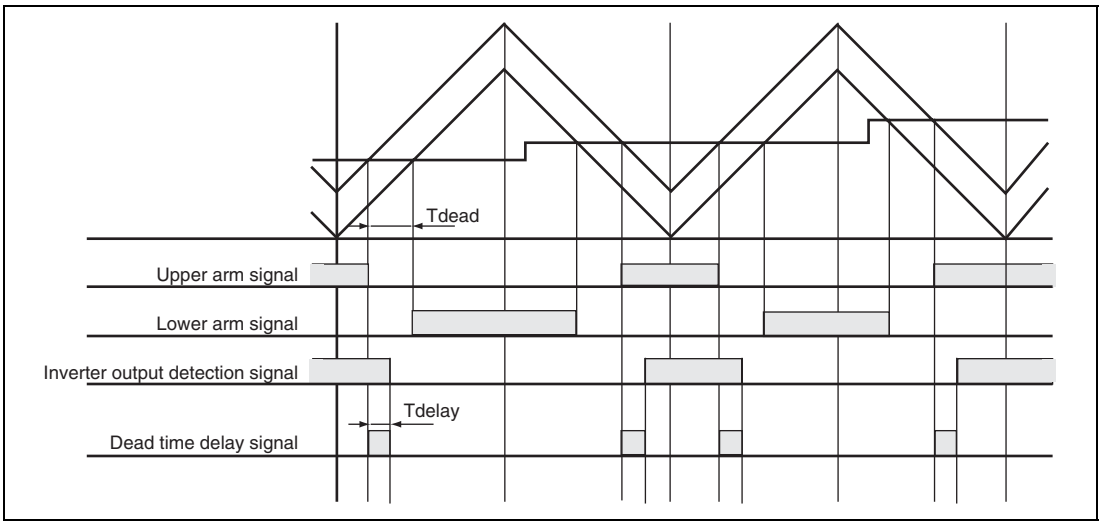
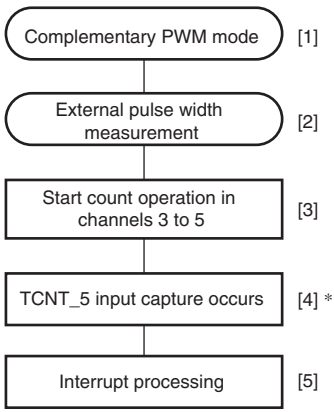


Figure 12.89 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 12.90 shows an example of dead time compensation setting procedure by using three counters in channel 5.



[1] Place channels 3 and 4 in complementary PWM mode. For details, refer to section 12.4.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in channel 5. For details, refer to section 12.4.11, External Pulse Width Measurement.

[3] Set bits CST3 and CST4 in TSTR and bits CST5U, CST5V, and CST5W in TSTR2 to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the TCNT_5 value is captured in TGR_5.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA_3) or trough (TCIV_4) in complementary PWM mode, read the TGRU_5 value, calculate the difference in time in TGRB_3, and write the corrected value to TGRD_3 in the interrupt processing.
For the V phase and W phase, read the TGRV_5 and TGRW_5 values and write the corrected values to TGRC_4 and TGRD_4, respectively, in the same way as for U-phase compensation.
The TCNT_5 value should be cleared through the TCNTCMPCLR setting or by software.

Notes: The PFC settings must be completed in advance.
* As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in TIER_5.

Figure 12.90 Example of Dead Time Compensation Setting Procedure

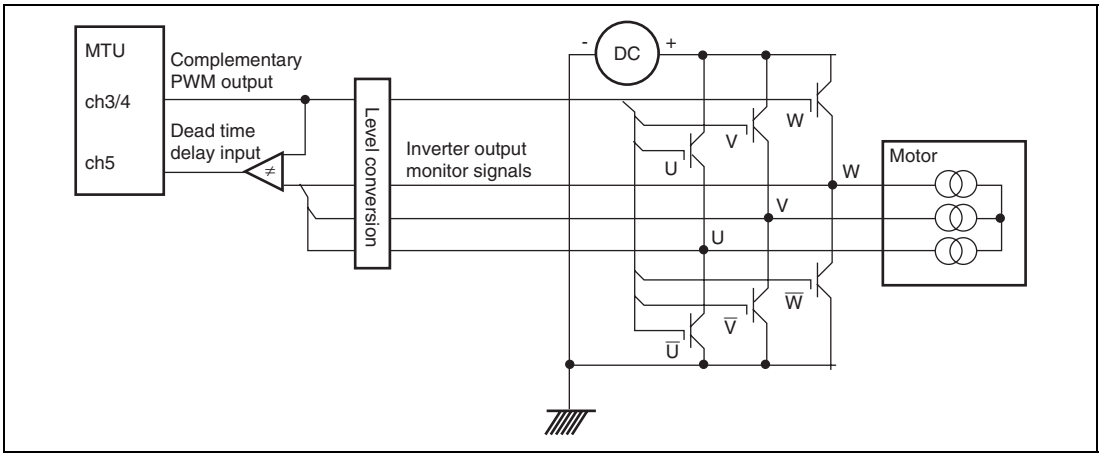


Figure 12.91 Example of Motor Control Circuit Configuration

12.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 12.92 shows an example in which TCNT is used as a free-running counter without being cleared, and the TCNT value is captured in TGR at the specified timing (either crest or trough, or both crest and trough).

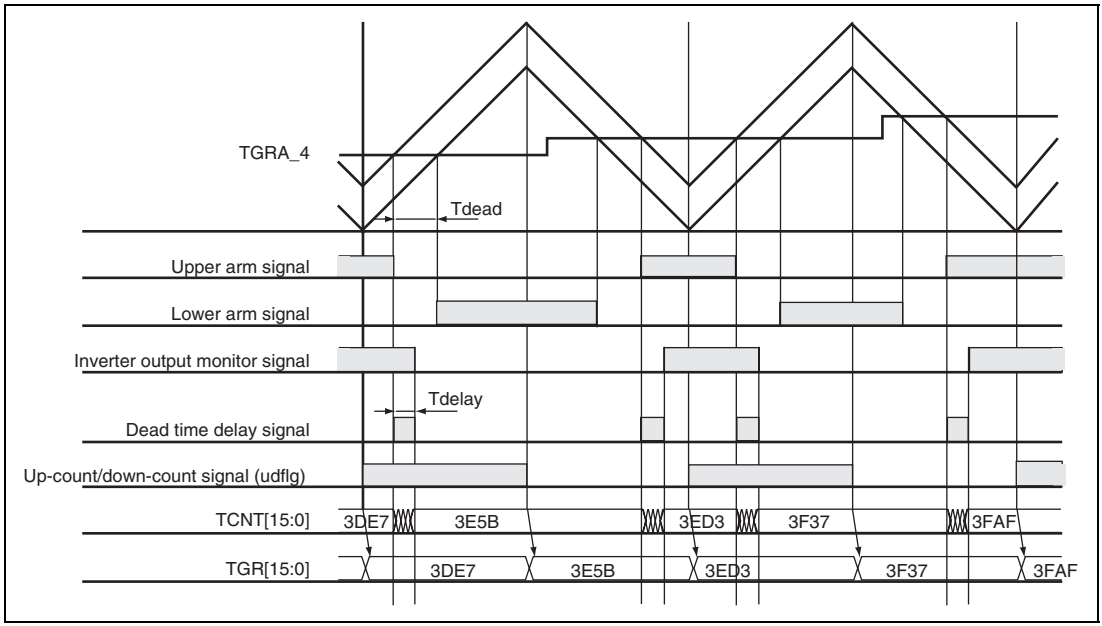


Figure 12.92 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

12.5 Interrupt Sources

12.5.1 Interrupt Sources and Priorities

There are three kinds of MTU2 interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 7, Interrupt Controller (INTC).

Table 12.57 lists the MTU2 interrupt sources.

Table 12.57 MTU2 Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DMAC Activation	Priority
0	TGIA_0	TGRA_0 input capture/compare match	TGFA_0	Possible	High
	TGIB_0	TGRB_0 input capture/compare match	TGFB_0	Not possible	
	TGIC_0	TGRC_0 input capture/compare match	TGFC_0	Not possible	
	TGID_0	TGRD_0 input capture/compare match	TGFD_0	Not possible	
	TCIV_0	TCNT_0 overflow	TCFV_0	Not possible	
	TGIE_0	TGRE_0 compare match	TGFE_0	Not possible	
	TGIF_0	TGRF_0 compare match	TGFF_0	Not possible	
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible	
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Not possible	
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible	
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible	
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible	
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Not possible	
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible	
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible	
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible	
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Not possible	
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Not possible	
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Not possible	
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible	
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible	
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Not possible	
	TGIC_4	TGRC_4 input capture/compare match	TGFC_4	Not possible	
	TGID_4	TGRD_4 input capture/compare match	TGFD_4	Not possible	
	TCIV_4	TCNT_4 overflow/underflow	TCFV_4	Not possible	
5	TGIU_5	TGRU_5 input capture/compare match	TGFU_5	Not possible	
	TGIV_5	TGRV_5 input capture/compare match	TGFV_5	Not possible	
	TGIW_5	TGRW_5 input capture/compare match	TGFW_5	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The MTU2 has 21 input capture/compare match interrupts, six for channel 0, four each for channels 3 and 4, two each for channels 1 and 2, and three for channel 5. The TGFE_0 and TGFF_0 flags in channel 0 are not set by the occurrence of an input capture.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The MTU2 has five overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU2 has two underflow interrupts, one each for channels 1 and 2.

12.5.2 DMAC and DTC Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel and the overflow interrupt of channel 4. For details, see section 9, Data Transfer Controller (DTC).

In the MTU2, a total of twenty input capture/compare match interrupts and overflow interrupts can be used as DTC activation sources, four each for channels 0 and 3, two each for channels 1 and 2, five for channel 4 and three for channel 5.

(2) DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 11, Direct Memory Access Controller (DMAC).

In the MTU2, a total of five TGRA input capture/compare match interrupts can be used as DMAC activation sources, one each for channels 0 to 4.

12.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 12.58 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at TCNT_4 Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in TIER_4 is set to 1, the A/D converter can be activated at the trough of TCNT_4 count (TCNT_4 = H'0000).

A/D converter start request signal TRGAN is issued to the A/D converter under either one of the following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel while the TTGE bit in TIER is set to 1
- When the TCNT_4 count reaches the trough (TCNT_4 = H'0000) during complementary PWM operation while the TTGE2 bit in TIER_4 is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between TCNT_0 and TGRE_0

The A/D converter can be activated by generating A/D converter start request signal TRG0N when a compare match occurs between TCNT_0 and TGRE_0 in channel 0.

When the TGFE flag in TSR2_0 is set to 1 by the occurrence of a compare match between TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0N from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 12.4.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4AN from the MTU2 is selected as the trigger in the A/D converter when TRG4AN is generated or if TRG4BN from the MTU2 is selected as the trigger in the A/D converter when TRG4BN is generated.

Table 12.58 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Request Signal
TGRA_0 and TCNT_0	Input capture/compare match	TRGAN
TGRA_1 and TCNT_1		
TGRA_2 and TCNT_2		
TGRA_3 and TCNT_3		
TGRA_4 and TCNT_4		
TCNT_4	TCNT_4 trough in complementary PWM mode	
TGRE_0 and TCNT_0	Compare match	TRG0N
TADCORA and TCNT_4		TRG4AN
TADCORB and TCNT_4		TRG4BN

12.6 Operation Timing

12.6.1 Input/Output Timing

(1) TCNT Count Timing

Figures 12.93 and 94 show TCNT count timing in internal clock operation, and figure 12.95 shows TCNT count timing in external clock operation (normal mode), and figure 12.96 shows TCNT count timing in external clock operation (phase counting mode).

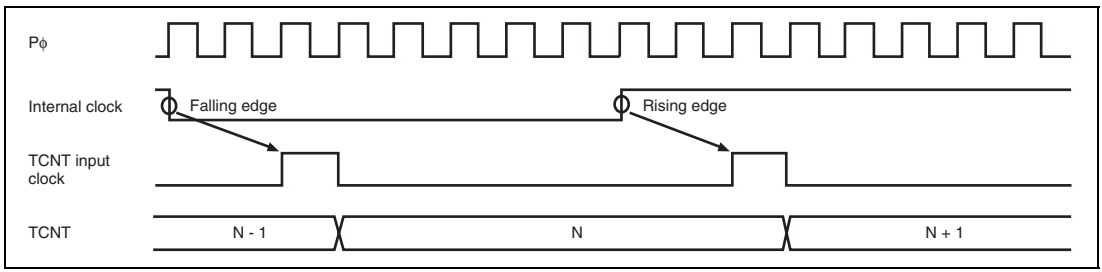


Figure 12.93 Count Timing in Internal Clock Operation (Channels 0 to 4)

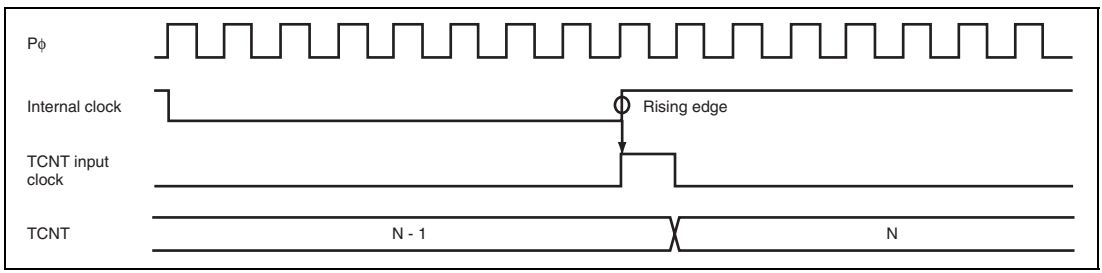


Figure 12.94 Count Timing in Internal Clock Operation (Channel 5)

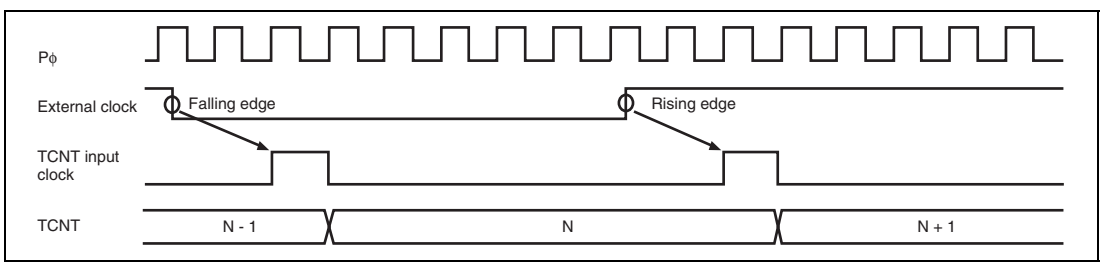


Figure 12.95 Count Timing in External Clock Operation (Channels 0 to 4)

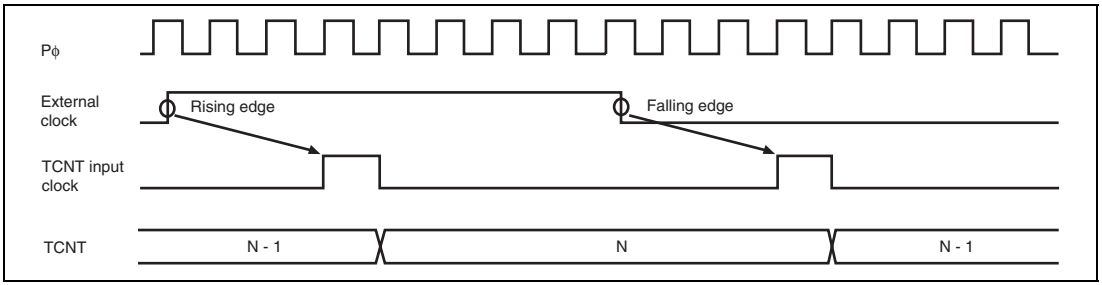


Figure 12.96 Count Timing in External Clock Operation (Phase Counting Mode)

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 12.97 shows output compare output timing (normal mode and PWM mode) and figure 12.98 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

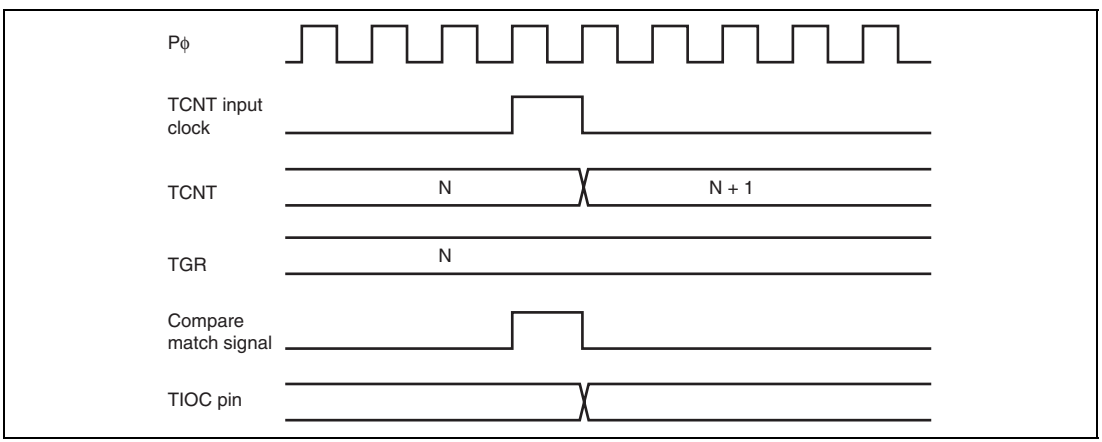


Figure 12.97 Output Compare Output Timing (Normal Mode/PWM Mode)

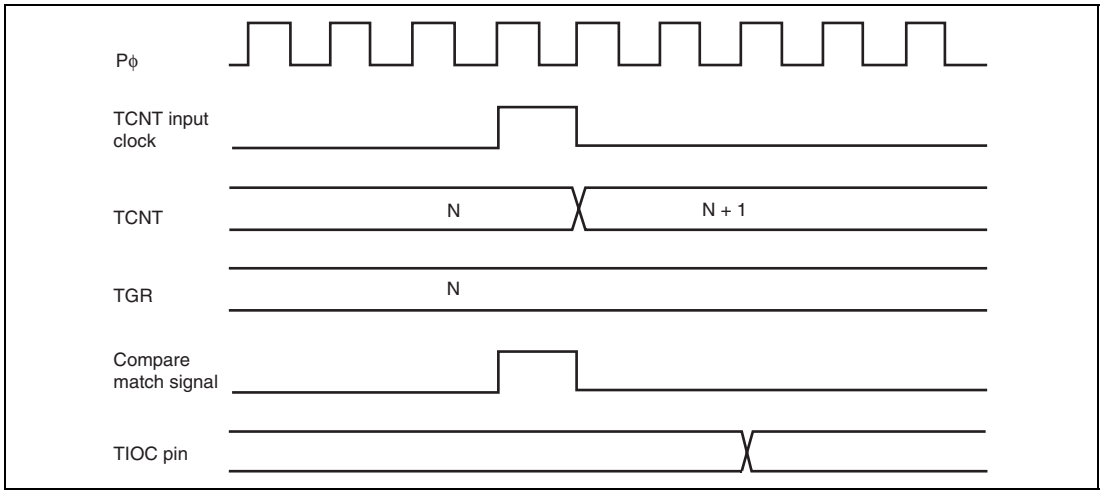


Figure 12.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 12.99 shows input capture signal timing.

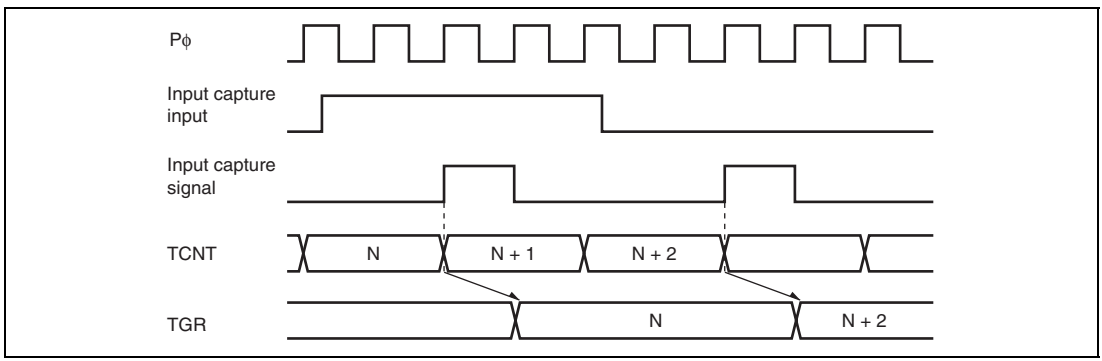


Figure 12.99 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figures 12.100 and 101 show the timing when counter clearing on compare match is specified, and figure 12.102 shows the timing when counter clearing on input capture is specified.

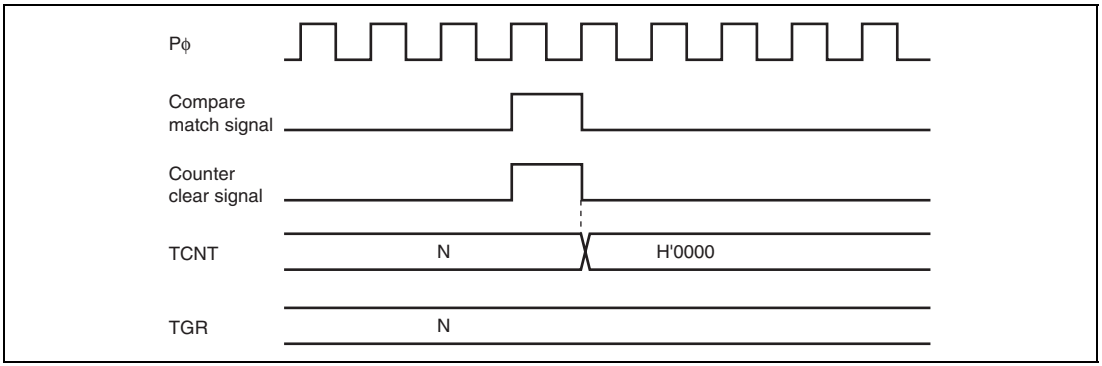


Figure 12.100 Counter Clear Timing (Compare Match) (Channels 0 to 4)

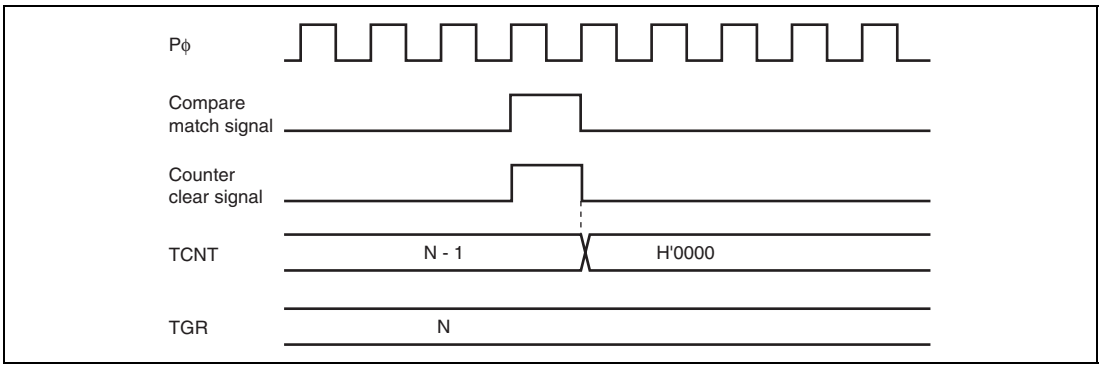


Figure 12.101 Counter Clear Timing (Compare Match) (Channel 5)

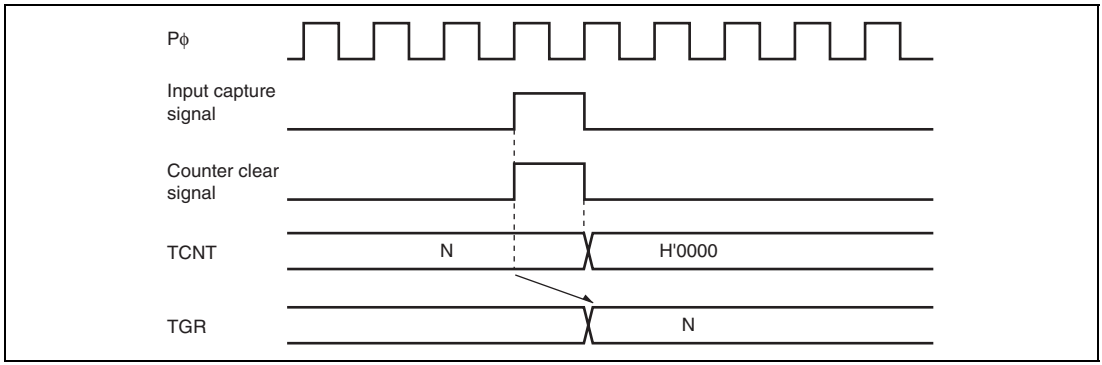


Figure 12.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)

(5) Buffer Operation Timing

Figures 12.103 to 12.105 show the timing in buffer operation.

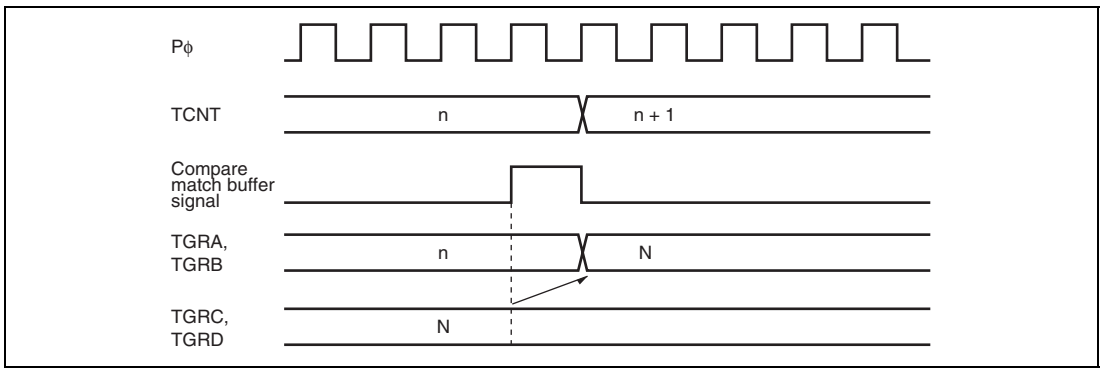


Figure 12.103 Buffer Operation Timing (Compare Match)

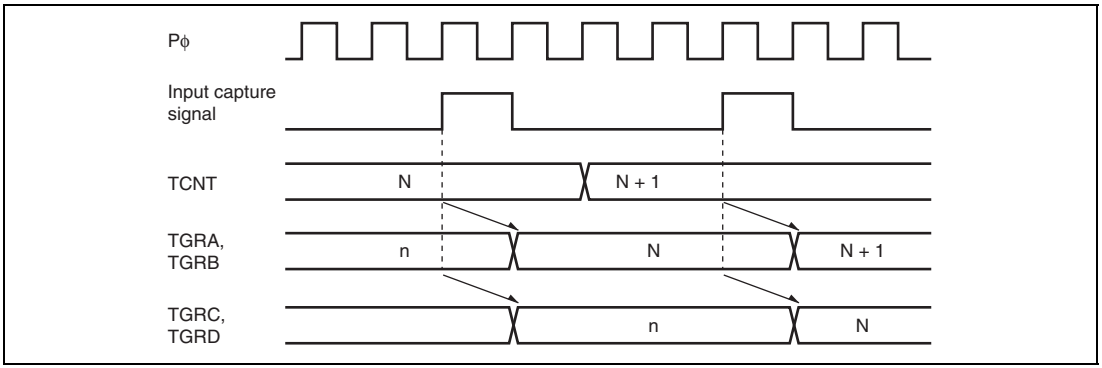


Figure 12.104 Buffer Operation Timing (Input Capture)

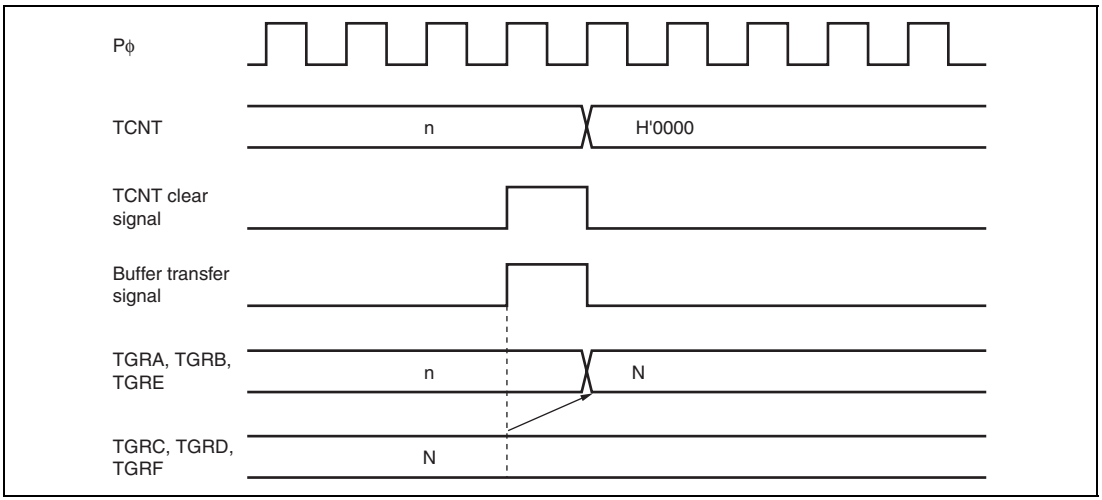


Figure 12.105 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 12.106 to 12.108 show the buffer transfer timing in complementary PWM mode.

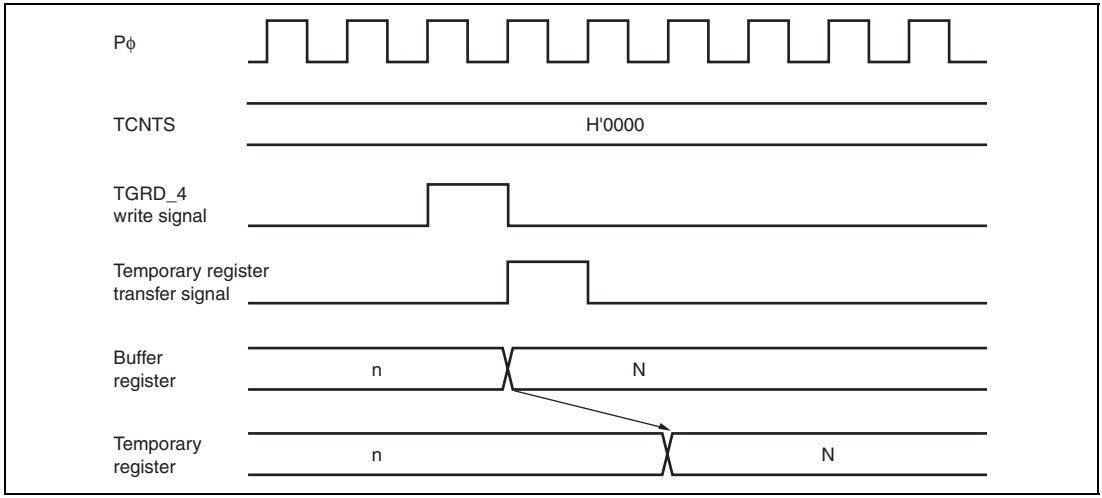


Figure 12.106 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

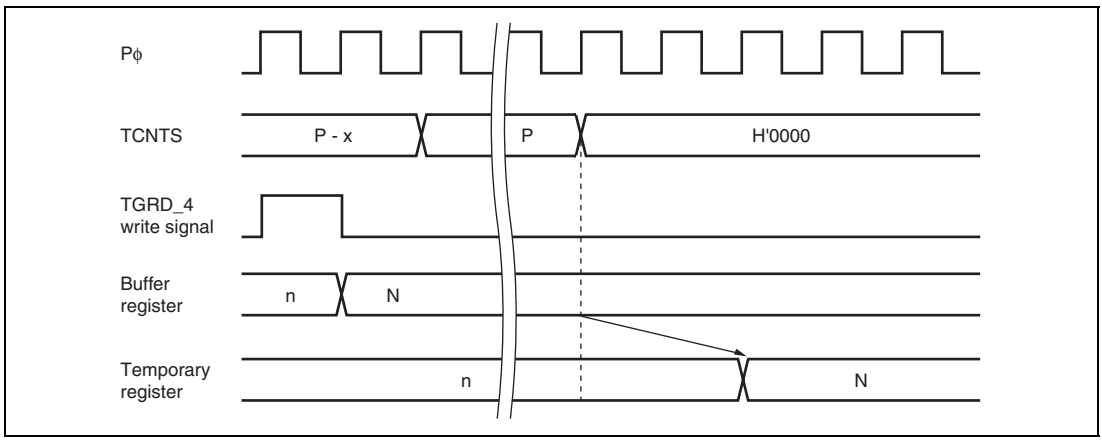


Figure 12.107 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

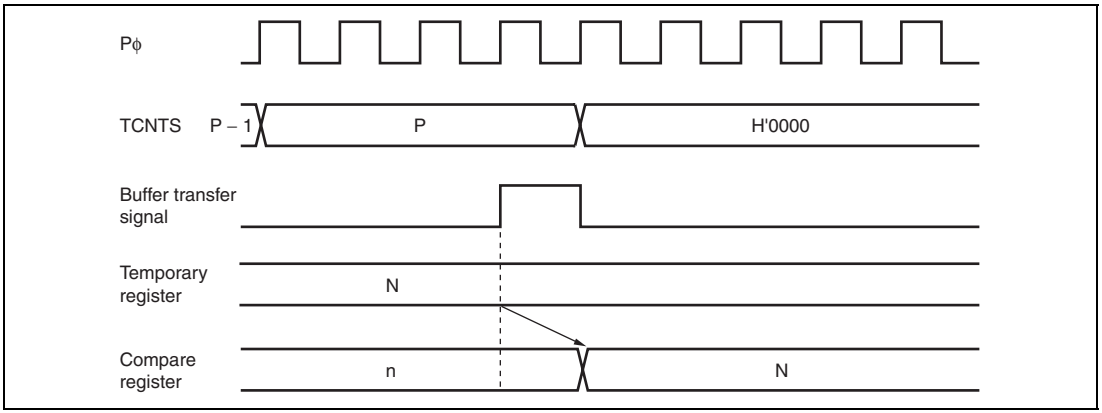


Figure 12.108 Transfer Timing from Temporary Register to Compare Register

12.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figures 12.109 and 110 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

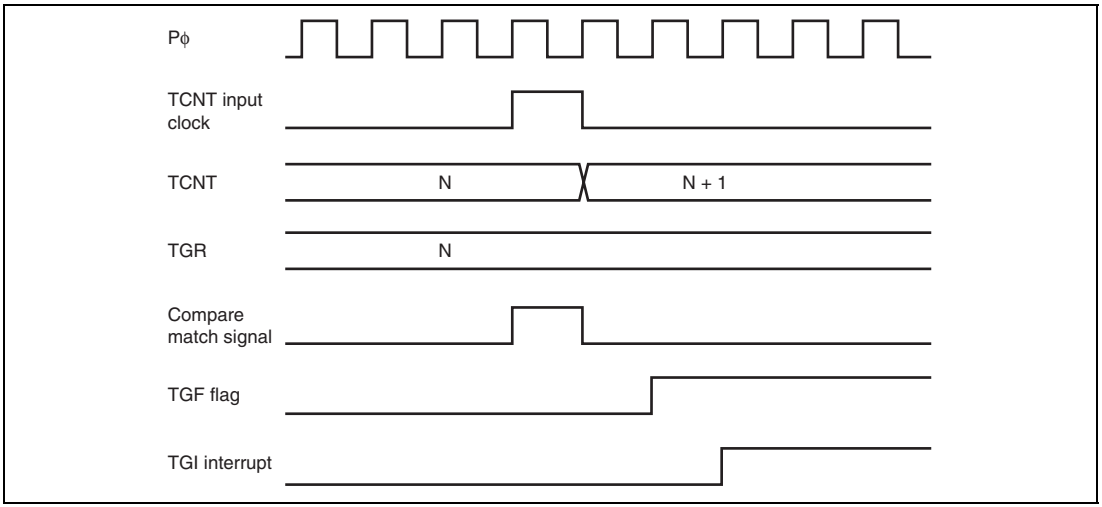


Figure 12.109 TGI Interrupt Timing (Compare Match) (Channels 0 to 4)

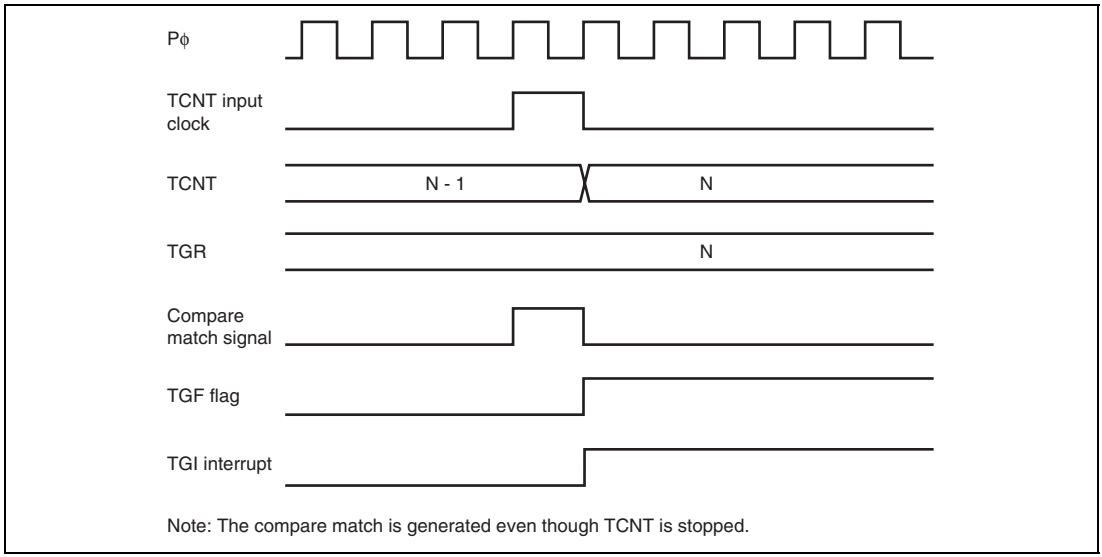


Figure 12.110 TGI Interrupt Timing (Compare Match) (Channel 5)

(2) TGF Flag Setting Timing in Case of Input Capture

Figures 12.111 and 112 show the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

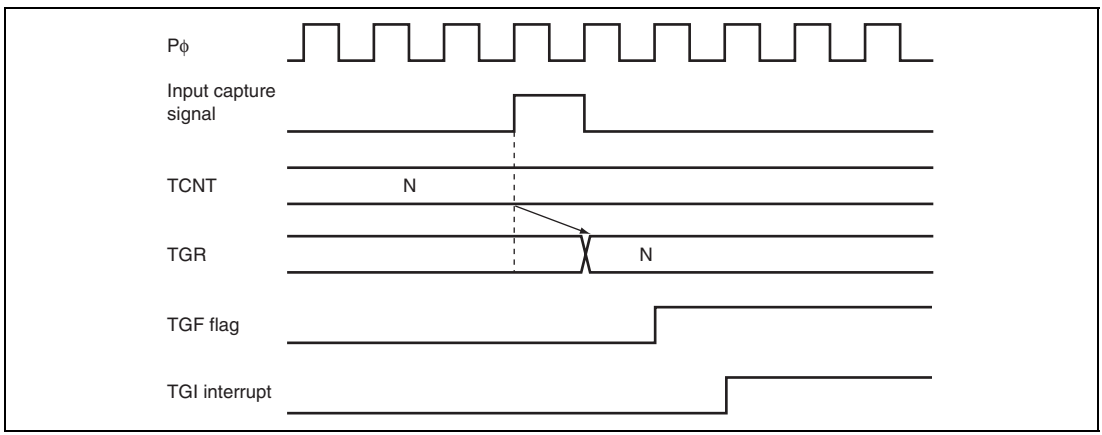


Figure 12.111 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)

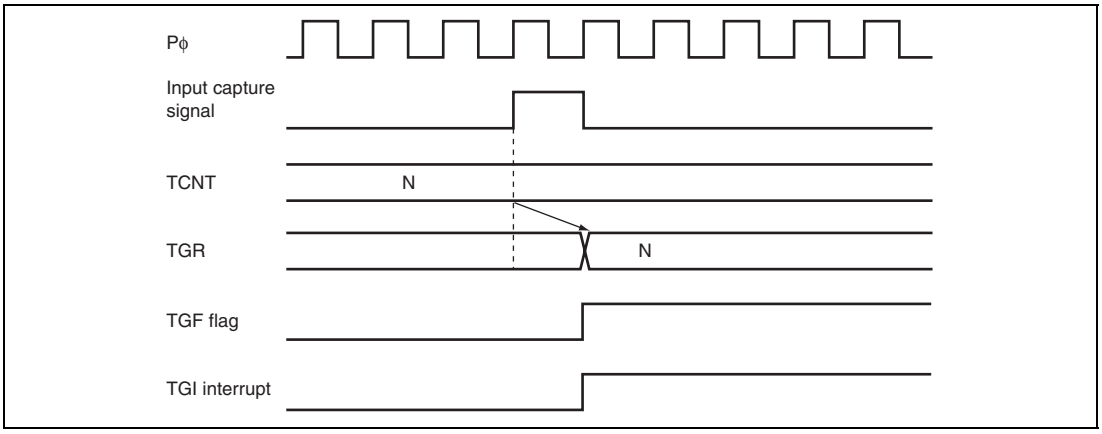


Figure 12.112 TGI Interrupt Timing (Input Capture) (Channel 5)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 12.113 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 12.114 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

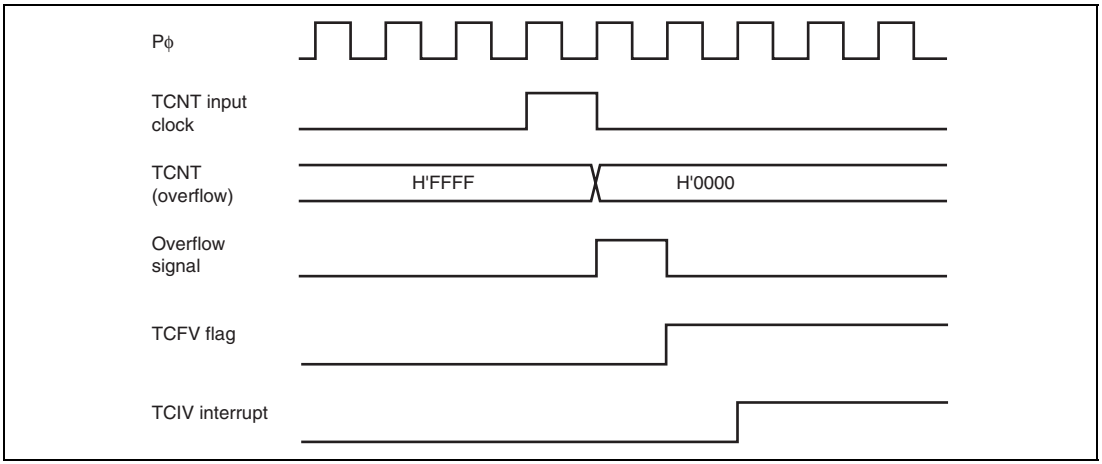


Figure 12.113 TCIV Interrupt Setting Timing

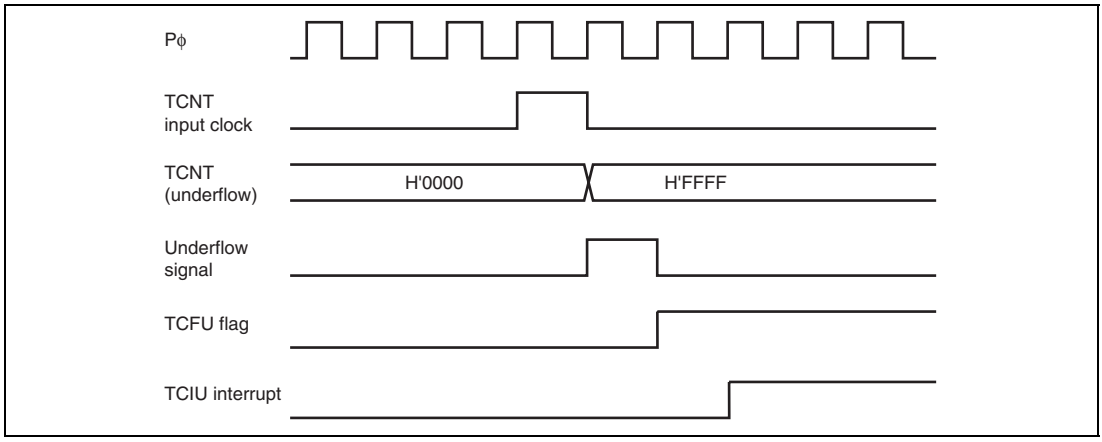


Figure 12.114 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC/DMAC is activated, the flag is cleared automatically. Figures 12.115 and 116 show the timing for status flag clearing by the CPU, and figures 12.117 to 12.119 show the timing for status flag clearing by the DTC/DMAC.

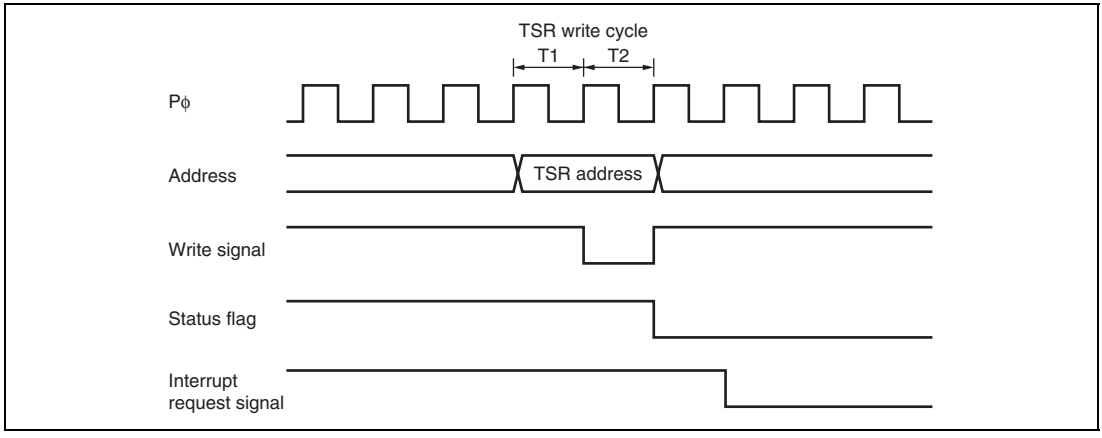


Figure 12.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)

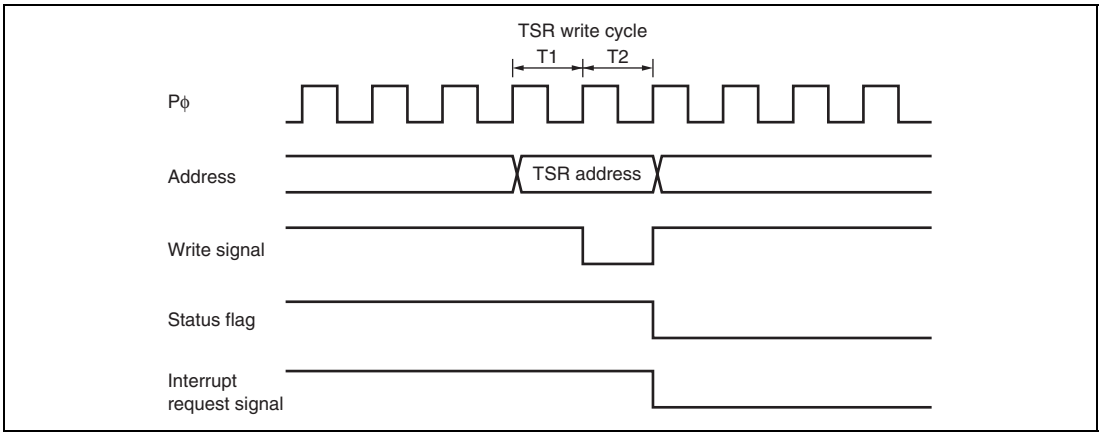


Figure 12.116 Timing for Status Flag Clearing by CPU (Channel 5)

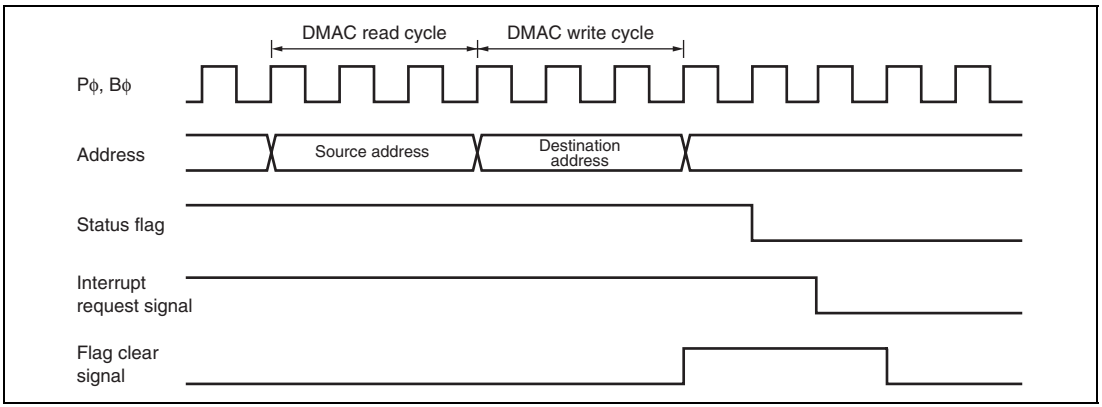


Figure 12.117 Timing for Status Flag Clearing by DTC Activation (Channels 0 to 4)

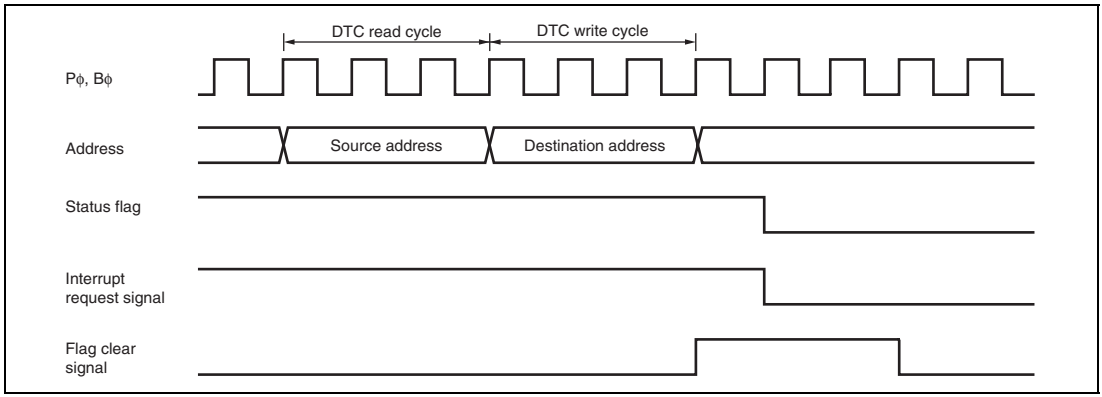


Figure 12.118 Timing for Status Flag Clearing by DTC Activation (Channel 5)

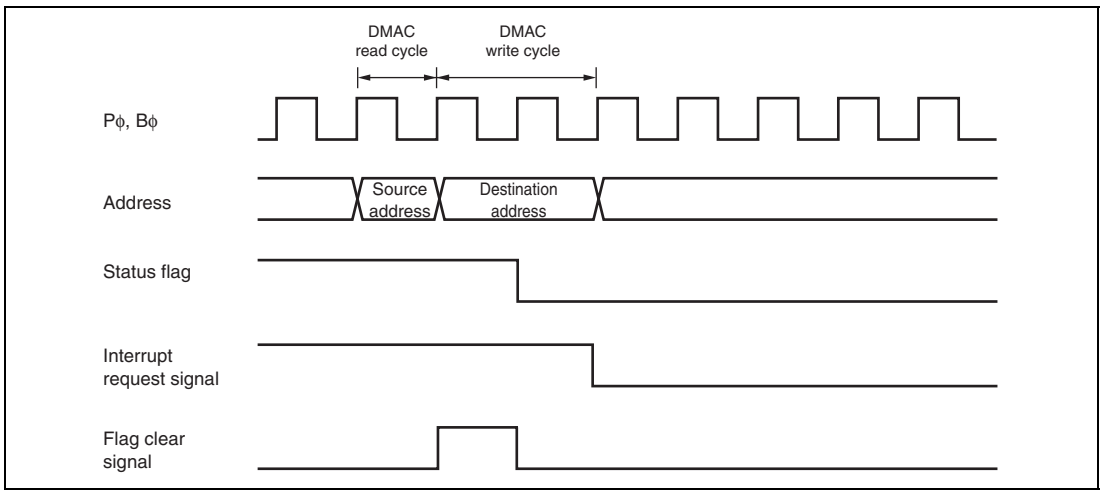


Figure 12.119 Timing for Status Flag Clearing by DMAC Activation

12.7 Usage Notes

12.7.1 Module Standby Mode Setting

MTU2 operation can be disabled or enabled using the standby control register. The initial setting is for MTU2 operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 32, Power-Down Modes.

12.7.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The MTU2 will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 12.120 shows the input clock conditions in phase counting mode.

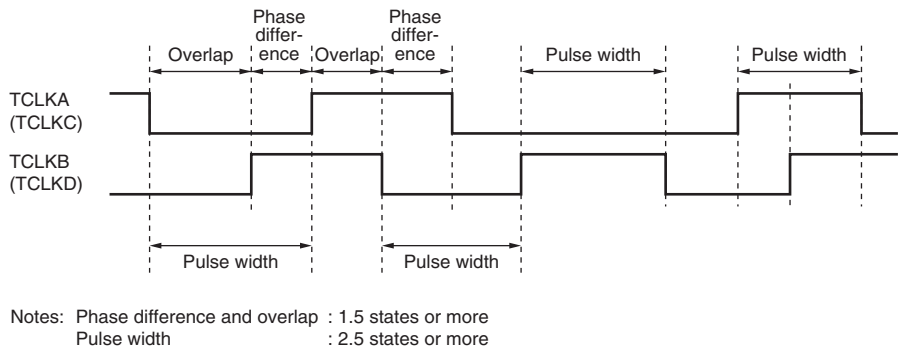


Figure 12.120 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

12.7.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

- Channels 0 to 4

$$f = \frac{P\phi}{(N + 1)}$$

- Channel 5

$$f = \frac{P\phi}{N}$$

Where f: Counter frequency
Pϕ: MTU2 clock operating frequency
N: TGR set value

12.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 12.121 shows the timing in this case.

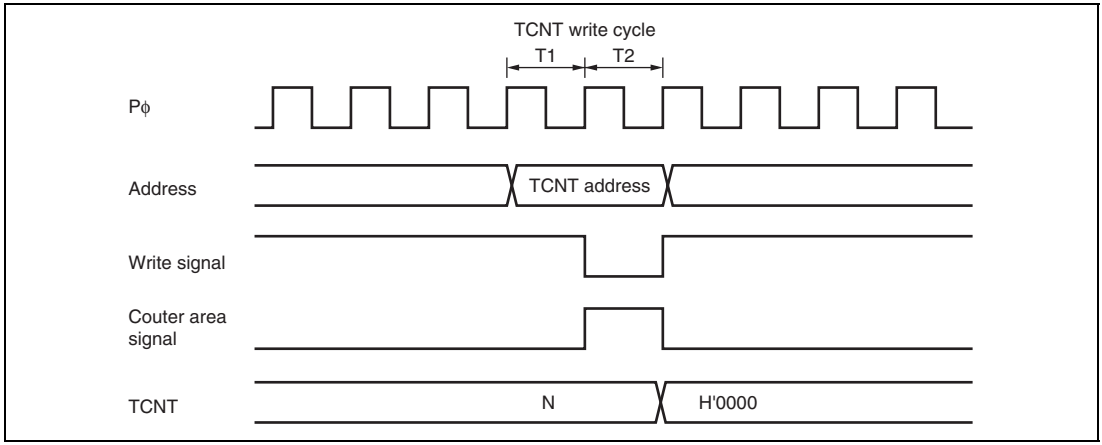


Figure 12.121 Contention between TCNT Write and Clear Operations

12.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 12.122 shows the timing in this case.

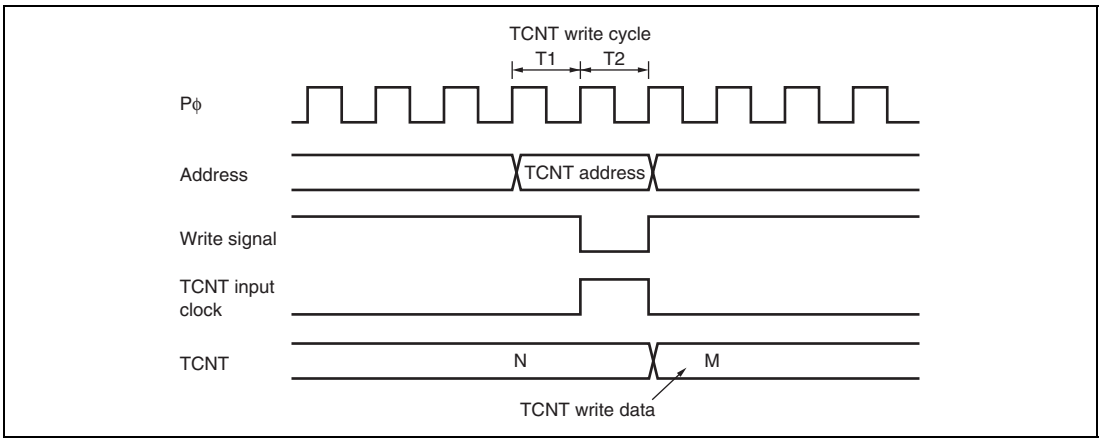


Figure 12.122 Contention between TCNT Write and Increment Operations

12.7.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is also generated.

Figure 12.123 shows the timing in this case.

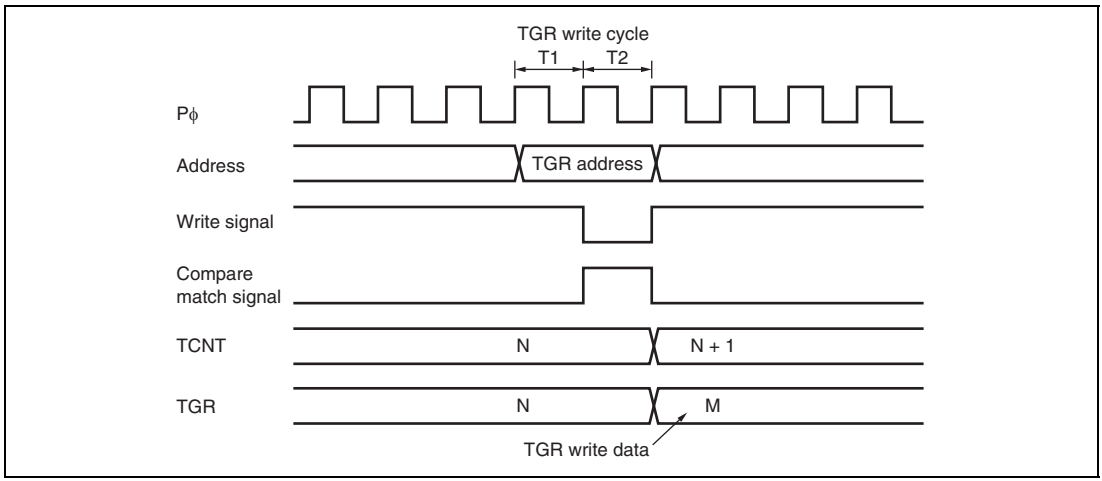


Figure 12.123 Contention between TGR Write and Compare Match

12.7.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data after write.

Figure 12.124 shows the timing in this case.

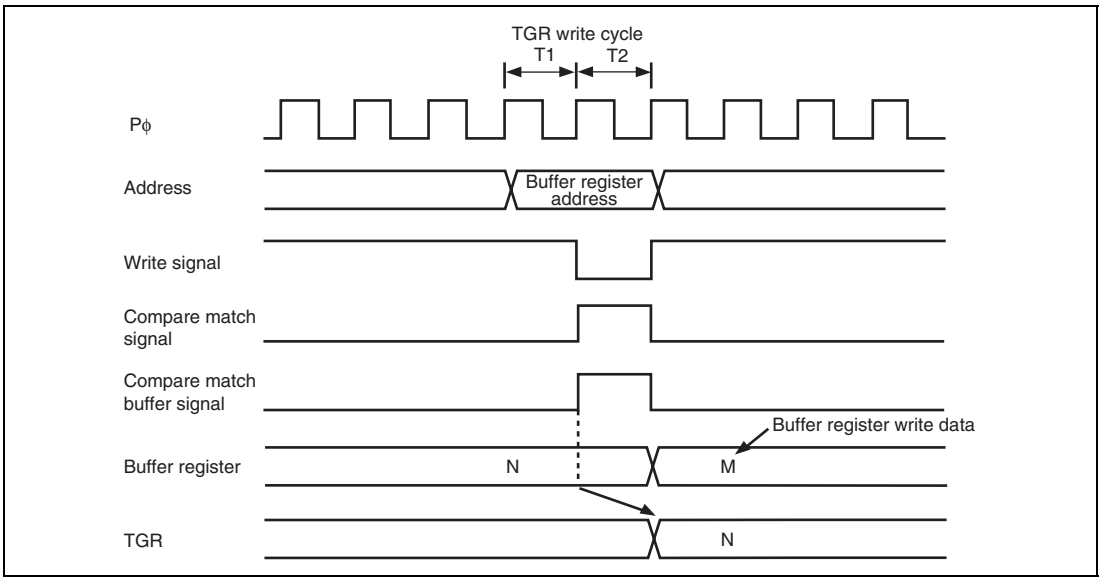


Figure 12.124 Contention between Buffer Register Write and Compare Match

12.7.8 Contention between Buffer Register Write and TCNT Clear

When the buffer transfer timing is set at the TCNT clear by the buffer transfer mode register (TBTM), if TCNT clear occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data before write.

Figure 12.125 shows the timing in this case.

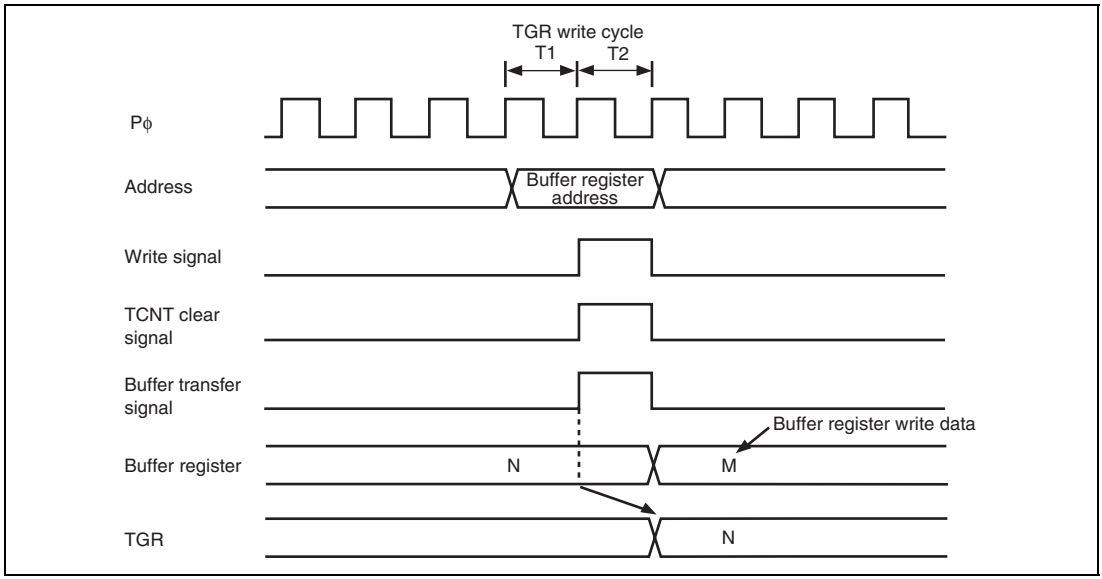


Figure 12.125 Contention between Buffer Register Write and TCNT Clear

12.7.9 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data in the buffer before input capture transfer for channels 0 to 4, and the data after input capture transfer for channel 5.

Figures 12.126 and 127 show the timing in this case.

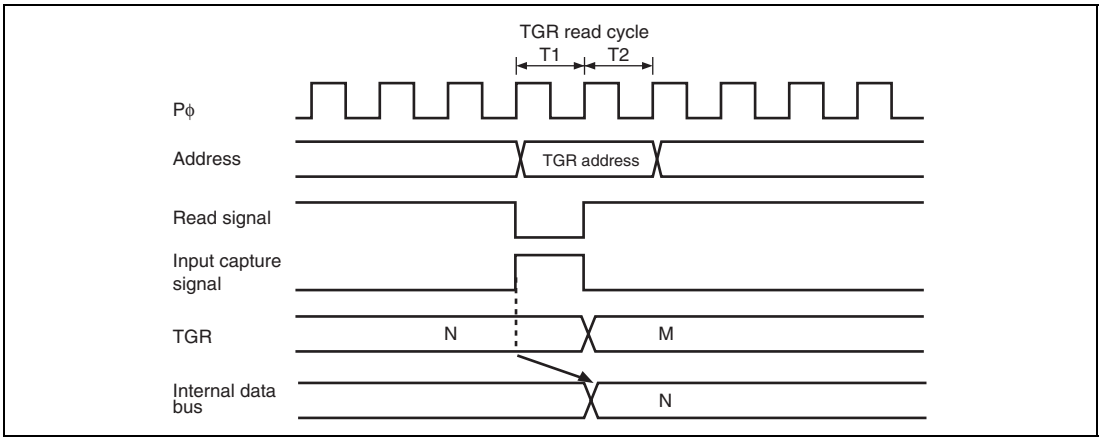


Figure 12.126 Contention between TGR Read and Input Capture (Channels 0 to 4)

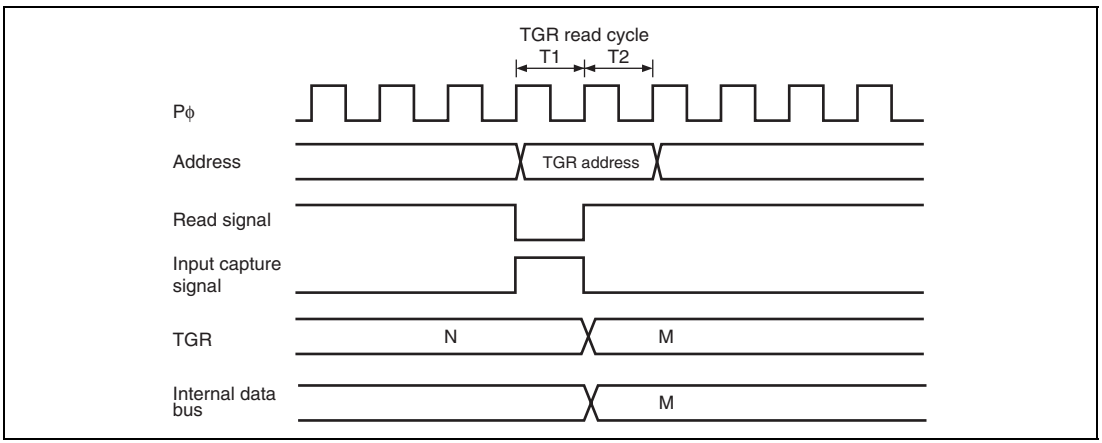


Figure 12.127 Contention between TGR Read and Input Capture (Channel 5)

12.7.10 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed for channels 0 to 4. For channel 5, write to TGR is performed and the input capture signal is generated.

Figures 12.128 and 129 show the timing in this case.

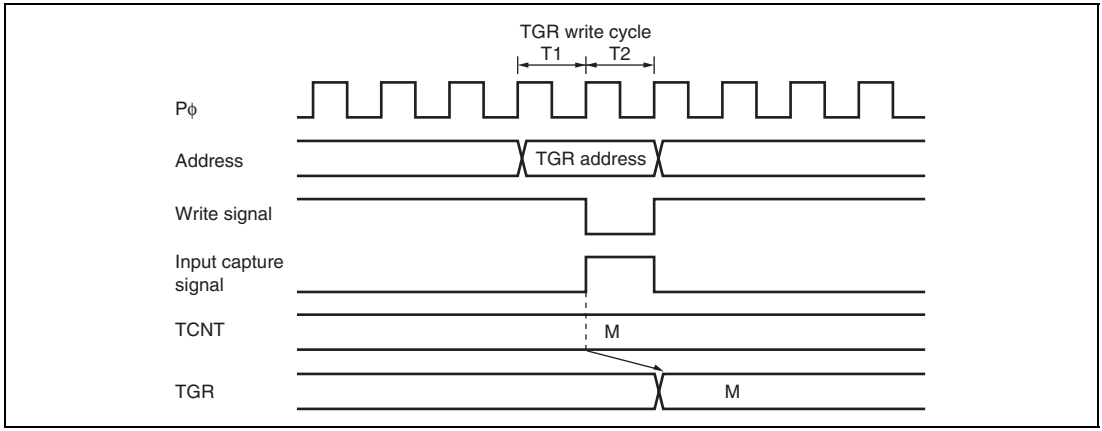


Figure 12.128 Contention between TGR Write and Input Capture (Channels 0 to 4)

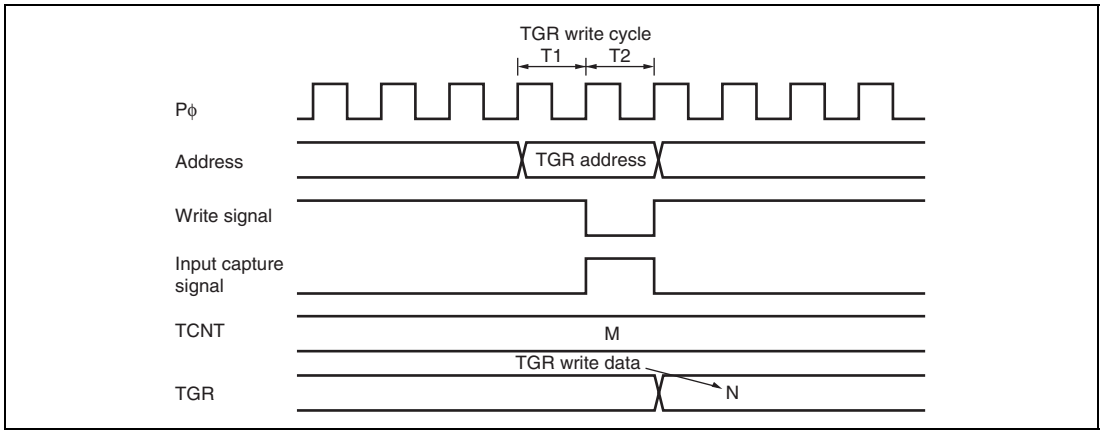


Figure 12.129 Contention between TGR Write and Input Capture (Channel 5)

12.7.11 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 12.130 shows the timing in this case.

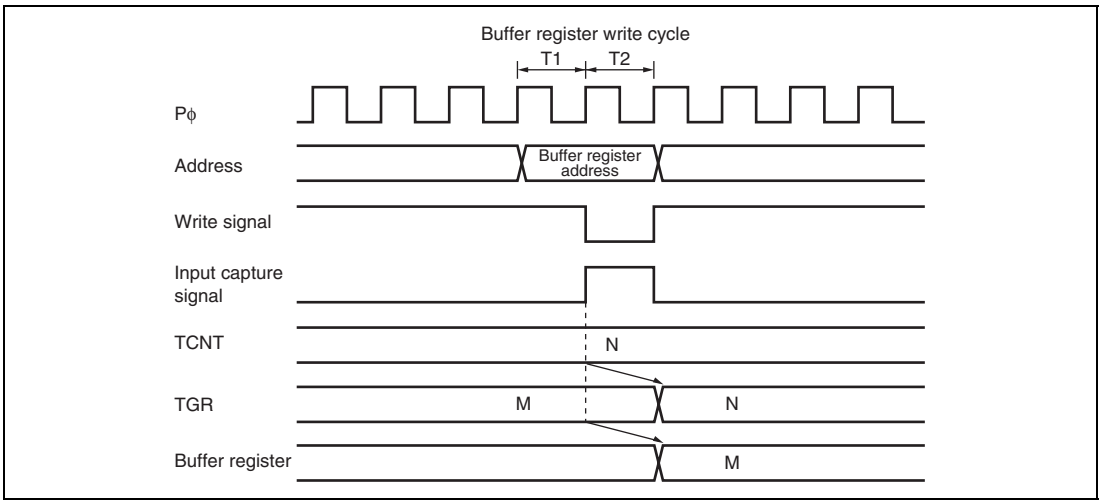


Figure 12.130 Contention between Buffer Register Write and Input Capture

12.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention occurs during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T_2 state of the TCNT_2 write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disabled. At this point, if there is match with TGRA_1 and the TCNT_1 value, a compare signal is issued. Furthermore, when the TCNT_1 count clock is selected as the input capture source of channel 0, TGRA_0 to D_0 carry out the input capture operation. In addition, when the compare match/input capture is selected as the input capture source of TGRB_1, TGRB_1 carries out input capture operation. The timing is shown in figure 12.131.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

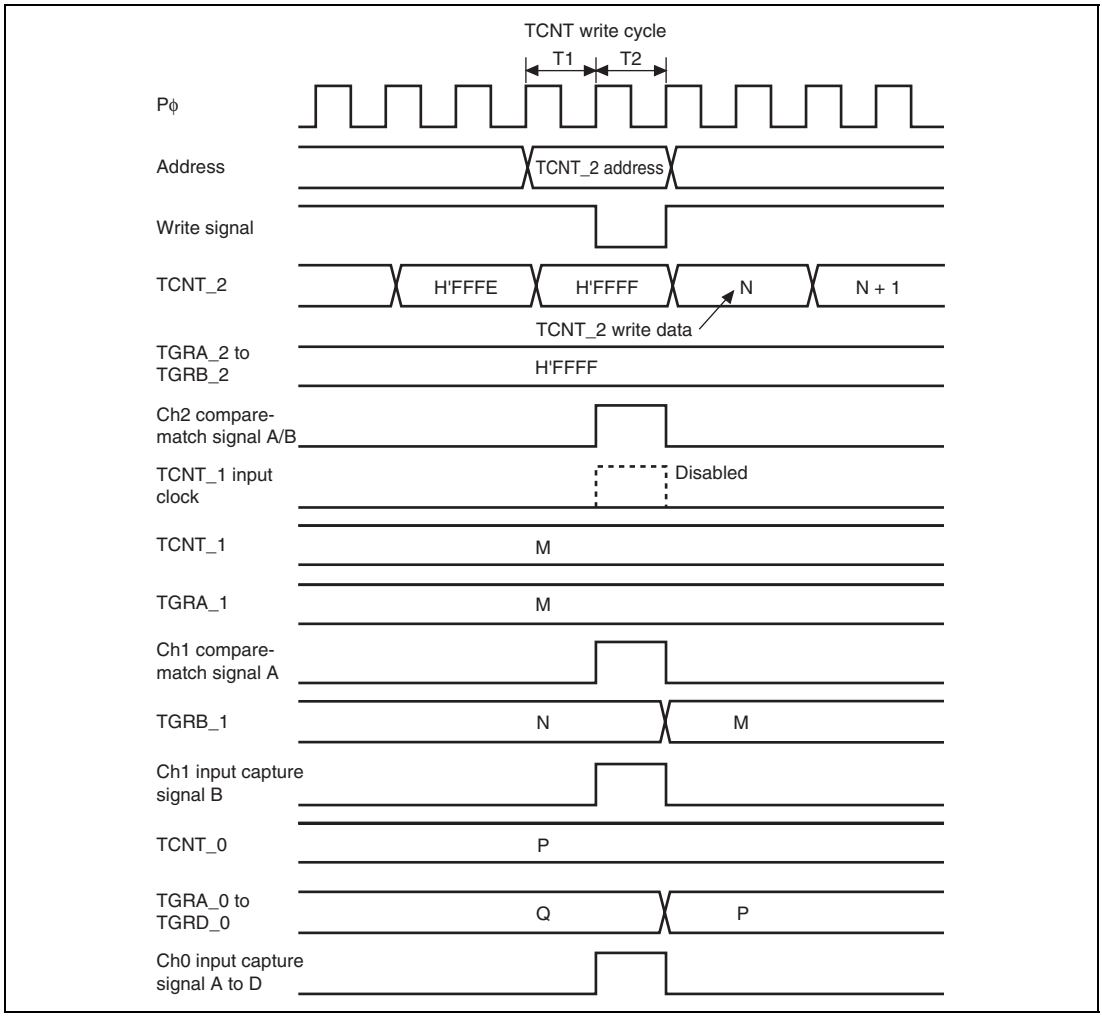


Figure 12.131 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

12.7.13 Counter Value during Complementary PWM Mode Stop

When counting operation is suspended with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is held at H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in figure 12.132.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

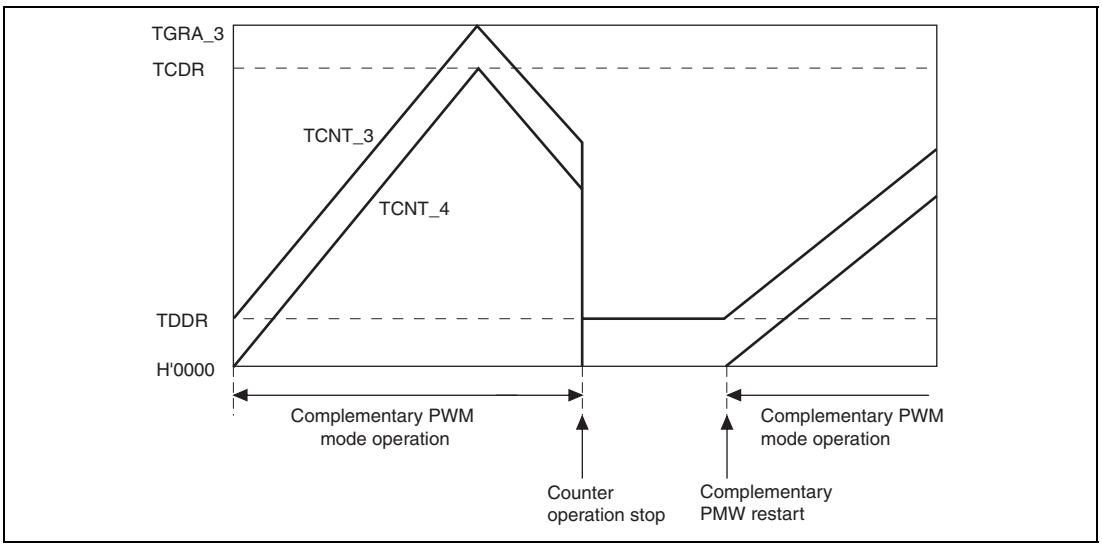


Figure 12.132 Counter Value during Complementary PWM Mode Stop

12.7.14 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_3, TGRA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 functions as a buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4, and TCBR functions as the TCDR's buffer register.

12.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR_4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR_4 is set to 1.

In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR_3. For example, if the BFA bit of TMDR_3 is set to 1, TGRC_3 functions as the buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4.

The TGFC bit and TGFD bit of TSR_3 and TSR_4 are not set when TGRC_3 and TGRD_3 are operating as buffer registers.

Figure 12.133 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4, with TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

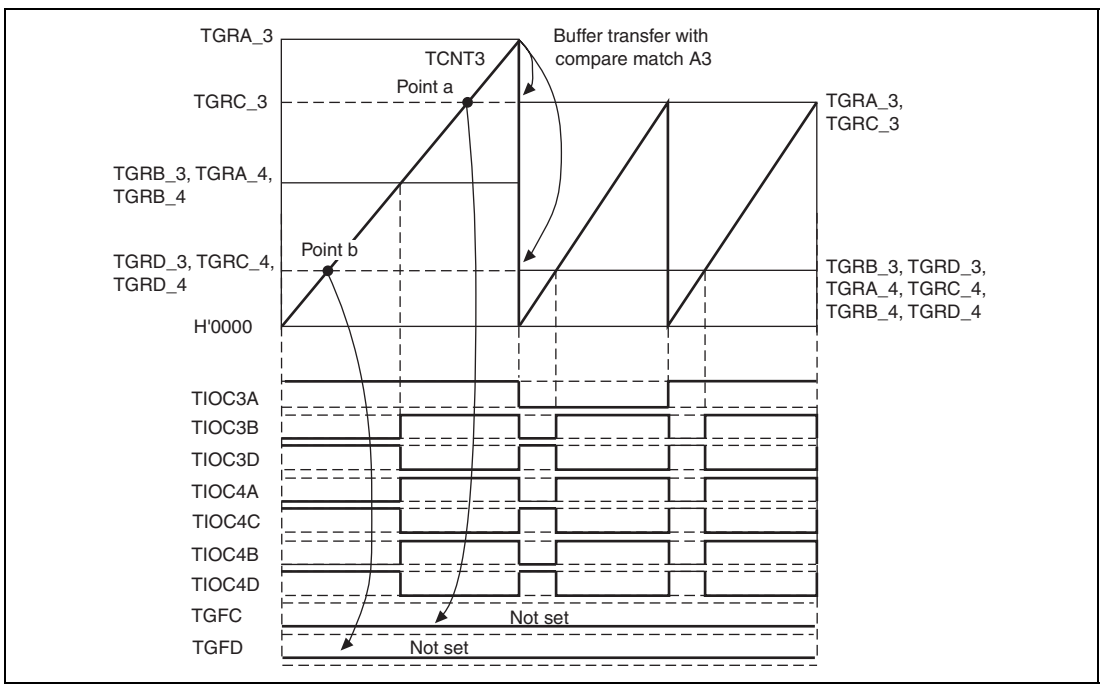


Figure 12.133 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

12.7.16 Overflow Flags in Reset Synchronous PWM Mode

When set to reset synchronous PWM mode, TCNT_3 and TCNT_4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT_4's count clock source and count edge obey the TCR_3 setting.

In reset synchronous PWM mode, with cycle register TGRA_3's set value at H'FFFF, when specifying TGR3A compare-match for the counter clear source, TCNT_3 and TCNT_4 count up to H'FFFF, then a compare-match occurs with TGRA_3, and TCNT_3 and TCNT_4 are both cleared. At this point, TSR's overflow flag TCFV bit is not set.

Figure 12.134 shows a TCFV bit operation example in reset synchronous PWM mode with a set value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been specified without synchronous setting for the counter clear source.

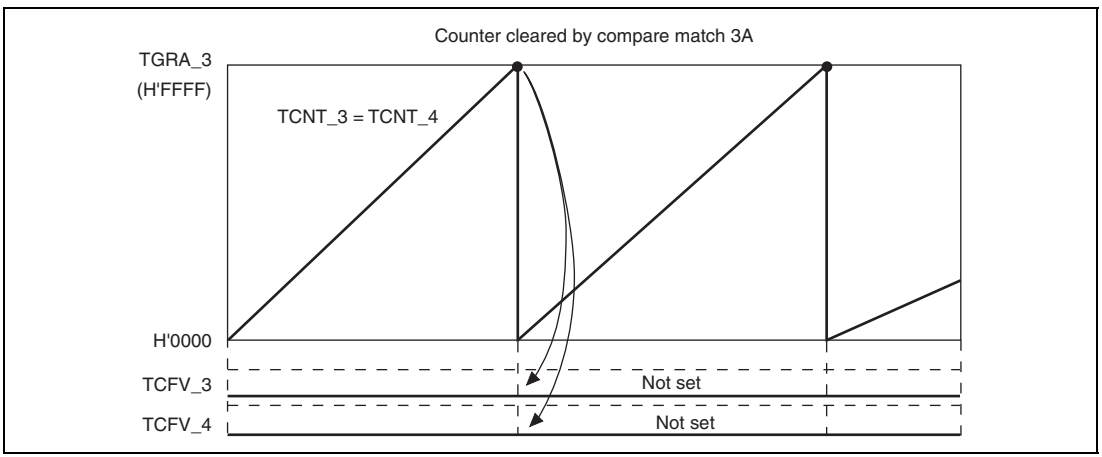


Figure 12.134 Reset Synchronous PWM Mode Overflow Flag

12.7.17 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 12.135 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

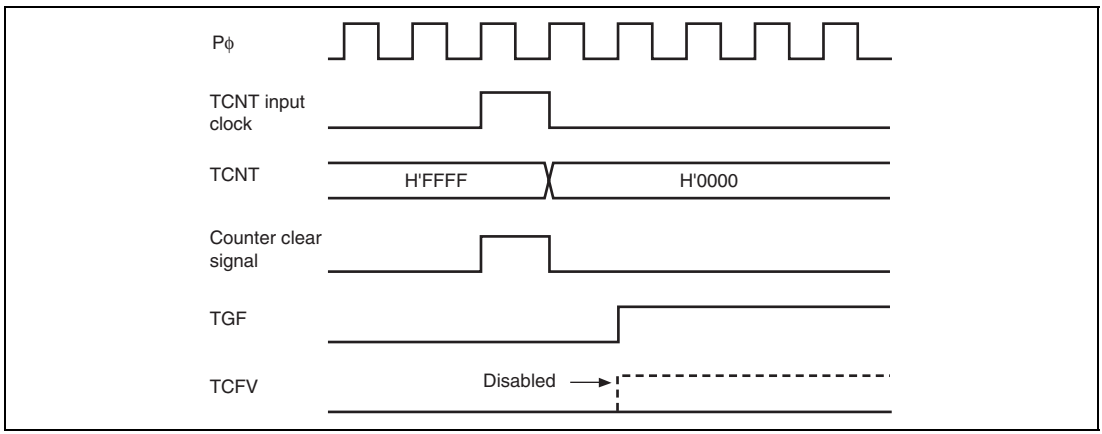


Figure 12.135 Contention between Overflow and Counter Clearing

12.7.18 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 12.136 shows the operation timing when there is contention between TCNT write and overflow.

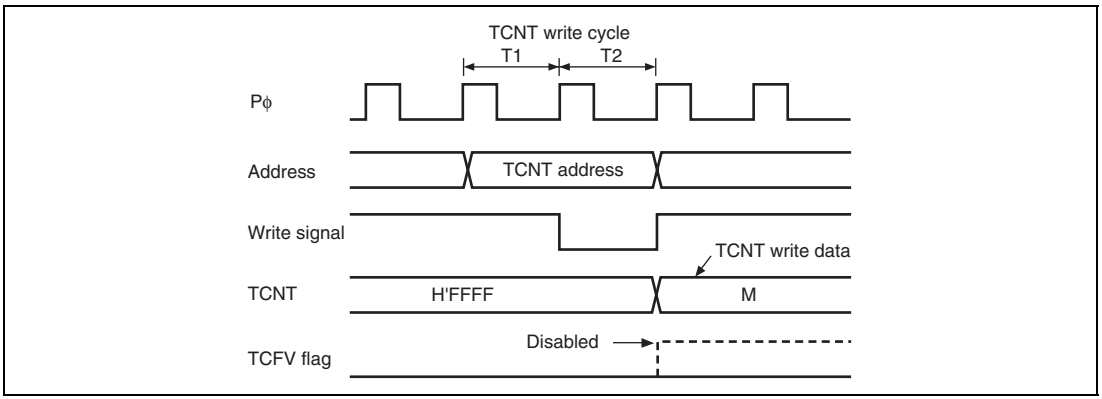


Figure 12.136 Contention between TCNT Write and Overflow

12.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transition to reset-synchronized PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronized PWM mode, write H'11 to registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronized PWM mode.

12.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to H'00.

12.7.21 Interrupts in Module Standby Mode

If module standby mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC activation source. Interrupts should therefore be disabled before entering module standby mode.

12.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection

When timer counters 1 and 2 (TCNT_1 and TCNT_2) are operated as a 32-bit counter in cascade connection, the cascade counter value cannot be captured successfully even if input-capture input is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is because the input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same when external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchronization with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not capture the count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures the count value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 = H'0000 should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the values of TCNT_1 = H'FFF0 and TCNT_2 = H'0000 are erroneously transferred.

The MTU2 has a new function that allows simultaneous capture of TCNT_1 and TCNT_2 with a single input-capture as the trigger. This function allows reading of the 32-bit counter such that TCNT_1 and TCNT_2 are captured at the same time. For details, see section 12.3.8, Timer Input Capture Control Register (TICCR).

12.7.23 Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Satisfaction of either of the following conditions (1 or 2) while output waveform control is effective at the time of synchronous counter clearing in complementary PWM mode (WRE = 1 in TWCR) leads to effects (1 and 2). Therefore, use this countermeasure to avoid these effects.

Condition 1: Synchronous clearing of the PWM output occurs during the dead time portion of the suppression interval 10 for initial output (see figure 12.137).

Condition 2: Synchronous clearing occurs while any of TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, and TGRB_4 ≤ TDDR is met in suppression interval 10 or 11 for initial output (see figure 12.138).

Effect 1: The dead time on the PWM output pin is shortened (or omitted).

Effect 2: The active level is output on the PWM negative-phase output pin beyond the output period for the active level.

Countermeasure: Perform synchronous clearing while TGRB_3 ≥ TDDR × 2, TGRA_4 ≥ TDDR × 2, and TGRB_4 ≥ TDDR × 2 are all met.

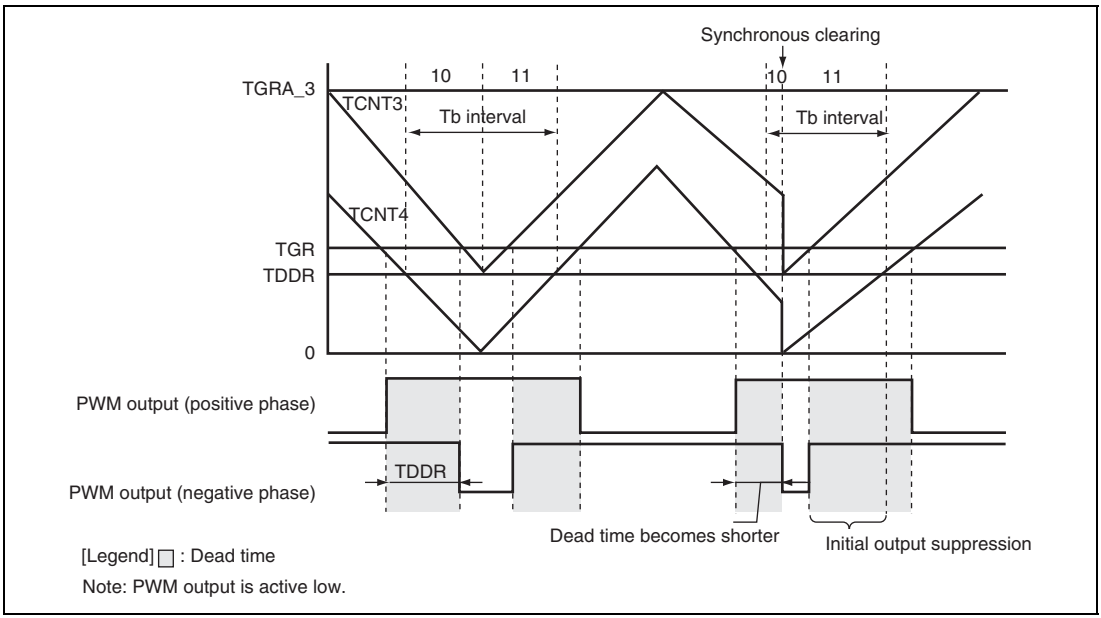


Figure 12.137 Example of Synchronous Clearing for Condition 1

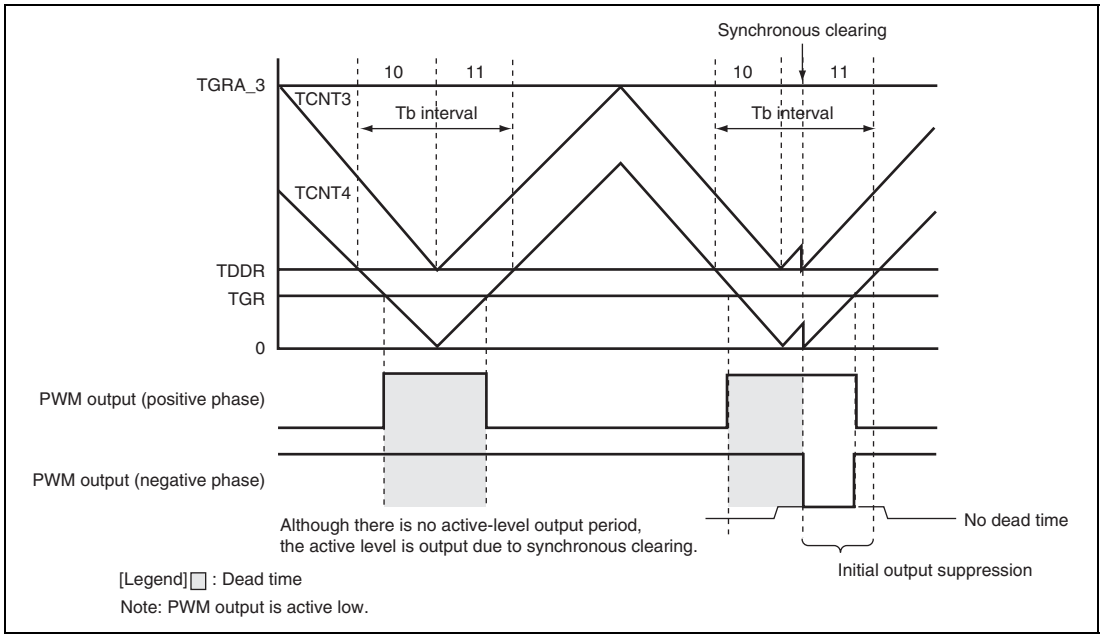


Figure 12.138 Example of Synchronous Clearing for Condition 2

12.8 MTU2 Output Pin Initialization

12.8.1 Operating Modes

The MTU2 has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 to 4)
- PWM mode 1 (channels 0 to 4)
- PWM mode 2 (channels 0 to 2)
- Phase counting modes 1 to 4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronized PWM mode (channels 3 and 4)

The MTU2 output pin initialization method for each of these modes is described in this section.

12.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a reset and in standby mode. Since MTU2 pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU2 pin states at that point are output to the ports. When MTU2 output is selected by the PFC immediately after a reset, the MTU2 output initial level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the PFC setting should be made after initialization of the MTU2 output pins is completed.

Note: Channel number and port notation are substituted for *.

12.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc.

If an error occurs during MTU2 operation, MTU2 output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. For large-current pins, output can also be cut by hardware, using port output enable (POE). The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU2 has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in table 12.59.

Table 12.59 Mode Transition Combinations

			After			
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPWM**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23) (24)	(25)
RPWM	(26)	(27)	None	None	(28)	(29)

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

12.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Note: Channel number is substituted for * indicated in this article.

Pin initialization procedures are described below for the numbered combinations in table 12.59. The active level is assumed to be low.

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode

Figure 12.139 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

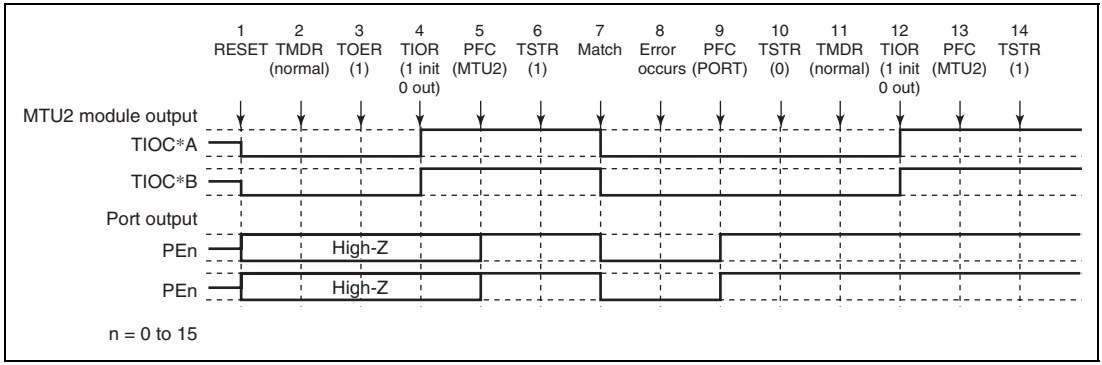


Figure 12.139 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(2) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 12.140 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

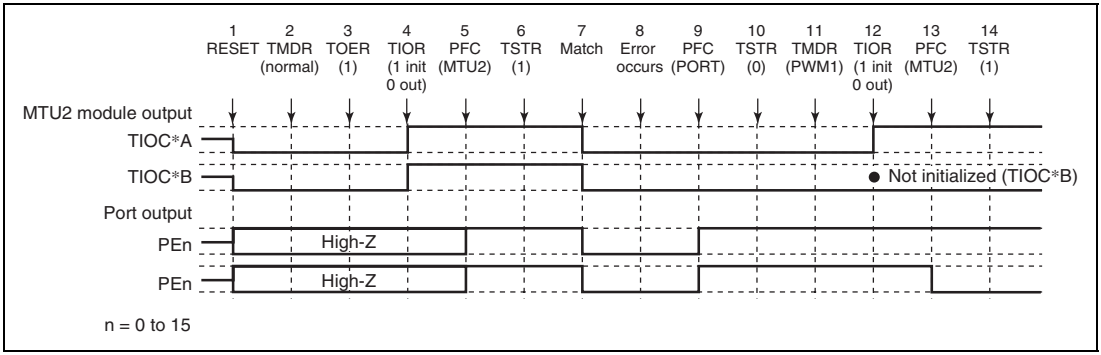


Figure 12.140 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.139.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(3) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 12.141 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

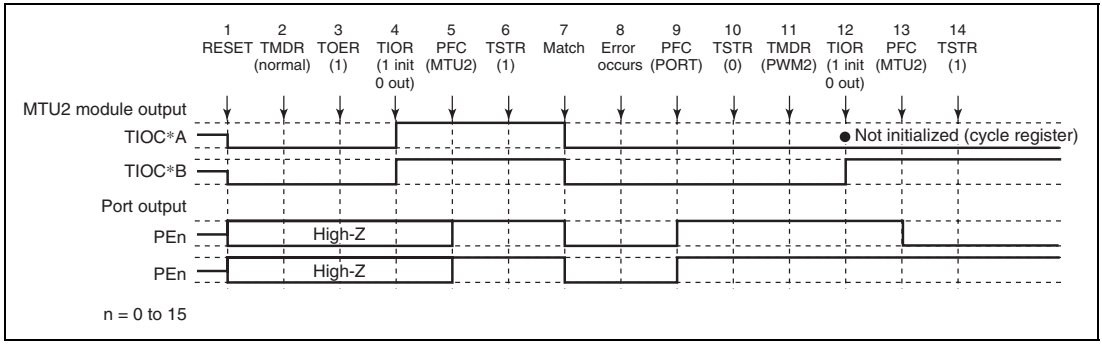


Figure 12.141 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.139.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 12.142 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

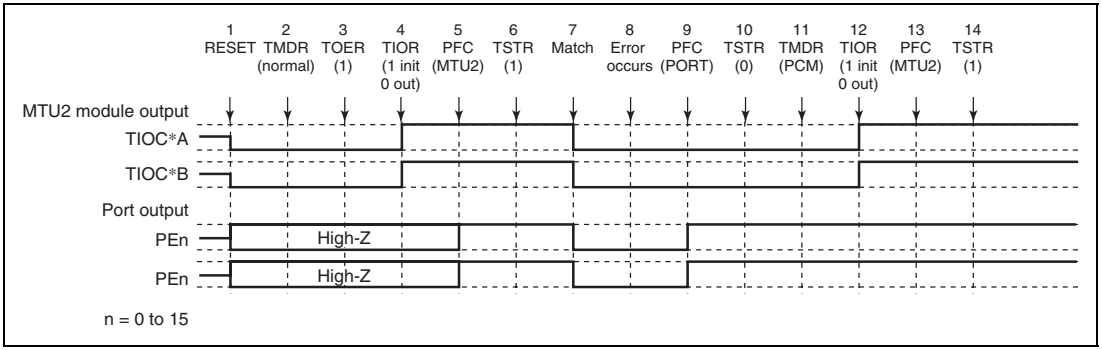


Figure 12.142 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.139.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(5) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 12.143 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

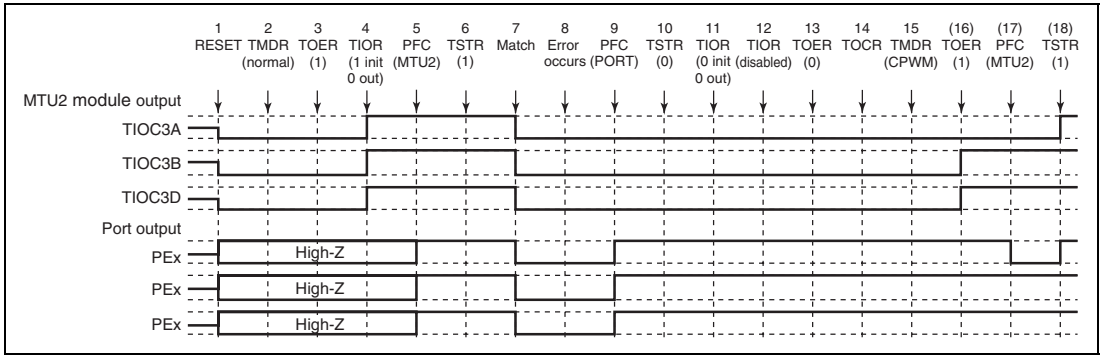


Figure 12.143 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.139.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 output with the PFC.
18. Operation is restarted by TSTR.

(6) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 12.144 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

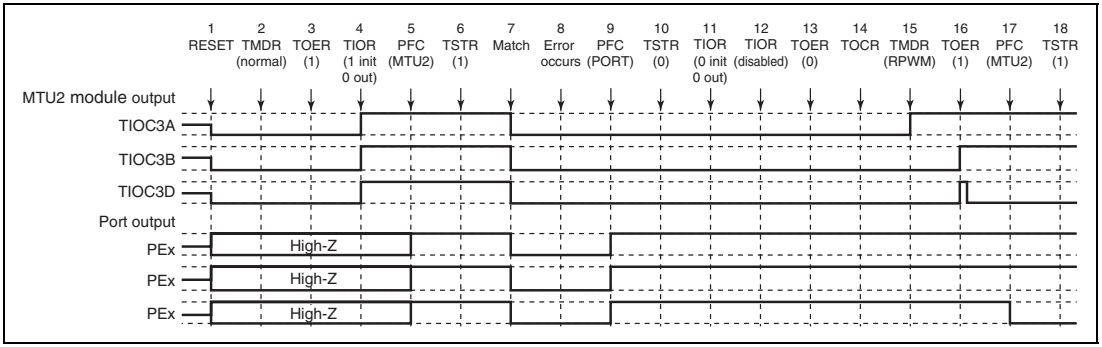


Figure 12.144 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 12.139.

14. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronized PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 output with the PFC.
18. Operation is restarted by TSTR.

(7) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode

Figure 12.145 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

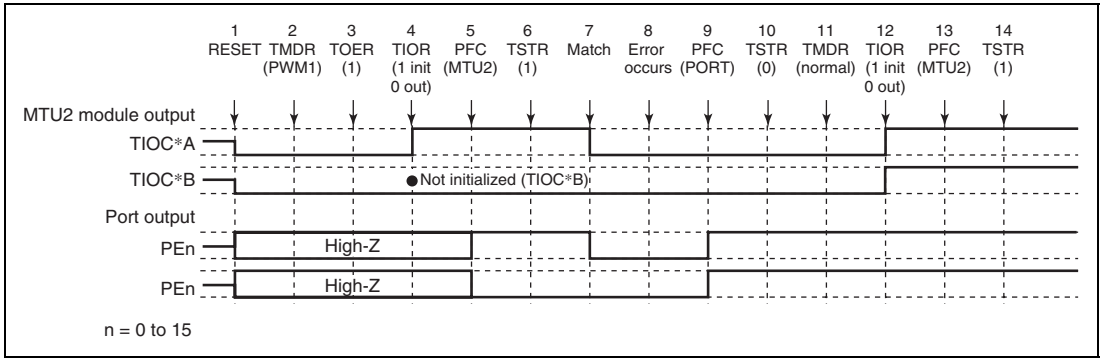


Figure 12.145 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(8) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1

Figure 12.146 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

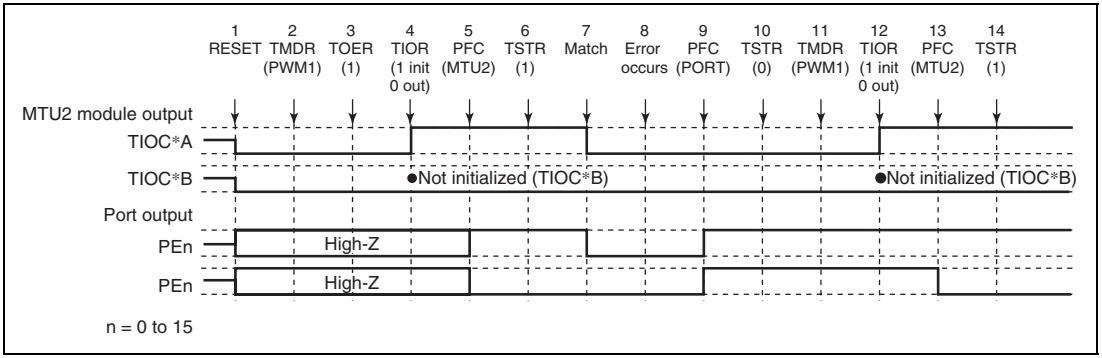


Figure 12.146 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.145.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(9) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2

Figure 12.147 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

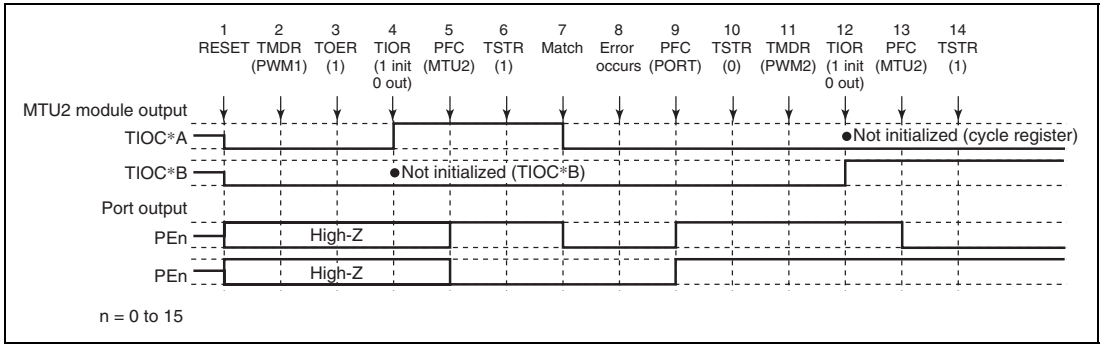


Figure 12.147 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.145.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

(10) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode

Figure 12.148 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

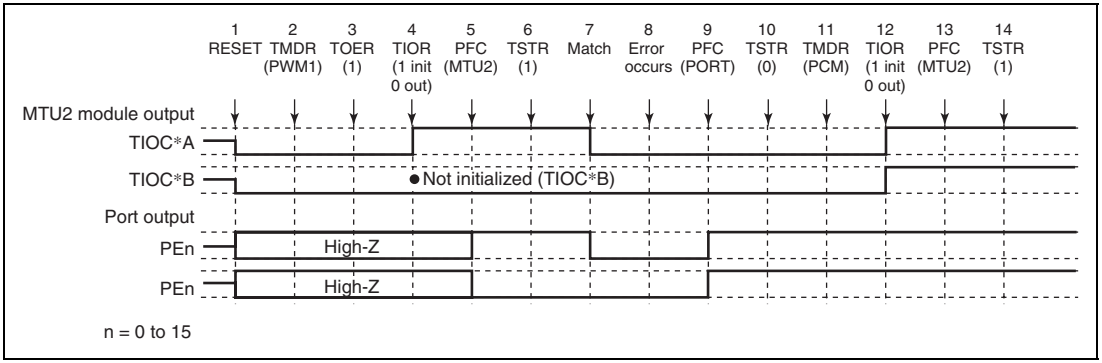


Figure 12.148 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.145.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(11) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode

Figure 12.149 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

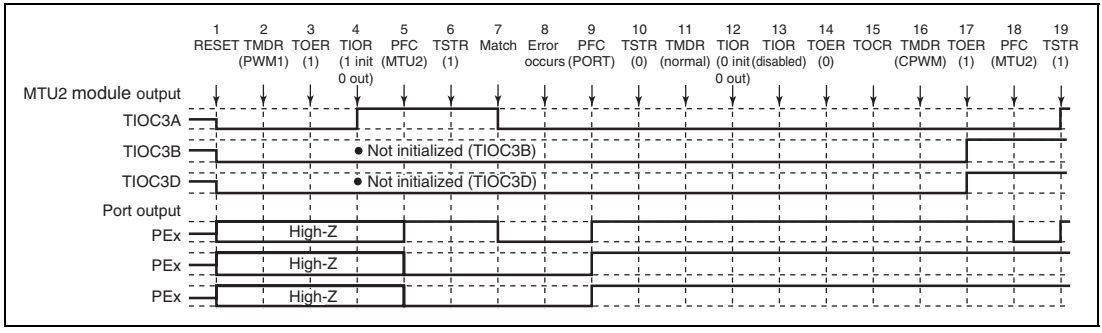


Figure 12.149 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.145.

11. Set normal mode for initialization of the normal mode waveform generation section.
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 output with the PFC.
19. Operation is restarted by TSTR.

(12) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 12.150 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

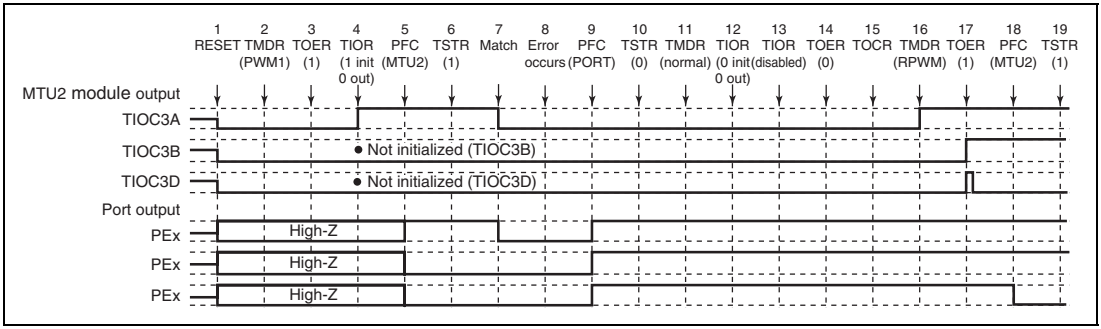


Figure 12.150 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 12.149.

15. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronized PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 output with the PFC.
19. Operation is restarted by TSTR.

(13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode

Figure 12.151 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

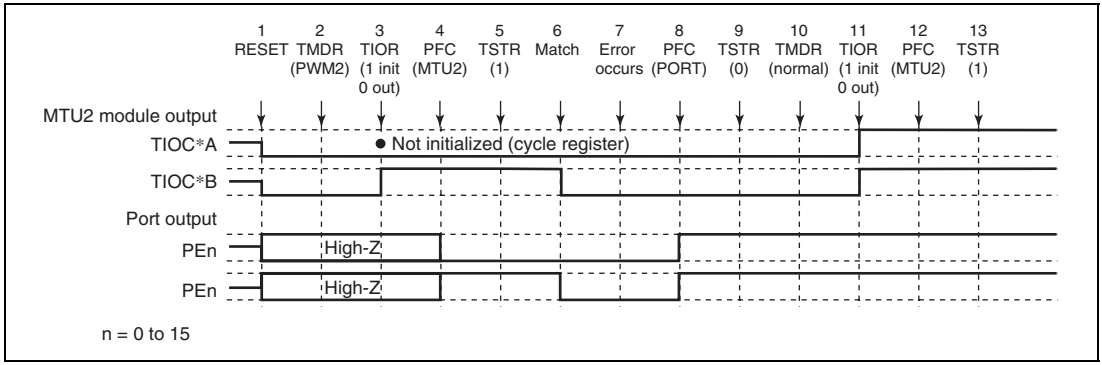


Figure 12.151 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(14) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1

Figure 12.152 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

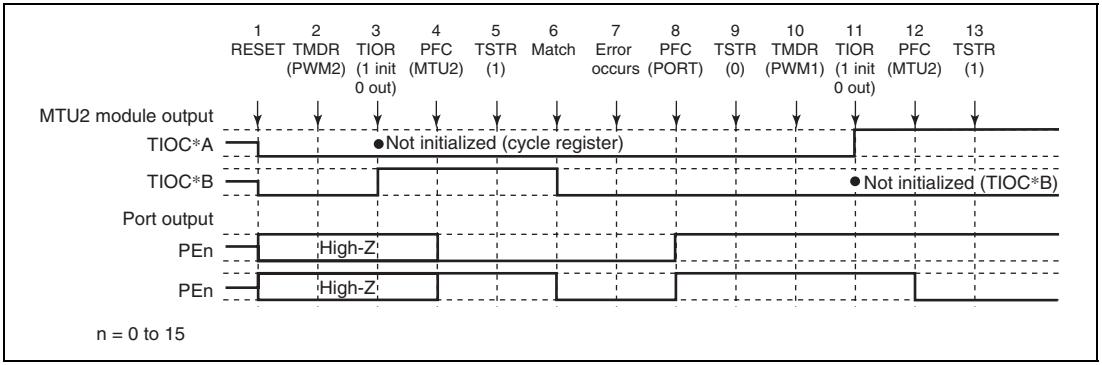


Figure 12.152 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.151.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(15) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2

Figure 12.153 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

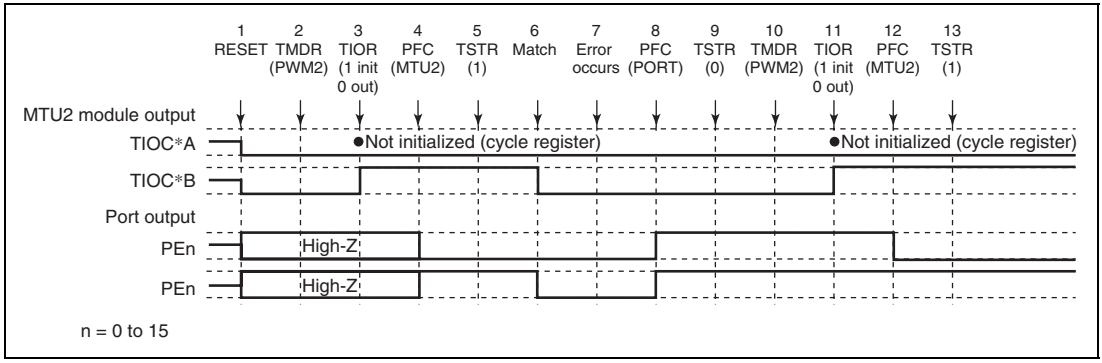


Figure 12.153 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.151.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(16) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode

Figure 12.154 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

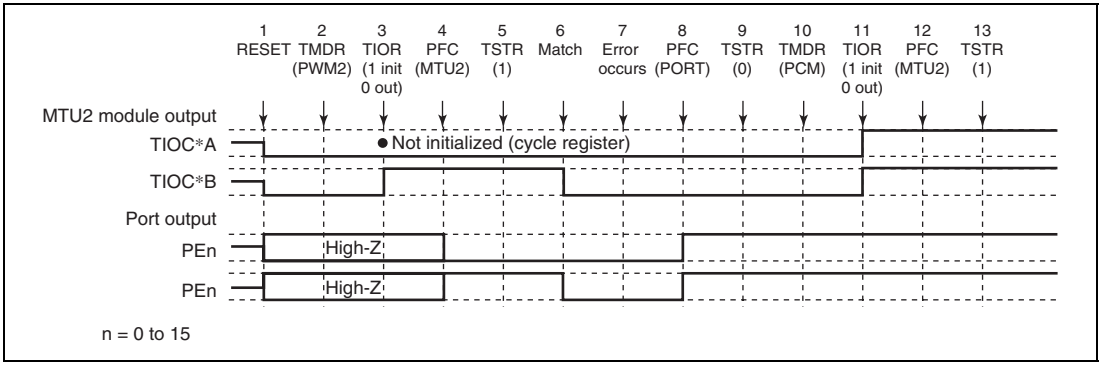


Figure 12.154 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.151.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode

Figure 12.155 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

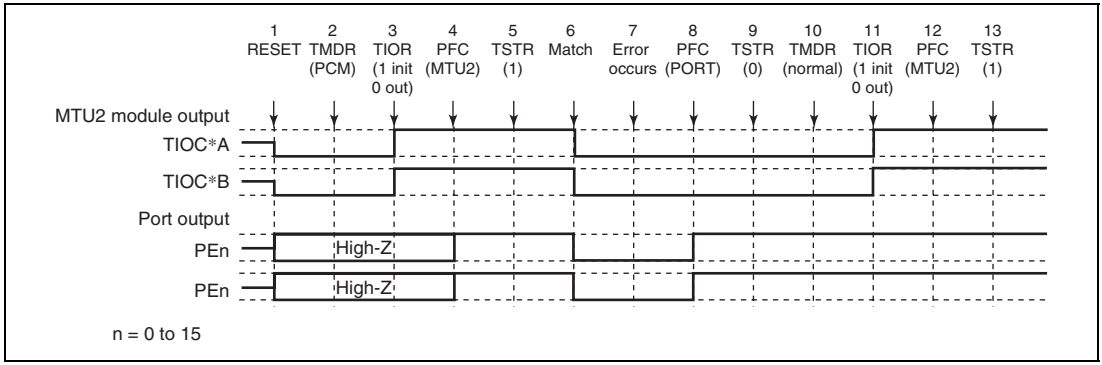


Figure 12.155 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(18) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 12.156 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

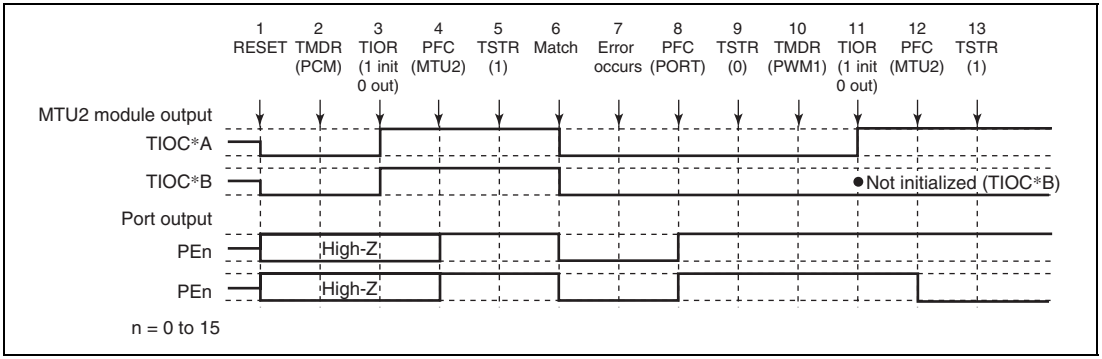


Figure 12.156 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.155.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(19) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 12.157 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

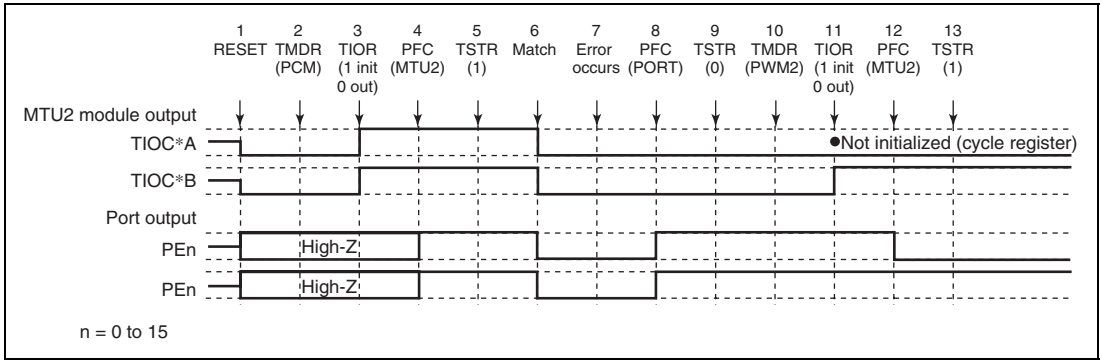


Figure 12.157 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.155.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(20) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 12.158 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

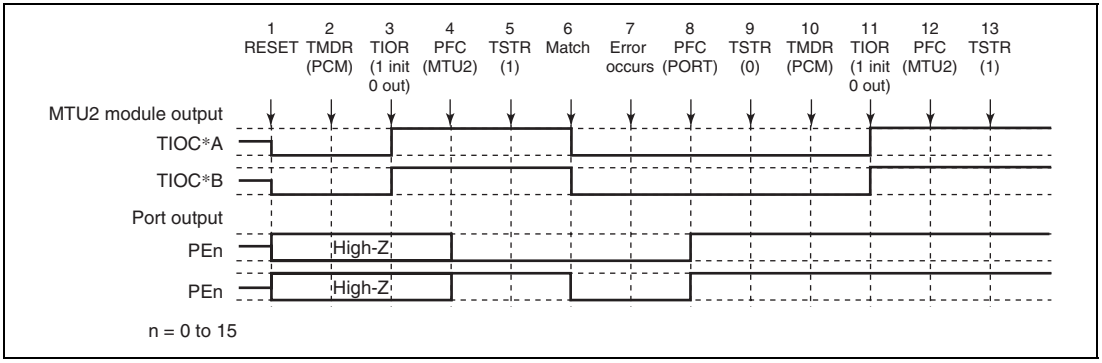


Figure 12.158 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.155.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 12.159 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

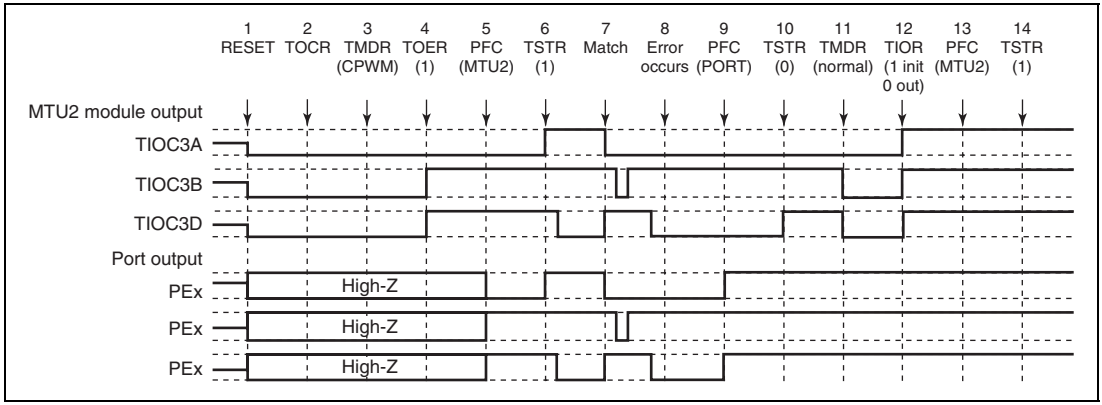


Figure 12.159 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU2 output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(22) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 12.160 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

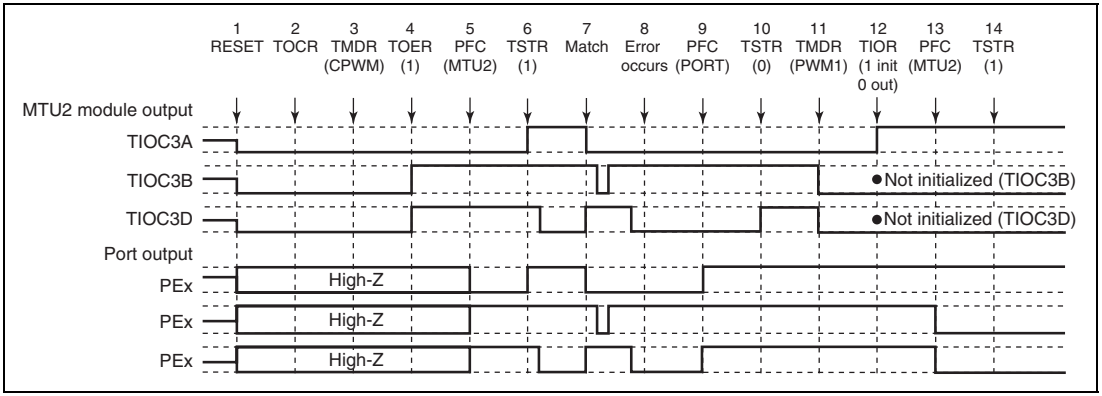


Figure 12.160 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.159.

11. Set PWM mode 1. (MTU2 output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(23) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 12.161 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

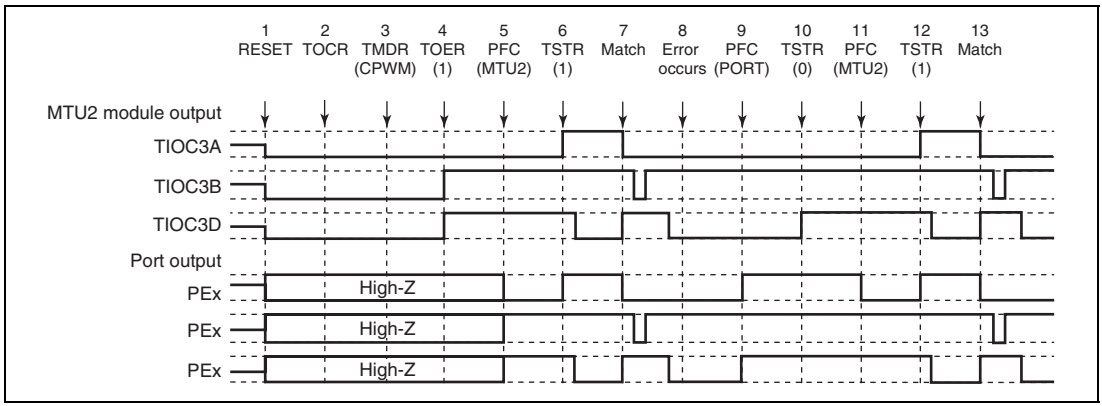


Figure 12.161 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.159.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

(24) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 12.162 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

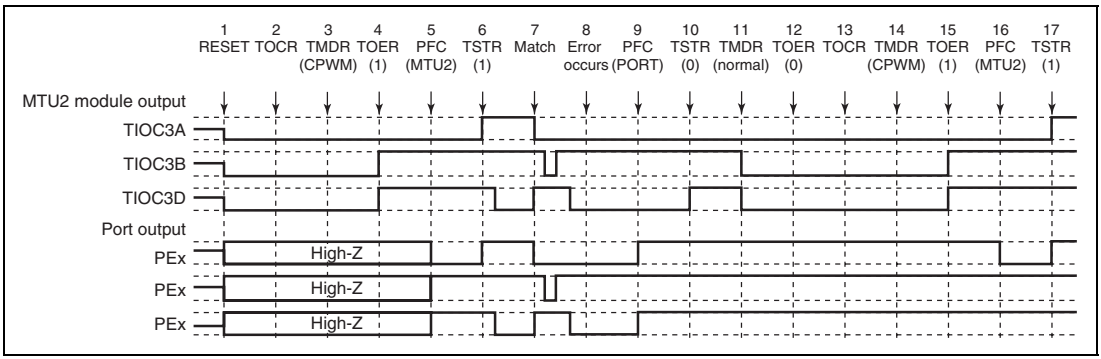


Figure 12.162 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.159.

11. Set normal mode and make new settings. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

(25) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 12.163 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

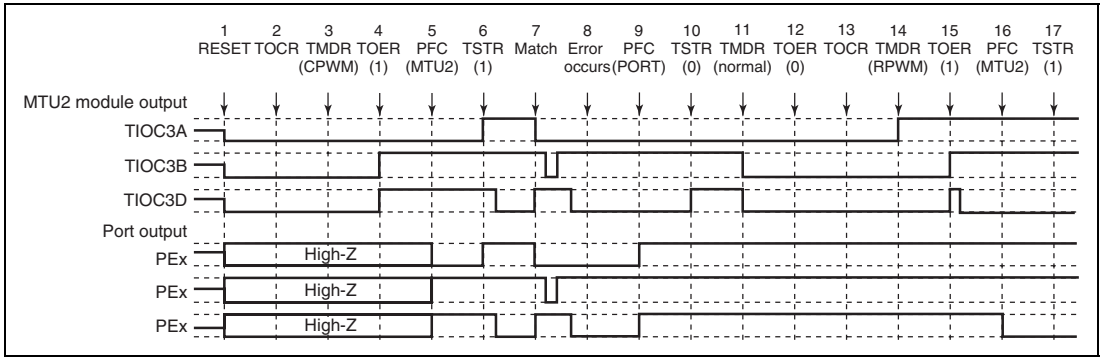


Figure 12.163 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 12.159.

11. Set normal mode. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronized PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronized PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

(26) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 12.164 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

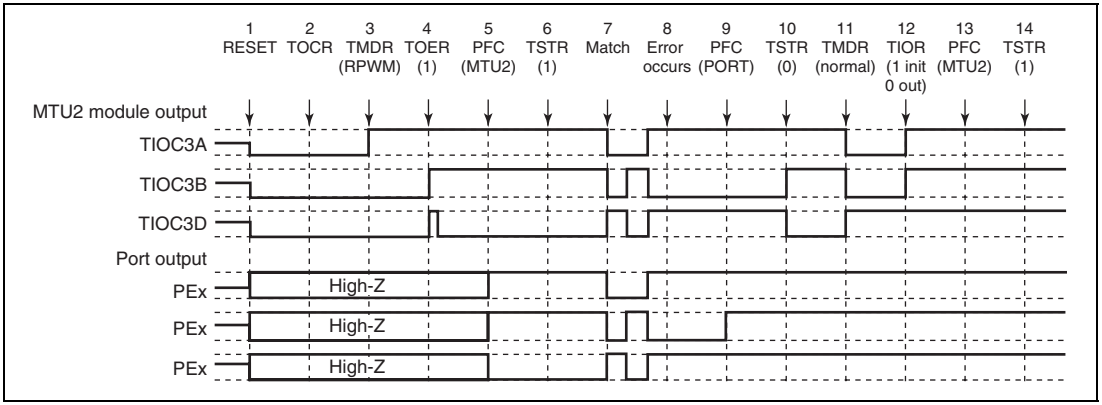


Figure 12.164 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronized PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronized PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the reset-synchronized PWM output initial value.)
11. Set normal mode. (MTU2 positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(27) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 12.165 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

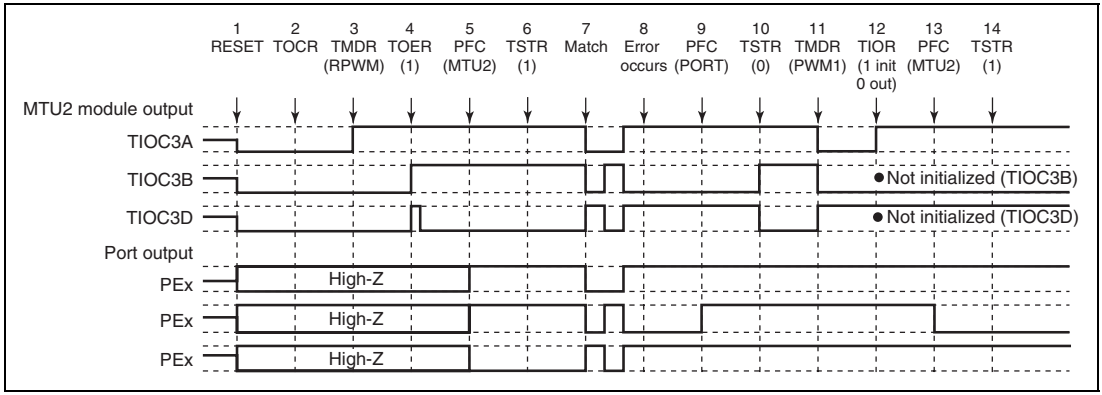


Figure 12.165 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.164.

11. Set PWM mode 1. (MTU2 positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(28) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 12.166 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

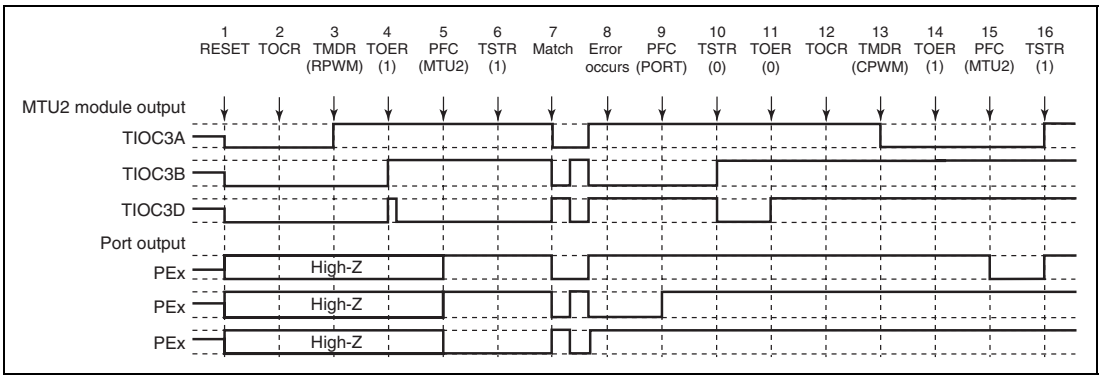


Figure 12.166 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.164.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
13. Set complementary PWM. (The MTU2 cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU2 output with the PFC.
16. Operation is restarted by TSTR.

(29) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 12.167 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

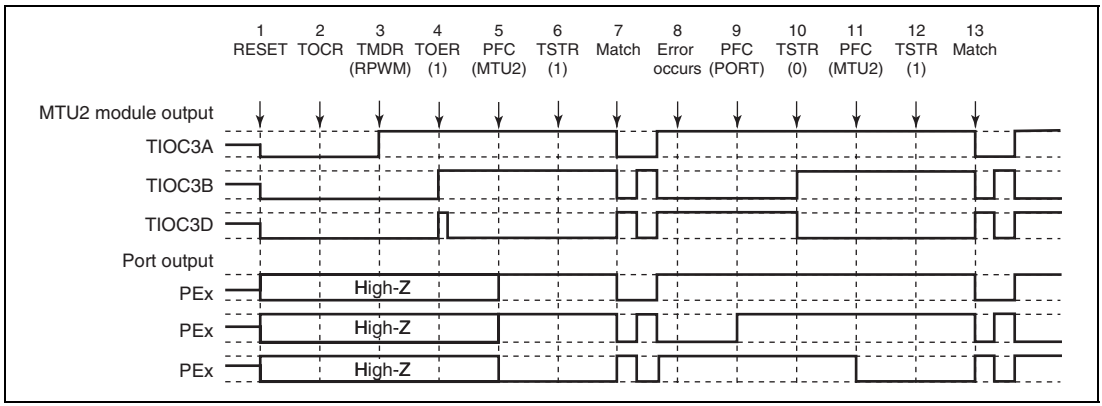


Figure 12.167 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 12.164.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronized PWM waveform is output on compare-match occurrence.

Section 13 Multi-Function Timer Pulse Unit 2S (MTU2S)

This LSI has an on-chip multi-function timer pulse unit 2S (MTU2S) that comprises three 16-bit timer channels. The MTU2S includes channels 3 to 5 of the MTU2. For details, refer to section 12, Multi-Function Timer Pulse Unit 2 (MTU2). To distinguish from the MTU2, "S" is added to the end of the MTU2S input/output pin and register names. For example, TIOC3A is called TIOC3AS and TGRA_3 is called TGRA_3S in this section.

The MTU2S can operate at 100 MHz max. for complementary PWM output functions or at 50 MHz max. for the other functions.

Table 13.1 MTU2S Functions

Item		Channel 3	Channel 4	Channel 5
Count clock		Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 Mϕ/1024	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 Mϕ/1024	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64
General registers		TGRA_3S TGRB_3S	TGRA_4S TGRB_4S	TGRU_5S TGRV_5S TGRW_5S
General registers/ buffer registers		TGRC_3S TGRD_3S	TGRC_4S TGRD_4S	—
I/O pins		TIOC3AS TIOC3BS TIOC3CS TIOC3DS	TIOC4AS TIOC4BS TIOC4CS TIOC4DS	Input pins TIC5US TIC5VS TIC5WS
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	√	√	—
	1 output	√	√	—
	Toggle output	√	√	—
Input capture function		√	√	√
Synchronous operation		√	√	—

Item	Channel 3	Channel 4	Channel 5
PWM mode 1	√	√	—
PWM mode 2	—	—	—
Complementary PWM mode	√	√	—
Reset PWM mode	√	√	—
AC synchronous motor drive mode	—	—	—
Phase counting mode	—	—	—
Buffer operation	√	√	—
Counter function of compensation for dead time	—	—	√
DTC activation	TGR compare match or input capture	TGR compare match or input capture, or TCNT overflow or underflow	TGR compare match or input capture
A/D converter start trigger	TGRA_3S compare match or input capture	TGRA_4S compare match or input capture TCNT_4S underflow (trough) in complementary PWM mode	—
Interrupt sources	5 sources • Compare match or input capture 3AS • Compare match or input capture 3BS • Compare match or input capture 3CS • Compare match or input capture 3DS • Overflow	5 sources • Compare match or input capture 4AS • Compare match or input capture 4BS • Compare match or input capture 4CS • Compare match or input capture 4DS • Overflow or underflow	3 sources • Compare match or input capture 5US • Compare match or input capture 5VS • Compare match or input capture 5WS

Item	Channel 3	Channel 4	Channel 5
A/D converter start request delaying function	—	• A/D converter start request at a match between TADCORA_4S and TCNT_4S • A/D converter start request at a match between TADCORB_4S and TCNT_4S	—
Interrupt skipping function	• Skips TGRA_3S compare match interrupts	• Skips TCIV_4S interrupts	—

[Legend]
√: Possible
—: Not possible

13.1 Input/Output Pins

Table 13.2 Pin Configuration

Channel	Symbol	I/O	Function
3	TIOC3AS	I/O	TGRA_3S input capture input/output compare output/PWM output pin
	TIOC3BS	I/O	TGRB_3S input capture input/output compare output/PWM output pin
	TIOC3CS	I/O	TGRC_3S input capture input/output compare output/PWM output pin
	TIOC3DS	I/O	TGRD_3S input capture input/output compare output/PWM output pin
4	TIOC4AS	I/O	TGRA_4S input capture input/output compare output/PWM output pin
	TIOC4BS	I/O	TGRB_4S input capture input/output compare output/PWM output pin
	TIOC4CS	I/O	TGRC_4S input capture input/output compare output/PWM output pin
	TIOC4DS	I/O	TGRD_4S input capture input/output compare output/PWM output pin
5	TIC5US	Input	TGRU_5S input capture input/external pulse input pin
	TIC5VS	Input	TGRV_5S input capture input/external pulse input pin
	TIC5WS	Input	TGRW_5S input capture input/external pulse input pin

13.2 Register Descriptions

The MTU2S has the following registers. For details on register addresses and register states during each process, refer to section 34, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 3 is expressed as TCR_3S.

Table 13.3 Register Configuration

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer control register_3S	TCR_3S	R/W	H'00	H'FFFE4A00	8, 16, 32
Timer control register_4S	TCR_4S	R/W	H'00	H'FFFE4A01	8
Timer mode register_3S	TMDR_3S	R/W	H'00	H'FFFE4A02	8, 16
Timer mode register_4S	TMDR_4S	R/W	H'00	H'FFFE4A03	8
Timer I/O control register H_3S	TIORH_3S	R/W	H'00	H'FFFE4A04	8, 16, 32
Timer I/O control register L_3S	TIORL_3S	R/W	H'00	H'FFFE4A05	8
Timer I/O control register H_4S	TIORH_4S	R/W	H'00	H'FFFE4A06	8, 16
Timer I/O control register L_4S	TIORL_4S	R/W	H'00	H'FFFE4A07	8
Timer interrupt enable register_3S	TIER_3S	R/W	H'00	H'FFFE4A08	8, 16
Timer interrupt enable register_4S	TIER_4S	R/W	H'00	H'FFFE4A09	8
Timer output master enable register S	TOERS	R/W	H'C0	H'FFFE4A0A	8
Timer gate control register S	TGCRS	R/W	H'80	H'FFFE4A0D	8
Timer output control register 1S	TOCR1S	R/W	H'00	H'FFFE4A0E	8, 16
Timer output control register 2S	TOCR2S	R/W	H'00	H'FFFE4A0F	8
Timer counter_3S	TCNT_3S	R/W	H'0000	H'FFFE4A10	16, 32
Timer counter_4S	TCNT_4S	R/W	H'0000	H'FFFE4A12	16
Timer cycle data register S	TCDRS	R/W	H'FFFF	H'FFFE4A14	16, 32
Timer dead time data register S	TDDRS	R/W	H'FFFF	H'FFFE4A16	16
Timer general register A_3S	TGRA_3S	R/W	H'FFFF	H'FFFE4A18	16, 32
Timer general register B_3S	TGRB_3S	R/W	H'FFFF	H'FFFE4A1A	16
Timer general register A_4S	TGRA_4S	R/W	H'FFFF	H'FFFE4A1C	16, 32
Timer general register B_4S	TGRB_4S	R/W	H'FFFF	H'FFFE4A1E	16
Timer subcounter S	TCNTSS	R	H'0000	H'FFFE4A20	16, 32
Timer cycle buffer register S	TCBRS	R/W	H'FFFF	H'FFFE4A22	16

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer general register C_3S	TGRC_3S	R/W	H'FFFF	H'FFFE4A24	16, 32
Timer general register D_3S	TGRD_3S	R/W	H'FFFF	H'FFFE4A26	16
Timer general register C_4S	TGRC_4S	R/W	H'FFFF	H'FFFE4A28	16, 32
Timer general register D_4S	TGRD_4S	R/W	H'FFFF	H'FFFE4A2A	16
Timer status register_3S	TSR_3S	R/W	H'C0	H'FFFE4A2C	8, 16
Timer status register_4S	TSR_4S	R/W	H'C0	H'FFFE4A2D	8
Timer interrupt skipping set register S	TITCRS	R/W	H'00	H'FFFE4A30	8, 16
Timer interrupt skipping counter S	TITCNTS	R	H'00	H'FFFE4A31	8
Timer buffer transfer set register S	TBTERS	R/W	H'00	H'FFFE4A32	8
Timer dead time enable register S	TDERS	R/W	H'01	H'FFFE4A34	8
Timer output level buffer register S	TOLBRS	R/W	H'00	H'FFFE4A36	8
Timer buffer operation transfer mode register_3S	TBTM_3S	R/W	H'00	H'FFFE4A38	8, 16
Timer buffer operation transfer mode register_4S	TBTM_4S	R/W	H'00	H'FFFE4A39	8
Timer A/D converter start request control register S	TADCRS	R/W	H'0000	H'FFFE4A40	16
Timer A/D converter start request cycle set register A_4S	TADCORA_4S	R/W	H'FFFF	H'FFFE4A44	16, 32
Timer A/D converter start request cycle set register B_4S	TADCORB_4S	R/W	H'FFFF	H'FFFE4A46	16
Timer A/D converter start request cycle set buffer register A_4S	TADCOBRA_4S	R/W	H'FFFF	H'FFFE4A48	16, 32
Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFE4A4A	16
Timer synchronous clear register S	TSYCRS	R/W	H'00	H'FFFE4A50	8
Timer waveform control register S	TWCRS	R/W	H'00	H'FFFE4A60	8
Timer start register S	TSTRS	R/W	H'00	H'FFFE4A80	8, 16
Timer synchronous register S	TSYRS	R/W	H'00	H'FFFE4A81	8
Timer read/write enable register S	TRWERS	R/W	H'01	H'FFFE4A84	8

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer counter U_5S	TCNTU_5S	R/W	H'0000	H'FFFE4880	16, 32
Timer general register U_5S	TGRU_5S	R/W	H'FFFF	H'FFFE4882	16
Timer control register U_5S	TCRU_5S	R/W	H'00	H'FFFE4884	8
Timer I/O control register U_5S	TIORU_5S	R/W	H'00	H'FFFE4886	8
Timer counter V_5S	TCNTV_5S	R/W	H'0000	H'FFFE4890	16, 32
Timer general register V_5S	TGRV_5S	R/W	H'FFFF	H'FFFE4892	16
Timer control register V_5S	TCRV_5S	R/W	H'00	H'FFFE4894	8
Timer I/O control register V_5S	TIORV_5S	R/W	H'00	H'FFFE4896	8
Timer counter W_5S	TCNTW_5S	R/W	H'0000	H'FFFE48A0	16, 32
Timer general register W_5S	TGRW_5S	R/W	H'FFFF	H'FFFE48A2	16
Timer control register W_5S	TCRW_5S	R/W	H'00	H'FFFE48A4	8
Timer I/O control register W_5S	TIORW_5S	R/W	H'00	H'FFFE48A6	8
Timer status register_5S	TSR_5S	R/W	H'00	H'FFFE48B0	8
Timer interrupt enable register_5S	TIER_5S	R/W	H'00	H'FFFE48B2	8
Timer start register_5S	TSTR_5S	R/W	H'00	H'FFFE48B4	8
Timer compare match clear register S	TCNTCMPCLRS	R/W	H'00	H'FFFE48B6	8

Section 14 Port Output Enable 2 (POE2)

The port output enable 2 (POE2) module can be used to place the large-current pins (multiplexed with TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D of the MTU2, and with TIOC3BS, TIOC3DS, TIOC4AS, TIOC4BS, TIOC4CS, and TIOC4DS of the MUT2S) and the pins for channel 0 of the MTU2 (multiplexed with TIOC0A, TIOC0B, TIOC0C, and TIOC0D) in the high-impedance state in response to changes in the levels on the $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ input pins, in the states of output on the large-current pins, and in register settings. It can also simultaneously generate interrupt requests.

14.1 Features

- Each of the $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ input pins can be set for falling edge, Pϕ/8 × 16, Pϕ/16 × 16, or Pϕ/128 × 16 low-level sampling.
- Large-current pins and the pins for channel 0 of the MTU2 can be placed in high-impedance state by $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ pins falling-edge or low-level sampling.
- Large-current pins can be placed in high-impedance state when the large-current pin output levels are compared and simultaneous active-level output continues for one cycle or more.
- Large-current pins and the pins for channel 0 of the MTU2 can be placed in high-impedance state by modifying the POE2 register settings.
- Interrupts can be generated by input-level sampling or output-level comparison results.

The POE2 has input level detection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in the block diagram of figure 14.1. It is also capable of placing large-current pins in the high-impedance state when an oscillator halts or while this LSI is on software standby. For details, see section 22.1.29, Large Current Port Control Register (HCPCR), and Appendix, A. Pin States.

Figure 14.1 shows a block diagram of the POE2.

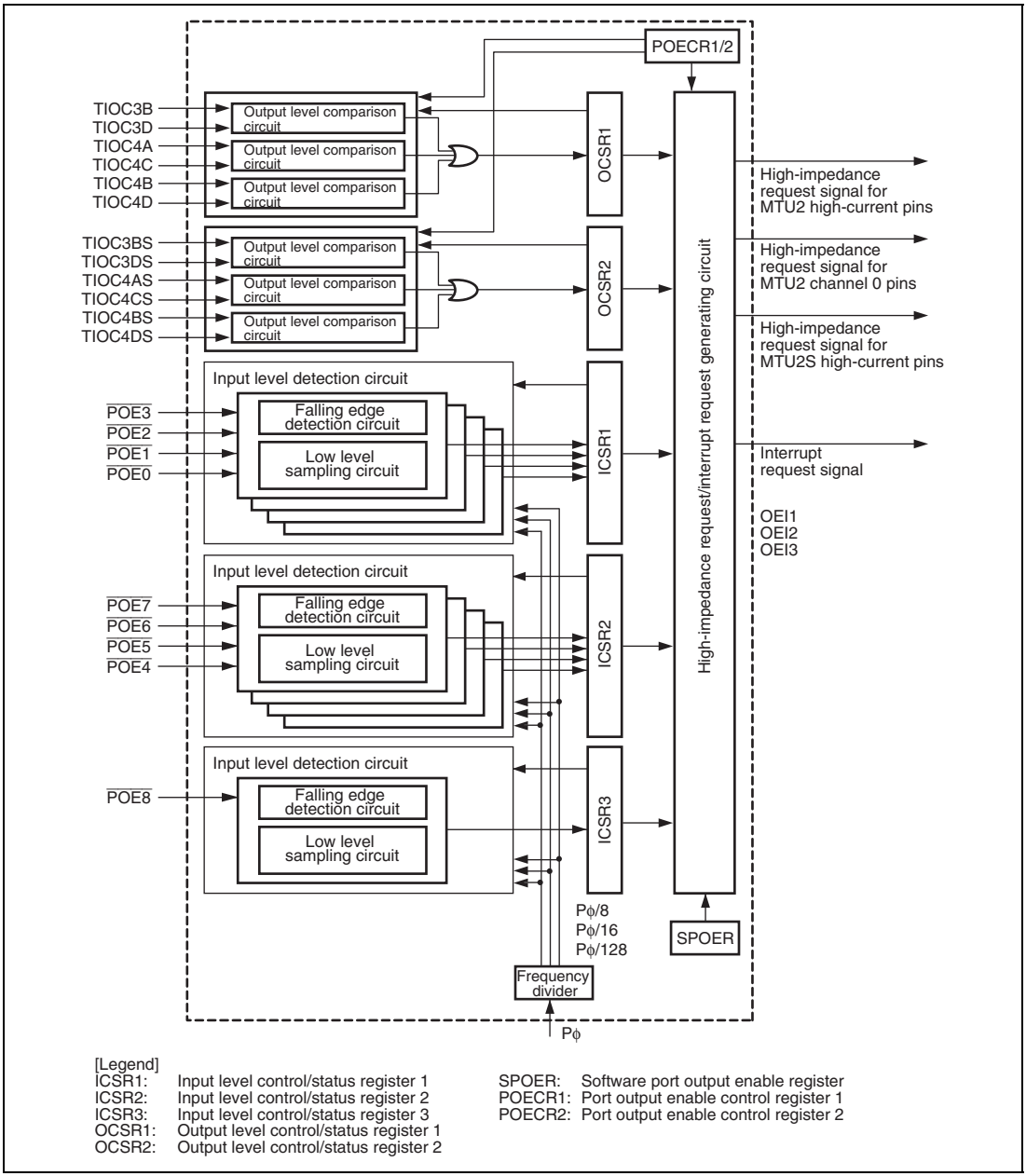


Figure 14.1 Block Diagram of POE2

14.2 Input/Output Pins

Table 14.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Port output enable input pins 0 to 3	$\overline{\text{POE0}}$ to $\overline{\text{POE3}}$	Input	Input request signals to place large-current pins for MTU2 in high-impedance state
Port output enable input pins 4 to 7	$\overline{\text{POE4}}$ to $\overline{\text{POE7}}$	Input	Input request signals to place large-current pins for MTU2S in high-impedance state
Port output enable input pin 8	$\overline{\text{POE8}}$	Input	Inputs a request signal to place pins for channel 0 in MTU2 in high-impedance state

Table 14.2 shows output-level comparisons with pin combinations.

Table 14.2 Pin Combinations

Pin Combination	I/O	Description
TIOC3B and TIOC3D TIOC4A and TIOC4C TIOC4B and TIOC4D	Output	The large-current pins for the MTU2 are placed in high-impedance state when the pins simultaneously output an active level for one or more cycles of the peripheral clock (Pϕ). (In the case of TOCS = 0 in timer output control register 1 (TOCR1) in the MTU2, low level when the output level select P (OLSP) bit is 0, or high level when the OLSP bit is 1. In the case of TOCS = 1, low level when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0 in TOCR2, or high level when these bits are 1.) This active level comparison is done when the MTU2 output function or general output function is selected in the pin function controller. If another function is selected, the output level is not checked. Pin combinations for output comparison and high-impedance control can be selected by POE2 registers.
TIOC3BS and TIOC3DS TIOC4AS and TIOC4CS TIOC4BS and TIOC4DS	Output	The large-current pins for the MTU2S are placed in high-impedance state when the pins simultaneously output an active level for one or more cycles of the peripheral clock (Pϕ). (In the case of TOCS = 0 in timer output control register 1S (TOCR1S) in the MTU2S, low level when the output level select P (OLSP) bit is 0, or high level when the OLSP bit is 1. In the case of TOCS = 1, low level when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0 in TOCR2S, or high level when these bits are 1.) This active level comparison is done when the MTU2S output function or general output function is selected in the pin function controller. If another function is selected, the output level is not checked. Pin combinations for output comparison and high-impedance control can be selected by POE2 registers.

14.3 Register Descriptions

The POE2 has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 14.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Input level control/status register 1	ICSR1	R/W	H'0000	H'FFFE5000	16
Output level control/status register 1	OCSR1	R/W	H'0000	H'FFFE5002	16
Input level control/status register 2	ICSR2	R/W	H'0000	H'FFFE5004	16
Output level control/status register 2	OCSR2	R/W	H'0000	H'FFFE5006	16
Input level control/status register 3	ICSR3	R/W	H'0000	H'FFFE5008	16
Software port output enable register	SPOER	R/W	H'00	H'FFFE500A	8
Port output enable control register 1	POECR1	R/W	H'00	H'FFFE500B	8
Port output enable control register 2	POECR2	R/W	H'7700	H'FFFE500C	16

14.3.1 Input Level Control/Status Register 1 (ICSR1)

ICSR1 is a 16-bit readable/writable register that selects the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE3F	POE2F	POE1F	POE0F	-	-	-	PIE1	POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	POE3F	0	R/(W)*1	POE3 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin. [Clearing condition] • By writing 0 to POE3F after reading POE3F = 1 [Setting condition] • When the input set by bits 7 and 6 in ICSR1 occurs at the $\overline{\text{POE3}}$ pin
14	POE2F	0	R/(W)*1	POE2 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin. [Clearing condition] • By writing 0 to POE2F after reading POE2F = 1 [Setting condition] • When the input set by bits 5 and 4 in ICSR1 occurs at the $\overline{\text{POE2}}$ pin
13	POE1F	0	R/(W)*1	POE1 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin. [Clearing condition] • By writing 0 to POE1F after reading POE1F = 1 [Setting condition] • When the input set by bits 3 and 2 in ICSR1 occurs at the $\overline{\text{POE1}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
12	POE0F	0	R/(W)*1	POE0 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin. [Clear condition] • By writing 0 to POE0F after reading POE0F = 1 [Set condition] • When the input set by bits 1 and 0 in ICSR1 occurs at the $\overline{\text{POE0}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PIE1	0	R/W	Port Interrupt Enable 1 Enables or disables interrupt requests when any one of the POE0F to POE3F bits of the ICSR1 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE3M[1:0]	00	R/W*2	POE3 Mode These bits select the input mode of the $\overline{\text{POE3}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE3}}$ input 01: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.
5, 4	POE2M[1:0]	00	R/W*2	POE2 Mode These bits select the input mode of the $\overline{\text{POE2}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE2}}$ input 01: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	POE1M[1:0]	00	R/W*2	POE1 Mode These bits select the input mode of the $\overline{\text{POE1}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE1}}$ input 01: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.
1, 0	POE0M[1:0]	00	R/W*2	POE0 Mode These bits select the input mode of the $\overline{\text{POE0}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE0}}$ input 01: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

14.3.2 Output Level Control/Status Register 1 (OCSR1)

OCSR1 is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF1	-	-	-	-	-	OCE1	OIE1	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	OSF1	0	R/(W)*1	Output Short Flag 1 Indicates that any one of the three pairs of MTU2 2-phase outputs to be compared has simultaneously become an active level. [Clearing condition] • By writing 0 to OSF1 after reading OSF1 = 1 [Setting condition] • When any one of the three pairs of 2-phase outputs has simultaneously become an active level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	OCE1	0	R/W*2	Output Short High-Impedance Enable 1 Specifies whether to place the pins in high-impedance state when the OSF1 bit in OCSR1 is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
8	OIE1	0	R/W	Output Short Interrupt Enable 1 Enables or disables interrupt requests when the OSF1 bit in OCSR is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

14.3.3 Input Level Control/Status Register 2 (ICSR2)

ICSR2 is a 16-bit readable/writable register that selects the $\overline{\text{POE4}}$ to $\overline{\text{POE7}}$ pin input modes, controls the enabling/disabling of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE7F	POE6F	POE5F	POE4F	-	-	-	PIE2	POE7M[1:0]		POE6M[1:0]		POE5M[1:0]		POE4M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	POE7F	0	R/(W)*1	POE7 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE7}}$ pin. [Clearing condition] • By writing 0 to POE7F after reading POE7F = 1 [Setting condition] When the input condition set by bits 7 and 6 in ICSR2 occurs at the $\overline{\text{POE7}}$ pin
14	POE6F	0	R/(W)*1	POE6 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE6}}$ pin. [Clearing condition] • By writing 0 to POE6F after reading POE6F = 1 [Setting condition] • When the input condition set by bits 5 and 4 in ICSR2 occurs at the $\overline{\text{POE6}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
13	POE5F	0	R/(W)*[1]	POE5 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE5}}$ pin. [Clearing condition] • By writing 0 to POE5F after reading POE5F = 1 [Setting condition] When the input condition set by bits 3 and 2 in ICSR2 occurs at the $\overline{\text{POE5}}$ pin
12	POE4F	0	R/(W)*[1]	POE4 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE4}}$ pin. [Clearing condition] • By writing 0 to POE4F after reading POE4F = 1 [Setting condition] • When the input condition set by bits 1 and 0 in ICSR2 occurs at the $\overline{\text{POE4}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PIE2	0	R/W	Port Interrupt Enable 2 Enables or disables interrupt requests when one or more of the POE4F to POE7F bits in the ICSR2 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE7M [1:0]	00	R/W*[2]	POE7 Mode These bits select the input mode of the $\overline{\text{POE7}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE7}}$ input 01: Accept request when $\overline{\text{POE7}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level. 10: Accept request when $\overline{\text{POE7}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level. 11: Accept request when $\overline{\text{POE7}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	POE6M [1:0]	00	R/W*2	POE6 Mode These bits select the input mode of the $\overline{\text{POE6}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE6}}$ input 01: Accept request when $\overline{\text{POE6}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level. 10: Accept request when $\overline{\text{POE6}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level. 11: Accept request when $\overline{\text{POE6}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level.
3, 2	POE5M [1:0]	00	R/W*2	POE5 Mode These bits select the input mode of the $\overline{\text{POE5}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE5}}$ input 01: Accept request when $\overline{\text{POE5}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level. 10: Accept request when $\overline{\text{POE5}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level. 11: Accept request when $\overline{\text{POE5}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level.
1, 0	POE4M [1:0]	00	R/W*2	POE4 Mode These bits select the input mode of the $\overline{\text{POE4}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE4}}$ input 01: Accept request when $\overline{\text{POE4}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level. 10: Accept request when $\overline{\text{POE4}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level. 11: Accept request when $\overline{\text{POE4}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

14.3.4 Output Level Control/Status Register 2 (OCSR2)

OCSR2 is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF2	-	-	-	-	-	OCE2	OIE2	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	OSF2	0	R/(W)*1	Output Short Flag 2 Indicates that any one of the three pairs of MTU2S 2-phase outputs to be compared has simultaneously become an active level. [Clearing condition] • By writing 0 to OSF2 after reading OSF2 = 1 [Setting condition] • When any one of the three pairs of 2-phase outputs has simultaneously become an active level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	OCE2	0	R/W*2	Output Short High-Impedance Enable 2 Specifies whether to place the pins in high-impedance state when the OSF2 bit in OCSR2 is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
8	OIE2	0	R/W	Output Short Interrupt Enable 2 Enables or disables interrupt requests when the OSF2 bit in OCSR2 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

14.3.5 Input Level Control/Status Register 3 (ICSR3)

ICSR3 is a 16-bit readable/writable register that selects the $\overline{\text{POE8}}$ pin input mode, controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	POE8F	-	-	POE8E	PIE3	-	-	-	-	-	-	POE8M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*1	R	R	R/W*2	R/W	R	R	R	R	R	R	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	POE8F	0	R/(W)*1	POE8 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE8}}$ pin. [Clearing condition] • By writing 0 to POE8F after reading POE8F = 1 [Setting condition] • When the input condition set by bits 1 and 0 in ICSR3 occurs at the $\overline{\text{POE8}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	POE8E	0	R/W*2	POE8 High-Impedance Enable Specifies whether to place the pins in high-impedance state when the POE8F bit in ICSR3 is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
8	PIE3	0	R/W	Port Interrupt Enable 3 Enables or disables interrupt requests when the POE8F bit in ICSR3 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	POE8M[1:0]	00	R/W*2	POE8 Mode These bits select the input mode of the $\overline{\text{POE8}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE8}}$ input 01: Accept request when $\overline{\text{POE8}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE8}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE8}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

14.3.6 Software Port Output Enable Register (SPOER)

SPOER is an 8-bit readable/writable register that controls high-impedance state of the pins.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	MTU2SHIZ	0	R/W	MTU2S Output High-Impedance Specifies whether to place the large-current pins for the MTU2S in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2SHIZ after reading MTU2SHIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2SHIZ
1	MTU2CH0HIZ	0	R/W	MTU2 Channel 0 Output High-Impedance Specifies whether to place the pins for channel 0 in the MTU2 in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH0HIZ after reading MTU2CH0HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH0HIZ

Bit	Bit Name	Initial Value	R/W	Description
0	MTU2CH34HIZ	0	R/W	MTU2 Channels 3 and 4 Output High-Impedance Specifies whether to place the large-current pins for the MTU2 in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH34HIZ after reading MTU2CH34HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH34HIZ

14.3.7 Port Output Enable Control Register 1 (POECR1)

POECR1 is an 8-bit readable/writable register that controls high-impedance state of the pins.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W*	R/W*	R/W*	R/W*

Note: * Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MTU2PE3ZE	0	R/W*	MTU2PE3 High-Impedance Enable Specifies whether to place the PEx/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
2	MTU2PE2ZE	0	R/W*	MTU2PE2 High-Impedance Enable Specifies whether to place the PEx/TIOC0C pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
1	MTU2PE1ZE	0	R/W*	MTU2PE1 High-Impedance Enable Specifies whether to place the PEx/TIOC0B pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
0	MTU2PE0ZE	0	R/W*	MTU2PE0 High-Impedance Enable Specifies whether to place the PEx/TIOC0A pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

Note: * Can modified only once after a power-on reset.

14.3.8 Port Output Enable Control Register 2 (POECR2)

POECR2 is a 16-bit readable/writable register that controls high-impedance state of the pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	-	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE	-	-	-	-	-	-	-	-
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R	R	R	R	R	R	R

Note: * Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	MTU2P1CZE	1	R/W*	MTU2 Port 1 High-Impedance Enable Specifies whether to place MTU2 large-current PEx/TIOC3B and PEx/TIOC3D pins in high-impedance state when any one of the OSF1, POE0F to POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
13	MTU2P2CZE	1	R/W*	MTU2 Port 2 High-Impedance Enable Specifies whether to place the MTU2 large-current PEx/TIOC4A and PEx/TIOC4C pins in high-impedance state when any one of the OSF1, POE0F to POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
12	MTU2P3CZE	1	R/W*	MTU2 Port 3 High-Impedance Enable Specifies whether to place the MTU2 large-current PEx/TIOC4B and PEx/TIOC4D in high-impedance state when any one of the OSF1, POE0F to POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	MTU2SP1CZE	1	R/W*	MTU2S Port 1 High-Impedance Enable Specifies whether to place the MTU2S large-current PEx/TIOC3BS and PEx/TIOC3DS pins in high-impedance state when any one of the OSF2, POE4F to POE7F, and MTU2SHIZ bits is set to 1. 0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.
9	MTU2SP2CZE	1	R/W*	MTU2S Port 2 High-Impedance Enable Specifies whether to place the MTU2S large-current PEx/TIOC4AS and PEx/TIOC4CS pins in high-impedance state when any one of the OSF2, POE4F to POE7F, and MTU2SHIZ bits is set to 1. 0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.
8	MTU2SP3CZE	1	R/W*	MTU2S Port 3 High-Impedance Enable Specifies whether to place the MTU2S large-current PEx/TIOC4BS and PEx/TIOC4DS pins in high-impedance state when any one of the OSF2, POE4F to POE7F, and MTU2SHIZ bits is set to 1. 0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Can be modified only once after a power-on reset.

14.4 Operation

Table 14.4 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 14.4 Target Pins and Conditions for High-Impedance Control

Pins	Conditions	Detailed Conditions
MTU2 large-current pins (PEx/TIOC3B and PEx/TIOC3D)	Input level detection for pins $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$, output level comparison for pins PEx/TIOC3B and PEx/TIOC3D, or SPOER setting	MTU2P1CZE ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 large-current pins (PEx/TIOC4A and PEx/TIOC4C)	Input level detection for pins $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$, output level comparison for pins PEx/TIOC4A and PEx/TIOC4C, or SPOER setting	MTU2P2CZE ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 large-current pins (PEx/TIOC4B and PEx/TIOC4D)	Input level detection for pins $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$, output level comparison for pins PEx/TIOC4B and PEx/TIOC4D, or SPOER setting	MTU2P3CZE ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2S large-current pins (PEx/TIOC3BS and PEx/TIOC3DS)	Input level detection for pins $\overline{\text{POE4}}$ to $\overline{\text{POE7}}$, output level comparison for pins PEx/TIOC3BS and PEx/TIOC3DS, or SPOER setting	MTU2SP1CZE ((POE4F+POE5F+POE6F+POE7F) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S large-current pins (PEx/TIOC4AS and PEx/TIOC4CS)	Input level detection for pins $\overline{\text{POE4}}$ to $\overline{\text{POE7}}$, output level comparison for pins PEx/TIOC4AS and PEx/TIOC4CS, or SPOER setting	MTU2SP2CZE ((POE4F+POE5F+POE6F+POE7F) + (OSF2 • OCE2) + (MTU2SHIZ))

Pins	Conditions	Detailed Conditions
MTU2S large-current pins (PEx/TIOC4BS and PEx/TIOC4DS)	Input level detection for pins $\overline{\text{POE4}}$ to $\overline{\text{POE7}}$, output level comparison for pins PEx/TIOC4BS and PEx/TIOC4DS, or SPOER setting	MTU2SP3CZE ((POE4F+POE5F+POE6F+POE7F) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2 CH0 pin (PEx/TIOC0A)	Input level detection for pin $\overline{\text{POE8}}$ or SPOER setting	MTU2PE0ZE ((POE8F • POE8E) + (MTU2CH0HIZ))
MTU2 CH0 pin (PEx/TIOC0B)	Input level detection for pin $\overline{\text{POE8}}$ or SPOER setting	MTU2PE1ZE ((POE8F • POE8E) + (MTU2CH0HIZ))
MTU2 CH0 pin (PEx/TIOC0C)	Input level detection for pin $\overline{\text{POE8}}$ or SPOER setting	MTU2PE2ZE ((POE8F • POE8E) + (MTU2CH0HIZ))
MTU2 CH0 pin (PEx/TIOC0D)	Input level detection for pin $\overline{\text{POE8}}$ or SPOER setting	MTU2PE3ZE ((POE8F • POE8E) + (MTU2CH0HIZ))

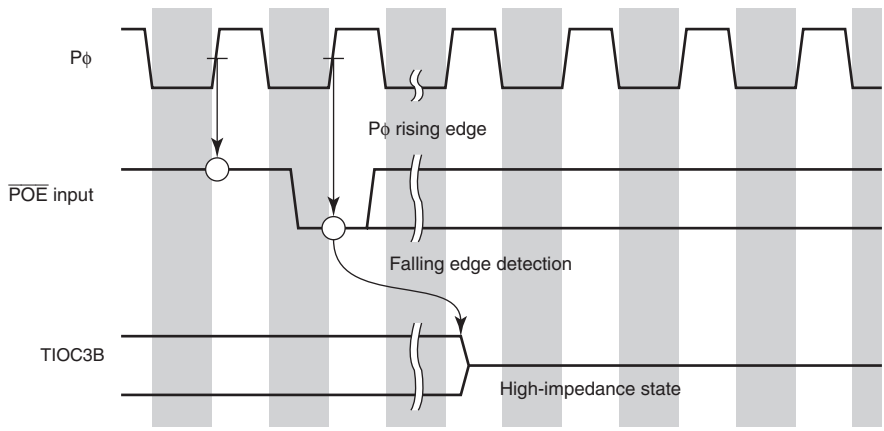
14.4.1 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR3 occur on the $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ pins, the large-current pins and the pins for channel 0 of the MTU2 are placed in high-impedance state. Note however, that these large-current and MTU2 pins enter high-impedance state only when general input/output function, MTU2 function, or MTU2S function is selected for these pins.

(1) Falling Edge Detection

When a change from a high to low level is input to the $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ pins, the large-current pins and the pins for channel 0 of the MTU2 are placed in high-impedance state.

Figure 14.2 shows the sample timing after the level changes in input to the $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ pins until the respective pins enter high-impedance state.



Note: The other high-current pins and MTU2 channel 0 pins also enter the high-impedance state in the similar timing.

Figure 14.2 Falling Edge Detection

(2) Low-Level Detection

Figure 14.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 to ICSR3. If even one high level is detected during this interval, the low level is not accepted.

The timing when the large-current pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

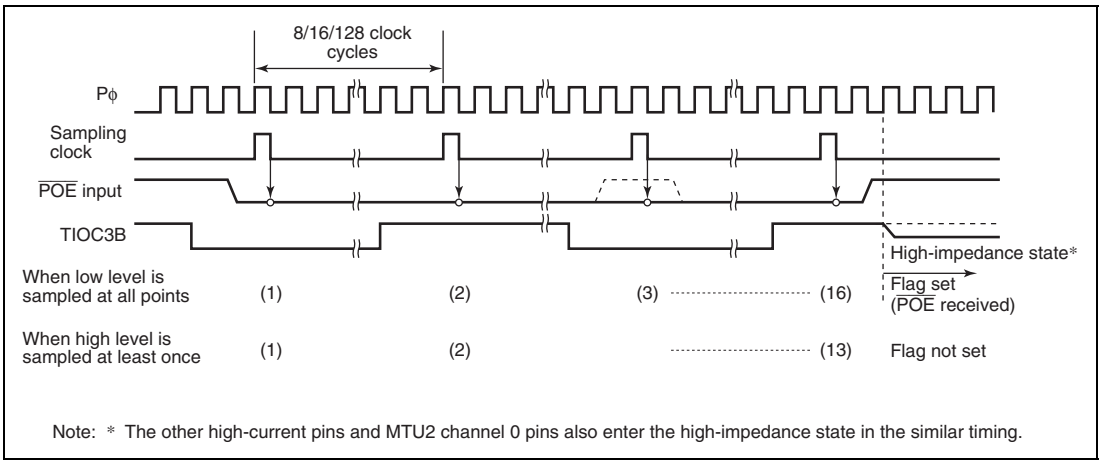


Figure 14.3 Low-Level Detection Operation

14.4.2 Output-Level Compare Operation

Figure 14.4 shows an example of the output-level compare operation for the combination of TIOC3B and TIOC3D. The operation is the same for the other pin combinations.

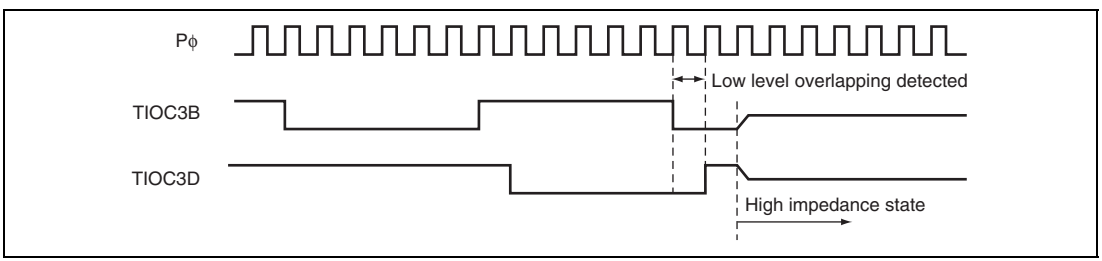


Figure 14.4 Output-Level Compare Operation

14.4.3 Release from High-Impedance State

Large-current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the flags in bits 15 to 12 (POE8F to POE0F) of ICSR1 to ICSR3. However, note that when low-level sampling is selected by bits 7 to 0 in ICSR1 to ICSR3, just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared by writing 0 to it only after a high level is input to one of the $\overline{\text{POE0}}$ to $\overline{\text{POE8}}$ pins and is sampled.

Large-current pins that have entered high-impedance state due to output-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing the flag in bit 15 (OCF1 and OCF2) in OCSR1 and OCSR2. However, note that just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared only after an inactive level is output from the large-current pins. Inactive-level outputs can be achieved by setting the MTU2 and MTU2S internal registers.

14.5 Interrupts

The POE2 issues a request to generate an interrupt when the specified condition is satisfied during input level detection or output level comparison. Table 14.5 shows the interrupt sources and their conditions.

Table 14.5 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE3F, POE2F, POE1F, POE0F, and OSF1	PIE1 • (POE3F + POE2F + POE1F + POE0F) + OIE1 • OSF1
OEI2	Output enable interrupt 2	POE8F	PIE3 • POE8F
OEI3	Output enable interrupt 3	POE4F, POE5F, POE6F, POE7F, OSF2	PIE2 • (POE4F + POE5F + POE6F + POE7F) + OIE2 • OSF2

14.6 Usage Note

14.6.1 Pin State when a Power-On Reset is Issued from the Watchdog Timer

When a power-on reset is issued from the watchdog timer (WDT), initialization of the pin function controller (PFC) sets initial values that select the general input function for the I/O ports. However, when a power-on reset is issued from the WDT while a pin is being handled as high impedance by the port output enable 2 (POE2), the pin is placed in the output state for one cycle of the peripheral clock (Pϕ), after which the pin state is switched to general input.

This also occurs when a power-on reset is issued from the WDT for pins that are being handled as high impedance due to short-circuit detection by the MTU2 and MTU2S.

Figure 14.5 shows the state of a pin for which the $\overline{\text{POE}}$ input has selected high impedance handling with the timer output selected when a power-on reset is issued from the WDT.

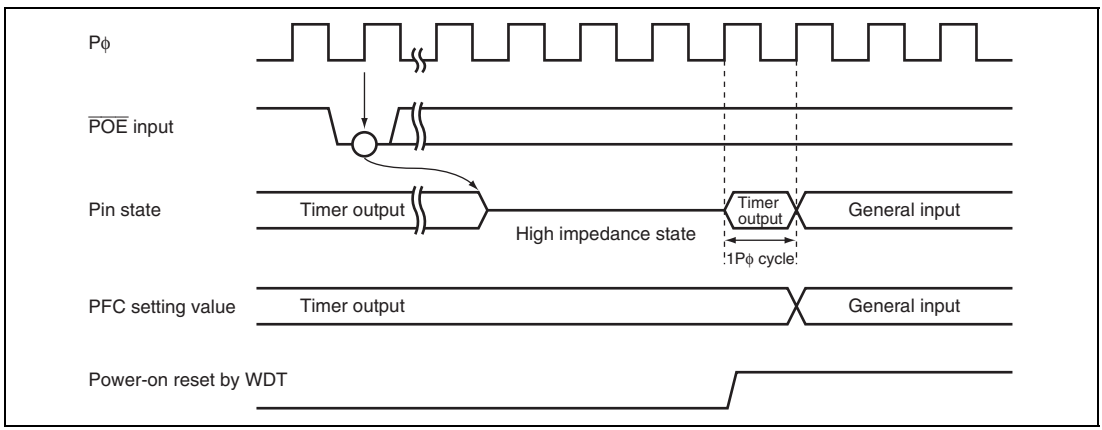


Figure 14.5 Pin State when a Power-On Reset is Issued from the Watchdog Timer

Section 15 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a two-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

15.1 Features

- Independent selection of four counter input clocks at two channels
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected.
- Selection of DTC/DMA transfer request or interrupt request generation on compare match by DTC/DMA setting
- When not in use, the CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 15.1 shows a block diagram of CMT.

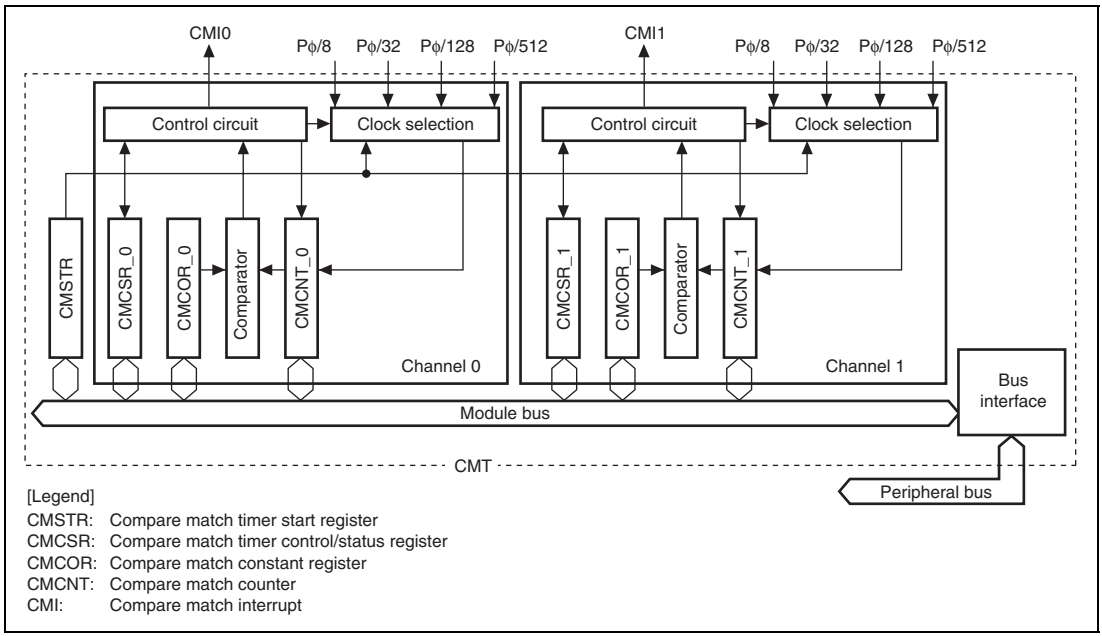


Figure 15.1 Block Diagram of CMT

15.2 Register Descriptions

The CMT has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 15.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFEC000	16
0	Compare match timer control/ status register_0	CMCSR_0	R/W	H'0000	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	R/W	H'0000	H'FFFEC004	16
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFEC006	16
1	Compare match timer control/ status register_1	CMCSR_1	R/W	H'0000	H'FFFEC008	16
	Compare match counter_1	CMCNT_1	R/W	H'0000	H'FFFEC00A	16
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFEC00C	16

15.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

CMSTR is initialized to H'0000 by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether compare match counter_1 operates or is stopped. 0: CMCNT_1 count is stopped 1: CMCNT_1 count is started
0	STR0	0	R/W	Count Start 0 Specifies whether compare match counter_0 operates or is stopped. 0: CMCNT_0 count is stopped 1: CMCNT_0 count is started

15.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables or disables interrupts, and selects the counter input clock.

CMCSR is initialized to H'0000 by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match. [Clearing conditions] • By a power-on reset and in module standby mode • When 0 is written to CMF after reading CMF = 1 • When data is transferred after the DTC has been activated by CMI (except when the DTC transfer counter value has become H'000). • When data is transferred after the DMAC has been activated by CMI 1: CMCNT and CMCOR values match [Setting condition] • When CMCNT and CMCOR values match

Bit	Bit Name	Initial Value	R/W	Description
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select These bits select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS[1:0]. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

Note: * Only 0 can be written to clear the flag after 1 is read.

15.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS[1:0] in CMCSR, and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock. When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.3 Operation

15.3.1 Interval Count Operation

When a counter clock is selected with the CKS[1:0] bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CMIE bit in CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 15.2 shows the operation of the compare match counter.

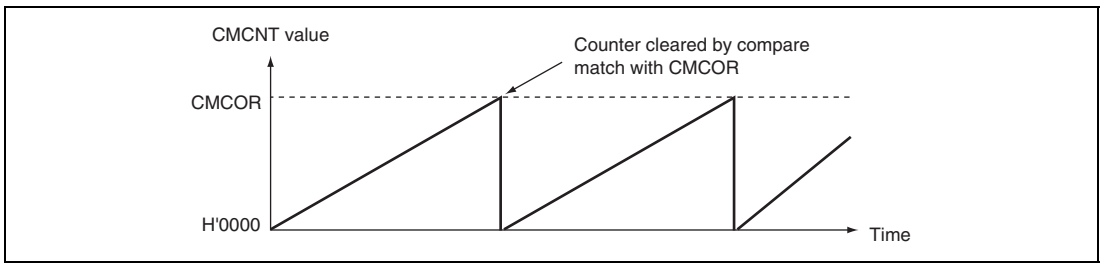


Figure 15.2 Counter Operation

15.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected with the CKS[1:0] bits in CMCSR. Figure 15.3 shows the timing.

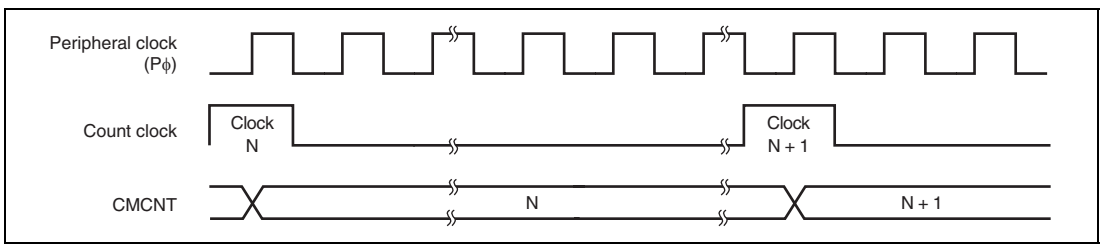


Figure 15.3 Count Timing

15.4 Interrupts

15.4.1 Interrupt Sources and DTC/DMA Transfer Requests

The CMT has two channels, and each of them to which a different vector address is allocated has a compare match interrupt as shown in table 15.2. When both the interrupt request flag (CMF) and the interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 7, Interrupt Controller (INTC).

Clear the CMF bit to 0 from within the user exception handling routine. If this is not done, the interrupt will be generated again. If the next compare match sets the CMF flag before 0 is written to this flag after it has been read as 1, writing 0 to this flag will not clear it. Therefore, read this flag as 1 again before writing 0 to it.

A compare match interrupt request can activate the direct memory access controller (DMAC) or data transfer controller. Since data transfer due to DMAC activation automatically clears the flag, the interrupt is not conveyed to the CPU. Since data transfer due to DTC activation automatically clears the flag while the DISEL bit of the DTC is 0 and the transfer counter value is not 0, an interrupt is not issued to the CPU in this case. However, if the DISEL bit and the transfer counter value are both 0, or the DISEL bit is 1, data transfer does not clear the flag, so an interrupt request for the CPU is generated after completion of the data transfer.

Table 15.2 Interrupt Sources

Channel	Interrupt Source	Interrupt Enable Bit	Interrupt Flag	DMAC/DTC Activation	Priority
0	CMI0	CMIE of CMCSR_0	CMF of CMCSR_0	Possible	High
1	CMI1	CMIE of CMCSR_1	CMF of CMCSR_1	Possible	Low

15.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated at the last state in which the values match (the timing when the CMCNT value is updated to H'0000) and the CMF bit in CMCSR is set to 1. That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 15.4 shows the timing of CMF bit setting.

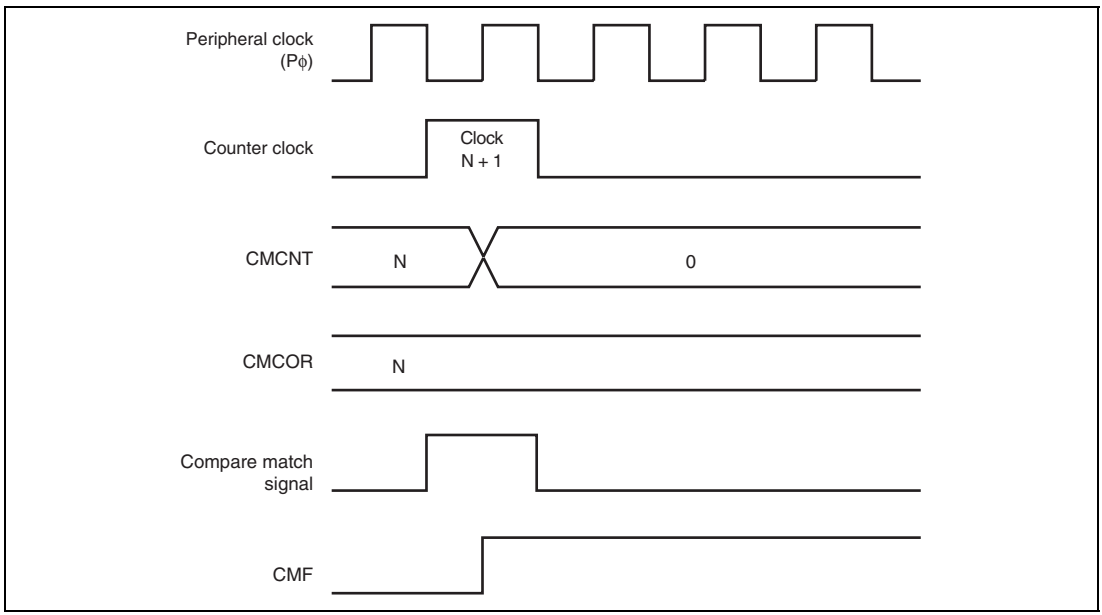


Figure 15.4 Timing of CMF Setting

15.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing of 0. However, in the case of the DMAC/DTC being activated, the CMF bit is automatically cleared to 0 when data is transferred by the DMAC/DTC.

15.5 Usage Notes

15.5.1 Setting of Module Standby Mode

The standby control register setting can enable or disable the CMT. The initial setting is for operation to be stopped. Release from module standby mode enables access to this register. For details, see section 32, Power-Down Modes.

15.5.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 15.5 shows the timing to clear the CMCNT counter.

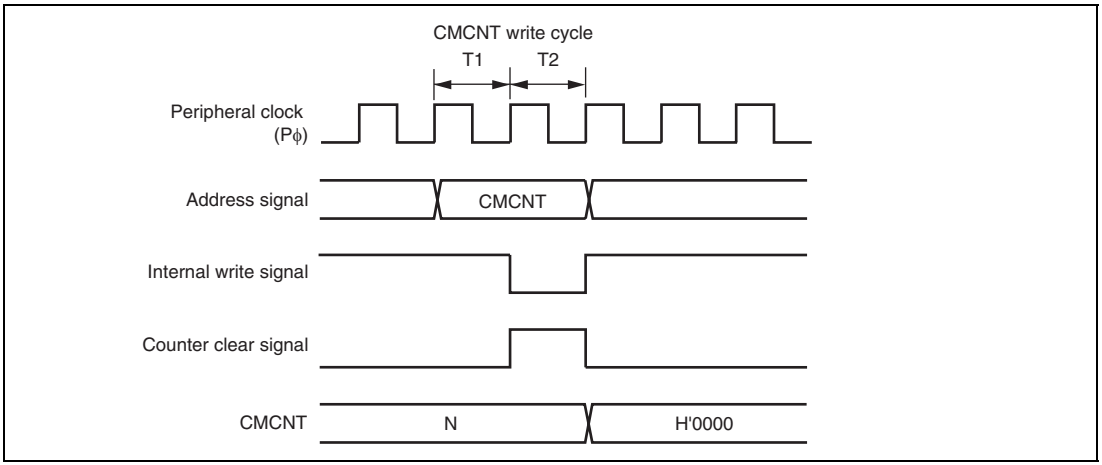


Figure 15.5 Conflict between Write and Compare Match Processes of CMCNT

15.5.3 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 15.6 shows the timing to write to CMCNT in words.

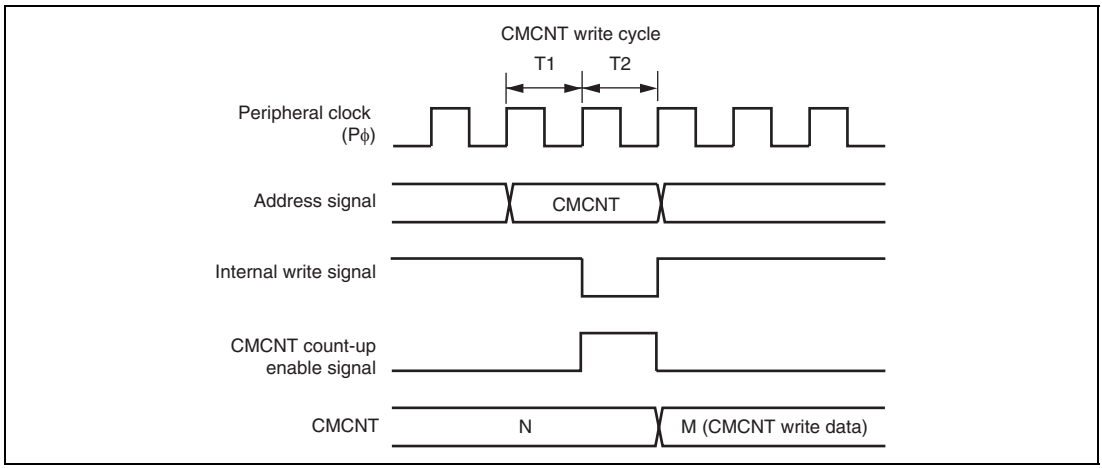


Figure 15.6 Conflict between Word-Write and Count-Up Processes of CMCNT

15.5.4 Compare Match between CMCNT and CMCOR

Do not set a same value to CMCNT and CMCOR while the count operation of CMCNT is stopped.

Section 16 Compare Match Timer 2 (CMT2)

This LSI includes one channel of 32-bit compare match timer 2 (CMT2).

16.1 Features

- Compare match of one channel
- Includes one channel of upcounter (16-bit/32-bit selectable).
- Up to two input capture input signals available.
- Up to two output compare output signals available.
- Allows selection from among four counter input clocks:
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected.
- Allows the counter to be cleared by input capture input and upon compare match.
- Five types of interrupts: There are five interrupt sources including input capture input interrupts and compare match interrupts, and each interrupt can be requested independently. The data transfer controller (DTC) or direct memory access controller (DMAC) can be activated by the interrupts to transfer data.
- Module standby mode can be set.

Figure 16.1 shows a block diagram of the CMT2.

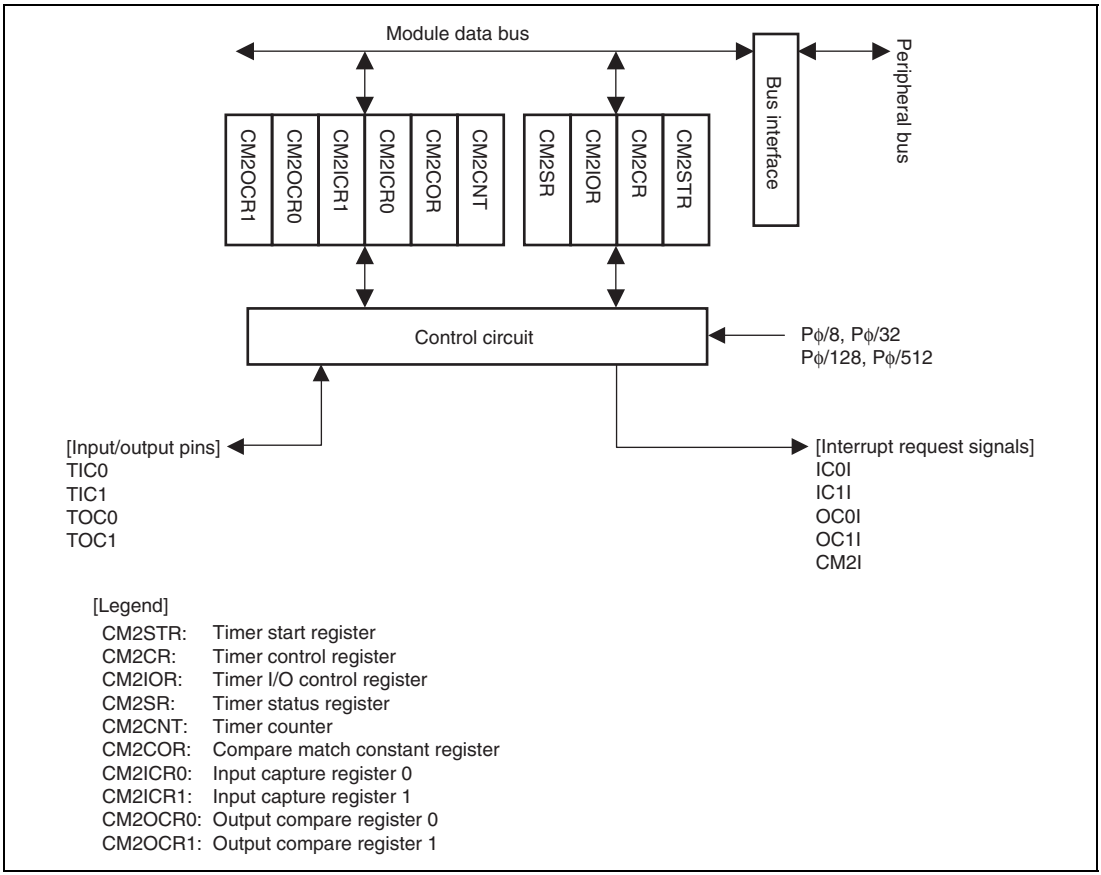


Figure 16.1 Block Diagram of CMT2

16.2 Input/Output Pin

Table 16.1 shows the CMT2 pin configuration.

Table 16.1 Pin Configuration

Pin Name	I/O	Function
TIC0	Input	Input capture input 0
TIC1	Input	Input capture input 1
TOC0	Output	Output compare output 0
TOC1	Output	Output compare output 1

16.3 Register Descriptions

The CMT2 has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 16.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Timer start register	CM2STR	R/W	H'0000	H'FFFEC100	16
Timer control register	CM2CR	R/W	H'0000	H'FFFEC104	16
Timer I/O control register	CM2IOR	R/W	H'0000	H'FFFEC108	16
Timer status register	CM2SR	R/W	H'0000	H'FFFEC10C	16
Timer counter	CM2CNT	R/W	H'00000000	H'FFFEC110	32
Compare match constant register	CM2COR	R/W	H'FFFFFFFF	H'FFFEC114	32
Input capture register 0	CM2ICR0	R	H'00000000	H'FFFEC118	32
Input capture register 1	CM2ICR1	R	H'00000000	H'FFFEC11C	32
Output compare register 0	CM2OCR0	R/W	H'FFFFFFFF	H'FFFEC120	32
Output compare register 1	CM2OCR1	R/W	H'FFFFFFFF	H'FFFEC124	32

16.3.1 Timer Start Register (CM2STR)

CM2STR is a 16-bit register that selects whether the timer counter (CM2CNT) operates or is stopped. CM2STR can only be accessed in word units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	STR	0	R/W	Counter Start Specifies whether the timer counter operates or is stopped. 0: CM2CNT count is stopped. 1: CM2CNT count is started.

16.3.2 Timer Control Register (CM2CR)

CM2CR is a 16-bit register that selects the counter clearing source and the counter input clock, and enables or disables interrupts. CM2CR can only be accessed in word units.

CM2CR should be set while the timer counter (CM2CNT) operation is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCLR[2:0]			-	-	-	CMS	-	OC1IE	OC0IE	IC1IE	IC0IE	CM2IE	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	CCLR[2:0]	000	R/W	Counter Clear Selects the CM2CNT counter clearing source. 000: The CM2CNT cleared by CM2COR compare match. 001: The CM2CNT not cleared. 010: The CM2CNT not cleared. 011: The CM2CNT not cleared. 100: The CM2CNT cleared by CM2ICR0 input capture. 101: The CM2CNT cleared by CM2ICR1 input capture. 110: The CM2CNT cleared by CM2OCR0 compare match. 111: The CM2CNT cleared by CM2OCR1 compare match.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	CMS	0	R/W	Timer Counter Size Selects either 16 or 32 bits as the size of the timer counter (CM2CNT). The number of bits selected with the CMS bit are valid in the compare match constant register (CM2COR), input capture registers (CM2ICR0 and CM2ICR1), and output compare registers (CM2OCR0 and CM2OCR1). 0: 32 bits 1: 16 bits
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7	OC1IE	0	R/W	Output Compare 1 Interrupt Enable Enables or disables output compare interrupt (OC1I) request generation when the OCF1 bit in CM2SR is set to 1. 0: Disables an output compare interrupt (OC1I) to be requested by the OCF1 bit. 1: Enables an output compare interrupt (OC1I) to be requested by the OCF1 bit.

Bit	Bit Name	Initial Value	R/W	Description
6	OC0IE	0	R/W	Output Compare 0 Interrupt Enable Enables or disables output compare interrupt (OC0I) request generation when the OCF0 bit in CM2SR is set to 1. 0: Disables an output compare interrupt (OC0I) to be requested by the OCF0 bit. 1: Enables an output compare interrupt (OC0I) to be requested by the OCF0 bit.
5	IC1IE	0	R/W	Input Capture 1 Interrupt Enable Enables or disables input capture interrupt (IC1I) request generation when the ICF1 bit in CM2SR is set to 1. 0: Disables an input capture interrupt (IC1I) to be requested by the ICF1 bit. 1: Enables an input capture interrupt (IC1I) to be requested by the ICF1 bit.
4	IC0IE	0	R/W	Input Capture 0 Interrupt Enable Enables or disables input capture interrupt (IC0I) request generation when the ICF0 bit in CM2SR is set to 1. 0: Disables an input capture interrupt (IC0I) to be requested by the ICF0 bit. 1: Enables an input capture interrupt (IC0I) to be requested by the ICF0 bit.
3	CM2IE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CM2I) request generation when CM2CNT and CM2COR values match. 0: Disables a compare match interrupt (CM2I) to be requested. 1: Enables a compare match interrupt (CM2I) to be requested.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	CKS[1:0]	00	R/W	Clock Select Selects the clock to be input to CM2CNT from four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit in CM2STR is set to 1, the CM2CNT starts counting based on the clock selected with the CKS[1:0] bits. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

16.3.3 Timer I/O Control Register (CM2IOR)

CM2IOR is a 16-bit register that controls CM2COR, CM2ICR0, CM2ICR1, CM2OCR0, and CM2OCR1. CM2IOR can only be accessed in word units.

CM2IOR should be set while the timer counter (CM2CNT) operation is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CM2E	-	OC1E	OC0E	OC1[1:0]		OC0[1:0]		-	-	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CM2E	0	R/W	Compare Match Enable Enables or disables the compare match operation using CM2COR. 0: Disables the compare match operation using CM2COR. 1: Disables the compare match operation using CM2COR.
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13	OC1E	0	R/W	Compare Match Enable 1 Enables or disables the compare match operation using CM2OCR1. 0: Disables the compare match operation using CM2OCR1. 1: Enables the compare match operation using CM2OCR1.
12	OC0E	0	R/W	Compare Match Enable 0 Enables or disables the compare match operation using CM2OCR0. 0: Disables the compare match operation using CM2OCR0. 1: Enables the compare match operation using CM2OCR0.
11, 10	OC1[1:0]	00	R/W	Output Compare Control 1 Sets the function of CM2OCR1. 00: Retains the output value. * 01: Initially outputs 0 and toggles the output value upon compare match. 10: Initially outputs 1 and toggles the output value upon compare match. 11: Setting prohibited
9, 8	OC0[1:0]	00	R/W	Output Compare Control 0 Sets the function of CM2OCR0. 00: Retains the output value.* 01: Initially outputs 0 and toggles the output value upon compare match. 10: Initially outputs 1 and toggles the output value upon compare match. 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	IC1E	0	R/W	Input Capture Enable 1 Enables or disables the input capture operation using the TIC1 pin. 0: Disables the input capture operation using the TIC1 pin. 1: Enables the input capture operation using the TIC1 pin.
4	IC0E	0	R/W	Input Capture Enable 1 Enables or disables the input capture operation using the TIC0 pin. 0: Disables the input capture operation using the TIC0 pin. 1: Enables the input capture operation using the TIC0 pin.
3, 2	IC1[1:0]	00	R/W	Input Capture Control 1 Sets the function of CM2ICR1. 00: Input capture at the rising edge. 01: Input capture at the falling edge. 10: Input capture at both edges. 11: Setting prohibited
1, 0	IC0[1:0]	00	R/W	Input Compare Control 0 Sets the function of CM2ICR0. 00: Input capture at the rising edge. 01: Input capture at the falling edge. 10: Input capture at both edges. 11: Setting prohibited

Note: * After power-on reset, 0 is output until CM2IOR is set.

16.3.4 Timer Status Register (CM2SR)

CM2SR is a 16-bit register that indicates generation of input capture, compare match, and overflow. CM2SR can only be accessed in word units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CM2F	OVF	-	-	OCF1	OCF0	ICF1	ICF0	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R

Note: * Only 0 can be written to after reading 1, to clear the flag.
When clearing a flag bit in the CM2SR register, only write 0 to the bit that is to be cleared, and write 1 to the other bits.

Bit	Bit Name	Initial Value	R/W	Description
15	CM2F	0	R/(W)*	Compare Match Flag Indicates that the CM2CNT and CM2COR values match. 0: The CM2CNT and CM2COR values do not match. [Clearing conditions] • Power-on reset or in module standby mode • A 0 is written to the CM2F bit after 1 is read from the bit. • The DMAC is activated by CM2I. • The DTC is activated by CM2I and the DISEL bit in MRB of the DTC is 0. 1: Indicates that the CM2CNT and CM2COR values match. [Setting condition] • The CM2CNT and CM2COR values match.

Bit	Bit Name	Initial Value	R/W	Description
14	OVF	0	R/(W)*	Overflow Flag Indicates that CM2CNT has overflowed (H'FFFFFFFF → H'00000000). This bit is valid only when CM2COR is set to H'FFFFFFFF (when the counter size is 32 bits) or H'0000FFFF (when the counter size is 16 bits). 0: CM2CNT has not overflowed. [Clearing conditions] • Power-on reset or in module standby mode • A 0 is written to the OVF bit after 1 is read from the bit. 1: CM2CNT has overflowed. [Setting condition] • CM2CNT overflows.
13, 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	OCF1	0	R/(W)*	Output Compare Flag 1 Indicates the CM2CNT and CM2OCR1 values match. 0: The CM2CNT and CM2OCR1 values do not match. [Clearing conditions] • Power-on reset or in module standby mode • A 0 is written to the OCF1 bit after 1 is read from the bit. • The DMAC is activated by OC1I. • The DTC is activated by OC1I and the DISEL bit in MRB of the DTC is 0. 1: The CM2CNT and CM2OCR1 values match. [Setting condition] • The CM2CNT and CM2OCR1 values match.

Bit	Bit Name	Initial Value	R/W	Description
10	OCF0	0	R/(W)*	Output Compare Flag 0 Indicates the CM2CNT and CM2OCR0 values match. 0: The CM2CNT and CM2OCR0 values do not match. [Clearing conditions] • Power-on reset or in module standby mode • A 0 is written to the OCF0 bit after 1 is read from the bit. • The DMAC is activated by OC0I. • The DTC is activated by OC0I and the DISEL bit in MRB of the DTC is 0. 1: The CM2CNT and CM2OCR0 values match. [Setting condition] • The CM2CNT and CM2OCR0 values match.
9	ICF1	0	R/(W)*	Input Capture Flag 1 Indicates that CM2ICR1 input capture has been generated. 0: Indicates that CM2ICR1 input capture has not been generated. [Clearing conditions] • Power-on reset or in module standby mode • A 0 is written to the ICF1 bit after 1 is read from the bit. • The DMAC is activated by IC1I. • The DTC is activated by IC1I and the DISEL bit in MRB of the DTC is 0. 1: Indicates that CM2ICR1 input capture has been generated. [Setting condition] • The CM2CNT value is transferred to CM2ICR1 by the input capture signal.

Bit	Bit Name	Initial Value	R/W	Description
8	ICF0	0	R/(W)*	Input Capture Flag 0 Indicates that CM2ICR0 input capture has been generated. 0: Indicates that CM2ICR0 input capture has not been generated. [Clearing conditions] • Power-on reset or in module standby mode • A 0 is written to the ICF0 bit after 1 is read from the bit. • The DMAC is activated by IC0I. • The DTC is activated by IC0I and the DISEL bit in MRB of the DTC is 0. 1: Indicates that CM2ICR0 input capture has been generated. [Setting condition] • The CM2CNT value is transferred to CM2ICR0 by the input capture signal.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Only 0 can be written to after reading 1, to clear the flag.
When clearing a flag bit in the CM2SR register, only write 0 to the bit that is to be cleared, and write 1 to the other bits.

16.3.5 Timer Counter (CM2CNT)

CM2CNT is a 32-bit register used as an upcounter.

Before starting counter operation, the timer control register (CM2CR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CM2CR), bits 0 to 15 in this register are valid. When writing to CM2CNT, a 32-bit value should be set with H'0000 in the upper bits. CM2CNT can only be accessed in longword units.

The initial value of CM2CNT is H'00000000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.6 Compare Match Constant Register (CM2COR)

CM2COR is a 32-bit register that specifies the time up to a compare match between the timer counter (CM2CNT) value and CM2COR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CM2CR), bits 0 to 15 in this register are valid. When writing to CM2COR, a 32-bit value should be set with H'0000 in the upper bits. CM2CNT can only be accessed in longword units. An overflow is detected when the CM2COR value is H'FFFFFFFF (when the counter size is 32 bits) or H'0000FFFF (when the counter size is 16 bits) and CM2CNT is cleared to 0. The initial value of CM2COR is H'FFFFFFFF.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.7 Input Capture Registers 0 and 1 (CM2ICR0 and CM2ICR1)

CM2ICR0 and CM2ICR1 are 32-bit registers in which the CM2CNT value is stored when an input capture is generated.

When the 16-bit counter size is selected with the CMS bit in the timer control register (CM2CR), bits 15 to 0 in these registers are valid. Writing to these registers is invalid. CM2ICR0 and CM2ICR1 can only be accessed in longword units. The initial value of CM2ICR0 and CM2ICR1 is H'00000000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16.3.8 Output Compare Registers 0 and 1 (CM2OCR0 and CM2OCR1)

CM2OCR0 and CM2OCR1 are 32-bit registers that set the value to be compared when an output compare is generated.

When the 16-bit counter size is selected with the CMS bit in the timer control register (CM2CR), bits 15 to 0 of these registers become valid. When writing to these registers, a 32-bit value should be set with H'0000 in the upper bits. CM2OCR0 and CM2OCR1 can only be accessed in longword units. The initial value of CM2OCR0 and CM2OCR1 is H'FFFFFFFF.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.4 Operation

When CM2CR is set and then the STR bit in CM2STR is set to 1, the CMT2 starts counter operation. Setting CM2IOR enables using the compare match function, input capture input function, and output compare output function.

16.4.1 Counter Operation

Writing 1 to the STR bit in CM2STR starts counter operation. When the CCLR[2:0] bits in CM2CR are set so that CM2CNT should be cleared by a specific counter clearing source and the counter clearing source is generated, CM2CNT is cleared to H'00000000 and continues incrementing. When the CCLR[2:0] bits are set so that CM2CNT should not be cleared by any specific counter clearing source, CM2CNT is cleared to H'00000000 only when an overflow is generated (H'FFFFFFFF → H'00000000 (when the counter size is 32 bits) or H'0000FFFF → H'00000000 (when the counter size is 16 bits)) and continues incrementing.

16.4.2 Compare Match Function

When the CM2CNT and CM2COR values match, the CM2F bit in CM2SR is set to 1.

To enable overflow detection, the CM2COR value should be set to H'FFFFFFFF (when the counter size is 32 bits) or H'0000FFFF (when the counter size is 16 bits). When the CM2CNT and CM2COR values match, CM2CNT is cleared to H'00000000 thus setting the CM2F and OVF bits in CM2SR to 1.

Figure 16.2 shows an example of procedure for setting compare match operation.

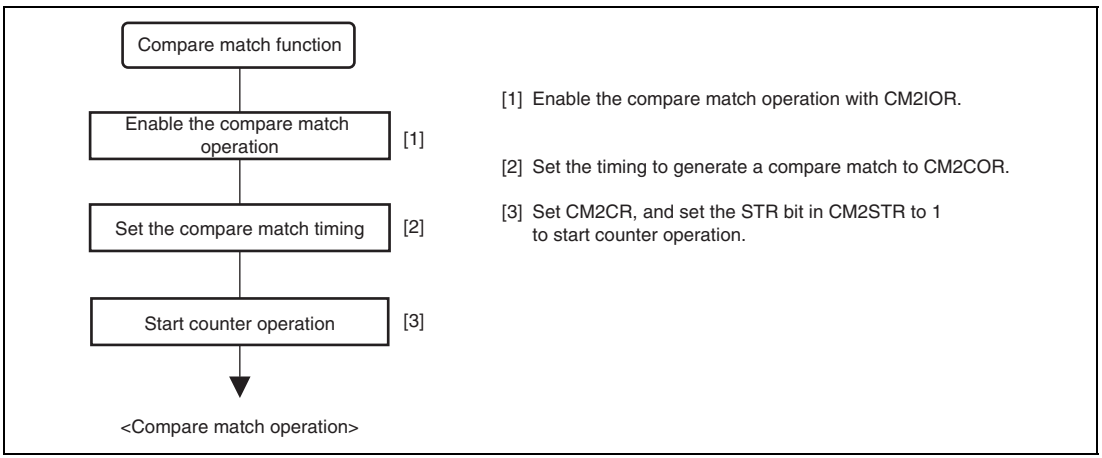


Figure 16.2 Procedure for Setting Compare Match Operation

Figure 16.3 shows an example when compare match with CM2COR is set as a counter clearing source.

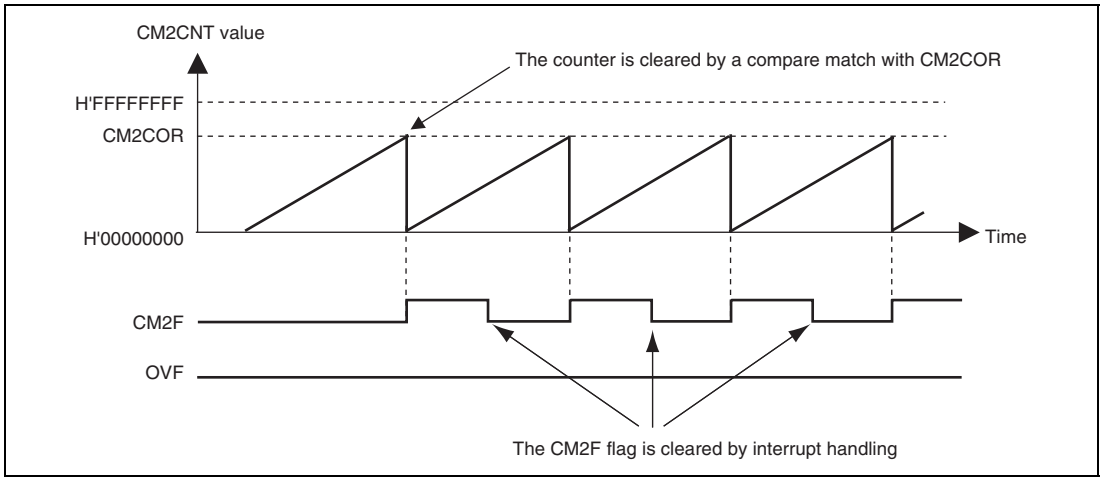


Figure 16.3 Example of Compare Match Operation

Figure 16.4 shows an example when CM2COR is set to H'FFFFFFFF and an overflow is detected.

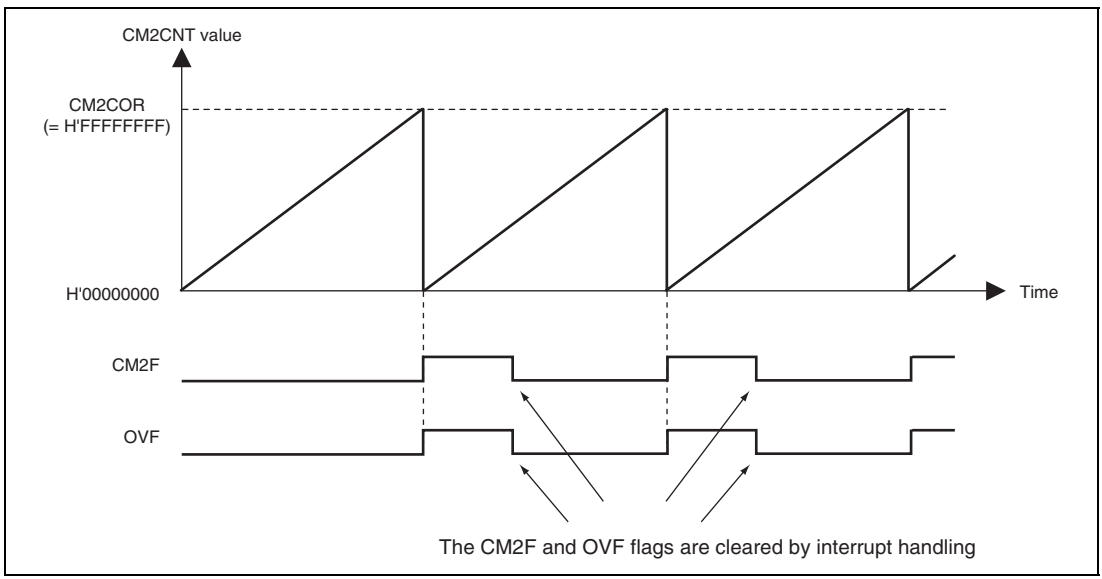


Figure 16.4 Example of Compare Match Operation (overflow detected)

16.4.3 Output Compare Function

Using the output compare function, toggle output from the relevant output pins can be provided. When the CM2CNT and CM2OCR0 or CM2OCR1 values match, the OCF0 or OCF1 bit in CM2SR is set to 1. Figure 16.5 shows an example of procedure for setting output compare operation.

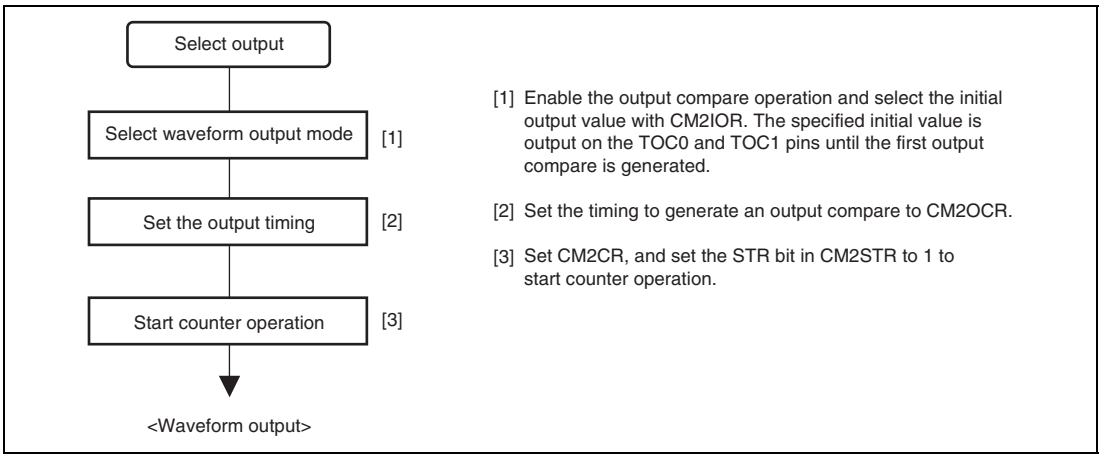


Figure 16.5 Procedure for Setting Output Compare Operation

Figure 16.6 shows an example when the counter is cleared upon compare match with CM2OCR1 and toggle outputs are provided from the TOC0 and TOC1 pins.

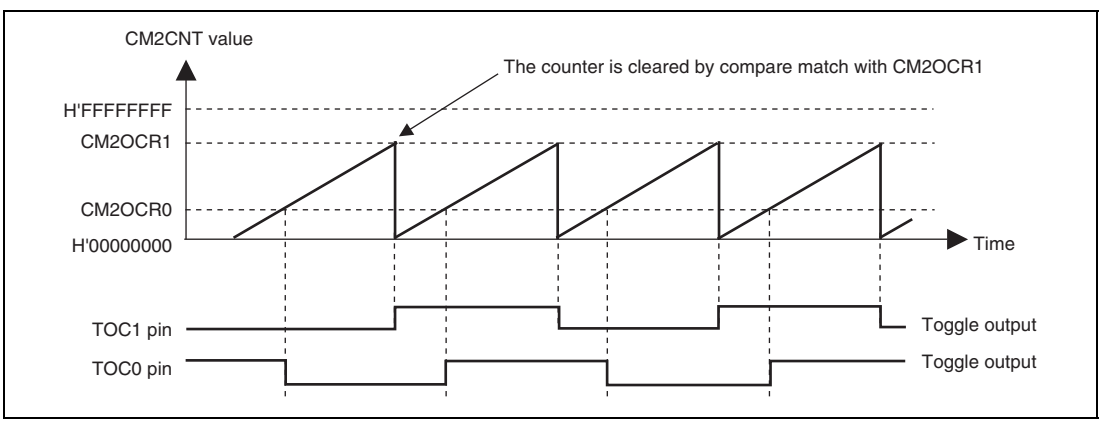


Figure 16.6 Example of Output Compare Operation

16.4.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CM2CNT value can be transferred to CM2ICR0 and CM2ICR1, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CM2CNT value is transferred to CM2ICR0 or CM2ICR1 using the input capture operation, the ICF0 or ICF1 bit in CM2SR is set to 1. Figure 16.7 shows an example of procedure for setting input capture operation.

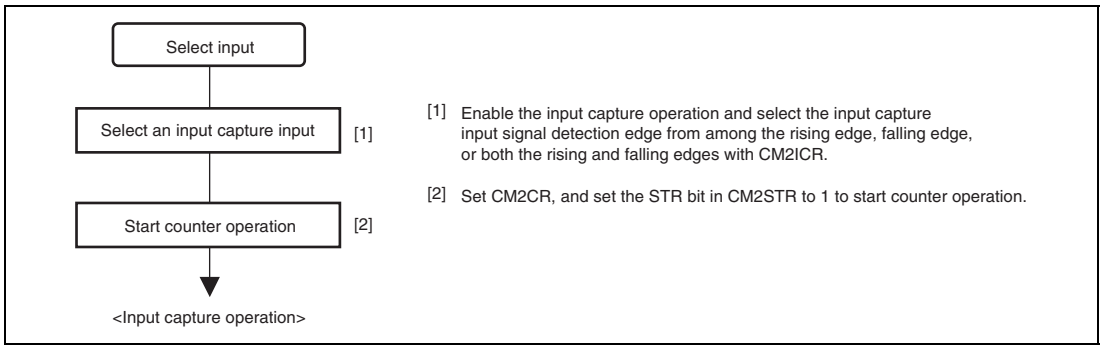


Figure 16.7 Procedure for Setting Input Capture Operation

Figure 16.8 shows an example in which both the rising and falling edges are selected for the TIC0 pin input capture input edge and the falling edge for the TIC1 pin, and CM2CNT is cleared by a CM2ICR1 input capture.

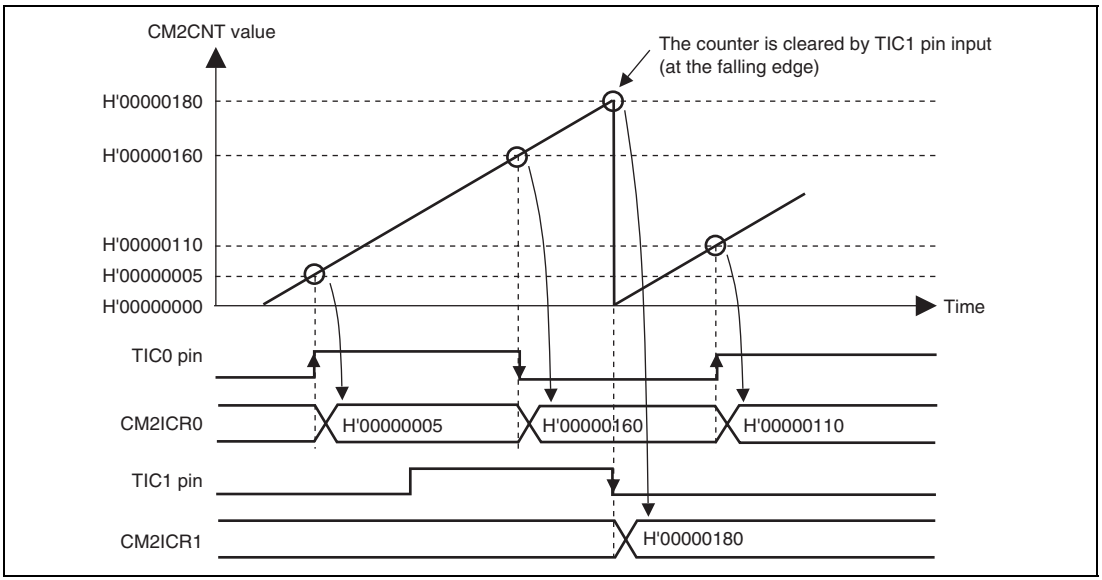


Figure 16.8 Example of Input Capture Operation

16.4.5 Counter Size

With the CMT2, either 16 or 32 bits can be selected as the counter size by using the CMS bit in CM2CR. When the counter is used as a 16-bit counter, a 32-bit value should be set to CM2COR with H'0000 in the upper bits; particularly, H'0000FFFF should be set to detect an overflow. Similarly, a 32-bit value should be set to CM2OCR0 and CM2OCR1with H'0000 in the upper bits. A 32-bit value with H'0000 in the upper bits is read from CM2ICR0 and CM2ICR1.

16.4.6 Count Timing based on CM2CNT

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected with the CKS[1:0] bits in CM2CR. Figure 16.9 shows the timing.

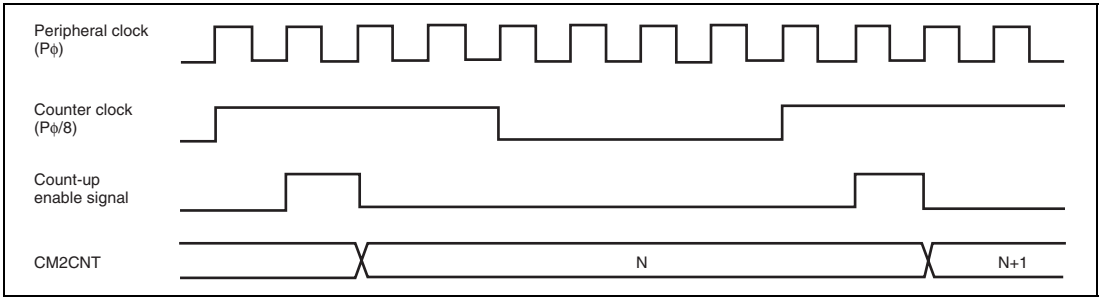


Figure 16.9 Count Timing (Pϕ/8)

16.4.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CM2OCR and CM2CNT values match (the CM2CNT value is updated immediately after the state). That is, the compare match signal is not generated if the CM2CNT counter clock is not input after a match between the CM2OCR and CM2CNT values. When a compare match signal is generated, the relevant value is output from the output compare output pin (TOC). Figure 16.10 shows output compare output timing.

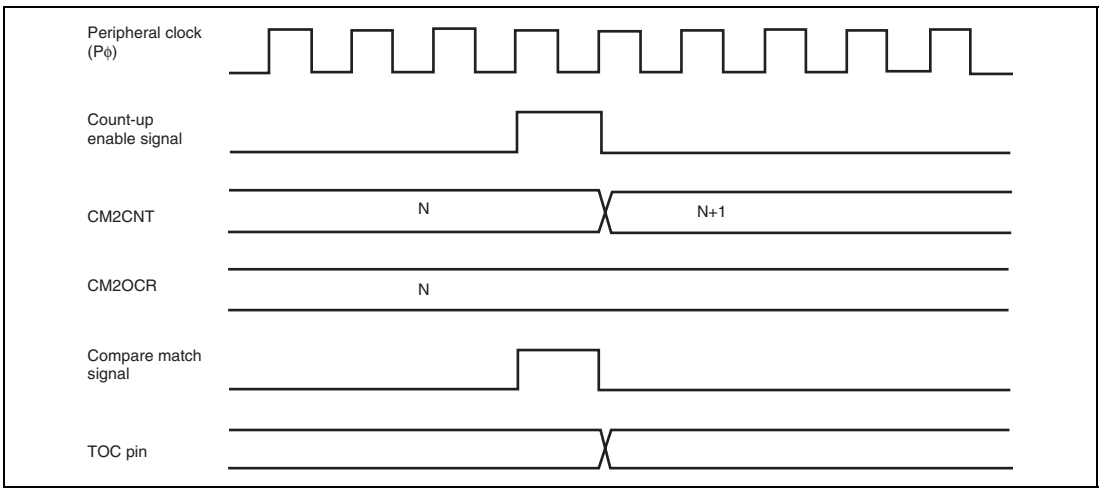


Figure 16.10 Output Compare Output Timing

16.4.8 Input Capture Signal Timing

Figure 16.11 shows the input capture timing.

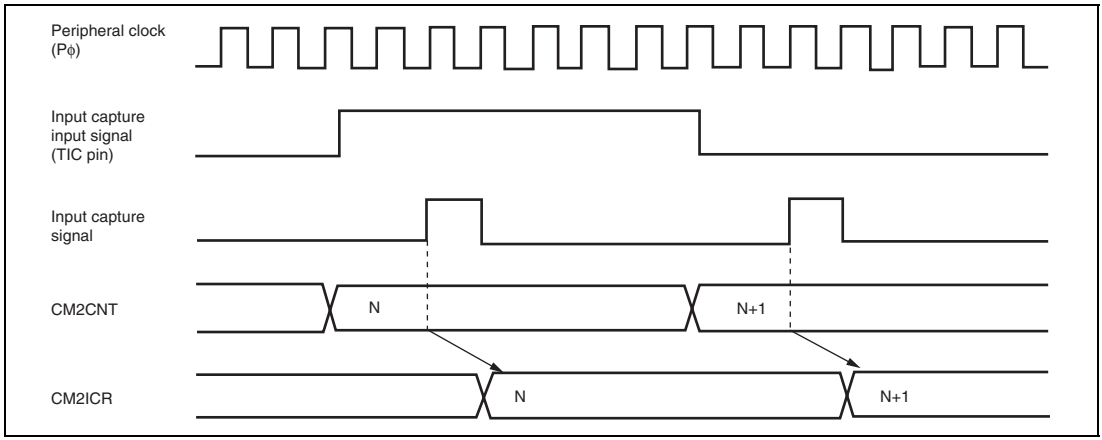


Figure 16.11 Input Capture Input Signal Timing

16.4.9 CM2CNT/CM2COR/CM2ICR/CM2OCR Access Timing

Since CM2CNT, CM2COR, CM2ICR, and CM2OCR are 32-bit registers but the peripheral data bus is 16 bits wide, a longword access to these registers is split up into an access to the upper 16 bits and an access to the lower 16 bits.

In writing, the upper bit data is temporarily stored in the write buffer, and then the lower bit data and stored upper bit data are written to the register simultaneously.

In reading, the upper bit data and lower bit data are read out simultaneously but the lower bit data is stored in the read buffer temporarily, and then the lower bit data is read from the read buffer in the lower bit data read cycle.

Figures 16.12 and 16.13 show CM2CNT write and read operation examples, respectively.

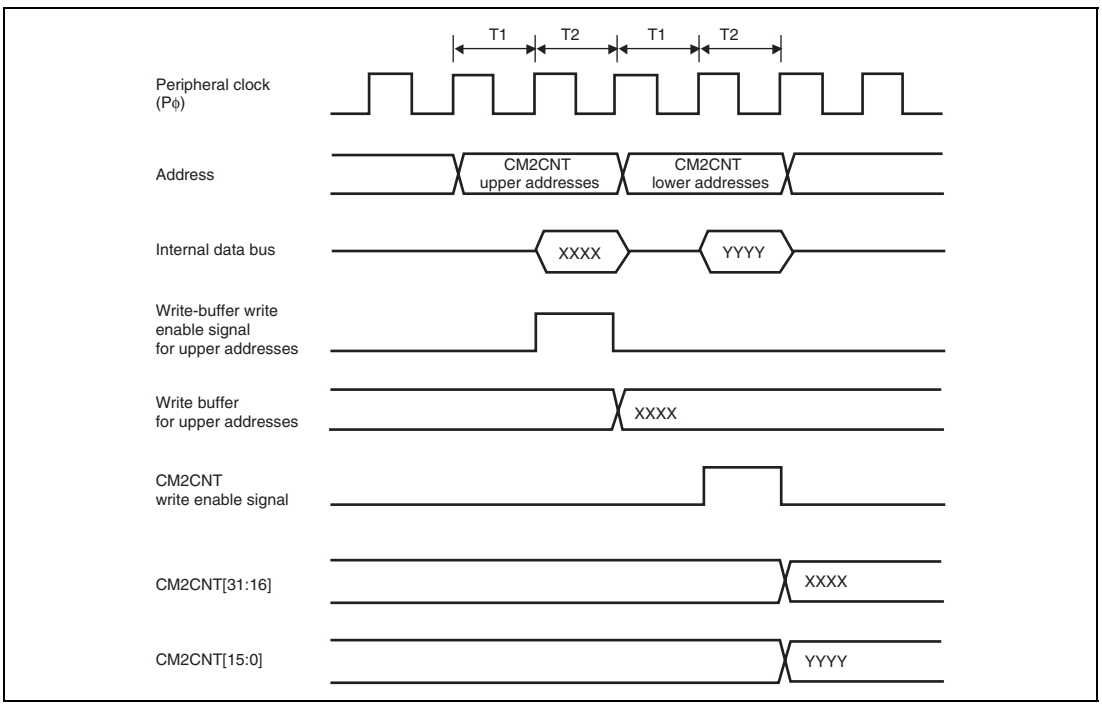


Figure 16.12 CM2CNT Write Operation Example

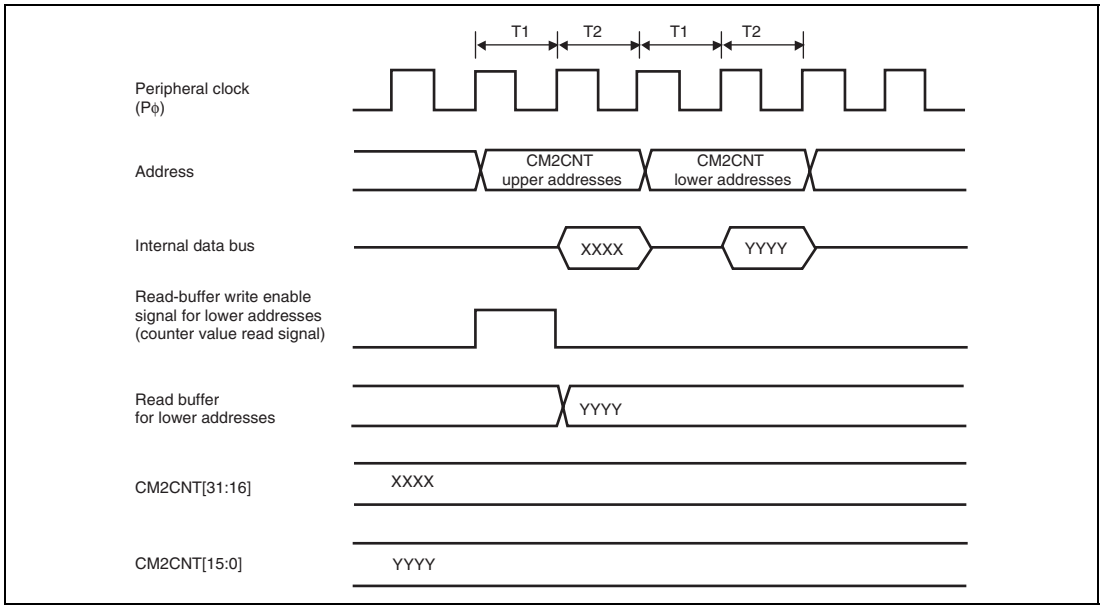


Figure 16.13 CM2CNT Read Operation Example

16.5 Interrupts

16.5.1 CMT2 Interrupt Sources and DTC/DMAC Transfer Requests

The CMT2 has five interrupt sources: two input capture interrupt requests (IC0I and IC1I), two output compare interrupt requests (OC0I and OC1I), and a compare match interrupt request (CM2I).

Table 16.3 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CM2IE bits in CM2CR and are separately issued to the interrupt controller. IC0I or IC1I is issued when the ICF0 or ICF1 flag in CM2SR is set to 1; OC0I or OC1I is issued when the OCF0 or OCF1 flag in CM2SR is set to 1; and CM2I is issued when the CM2F flag in CM2SR is set to 1. The relevant flag bit should be cleared by the user exception handling routine. If the bit is not cleared, the interrupt is requested again. If 1 is read from the flag bit and the flag is again set before 0 is written to the flag bit to clear the flag, writing 0 to the flag bit does not clear the flag; in this case, 1 should be read again from the flag bit and then 0 should be written.

Each interrupt request can activate the direct memory access controller (DMAC) or data transfer controller (DTC). When the DMAC is used for data transfer, the flag is automatically cleared on data transfer completion and thus an interrupt request is not issued to the CPU. When the DTC is used for data transfer, the flag is automatically cleared on data transfer completion and thus an interrupt request is not issued to the CPU if the DISEL bit is 0 and transfer counter value is not 0 in DTC. If both the DISEL bit and transfer counter value are 0 or if the DISEL bit is 1, the flag is not automatically cleared on data transfer completion and thus an interrupt request is issued to the CPU.

Table 16.3 CMT2 Interrupt Sources

Interrupt Source	Interrupt	Interrupt Enable Bit	DMAC/DTC Activation	Priority
CM2I	Interrupt caused by compare match (CM2F)	CM2IE	Possible	High
IC0I	Interrupt caused by input capture (ICF0)	IC0IE	Possible	↕
IC1I	Interrupt caused by input capture (ICF1)	ICI1E	Possible	
OC0I	Interrupt caused by output compare flag (OCF0)	OC0IE	Possible	
OC1I	Interrupt caused by output compare flag (OCF1)	OC1IE	Possible	Low

16.5.2 Timing of Compare Match Flag Setting

A compare match signal is generated in the last state in which the CM2COR and CM2CNT values match (the CM2CNT value is updated to H'00000000 immediately after the state) and the CM2F bit in CM2SR is set to 1. That is, the compare match signal is not generated if the CM2CNT counter clock is not input after a match between the CM2COR and CM2CNT values. Figure 16.14 shows the timing of CM2F bit setting.

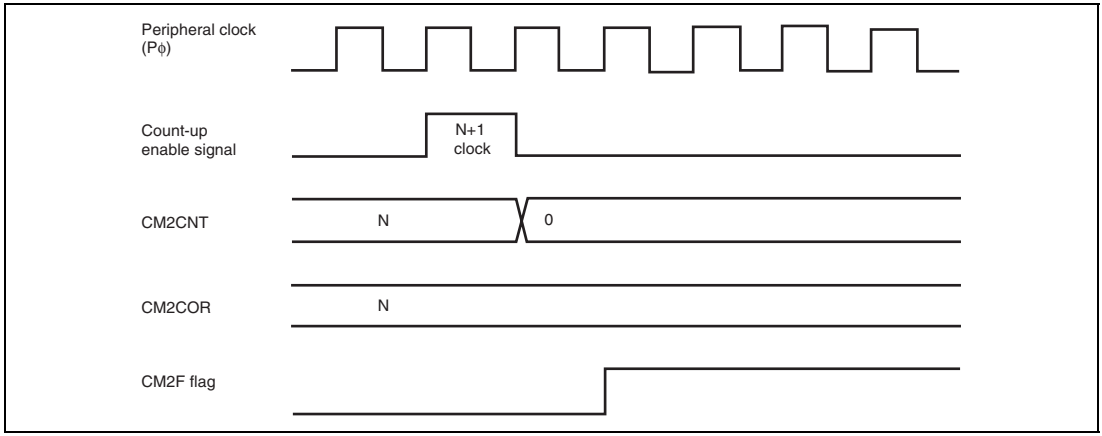


Figure 16.14 Timing of CM2F Flag Setting

16.5.3 Timing of Output Compare Flag Setting

Figure 16.15 shows the timing of setting the OCF0 and OCF1 flags in CM2SR when an output compare is generated.

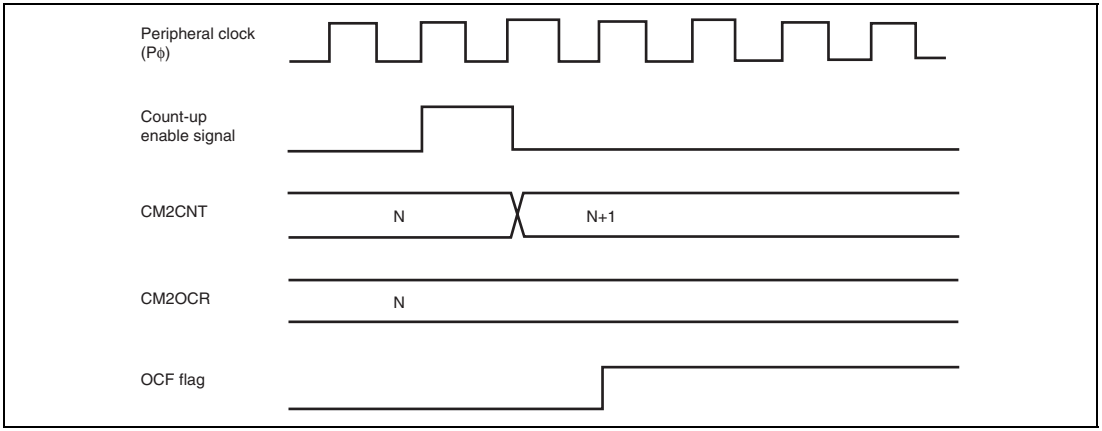


Figure 16.15 Timing of OCF Flag Setting

16.5.4 Timing of Input Capture Flag Setting

Figure 16.16 shows the timing of setting the ICF0 and ICF1 flags in CM2SR when an input capture is generated.

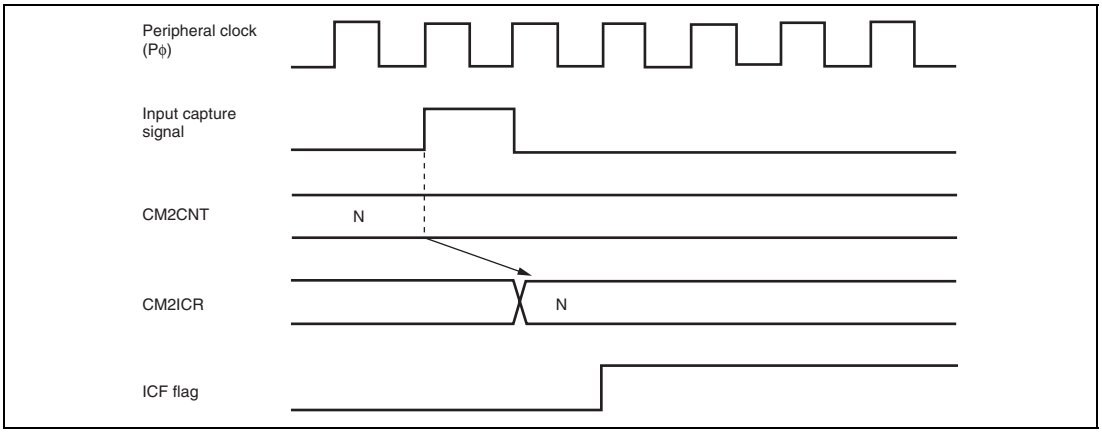


Figure 16.16 Timing of ICF Flag Setting

16.6 Usage Notes

16.6.1 Setting Module Standby Mode

The CMT2 module operation can be enabled and disabled using the standby control register. With the initial value, the CMT2 is halted. Register access is enabled by canceling module standby mode. For details, refer to section 32, Power-Down Modes.

16.6.2 Conflict between CM2CNT Writing and Incrementing or Clearing

Even if the CM2CNT incrementing or clearing process occurs in the T2 state for the lower address during the CM2CNT write cycle, neither incrementing or clearing is performed since writing to CM2CNT takes priority. Figure 16.17 shows the conflict between CM2CNT writing and incrementing.

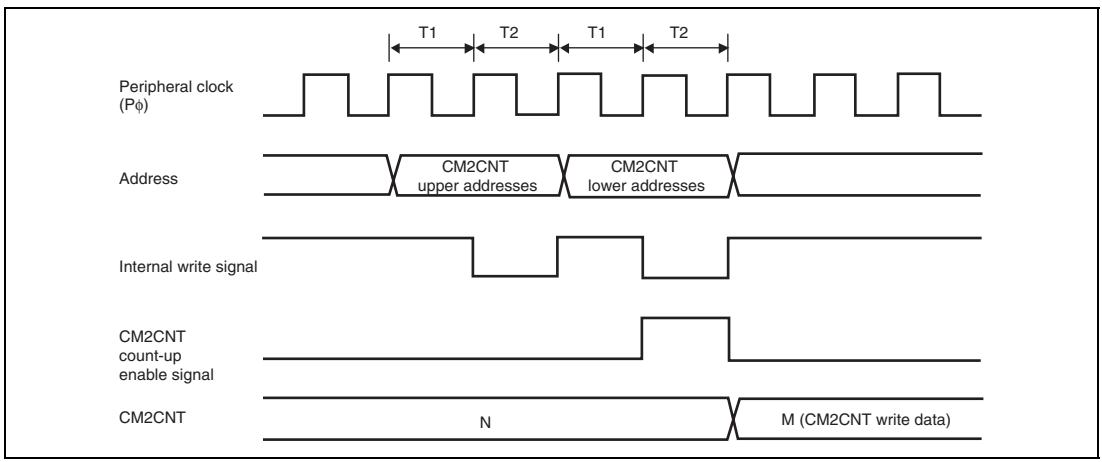


Figure 16.17 Conflict between CM2CNT Writing and Incrementing

16.6.3 Conflict between CM2COR Writing and Compare Match

If a compare match signal is generated in the T2 state for lower addresses during the CM2COR write cycle, writing to CM2COR takes priority, and a compare match signal is generated. Figure 16.18 shows the conflict between CM2COR writing and compare match.

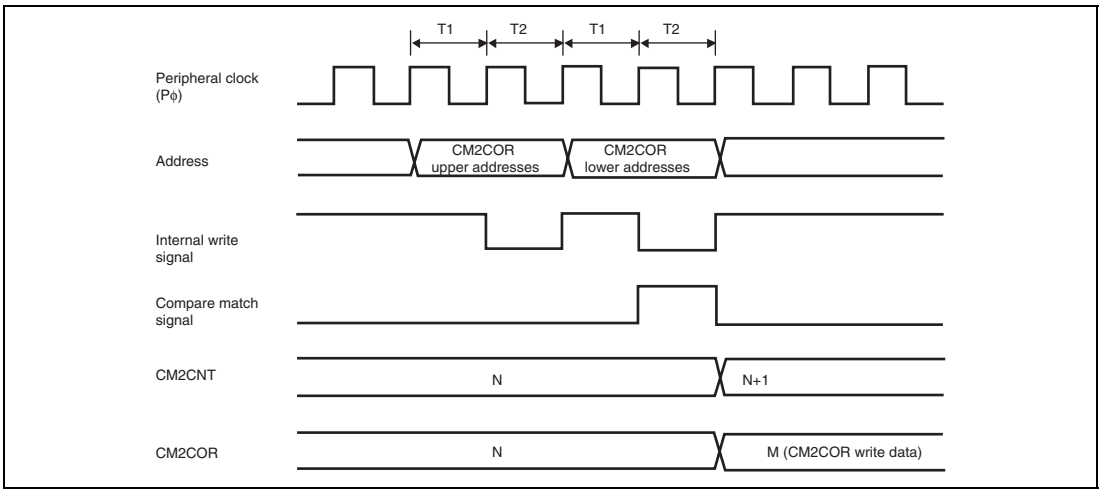


Figure 16.18 Conflict between CM2COR Writing and Compare Match

16.6.4 Conflict between CM2OCR Writing and Compare Match

If a compare match signal is generated in the T2 state for the lower addresses during the CM2OCR write cycle, writing to CM2OCR takes priority, and a compare match signal is generated. Figure 16.19 shows the conflict between CM2OCR writing and compare match.

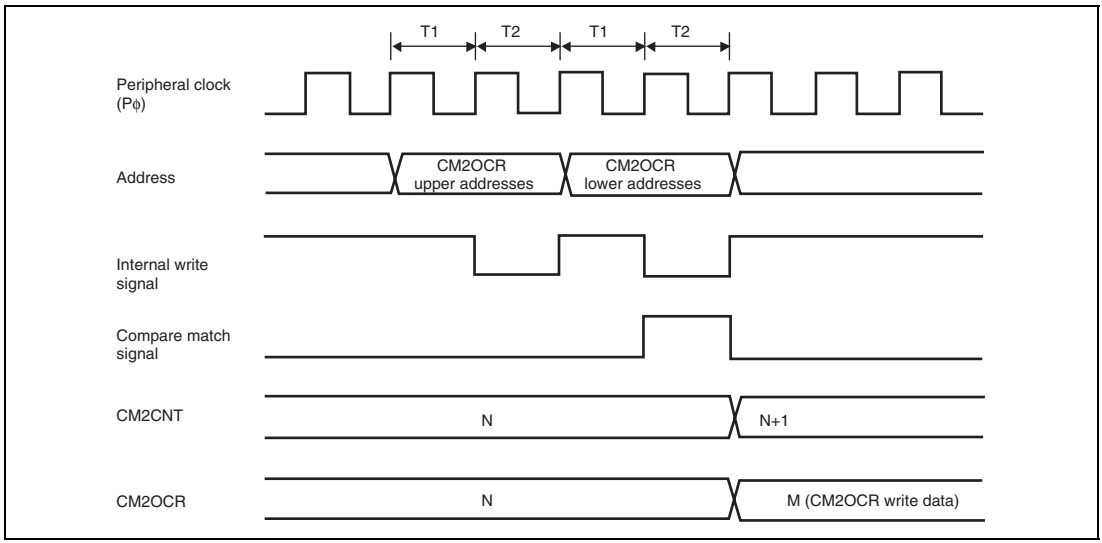


Figure 16.19 Conflict between CM2OCR Writing and Compare Match

16.6.5 Conflict between CM2CNT Reading and Incrementing or Clearing

If the CM2CNT incrementing or clearing process occurs at the same time that or after the upper address data is read, the value having been in CM2CNT before incremented or cleared is read as the lower address data. Figure 16.20 shows the conflict between CM2CNT reading and incrementing.

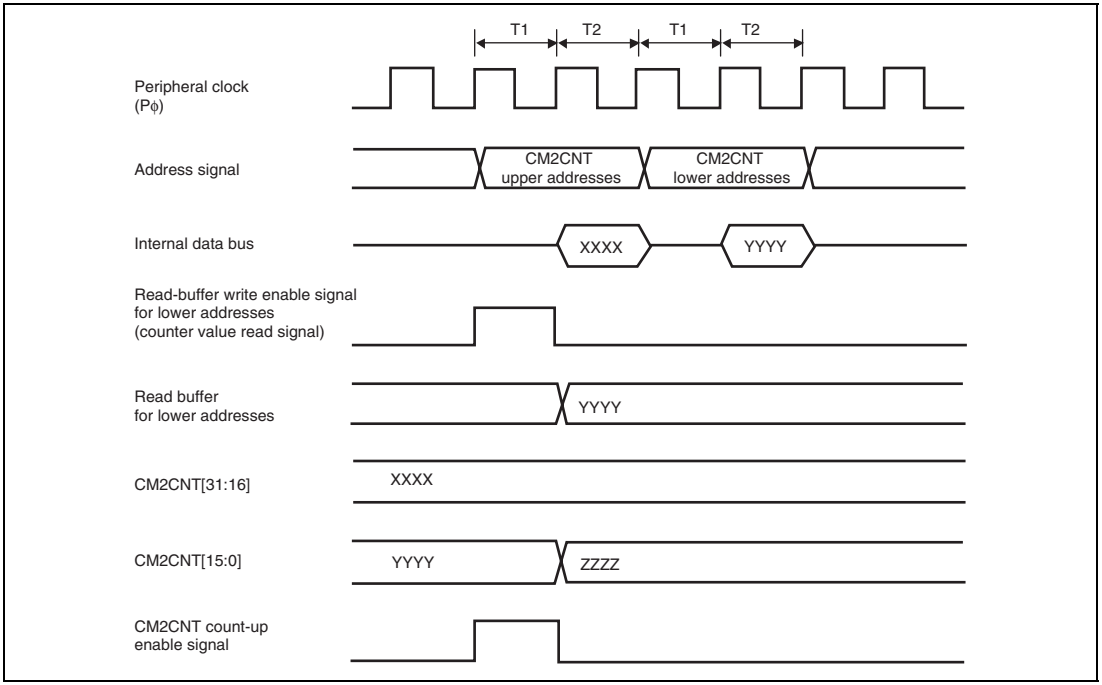


Figure 16.20 Conflict between CM2CNT Reading and Incrementing (when the upper address data reading and incrementing process occur simultaneously)

16.6.6 Conflict between CM2ICR Reading and Input Capture

If an input capture signal is generated at the same time that or after the CM2ICR upper address data is read, the value having been in CM2ICR before updated by input capture transfer is read as the lower address data. Figure 16.21 shows the conflict between CM2ICR reading and input capture.

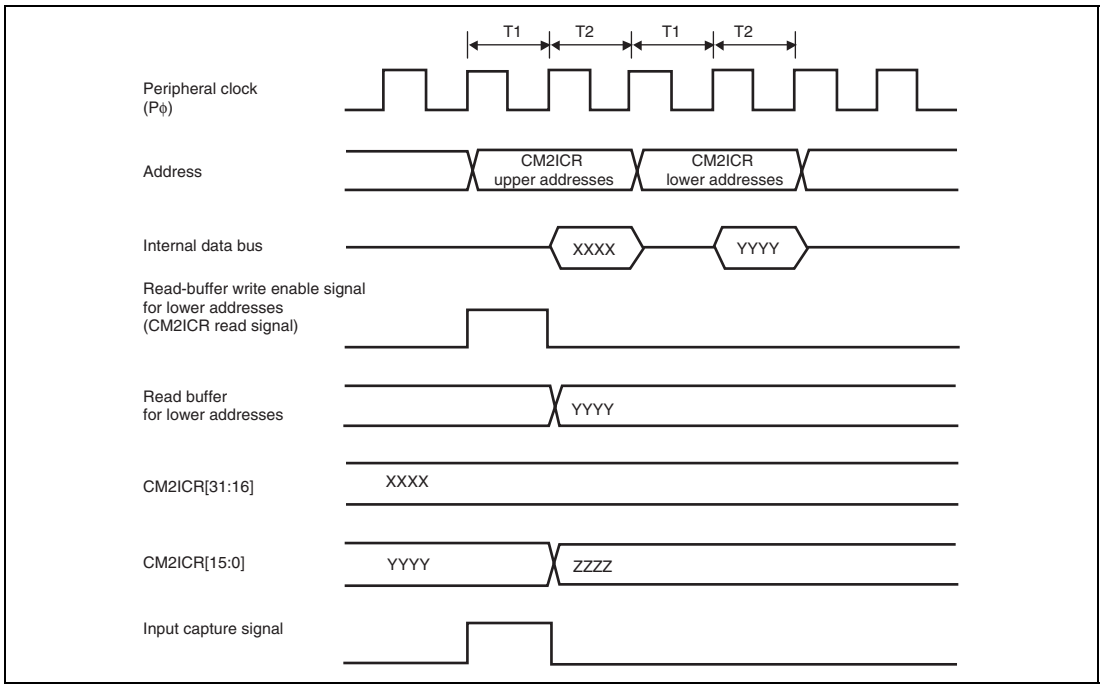


Figure 16.21 Conflict between CM2ICR Reading and Input Capture (when upper address data reading and input capture signal generation occur simultaneously)

16.6.7 Note on Clearing of Flag Bits in CM2SR Register

When clearing a flag in the CM2SR register, only write 0 to the bit that is to be cleared, and write 1 to the other bits. For example, when the ICF0 flag is to be cleared to 0, write H'CE00 to the CM2SR register.

Section 17 Watchdog Timer (WDT)

This LSI has a single 8-bit watchdog timer (WDT), which externally outputs an overflow signal ($\overline{\text{WDTOVF}}$) if its counter overflows when the value of the counter has not been updated because of a system malfunction. The WDT can simultaneously generate an internal reset signal for the entire LSI.

The WDT also uses a single timer to count the clock oscillation settling period for release from software standby mode. This timer can also be used as a general watchdog timer or an interval timer.

17.1 Features

- Can be used to ensure the clock oscillation settling time
  The WDT is used when this LSI is released from software standby mode.
- Can switch between watchdog timer mode and interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ signal in watchdog timer mode
  When the counter overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. It is possible to select whether to reset the LSI internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset type.
- Interrupt generation in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (Pϕ × 1 to Pϕ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

Figure 17.1 shows a block diagram of the WDT.

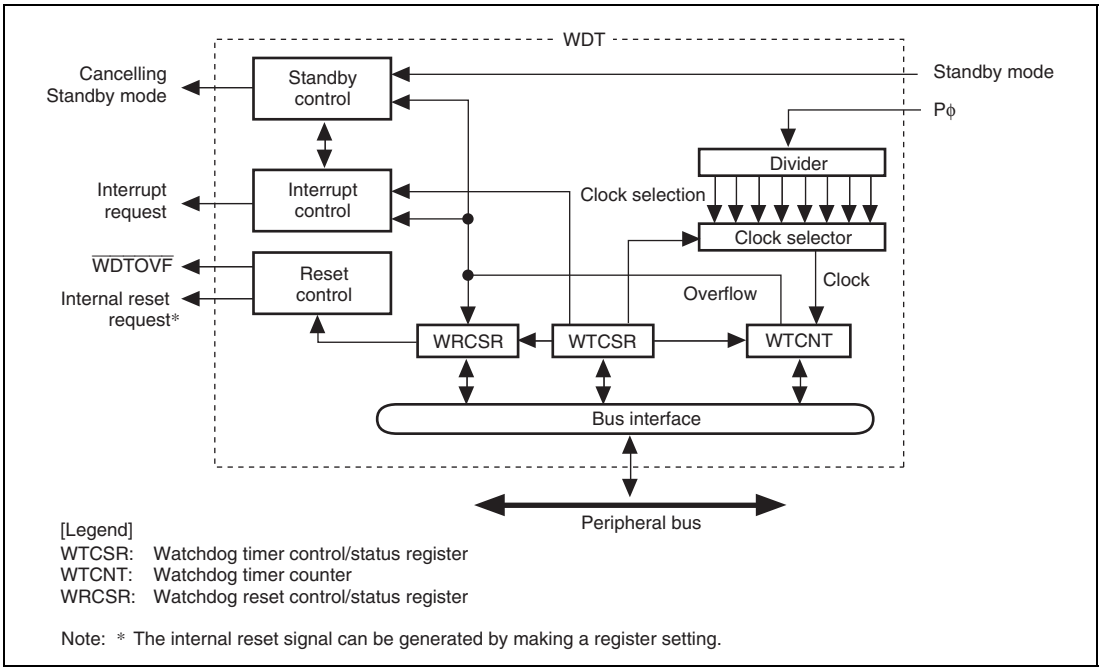


Figure 17.1 Block Diagram of WDT

17.2 Input/Output Pin

Table 17.1 shows the pin configuration of the WDT.

Table 17.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

17.3 Register Descriptions

The WDT has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 17.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Watchdog timer counter	WTCNT	R/W	H'00	H'FFFE0002	16*
Watchdog timer control/status register	WTCSR	R/W	H'18	H'FFFE0000	16*
Watchdog reset control/status register	WRCSR	R/W	H'1F	H'FFFE0004	16*

Note: * For the access size, see section 17.3.4, Notes on Register Access.

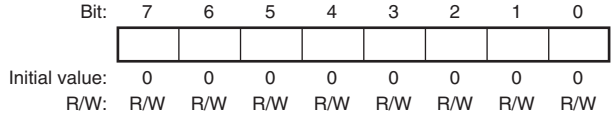
17.3.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that is incremented by cycles of the selected clock signal. When an overflow occurs, it generates a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) in watchdog timer mode and an interrupt in interval timer mode.

WTCNT is initialized to H'00 by a power-on reset caused by the $\overline{\text{RES}}$ pin or an internal reset issued for release from deep software standby mode.

Use word access to write to WTCNT, writing H'5A in the upper byte. Use byte access to read from WTCNT.

Note: The method for writing to WTCNT differs from that for other registers to prevent erroneous writes. See section 17.3.4, Notes on Register Access, for details.



17.3.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the count, overflow flags, and timer enable bit.

WTCSR is initialized to H'18 by a power-on reset caused by the $\overline{\text{RES}}$ pin or an internal reset issued for release from deep software standby mode.

When this register is used to count the clock oscillation settling period for release from software standby mode, its value is retained after the counter overflows.

Use word access to write to WTCSR, writing H'A5 in the upper byte. Use byte access to read from WTCSR.

Note: The method for writing to WTCSR differs from that for other registers to prevent erroneous writes. See section 17.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{\text{IT}}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IOVF	0	R/(W)	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: No overflow [Clearing condition] • When 0 is written to IOVF after reading IOVF = 1 [Setting condition] 1: WTCNT overflow in interval timer mode • When WTCNT overflows in interval timer mode

Bit	Bit Name	Initial Value	R/W	Description
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: When the WTCNT overflows in watchdog timer mode, the $\overline{WDTOVF}$ signal is output externally. If this bit is modified when the WDT is running, the up-count may not be performed correctly.
5	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in software standby mode. 0: Timer disabled Count-up stops and WTCNT value is retained 1: Timer enabled
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CKS[2:0]	000	R/W	Clock Select These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow period that is shown in the table is the value when the peripheral clock (Pϕ) is 50 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 × Pϕ	5.12 µs
001:	1/64 × Pϕ	328 µs
010:	1/128 × Pϕ	655 µs
011:	1/256 × Pϕ	1.31 ms
100:	1/512 × Pϕ	2.62 ms
101:	1/1024 × Pϕ	5.24 ms
110:	1/4096 × Pϕ	21.0 ms
111:	1/16384 × Pϕ	83.9 ms

Note: If bits CKS[2:0] are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

17.3.3 Watchdog Reset Control/Status Register (WRCSR)

WRCSR is an 8-bit readable/writable register that controls output of the internal reset signal generated by watchdog timer counter (WTCNT) overflow.

WRCSR is initialized to H'1F by a power-on reset caused by the $\overline{\text{RES}}$ pin or an internal reset issued for release from deep software standby mode.

Note: The method for writing to WRCSR differs from that for other registers to prevent erroneous writes. See section 17.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow [Clearing condition] • When 0 is written to WOVF after reading WOVF = 1 1: WTCNT has overflowed in watchdog timer mode [Setting condition] • When WTCNT has overflowed in watchdog timer mode
6	RSTE	0	R/W	Reset Enable Selects whether to generate a signal to reset the LSI internally if WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Not reset when WTCNT overflows* 1: Reset when WTCNT overflows Note: * LSI not reset internally, but WTCNT and WTCSR reset within WDT.

Bit	Bit Name	Initial Value	R/W	Description
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset
4 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

17.3.4 Notes on Register Access

The watchdog timer counter (WTCNT), watchdog timer control/status register (WTCSR), and watchdog reset control/status register (WRCSR) are more difficult to write to than other registers. The procedures for reading or writing to these registers are given below.

(1) Writing to WTCNT and WTCSR

These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 17.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

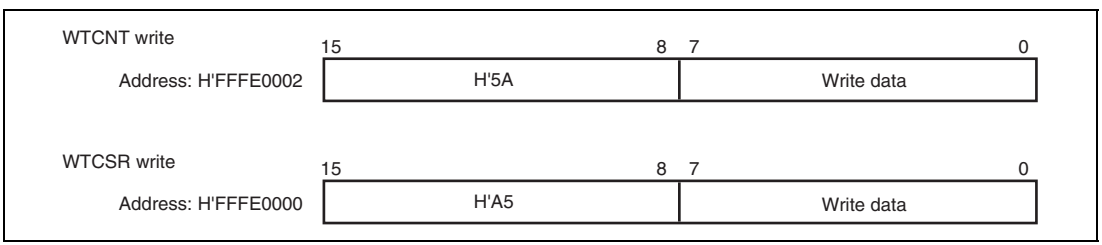


Figure 17.2 Writing to WTCNT and WTCSR

(2) Writing to WRCSR

WRCSR must be written by a word access to address H'FFFE0004. It cannot be written by byte transfer or longword transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 17.3.

To write 0 to the WOVF bit, write H'A5 to the upper byte and write the write data to the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

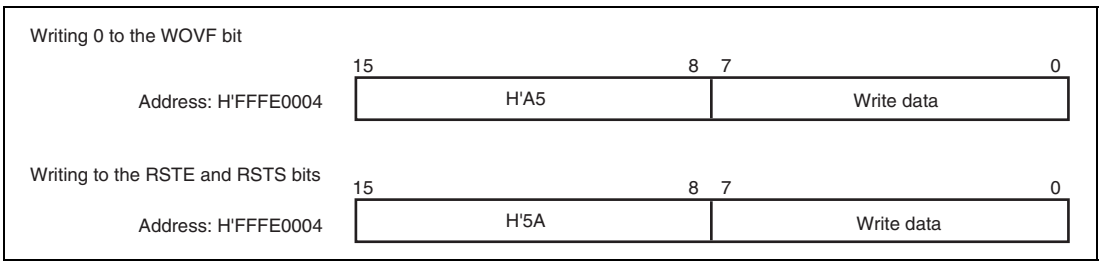


Figure 17.3 Writing to WRCSR

(3) Reading from WTCNT, WTCSR, and WRCSR

WTCNT, WTCSR, and WRCSR are read in a method similar to other registers. WTCSR is allocated to address H'FFFE0000, WTCNT to address H'FFFE0002, and WRCSR to address H'FFFE0004. Byte transfer instructions must be used for reading from these registers.

17.4 WDT Usage

17.4.1 Canceling Software Standby Mode

The WDT can be used to cancel software standby mode with an interrupt such as an NMI interrupt. The procedure is described below. (The WDT does not operate when resets are used for canceling, so keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until clock oscillation settles.)

1. Before making a transition to software standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS[2:0] bits in WTCSR and the initial value of the counter in WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. After setting the STBY bit in the standby control register (STBCR) to 1 and the DPSTBY bit in the deep standby control register (DPSTBCR) to 0 (see section 32, Power- Down Modes for the both registers), the execution of a SLEEP instruction puts the system in software standby mode and clock operation then stops.
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and this LSI resumes operation. The WOVF flag in WRCSR is not set when this happens.

17.4.2 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in WTCSR to 1, the type of count clock in the CKS[2:0] bits in WTCSR, whether this LSI is to be reset internally or not in the RSTE bit in WRCSR, the reset type if it is generated in the RSTS bit in WRCSR, and the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WRCSR to 1, and the $\overline{\text{WDTOVF}}$ signal is output externally (figure 17.4). The $\overline{\text{WDTOVF}}$ signal can be used to reset the system. The $\overline{\text{WDTOVF}}$ signal is output for $64 \times P\phi$ clock cycles.
5. If the RSTE bit in WRCSR is set to 1, a signal to reset the inside of this LSI can be generated simultaneously with the $\overline{\text{WDTOVF}}$ signal. Either power-on reset or manual reset can be selected for this interrupt by the RSTS bit in WRCSR. The internal reset signal is output for $128 \times P\phi$ clock cycles.

6. When a WDT overflow reset is generated simultaneously with a reset input on the $\overline{\text{RES}}$ pin, the $\overline{\text{RES}}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.

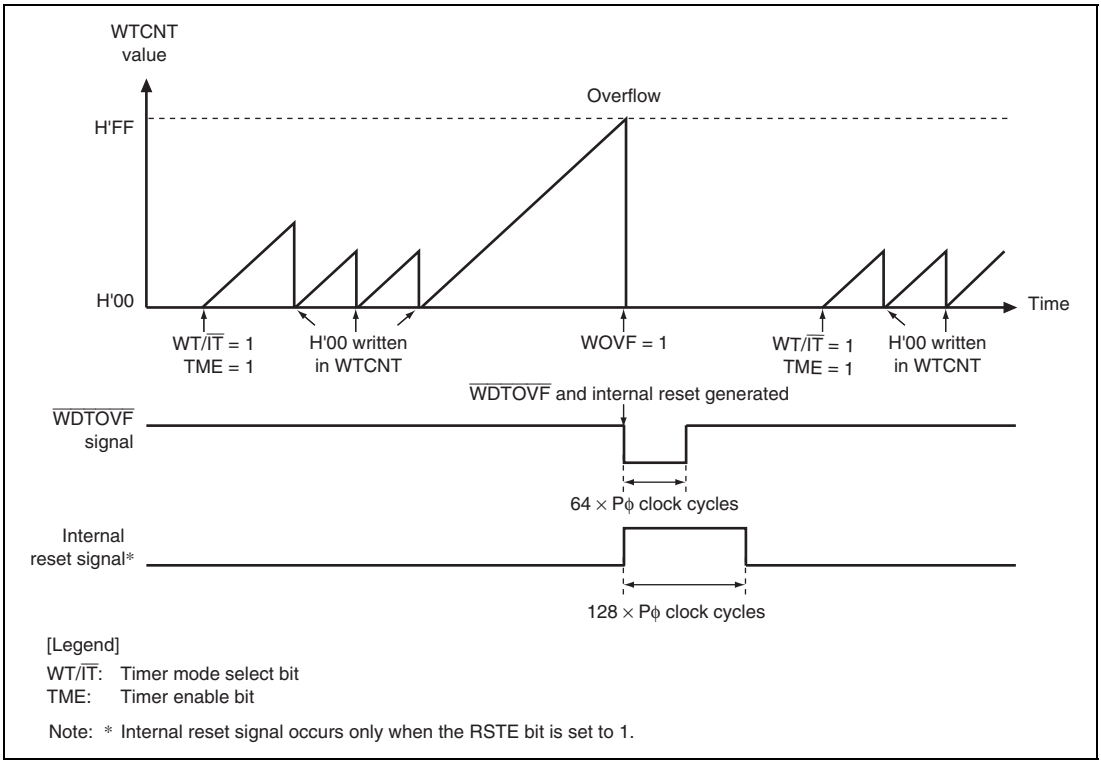


Figure 17.4 Operation in Watchdog Timer Mode

17.4.3 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in WTCSR to 0, set the type of count clock in the CKS[2:0] bits in WTCSR, and set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF bit in WTCSR to 1 and an interval timer interrupt request is sent to the INTC. The counter then resumes counting.

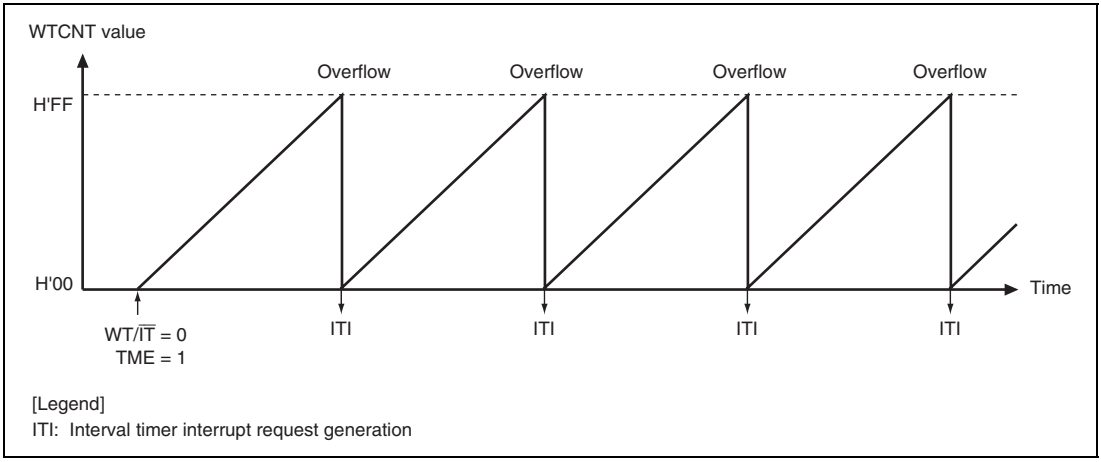


Figure 17.5 Operation in Interval Timer Mode

17.5 Interrupt Sources

Table 17.3 gives details on the interrupt source.

The interval timer interrupt (ITI) is generated when the interval timer overflow flag (IOVF) in the watchdog timer control/status register (WTCSR) is set to 1.

Clearing the interrupt flag bit to 0 cancels the interrupt request.

Table 17.3 Interrupt Source

Abbreviation	Interrupt Source	Interrupt Enable Bit	Interrupt Flag
ITI	Interval timer interrupt	—	Interval timer overflow flag (IOVF)

17.6 Usage Notes

Pay attention to the following points when using the WDT in either the interval timer or watchdog timer mode.

17.6.1 Timer Variation

After timer operation has started, the period from the power-on reset point to the first count up timing of WTCNT varies depending on the time period that is set by the TME bit of WTCSR. The shortest such time period is thus one cycle of the peripheral clock, Pϕ, while the longest is the result of frequency division according to the value in the CKS[2:0] bits. The timing of subsequent incrementation is in accord with the selected frequency division ratio. Accordingly, this time difference is referred to as timer variation.

This also applies to the timing of the first incrementation after WTCNT has been written to during timer operation.

17.6.2 Prohibition against Setting H'FF to WTCNT

When the value in WTCNT reaches H'FF, the WDT assumes that an overflow has occurred. Accordingly, when H'FF is set in WTCNT, an interval timer interrupt or WDT reset will occur immediately, regardless of the current clock selection by the CKS[2:0] bits.

17.6.3 Interval Timer Overflow Flag

When the value of WTCNT is H'FF, clearing of the IOVF flag is not possible. Clear the IOVF flag after the value of WTCNT has become H'00 or after writing a value other than H'FF to WTCNT.

17.6.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circuits. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 17.6.

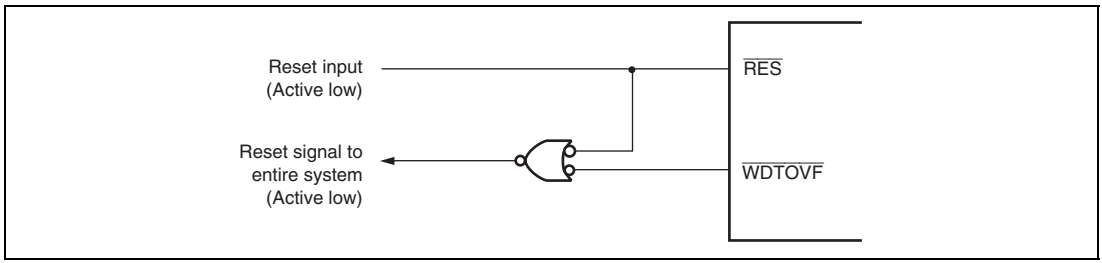


Figure 17.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

17.6.5 Manual Reset in Watchdog Timer Mode

When a manual reset occurs in watchdog timer mode, the current bus cycle is continued. If a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception handling will be pended until the CPU acquires the bus mastership.

However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than 128 cycles of Pϕ (the internal manual reset interval cycles), the internal manual reset source is ignored instead of being deferred, and manual reset exception handling is not executed.

17.6.6 Usage Note on the $\overline{\text{WDTOVF}}$ Pin

In general, do not pull the $\overline{\text{WDTOVF}}$ pin down. If pulling down is required, the value of the resistor must be at least 1 MΩ.

Section 18 Serial Communication Interface (SCI)

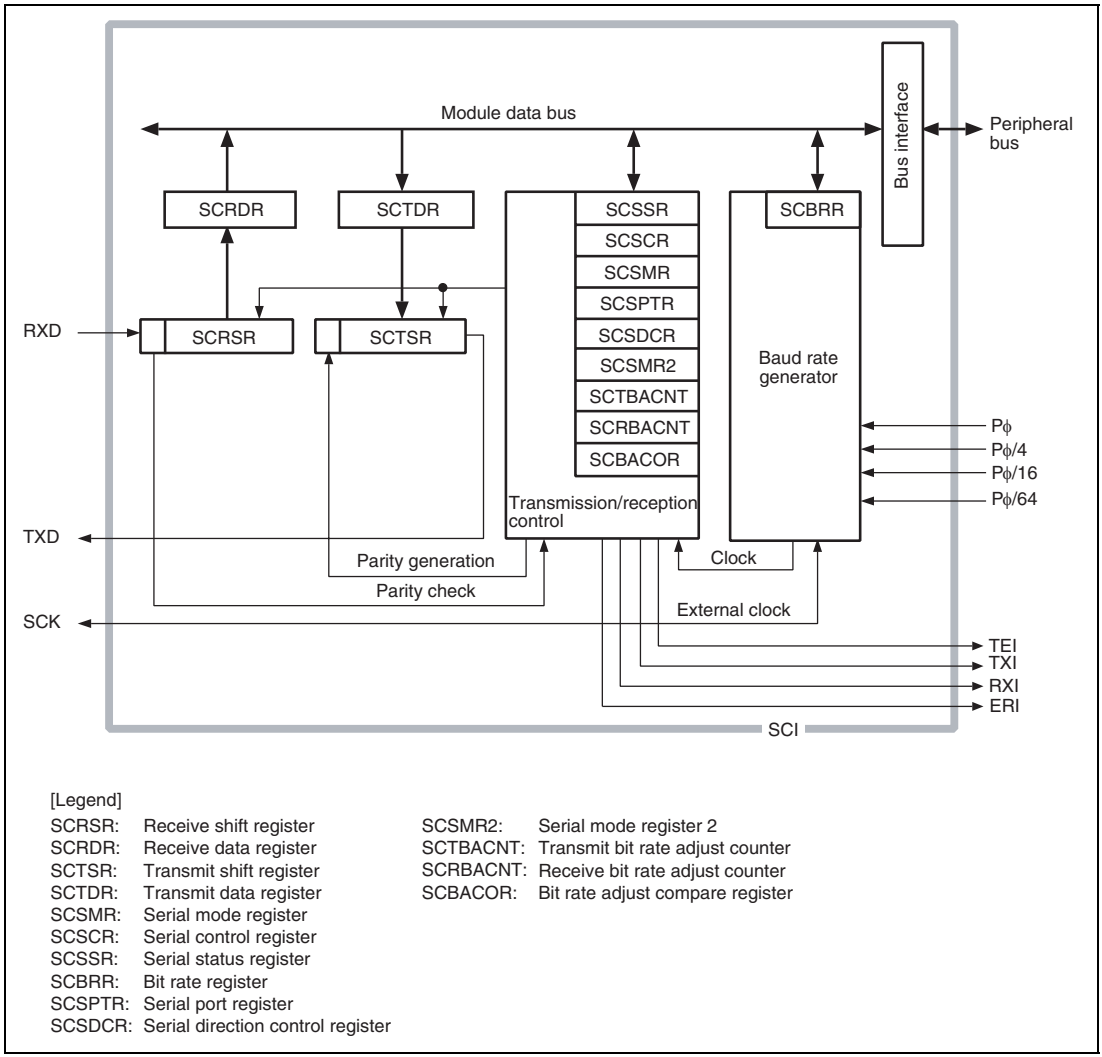
This LSI has four channels of independent serial communication interface (SCI). The SCI can handle both asynchronous and clock synchronous serial communication. In asynchronous serial communication mode, serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

18.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Asynchronous mode:
  - Serial data communication is performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are twelve selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Multiprocessor communications
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break is detected by reading the RXD pin level directly when a framing error occurs.
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal clock) or SCK pin (external clock)

- Choice of LSB-first or MSB-first data transfer (except for 7-bit data in asynchronous mode)
- Fine adjustment of bit rate (only in asynchronous mode)
- Four types of interrupts: There are four interrupt sources including transmit-data-empty, transmit end, receive-data-full, and receive error interrupts, and each interrupt can be requested independently. The direct memory access controller (DMAC) or data transfer controller (DTC) can be activated by the transmit-data-empty interrupt or receive-data-full interrupt to transfer data.
- Module standby mode can be set

Figure 18.1 shows a block diagram of the SCI.



[Legend]

SCRSR: Receive shift register
SCRDR: Receive data register
SCTSR: Transmit shift register
SCTDR: Transmit data register
SCSMR: Serial mode register
SCSCR: Serial control register
SCSSR: Serial status register
SCBRR: Bit rate register
SCSPTR: Serial port register
SCSDCR: Serial direction control register
SCSMR2: Serial mode register 2
SCTBACNT: Transmit bit rate adjust counter
SCRBACNT: Receive bit rate adjust counter
SCBACOR: Bit rate adjust compare register

Figure 18.1 Block Diagram of SCI

18.2 Input/Output Pins

The SCI has the serial pins summarized in table 18.1.

Table 18.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	SCI0 clock input/output
	RXD0	Input	SCI0 receive data input
	TXD0	Output	SCI0 transmit data output
1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
2	SCK2	I/O	SCI2 clock input/output
	RXD2	Input	SCI2 receive data input
	TXD2	Output	SCI2 transmit data output
3	SCK3	I/O	SCI3 clock input/output
	RXD3	Input	SCI3 receive data input
	TXD3	Output	SCI3 transmit data output

Note: * Pin names SCK, RXD, and TXD are used in the description for all channels, omitting the channel designation.

18.3 Register Descriptions

The SCI has the following registers for each channel. For details on register addresses and register states during each processing, refer to section 34, List of Registers.

Table 18.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'00	H'FFFF8000	8
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'FFFF8002	8
	Serial control register_0	SCSCR_0	R/W	H'00	H'FFFF8004	8
	Transmit data register_0	SCTDR_0	W	H'xx	H'FFFF8006	8
	Serial status register_0	SCSSR_0	R/W	H'84	H'FFFF8008	8
	Receive data register_0	SCRDR_0	R	H'xx	H'FFFF800A	8
	Serial direction control register_0	SCSDCR_0	R/W	H'F2	H'FFFF800C	8
	Serial port register_0	SCSPTR_0	R/W	H'0x	H'FFFF800E	8
	Serial mode register 2_0	SCSMR2_0	R/W	H'00	H'FFFF8010	8
	Transmit bit rate adjust counter_0	SCTBACNT_0	R	H'00	H'FFFF8018	8
	Receive bit rate adjust counter_0	SCRBACNT_0	R	H'00	H'FFFF801A	8
	Bit rate adjust compare register_0	SCBACOR_0	R/W	H'00	H'FFFF8014	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
1	Serial mode register_1	SCSMR_1	R/W	H'00	H'FFFF8800	8
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFF8802	8
	Serial control register_1	SCSCR_1	R/W	H'00	H'FFFF8804	8
	Transmit data register_1	SCTDR_1	W	H'xx	H'FFFF8806	8
	Serial status register_1	SCSSR_1	R/W	H'84	H'FFFF8808	8
	Receive data register_1	SCRDR_1	R	H'xx	H'FFFF880A	8
	Serial direction control register_1	SCSDCR_1	R/W	H'F2	H'FFFF880C	8
	Serial port register_1	SCSPTR_1	R/W	H'0x	H'FFFF880E	8
	Serial mode register 2_1	SCSMR2_1	R/W	H'00	H'FFFF8810	8
	Transmit bit rate adjust counter_1	SCTBACNT_1	R	H'00	H'FFFF8818	8
	Receive bit rate adjust counter_1	SCRBACNT_1	R	H'00	H'FFFF881A	8
	Bit rate adjust compare register_1	SCBACOR_1	R/W	H'00	H'FFFF8814	8
2	Serial mode register_2	SCSMR_2	R/W	H'00	H'FFFF9000	8
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'FFFF9002	8
	Serial control register_2	SCSCR_2	R/W	H'00	H'FFFF9004	8
	Transmit data register_2	SCTDR_2	W	H'xx	H'FFFF9006	8
	Serial status register_2	SCSSR_2	R/W	H'84	H'FFFF9008	8
	Receive data register_2	SCRDR_2	R	H'xx	H'FFFF900A	8
	Serial direction control register_2	SCSDCR_2	R/W	H'F2	H'FFFF900C	8
	Serial port register_2	SCSPTR_2	R/W	H'0x	H'FFFF900E	8
	Serial mode register 2_2	SCSMR2_2	R/W	H'00	H'FFFF9010	8
	Transmit bit rate adjust counter_2	SCTBACNT_2	R	H'00	H'FFFF9018	8
	Receive bit rate adjust counter_2	SCRBACNT_2	R	H'00	H'FFFF901A	8
	Bit rate adjust compare register_2	SCBACOR_2	R/W	H'00	H'FFFF9014	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
3	Serial mode register_3	SCSMR_3	R/W	H'00	H'FFFF9800	8
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'FFFF9802	8
	Serial control register_3	SCSCR_3	R/W	H'00	H'FFFF9804	8
	Transmit data register_3	SCTDR_3	W	H'xx	H'FFFF9806	8
	Serial status register_3	SCSSR_3	R/W	H'84	H'FFFF9808	8
	Receive data register_3	SCRDR_3	R	H'xx	H'FFFF980A	8
	Serial direction control register_3	SCSDCR_3	R/W	H'F2	H'FFFF980C	8
	Serial port register_3	SCSPTR_3	R/W	H'0x	H'FFFF980E	8
	Serial mode register 2_3	SCSMR2_3	R/W	H'00	H'FFFF9810	8
	Transmit bit rate adjust counter_3	SCTBACNT_3	R	H'00	H'FFFF9818	8
	Receive bit rate adjust counter_3	SCRBACNT_3	R	H'00	H'FFFF981A	8
	Bit rate adjust compare register_3	SCBACOR_3	R/W	H'00	H'FFFF9814	8

18.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. The SCI converts the data input at the RXD pin and loaded into SCRSR to parallel form. When one byte has been received, it is automatically transferred to SCRDR. The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

18.3.2 Receive Data Register (SCRDR)

SCRDR is a register that stores serial receive data. After receiving one byte of serial data, the SCI transfers the received data from the receive shift register (SCRSR) into SCRDR for storage and completes operation. After that, SCRSR is ready to receive data.

Since SCRSR and SCRDR work as a double buffer in this way, data can be received continuously.

SCRDR is a read-only register and can be read but cannot be written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

18.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR) into SCTSR, then transmits the data serially from the TXD pin. After transmitting one data byte, the SCI automatically loads the next transmit data from SCTDR into SCTSR and starts transmitting again. If the TDRE flag in the serial status register (SCSSR) is set to 1, the SCI does not transfer data from SCTDR to SCTSR. The CPU cannot read or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

18.3.4 Transmit Data Register (SCTDR)

SCTDR is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCTDR into SCTSR and starts serial transmission. If the next transmit data has been written to SCTDR during serial transmission from SCTSR, the SCI can transmit data continuously. SCTDR can always be written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

18.3.5 Serial Mode Register (SCSMR)

SCSMR is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR.

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCI operates in asynchronous or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data in asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting. When 7-bit data is selected, the MSB (bit 7) of the transmit data register (SCTDR) is not transmitted. 0: 8-bit data 1: 7-bit data

Bit	Bit Name	Initial value	R/W	Description
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity 1: Odd parity If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added. 0: One stop bit*1 1: Two stop bits*2 When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. Notes: 1. When transmitting, a single 1 bit is added at the end of each transmitted character. 2. When transmitting, two 1 bits are added at the end of each transmitted character.
2	MP	0	R/W	Multiprocessor Mode (only in asynchronous mode) Enables or disables multiprocessor mode. The PE and O/$\overline{E}$ bit settings are ignored in multiprocessor mode. 0: Multiprocessor mode disabled 1: Multiprocessor mode enabled
1, 0	CKS[1:0]	00	R/W	Clock Select 1 and 0 Select the internal clock source of the on-chip baud rate generator. Four clock sources are available; Pϕ, Pϕ/4, Pϕ/16, and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 18.3.10, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

18.3.6 Serial Control Register (SCSCR)

SCSCR is an 8-bit register that enables or disables SCI transmission/reception, serial clock output in asynchronous mode, and interrupt requests and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables a transmit-data-empty interrupt (TXI) to be issued when the TDRE flag in the serial status register (SCSSR) is set to 1 after serial transmit data is sent from the transmit data register (SCTDR) to the transmit shift register (SCTSR). TXI can be canceled by clearing the TDRE flag to 0 after reading TDRE = 1 or by clearing the TIE bit to 0. 0: Transmit-data-empty interrupt request (TXI) is disabled 1: Transmit-data-empty interrupt request (TXI) is enabled
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables a receive-data-full interrupt (RXI) and a receive error interrupt (ERI) to be issued when the RDRF flag in SCSSR is set to 1 after the serial data received is transferred from the receive shift register (SCRSR) to the receive data register (SCRDR). RXI can be canceled by clearing the RDRF flag after reading RDRF =1. ERI can be canceled by clearing the FER, PER, or ORER flag to 0 after reading 1 from the flag. Both RXI and ERI can also be canceled by clearing the RIE bit to 0. 0: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled 1: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled

Bit	Bit Name	Initial value	R/W	Description
5	TE	0	R/W	Transmit Enable Enables or disables the SCI serial transmitter. 0: Transmitter disabled*1 1: Transmitter enabled*2 Notes: 1. The TDRE flag in SCSSR is fixed at 1. 2. Serial transmission starts after writing transmit data into SCTDR and clearing the TDRE flag in SCSSR to 0 while the transmitter is enabled. Select the transmit format in the serial mode register (SCSMR) before setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables the SCI serial receiver. 0: Receiver disabled*1 1: Receiver enabled*2 Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, and ORER). These flags retain their previous values. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in clock synchronous mode. Select the receive format in SCSMR before setting RE to 1.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (only when MP = 1 in SCSMR in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped and setting of the RDRF, FER, and ORER status flags in SCSSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared to 0 and normal receiving operation is resumed. For details, refer to section 18.4.4, Multiprocessor Communication Function.

Bit	Bit Name	Initial value	R/W	Description
2	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables a transmit end interrupt (TEI) to be issued when no valid transmit data is found in SCTDR during MSB data transmission. TEI can be canceled by clearing the TEND flag to 0 (by clearing the TDRE flag in SCSSR to 0 after reading TDRE = 1) or by clearing the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled 1: Transmit end interrupt request (TEI) is enabled
1, 0	CKE[1:0]	00	R/W	Clock Enable 1 and 0 Select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input. When selecting the clock output in clock synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1 and then set bits CKE1 and CKE0. For details on clock source selection, refer to table 18.11. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (The input signal is ignored.) 01: Internal clock, SCK pin used for clock output*[1] 10: External clock, SCK pin used for clock input*[2] 11: External clock, SCK pin used for clock input*[2] • Clock synchronous mode 00: Internal clock, SCK pin used for synchronous clock output 01: Internal clock, SCK pin used for synchronous clock output 10: External clock, SCK pin used for synchronous clock input 11: External clock, SCK pin used for synchronous clock input Notes: 1. The output clock frequency is 16 times the bit rate. 2. The input clock frequency is 16 times the bit rate.

18.3.7 Serial Status Register (SCSSR)

SCSSR is an 8-bit register that contains status flags to indicate the SCI operating state.

The CPU can always read and write to SCSSR, but cannot write 1 to status flags TDRE, RDRF, ORER, PER, and FER. These flags can be cleared to 0 only after 1 is read from the flags. The TEND flag is a read-only bit and cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether data has been transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR) and SCTDR has become ready to be written with next serial transmit data. 0: Indicates that SCTDR holds valid transmit data [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC is activated by a TXI interrupt and transmit data is transferred to SCTDR. • When the DTC is activated by a TXI interrupt and transmit data is transferred to SCTDR while the DISEL bit in MRB of the DTC is 0 (except when the DTC transfer counter value has become H'0000). 1: Indicates that SCTDR does not hold valid transmit data [Setting conditions] • By a power-on reset or in module standby mode • When the TE bit in SCSCR is 0 • When data is transferred from SCTDR to SCTSR and data can be written to SCTDR

Bit	Bit Name	Initial value	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in the receive data register (SCRDR). 0: Indicates that valid received data is not stored in SCRDR [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC is activated by an RXI interrupt and data is transferred from SCRDR. • When the DTC is activated by an RXI interrupt and data is transferred from SCRDR while the DISEL bit in MRB of the DTC is 0 (except when the DTC transfer counter value has become H'0000). 1: Indicates that valid received data is stored in SCRDR [Setting condition] • When serial reception ends normally and receive data is transferred from SCRSR to SCRDR Note: SCRDR and the RDRF flag are not affected and retain their previous states even if an error is detected during data reception or if the RE bit in the serial control register (SCSCR) is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the received data will be lost.

Bit	Bit Name	Initial value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*[1] [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to ORER after reading ORER = 1 1: Indicates that an overrun error occurred during reception*[2] [Setting condition] • When the next serial reception is completed while RDRF = 1 Notes: 1. The ORER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0. 2. The receive data prior to the overrun error is retained in SCRDR, and the data received subsequently is lost. Subsequent serial reception cannot be continued while the ORER flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
4	FER	0	R/(W)*	Framing Error Indicates that a framing error occurred during data reception in asynchronous mode, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*1 [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to FER after reading FER = 1 1: Indicates that a framing error occurred during reception [Setting condition] • When the SCI founds that the stop bit at the end of the received data is 0 after completing reception*2 Notes: 1. The FER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0. 2. In 2-stop-bit mode, only the first stop bit is checked for a value to 1; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to SCRDR but the RDRF flag is not set. Subsequent serial reception cannot be continued while the FER flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred during data reception in asynchronous mode, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*1 [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to PER after reading PER = 1 1: Indicates that a parity error occurred during reception*2 [Setting condition] • When the number of 1s in the received data and parity does not match the even or odd parity specified by the O/$\overline{E}$ bit in the serial mode register (SCSMR). Notes: 1. The PER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0. 2. If a parity error occurs, the receive data is transferred to SCRDR but the RDRF flag is not set. Subsequent serial reception cannot be continued while the PER flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
2	TEND	1	R	Transmit End Indicates that no valid data was in SCTDR during transmission of the last bit of the transmit character and transmission has ended. The TEND flag is read-only and cannot be modified. 0: Indicates that transmission is in progress [Clearing condition] • When 0 is written to TDRE after reading TDRE = 1 1: Indicates that transmission has ended [Setting conditions] • By a power-on reset or in module standby mode • When the TE bit in SCSCR is 0 • When TDRE = 1 during transmission of the last bit of a 1-byte serial transmit character Note: The TEND flag value becomes undefined if data is written to SCTDR by activating the DMAC or DTC by a TXI interrupt. In this case, do not use the TEND flag as the transmit end flag.
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit found in the receive data. When the RE bit in SCSCR is cleared to 0, its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Specifies the multiprocessor bit value to be added to the transmit frame.

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

18.3.8 Serial Port Register (SCSPTR)

SCSPTR is an 8-bit register that controls input/output and data for the ports multiplexed with the SCI function pins. Data to be output through the TXD pin can be specified to control break of serial transfer. Through bits 3 and 2, data reading and writing through the SCK pin can be specified. Bit 7 enables or disables RXI interrupts. The CPU can always read and write to SCSPTR. When reading the value on the SCI pins, use the respective port register. For details, refer to section 23, I/O Ports.

Bit:	7	6	5	4	3	2	1	0
	EIO	-	-	-	SPB1IO	SPB1DT	-	SPB0DT
Initial value:	0	0	0	0	0	Undefined	0	1
R/W:	R/W	R	R	R	R/W	R/W	R	W

Bit	Bit Name	Initial value	R/W	Description
7	EIO	0	R/W	Error Interrupt Only Enables or disables RXI interrupts. While the EIO bit is set to 1, the SCI does not request an RXI interrupt to the CPU even if the RIE bit is set to 1. 0: While the RIE bit is 1, RXI and ERI interrupts are sent to the INTC. 1: While the RIE bit is 1, only the ERI interrupt is sent to the INTC.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SPB1IO	0	R/W	Clock Port Input/Output in Serial Port Specifies the input/output direction of the SCK pin in the serial port. To output the data specified in the SPB1DT bit through the SCK pin as a port output pin, set the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0. 0: Does not output the SPB1DT bit value through the SCK pin. 1: Outputs the SPB1DT bit value through the SCK pin.

Bit	Bit Name	Initial value	R/W	Description
2	SPB1DT	Undefined	R/W	Clock Port Data in Serial Port Specifies the data output through the SCK pin in the serial port. Output should be enabled by the SPB1IO bit (for details, refer to the SPB1IO bit description). When output is enabled, the SPB1DT bit value is output through the SCK pin. 0: Low level is output 1: High level is output
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	SPB0DT	1	W	Serial Port Break Data Controls the TXD pin by the TE bit in SCSCR. However, TXD pin function should be selected by the pin function controller (PFC). This is a read-only bit. The read value is undefined.

TE bit setting in SCSCR	SPB0DT bit setting	TXD pin state
0	0	Low output
0	1	High output (initial state)
1	*	Transmit data output in accord with serial core logic.

Note: * Don't care

18.3.9 Serial Direction Control Register (SCSDCR)

The DIR bit in the serial direction control register (SCSDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	DIR	-	-	-
Initial value:	1	1	1	1	0	0	1	0
R/W:	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	DIR	0	R/W	Data Transfer Direction Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format. 0: SCTDR contents are transmitted in LSB-first order Receive data is stored in SCRDR in LSB-first 1: SCTDR contents are transmitted in MSB-first order Receive data is stored in SCRDR in MSB-first
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

18.3.10 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR.

The SCBRR setting is calculated as follows:

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 18.3.)

Table 18.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Tables 18.4 shows an example of SCBRR settings in asynchronous mode, and table 18.5 shows an example of SCBRR settings in clock synchronous mode.

Table 18.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1)

	Pϕ (MHz)																	
Bit Rate (bits/s)	10			12			14			16			18			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	-0.25	2	212	0.03	2	248	-0.17	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	32	-1.36	0	38	0.16	0	45	-0.93	0	51	0.16	0	58	-0.69	0	64	0.16
14400	0	21	-1.36	0	25	0.16	0	29	1.27	0	34	-0.79	0	38	0.16	0	42	0.94
19200	0	15	1.73	0	19	-2.34	0	22	-0.93	0	25	0.16	0	28	1.02	0	32	-1.36
28800	0	10	-1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	-2.34	0	21	-1.36
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	7	1.73	0	9	-2.34	0	10	3.57	0	12	0.16	0	14	-2.34	0	15	1.73
115200	0	2	-9.58	0	2	8.51	0	3	-5.06	0	3	8.51	0	4	-2.34	0	4	8.51
500000	0	0*	-37.5	0	0*	-25.0	0	0*	-12.5	0	0*	0.00	0	0*	12.5	0	0*	25.0

Table 18.4 Bit Rates and SCBRR Settings in Asynchronous Mode (2)

Bit Rate (bits/s)	Pϕ (MHz)																	
	22			24			26			28			30			32		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	97	-0.35	3	106	-0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	-0.54	3	77	0.16	3	84	-0.43	3	90	0.16	3	97	-0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	-0.54	2	77	0.16	2	84	-0.43	2	90	0.16	2	97	-0.35	2	103	0.16
1200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2400	1	71	-0.54	1	77	0.16	1	84	-0.43	1	90	0.16	1	97	-0.35	1	103	0.16
4800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9600	0	71	-0.54	0	77	0.16	0	84	-0.43	0	90	0.16	0	97	-0.35	0	103	0.16
14400	0	47	-0.54	0	51	0.16	0	55	0.76	0	60	-0.39	0	64	0.16	0	68	0.64
19200	0	35	-0.54	0	38	0.16	0	41	0.76	0	45	-0.93	0	48	-0.35	0	51	0.16
28800	0	23	-0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	-1.36	0	34	-0.79
31250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38400	0	17	-0.54	0	19	-2.34	0	20	0.76	0	22	-0.93	0	23	1.73	0	25	0.16
115200	0	5	-0.54	0	6	-6.99	0	6	0.76	0	7	-5.06	0	7	1.73	0	8	-3.55
500000	0	0*	37.5	0	1	-25.0	0	1	-18.8	0	1	-12.5	0	1	-6.25	0	1	0.00

Table 18.4 Bit Rates and SCBRR Settings in Asynchronous Mode (3)

Bit Rate (bits/s)	Pϕ (MHz) 34 n	N	Error (%)	36 n	N	Error (%)	38 n	N	Error (%)	40 n	N	Error (%)	42 n	N	Error (%)
110	3	150	-0.05	3	159	-0.12	3	168	-0.19	3	177	-0.25	3	185	0.23
150	3	110	-0.29	3	116	0.16	3	123	-0.24	3	129	0.16	3	136	-0.21
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16	3	67	0.53
600	2	110	-0.29	2	116	0.16	2	123	-0.24	2	129	0.16	2	136	-0.21
1200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16	2	67	0.53
2400	1	110	-0.29	1	116	0.16	1	123	-0.24	1	129	0.16	1	136	-0.21
4800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16	1	67	0.53
9600	0	110	-0.29	0	116	0.16	0	123	-0.24	0	129	0.16	0	136	-0.21
14400	0	73	-0.29	0	77	0.16	0	81	0.57	0	86	-0.22	0	90	0.16
19200	0	54	0.62	0	58	-0.69	0	61	-0.24	0	64	0.16	0	67	0.53
28800	0	36	-0.29	0	38	0.16	0	40	0.57	0	42	0.94	0	45	-0.93
31250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00	0	41	0.00
38400	0	27	-1.18	0	28	1.02	0	30	-0.24	0	32	-1.36	0	33	0.53
115200	0	8	2.48	0	9	-2.34	0	9	3.08	0	10	-1.36	0	10	3.57
500000	0	1	6.25	0	1	12.5	0	1	18.8	0	2	-16.7	0	2	-12.5

Table 18.4 Bit Rates and SCBRR Settings in Asynchronous Mode (4)

Bit Rate (bits/s)	Pϕ (MHz) 44			46			48			50		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	194	0.16	3	203	0.09	3	212	0.03	3	221	-0.02
150	3	142	0.16	3	149	-0.17	3	155	0.16	3	162	-0.15
300	3	71	-0.54	3	74	-0.17	3	77	0.16	3	80	0.47
600	2	142	0.16	2	149	-0.17	2	155	0.16	2	162	-0.15
1200	2	71	-0.54	2	74	-0.17	2	77	0.16	2	80	0.47
2400	1	142	0.16	1	149	-0.17	1	155	0.16	1	162	-0.15
4800	1	71	-0.54	1	74	-0.17	1	77	0.16	1	80	0.47
9600	0	142	0.16	0	149	-0.17	0	155	0.16	0	162	-0.15
14400	0	94	0.51	0	99	-0.17	0	103	0.16	0	108	-0.45
19200	0	71	-0.54	0	74	-0.17	0	77	0.16	0	80	0.47
28800	0	47	-0.54	0	49	-0.17	0	51	0.16	0	53	0.47
31250	0	43	0.00	0	45	0.00	0	47	0.00	0	49	0
38400	0	35	-0.54	0	36	1.18	0	38	0.16	0	40	-0.76
115200	0	11	-0.54	0	11	3.99	0	12	0.16	0	13	-3.12
500000	0	2	-8.33	0	2	-4.17	0	2	0.00	0	2	4.17

Table 18.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (1)

Bit Rate (bits/s)	Pϕ (MHz)											
	10		12		14		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1000	2	155	2	187	2	218	2	249	3	69	3	77
2500	1	249	2	74	2	87	2	99	2	112	2	124
5000	1	124	1	149	1	174	1	199	1	224	1	249
10000	0	249	1	74	1	87	1	99	1	112	1	124
25000	0	99	0	119	0	139	0	159	0	179	0	199
50000	0	49	0	59	0	69	0	79	0	89	0	99
100000	0	24	0	29	0	34	0	39	0	44	0	49
250000	0	9	0	11	0	13	0	15	0	17	0	19
500000	0	4	0	5	0	6	0	7	0	8	0	9
1000000	—	—	0	2	—	—	0	3	—	—	0	4
2500000	0	0*	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*

Table 18.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (2)

Bit Rate (bits/s)	Pϕ (MHz)											
	22		24		26		28		30		32	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1000	3	85	3	93	3	101	3	108	3	116	3	124
2500	2	137	2	149	2	162	2	174	2	187	2	199
5000	2	68	2	74	2	80	2	87	2	93	2	99
10000	1	137	1	149	1	162	1	174	1	187	1	199
25000	0	219	0	239	1	64	1	69	1	74	1	79
50000	0	109	0	119	0	129	0	139	0	149	0	159
100000	0	54	0	59	0	64	0	69	0	74	0	79
250000	0	21	0	23	0	25	0	27	0	29	0	31
500000	0	10	0	11	0	12	0	13	0	14	0	15
1000000	—	—	0	5	—	—	0	6	—	—	0	7
2500000	—	—	—	—	—	—	—	—	0	2	—	—
5000000	—	—	—	—	—	—	—	—	—	—	—	—

Table 18.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (3)

Bit Rate (bits/s)	Pϕ (MHz)									
	34		36		38		40		42	
	n	N	n	N	n	N	n	N	n	N
250										
500										
1000	3	132	3	140	3	147	3	155	3	163
2500	2	212	2	224	2	237	2	249	3	65
5000	2	105	2	112	2	118	2	124	2	130
10000	1	212	1	224	1	237	1	249	2	65
25000	1	84	1	89	1	94	1	99	1	104
50000	0	169	0	179	0	189	0	199	0	209
100000	0	84	0	89	0	94	0	99	0	104
250000	0	33	0	35	0	37	0	39	0	41
500000	0	16	0	17	0	18	0	19	0	20
1000000	—	—	0	8	—	—	0	9	—	—
2500000	—	—	—	—	—	—	0	3	—	—
5000000	—	—	—	—	—	—	0	1	—	—

Table 18.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (4)

Bit Rate (bits/s)	Pϕ (MHz)							
	44		46		48		50	
	n	N	n	N	n	N	n	N
250								
500								
1000	3	171	3	179	3	187	3	194
2500	3	68	3	71	3	74	3	77
5000	2	137	2	143	2	149	2	155
10000	2	68	2	71	2	74	2	77
25000	1	109	1	114	1	119	1	124
50000	0	219	0	229	0	239	0	249
100000	0	109	0	114	0	119	0	124
250000	0	43	0	45	0	47	0	49
500000	0	21	0	22	0	23	0	24
1000000	0	10	—	—	0	11	—	—
2500000	—	—	—	—	—	—	0	4
5000000	—	—	—	—	—	—	—	—

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmission/reception is disabled.

Note: Settings with an error of 1% or less are recommended.

Table 18.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 18.7 indicates the maximum bit rates in clock synchronous mode when the baud rate generator is used. Tables 18.8 and 18.9 list the maximum rates for external clock input.

Table 18.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	Discontinuous Transmission/Reception			Continuous Transmission/Reception		
	Maximum Bit Rate	Settings		Maximum Bit Rate	Settings	
Pϕ (MHz)	(bits/s)	n	N	(bits/s)	n	N
10	312500	0	0	156250	0	1
12	375000	0	0	187500	0	1
14	437500	0	0	218750	0	1
16	500000	0	0	250000	0	1
18	562500	0	0	281250	0	1
20	625000	0	0	312500	0	1
22	687500	0	0	343750	0	1
24	750000	0	0	375000	0	1
26	812500	0	0	406250	0	1
28	875000	0	0	437500	0	1
30	937500	0	0	468750	0	1
32	1000000	0	0	500000	0	1
34	1062500	0	0	531250	0	1
36	1125000	0	0	562500	0	1
38	1187500	0	0	593750	0	1
40	1250000	0	0	625000	0	1
42	1312500	0	0	656250	0	1
44	1375000	0	0	687500	0	1
46	1437500	0	0	718750	0	1
48	1500000	0	0	750000	0	1
50	1562500	0	0	781250	0	1

Table 18.7 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Clock Synchronous Mode)

	Discontinuous Transmission/Reception			Continuous Transmission/Reception		
		Settings			Settings	
Pϕ (MHz)	Maximum Bit Rate (bits/s)	n	N	Maximum Bit Rate (bits/s)	n	N
10	2500000	0	0	1250000	0	1
12	3000000	0	0	1500000	0	1
14	3500000	0	0	1750000	0	1
16	4000000	0	0	2000000	0	1
18	4500000	0	0	2250000	0	1
20	5000000	0	0	2500000	0	1
22	5500000	0	0	2750000	0	1
24	6000000	0	0	3000000	0	1
26	6500000	0	0	3250000	0	1
28	7000000	0	0	3500000	0	1
30	7500000	0	0	3750000	0	1
32	8000000	0	0	4000000	0	1
34	8500000	0	0	4250000	0	1
36	9000000	0	0	4500000	0	1
38	9500000	0	0	4750000	0	1
40	10000000	0	0	5000000	0	1
42	10500000	0	0	5250000	0	1
44	11000000	0	0	5500000	0	1
46	11500000	0	0	5750000	0	1
48	12000000	0	0	6000000	0	1
50	12500000	0	0	6250000	0	1

Table 18.8 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10	2.5	156250
12	3.0	187500
14	3.5	218750
16	4.0	250000
18	4.5	281250
20	5.0	312500
22	5.5	343750
24	6.0	375000
26	6.5	406250
28	7.0	437500
30	7.5	468750
32	8.0	500000
34	8.5	531250
36	9.0	562500
38	9.5	593750
40	10.0	625000
42	10.5	656250
44	11.0	687500
46	11.5	718750
48	12.0	750000
50	12.5	781250

Table 18.9 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10	1.6667	1666666
12	2.0000	2000000
14	2.3333	2333333
16	2.6667	2666666
18	3.0000	3000000
20	3.3333	3333333
22	3.6667	3666666
24	4.0000	4000000
26	4.3333	4333333
28	4.6667	4666666
30	5.0000	5000000
32	5.3333	5333333
34	5.6667	5666666
36	6.0000	6000000
38	6.3333	6333333
40	6.6667	6666666
42	7.0000	7000000
44	7.3333	7333333
46	7.6667	7666666
48	8.0000	8000000
50	8.3333	8333333

18.3.11 Serial Mode Register 2 (SCSMR2)

SCSMR2 is an 8-bit register that adjusts the bit rate in asynchronous mode and specifies the sampling point.

The CPU can always read and write to SCSMR2.

Bit:	7	6	5	4	3	2	1	0
	-	BAE	SPSEL[1:0]		-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	BAE	0	R/W	Bit Rate Adjustment Enable Enables or disables bit rate adjustment. Setting this bit is valid only in asynchronous mode. Setting the BAE bit to 1 enables the SPSEL[1:0] bits in SCSMR2, transmit bit rate adjust counter (SCTBCNT), receive bit rate adjust counter (SCRBACNT), and bit rate adjust compare register (SCBACOR). 0: Disables bit rate adjustment. 1: Enables bit rate adjustment.
5, 4	SPSEL[1:0]	00	R/W	Sampling Point Select 1, 0 Specifies the data sampling point to be received. Setting this bit is valid only in asynchronous mode. 00: Samples data at the rising edge on the eighth clock pulse. 01: Samples data at the rising edge on the ninth clock pulse. 10: Samples data at the rising edge on the tenth clock pulse. 11: Samples data at the rising edge on the eleventh clock pulse.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.12 Transmit Bit Rate Adjust Counter (SCTBACNT)

SCTBACNT is an 8-bit upcounter that is incremented based on the Pϕ clock. The initial value of SCTBACNT is H'00.

SCTBACNT is initialized when a compare match occurs between the transmit bit rate adjust counter (SCTBACNT) and SCBACOR during data transmission or when the TE bit in the serial control register (SCSCR) is cleared to 0. For details on counter settings and bit rates, refer to section 18.5, Bit Rate Adjustment.

Bit:	7	6	5	4	3	2	1	0
	TBACNT[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

18.3.13 Receive Bit Rate Adjust Counter (SCRBACNT)

SCRBACNT is an 8-bit upcounter that is incremented based on the Pϕ clock. The initial value of SCRBACNT is H'00.

SCRBACNT is initialized when a compare match occurs between the receive bit rate adjust counter (SCRBACNT) and SCBACOR during data reception or when the RE bit in the serial control register (SCSCR) is cleared to 0. For details on counter settings and bit rates, refer to section 18.5, Bit Rate Adjustment.

Bit:	7	6	5	4	3	2	1	0
	RBACNT[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

18.3.14 Bit Rate Adjust Compare Register (SCBACOR)

SCBACOR is an 8-bit register that specifies the bit rate adjustment period. The initial value of SCBACOR is H'00.

SCBACOR extends one frame period until a compare match occurs between SCBACOR and SCTBACNT (in transmission) or SCRBACNT (in reception). When SCBACOR is set to H'FF, the extension period is approximately 5 µs (Pϕ = 50 MHz). When the BAE bit in the serial mode register 2 (SCSMR2) is 0, any setting to SCBACOR is invalid.

Bit:	7	6	5	4	3	2	1	0
	BACOR[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.4 Operation

18.4.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

Asynchronous or clock synchronous mode is selected and the transmit format is specified in the serial mode register (SCSMR) as shown in table 18.10. The SCI clock source is selected by the combination of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in the serial control register (SCSCR) as shown in table 18.11.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, and overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the clock supplied by the on-chip baud rate generator and can output a clock with a frequency 16 times the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clock Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 18.10 SCSMR Settings and SCI Communication Formats

SCSMR Settings					SCI Communication Format		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clock synchronous	8-bit	Not set	None

[Legend]
x: Don't care

Table 18.11 SCSMR and SCSCR Settings and SCI Clock Source Selection

SCSMR	SCSCR Settings				
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCI does not use the SCK pin.
		1			Clock with a frequency 16 times the bit rate is output.
	1	0		External	Input a clock with frequency 16 times the bit rate.
		1			
1	0	0	Clock synchronous	Internal	Synchronous clock is output.
		1			
	1	0		External	Input the synchronous clock.
		1			

18.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 18.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first when LSB-first transfer is selected), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit. When the bit rate adjustment is used, the sampling point can be changed. For details on the bit rate adjustment, refer to section 18.5, Bit Rate Adjustment.

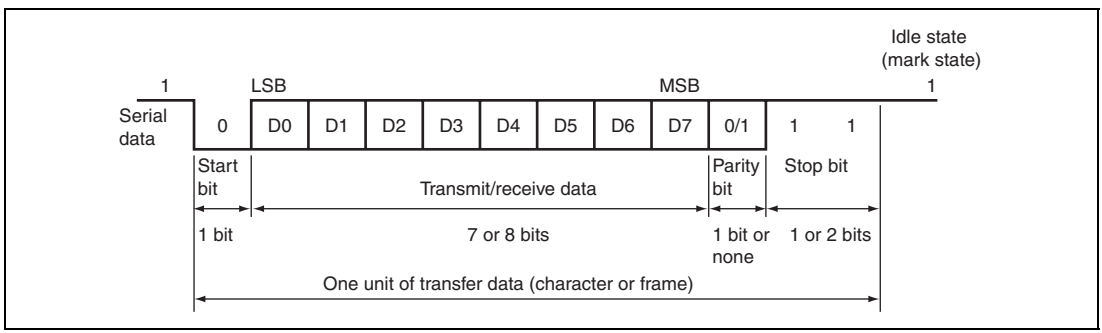


Figure 18.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits when LSB-First Transfer is Selected)

(1) Transmit/Receive Formats

Table 18.12 shows the transfer formats that can be selected in asynchronous mode. Any of 12 transfer formats can be selected according to the SCSMR settings.

Table 18.12 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings				Serial Transfer Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	x	1	0	S	8-bit data								MPB	STOP	
0	x	1	1	S	8-bit data								MPB	STOP	STOP
1	x	1	0	S	7-bit data							MPB	STOP		
1	x	1	1	S	7-bit data							MPB	STOP	STOP	

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit
x: Don't care

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR). For selection of the SCI clock source, see table 18.11.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to 16 times the desired bit rate.

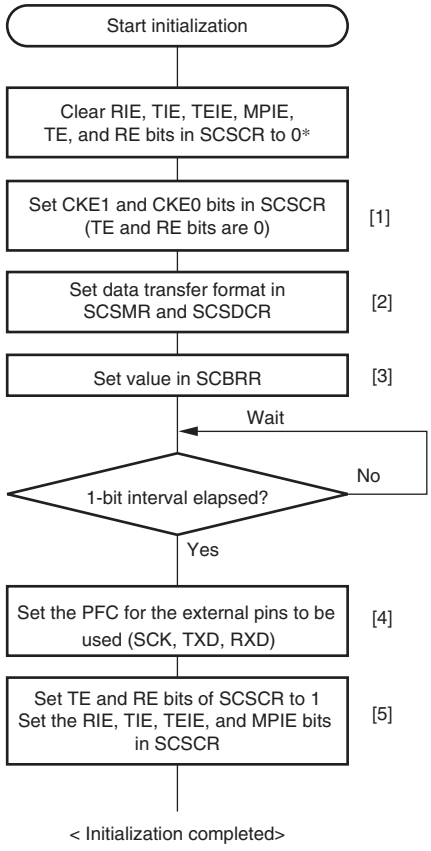
(3) Transmitting and Receiving Data

- SCI Initialization (Asynchronous Mode)

  Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

  When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR). Clearing the RE bit to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), which retain their previous contents.

  When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.



[1] Set the clock selection in SCSCR.

[2] Set the data transfer format in SCSMR and SCSDCR.

[3] Write a value corresponding to the bit rate to SCBRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RXD input during receiving and TXD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0. When CKE1 and CKE0 are 0 in asynchronous mode, setting the SCK pin is unnecessary. Outputting clocks from the SCK pin starts at synchronous clock output setting.

[5] Set the TE bit or RE bit in SCSCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting, and RXD pin is in an idle state for waiting the start bit during receiving.

Note : * In simultaneous transmit/receive operation, the TE and RE bits must be cleared to 0 or set to 1 simultaneously.

Figure 18.3 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Asynchronous Mode)

  Figure 18.4 shows a sample flowchart for serial transmission. Use the following procedure for serial data transmission after enabling the SCI for transmission.

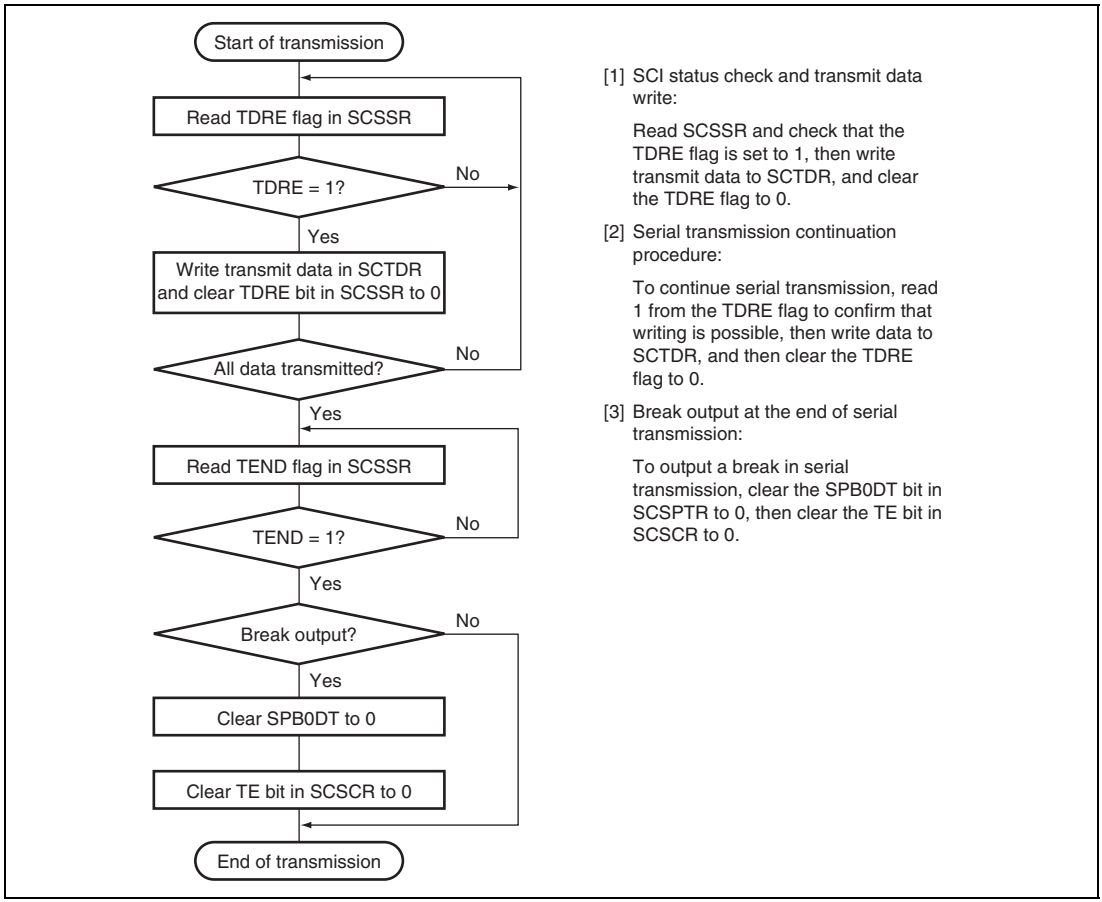


Figure 18.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in the serial status register (SCSSR). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR) and transfers the data from SCTDR to the transmit shift register (SCTSR).
2. After transferring data from SCTDR to SCTSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in the serial control register (SCSCR) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TXD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order (when LSB-first transfer is selected).
   C. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. (A format in which neither parity nor multiprocessor bit is output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCI checks the TDRE flag at the timing for sending the stop bit.
   If the TDRE flag is 0, the data is transferred from SCTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.
   If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE bit in SCSCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 18.5 shows an example of transmission in asynchronous mode.

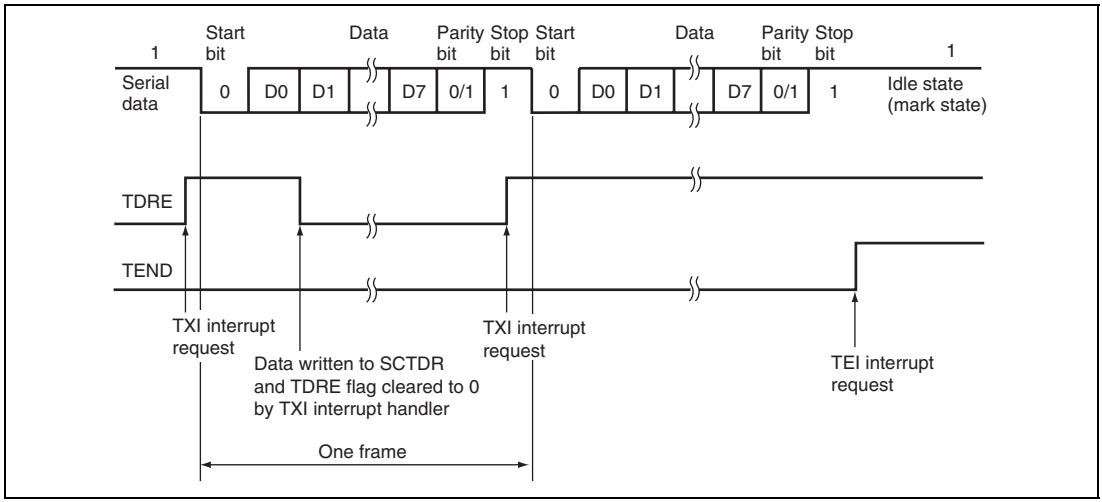
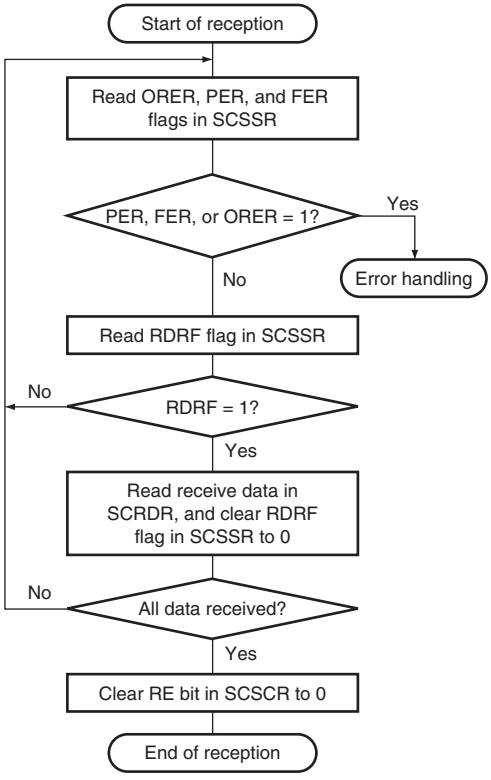


Figure 18.5 Example of Transmission in Asynchronous Mode (8-Bit Data with Parity and One Stop Bit when LSB-First Transfer is Selected)

- Receiving Serial Data (Asynchronous Mode)

Figure 18.6 shows a sample flowchart for serial reception. Use the following procedure for serial data reception after enabling the SCI for reception.



[1] Receive error handling and break detection:

If a receive error occurs, read the ORER, PER, and FER flags in SCSSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can also be detected by reading the value of the RXD pin.

[2] SCI status check and receive data read:

Read SCSSR and check that RDRF = 1, then read the receive data in SCRDR, and clear the RDRF flag to 0.

[3] Serial reception continuation procedure:

To continue serial reception, clear the RDRF flag to 0 before the stop bit for the current frame is received.

Figure 18.6 Sample Flowchart for Receiving Serial Data (1)

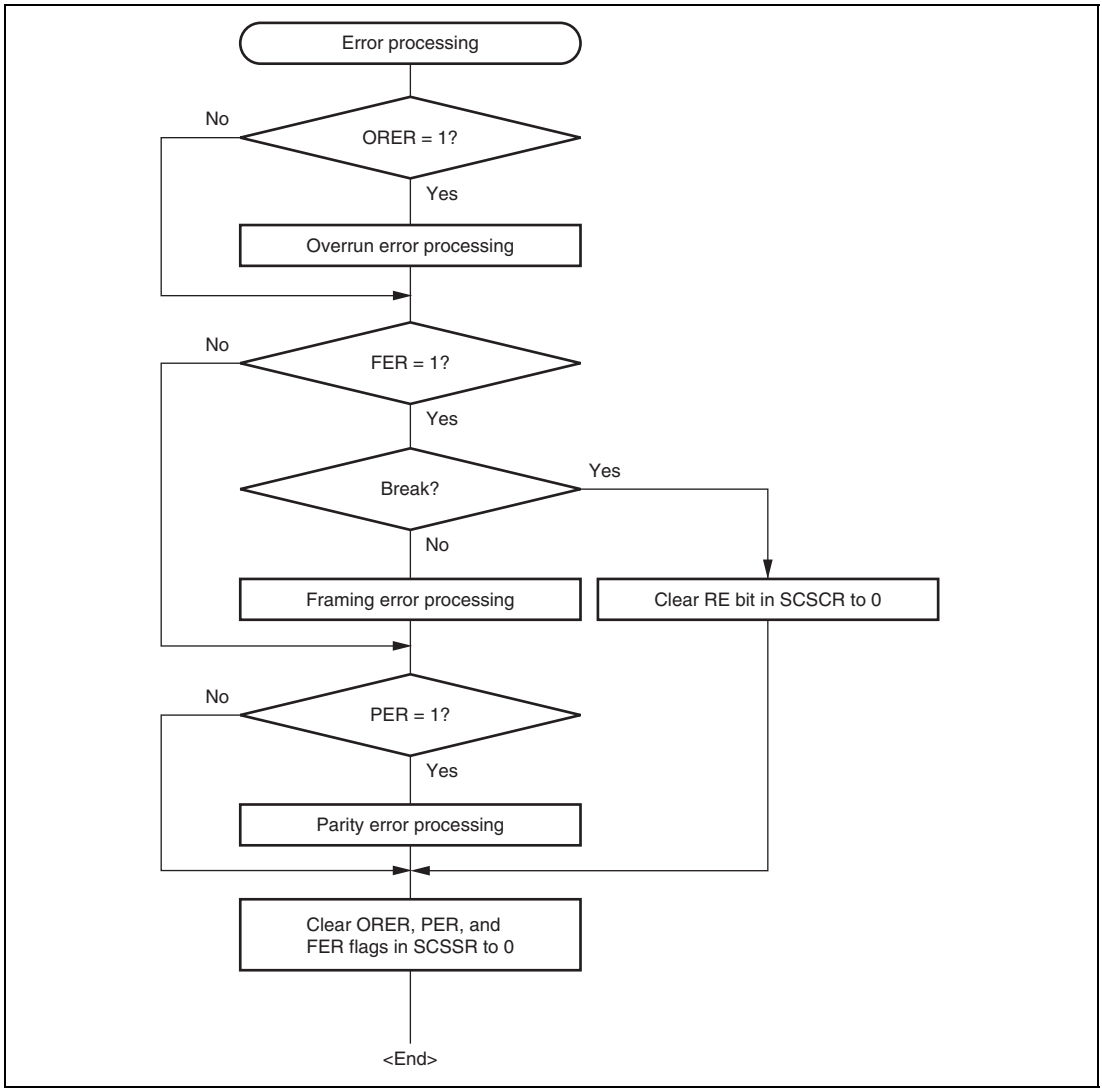


Figure 18.6 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order (when LSB-first transfer is selected).
3. The parity bit and stop bit are received.
   After receiving these bits, the SCI carries out the following checks.
   A. Parity check: The SCI counts the number of 1s in the received data and checks whether the count matches the even or odd parity specified by the O/$\overline{E}$ bit in the serial mode register (SCSMR).
   B. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   C. Status check: The SCI checks whether the RDRF flag is 0 and the received data can be transferred from the receive shift register (SCRSR) to SCRDR.

   If all the above checks are passed, the RDRF flag is set to 1 and the received data is stored in SCRDR. If a receive error is detected, the SCI operates as shown in table 18.13.

   Note: When a receive error occurs, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the error flag to 0.

4. If the EIO bit in SCSPTR is cleared to 0 and the RIE bit in SCSCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated. If the RIE bit in SCSCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 18.13 Receive Errors and Error Conditions

Receive Error	Abbreviation	Error Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SCSSR is set to 1	The received data is not transferred from SCRSR to SCRDR.
Framing error	FER	When the stop bit is 0	The received data is transferred from SCRSR to SCRDR.
Parity error	PER	When the received data does not match the even or odd parity specified in SCSMR	The received data is transferred from SCRSR to SCRDR.

Figure 18.7 shows an example of receive operation in asynchronous mode.

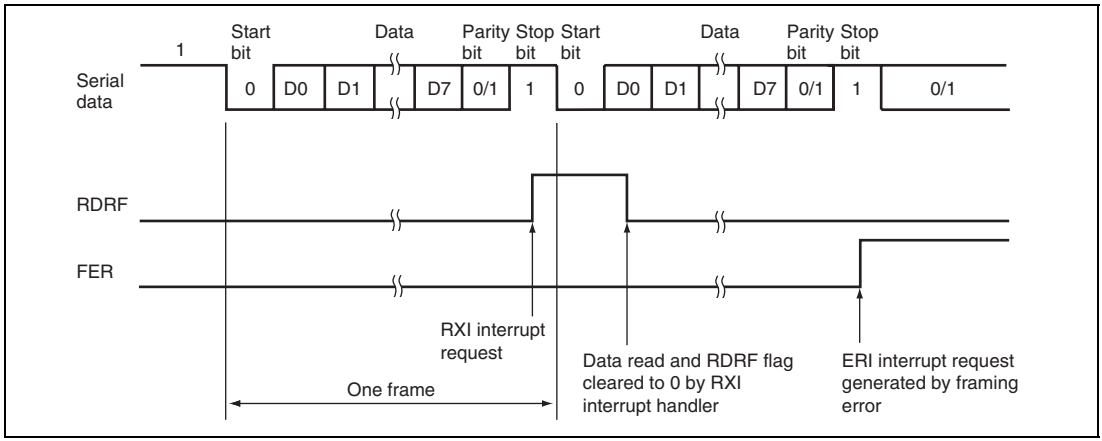


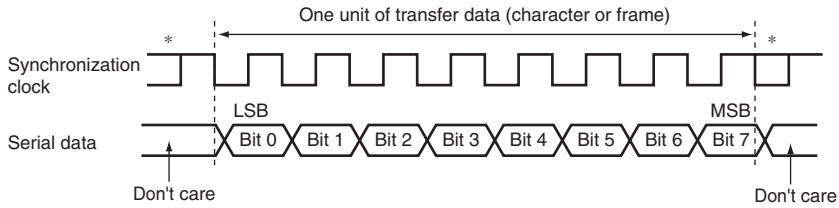
Figure 18.7 Example of SCI Receive Operation
(8-Bit Data with Parity and One Stop Bit when LSB-First Transfer is Selected)

18.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 18.8 shows the general format in clock synchronous serial communication.



Note: * High level except in continuous transfer

Figure 18.8 Data Format in Clock Synchronous Communication (when LSB-First Transfer is Selected)

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the synchronous clock to the next. Data is guaranteed valid at the rising edge of the synchronous clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB (when LSB-first transfer is selected).

In clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the synchronous clock.

(1) Communication Format

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external synchronous clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR). For selection of the SCI clock source, see table 18.11.

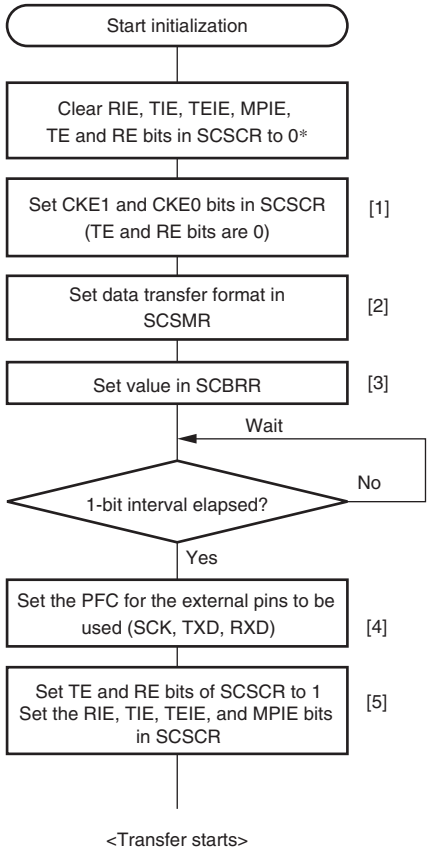
When the SCI operates on an internal clock, it outputs the synchronous clock signal at the SCK pin. Eight synchronous clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the synchronous clock signal remains in the high state. However, in reception-only operation, the synchronous clock is output until an overrun error occurs or the RE bit is cleared to 0. In operations for the reception of n characters, select the external clock as the clock source for the SCI. If the internal clock is to be used instead, set the RE and TE bits to 1, and then transmit n characters of dummy data during reception of the n characters to be received.

(3) Transmitting and Receiving Data

- SCI Initialization (Clock Synchronous Mode)

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI. Clearing TE to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 18.9 shows a sample flowchart for initializing the SCI.



[1] Set the clock selection in SCSCR.

[2] Set the data transfer format in SCSMR.

[3] Write a value corresponding to the bit rate to SCBRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RXD input during receiving and TXD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0.

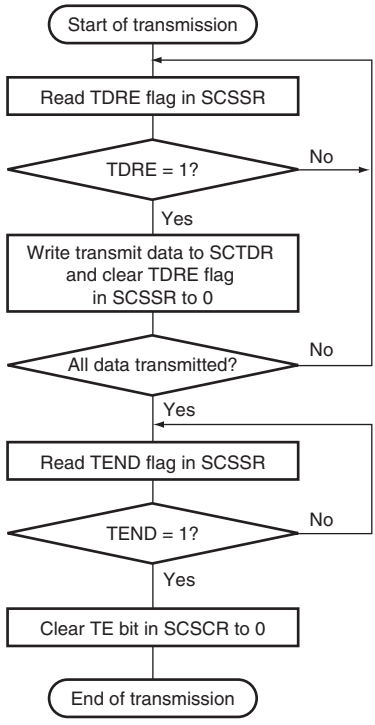
[5] Set the TE bit or RE bit in SCSCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting. When synchronous clock output (clock master) is set during receiving in clock synchronous mode, outputting clocks from the SCK pin starts.

Note: * In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 18.9 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Clock Synchronous Mode)

Figure 18.10 shows a sample flowchart for transmitting serial data. Use the following procedure for serial data transmission after enabling the SCI for transmission.



[1] SCI status check and transmit data write:

Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR, and clear the TDRE flag to 0.

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0.

Figure 18.10 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE flag in the serial status register (SCSSR). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR) and transfers the data from SCTDR to the transmit shift register (SCTSR).
2. After transferring data from SCTDR to SCTSR, the SCI sets the TDRE flag to 1 and starts transmission. If TIE in the serial control register (SCSCR) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7) (when LSB-first transfer is selected).
3. The SCI checks the TDRE flag at the timing for sending the LSB. If the TDRE flag is 0, the data is transferred from SCTDR to SCTSR and serial transmission of the next frame is started. If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the LSB is sent, and then the TXD pin holds the states.
   If the TEIE bit in SCSCR is set to 1 at this time, a TEI interrupt request is generated.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 18.11 shows an example of SCI transmit operation.

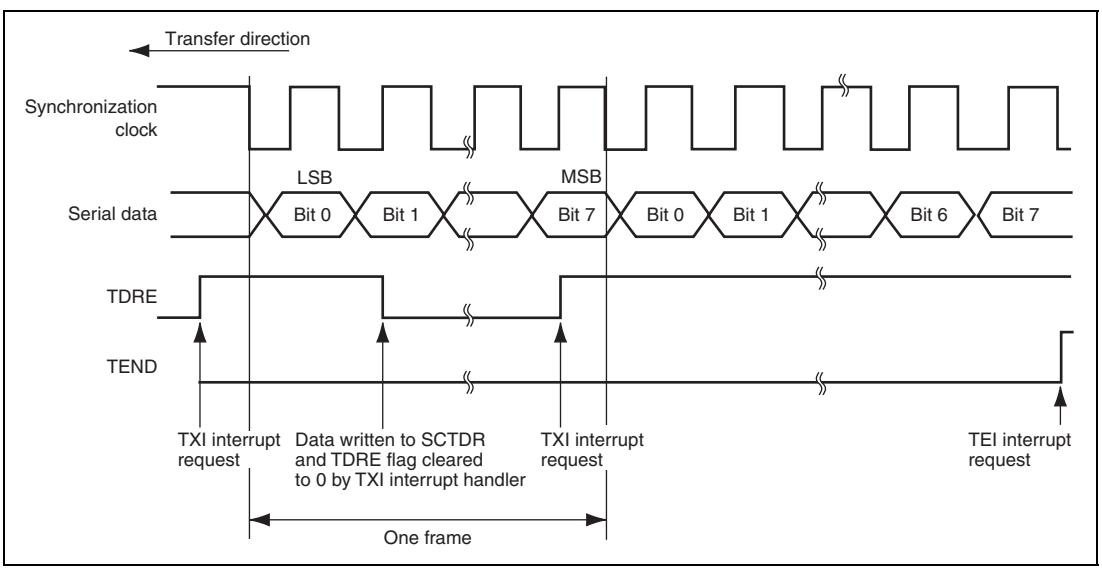
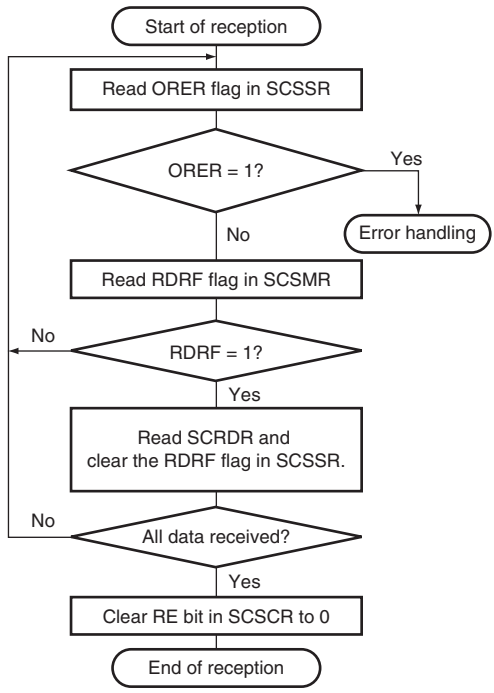


Figure 18.11 Example of SCI Transmit Operation (when LSB-First Transfer is Selected)

- Receiving Serial Data (Clock Synchronous Mode)

Figure 18.12 shows a sample flowchart for receiving serial data. Use the following procedure for serial data reception after enabling the SCI for reception.

When switching from asynchronous mode to clock synchronous mode, make sure that the ORER, PER, and FER flags are all cleared to 0. If the FER or PER flag is set to 1, the RDRF flag will not be set and data reception cannot be started.



[1] Receive error handling:

Read the ORER flag in SCSSR to identify any error and perform the appropriate error handling. Reception cannot be resumed while the ORER flag is set to 1.

[2] SCI status check and receive data read:

Read SCSSR and check that RDRF = 1, then read the receive data in SCRDR, and clear the RDRF flag to 0. The transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read the receive data register (SCRDR) and clear the RDRF flag to 0 before the MSB (bit 7) of the current frame is received.

Figure 18.12 Sample Flowchart for Receiving Serial Data (1)

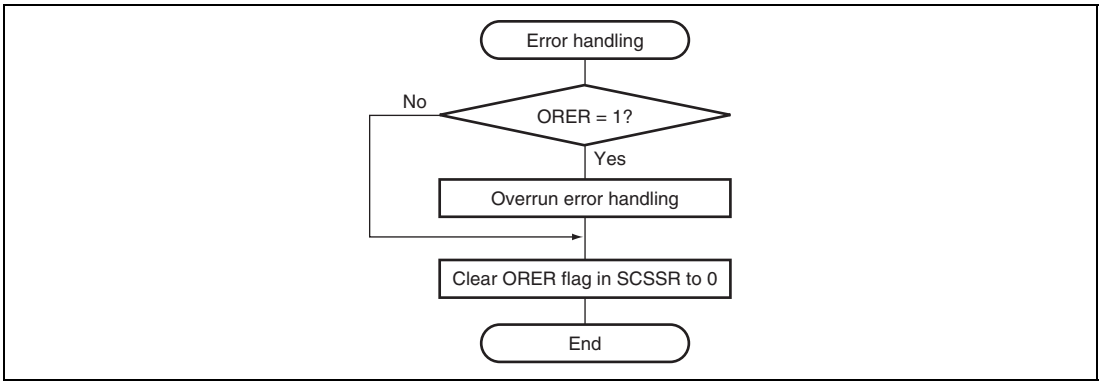


Figure 18.12 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with synchronous clock input or output and initializes internally.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB (when LSB-first transfer is selected). After receiving the data, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from SCRSR to SCRDR. If this check is passed, the SCI sets the RDRF flag to 1 and stores the received data in SCRDR. If a receive error is detected, the SCI operates as shown in table 18.13. In this state, subsequent transmission and reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the error flag to 0.
3. After setting RDRF to 1, if the RIE bit in SCSCR is set to 1, the SCI requests a receive-data-full interrupt (RXI). If the ORER flag is set to 1 and the RIE bit in SCSCR is also set to 1, the SCI requests a receive error interrupt (ERI).

Figure 18.13 shows an example of SCI receive operation.

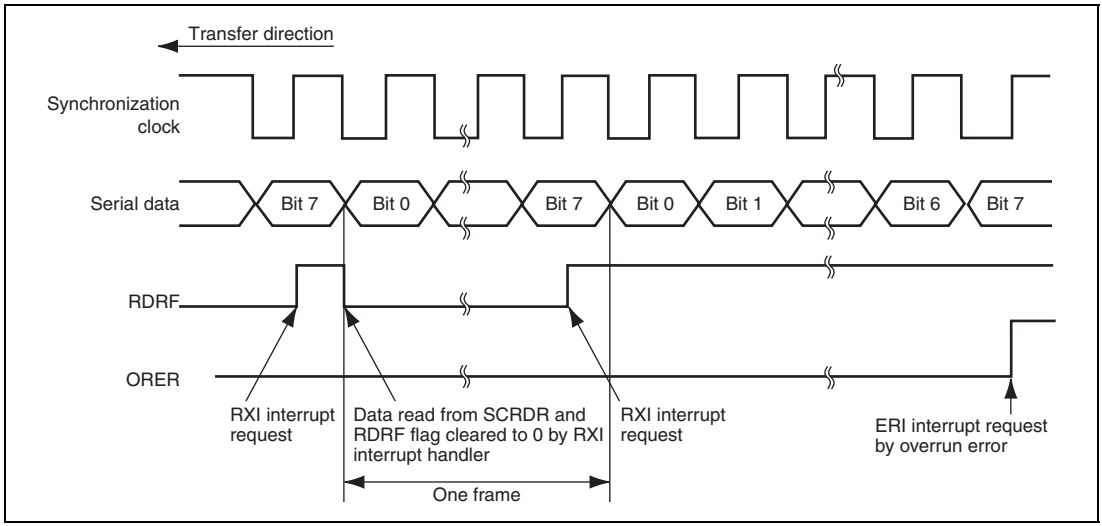
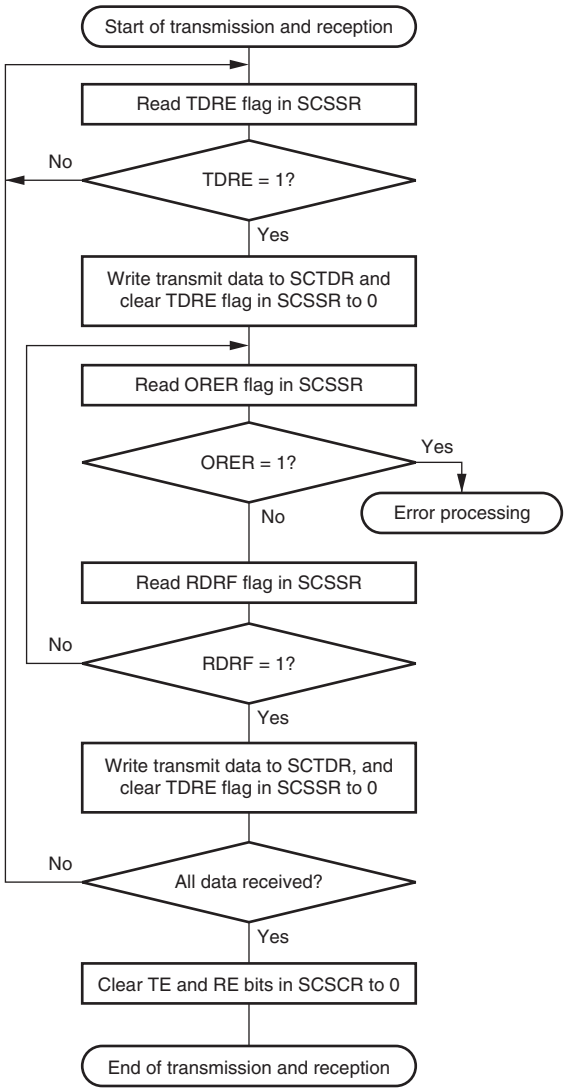


Figure 18.13 Example of SCI Receive Operation (when LSB-First Transfer is Selected)

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

Figure 18.14 shows a sample flowchart for transmitting and receiving serial data simultaneously. Use the following procedure for serial data transmission and reception after enabling the SCI for transmission and reception.



[1] SCI status check and transmit data write:
Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error processing:
If a receive error occurs, read the ORER flag in SCSSR, and after performing the appropriate error processing, clear the ORER flag to 0. Reception cannot be resumed if the ORER flag is set to 1.

[3] SCI status check and receive data read:
Read SCSSR and check that the RDRF flag is set to 1, then read the receive data in SCRDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading SCRDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to SCTDR and clear the TDRE flag to 0.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 18.14 Sample Flowchart for Transmitting/Receiving Serial Data

18.4.4 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle to the specified receiving station. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 18.15 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCSCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from SCRSR to SCRDR, error flag detection, and setting the SCSSR status flags, RDRF, FER, and OER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SCSSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCSCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

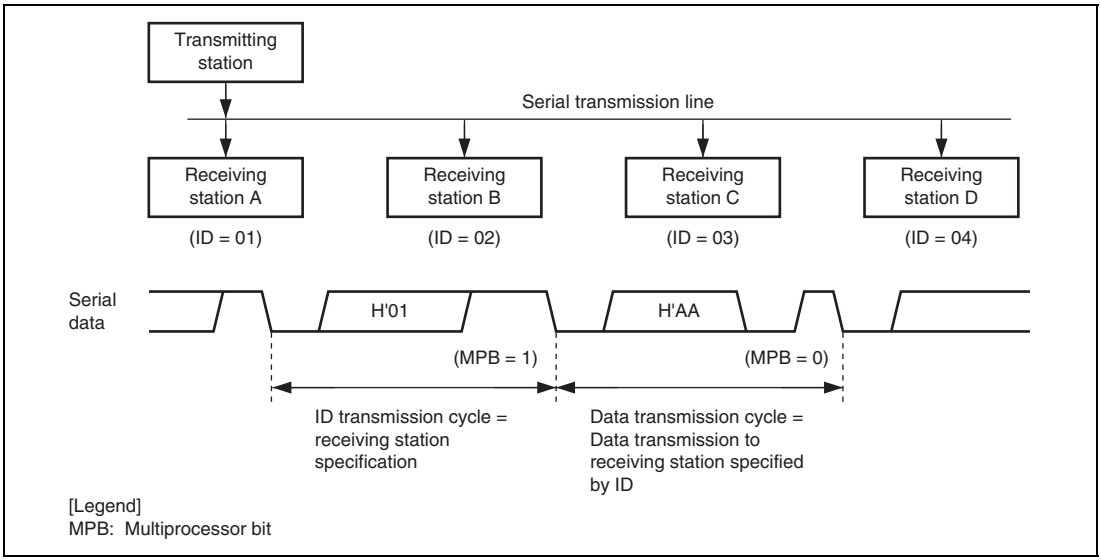


Figure 18.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

18.4.5 Multiprocessor Serial Data Transmission

Figure 18.16 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SCSSR to 1 before transmission. Keep MPBT at 1 until the ID is actually transmitted. For a data transmission cycle, clear the MPBT bit in SCSSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.

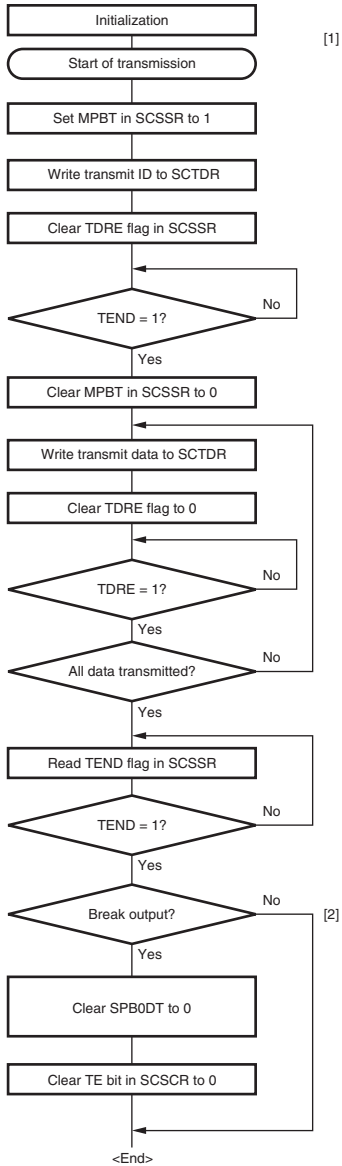


Figure 18.16 Sample Multiprocessor Serial Transmission Flowchart

18.4.6 Multiprocessor Serial Data Reception

Figure 18.18 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCSCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to SCRDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 18.17 shows an example of SCI operation for multiprocessor format reception.

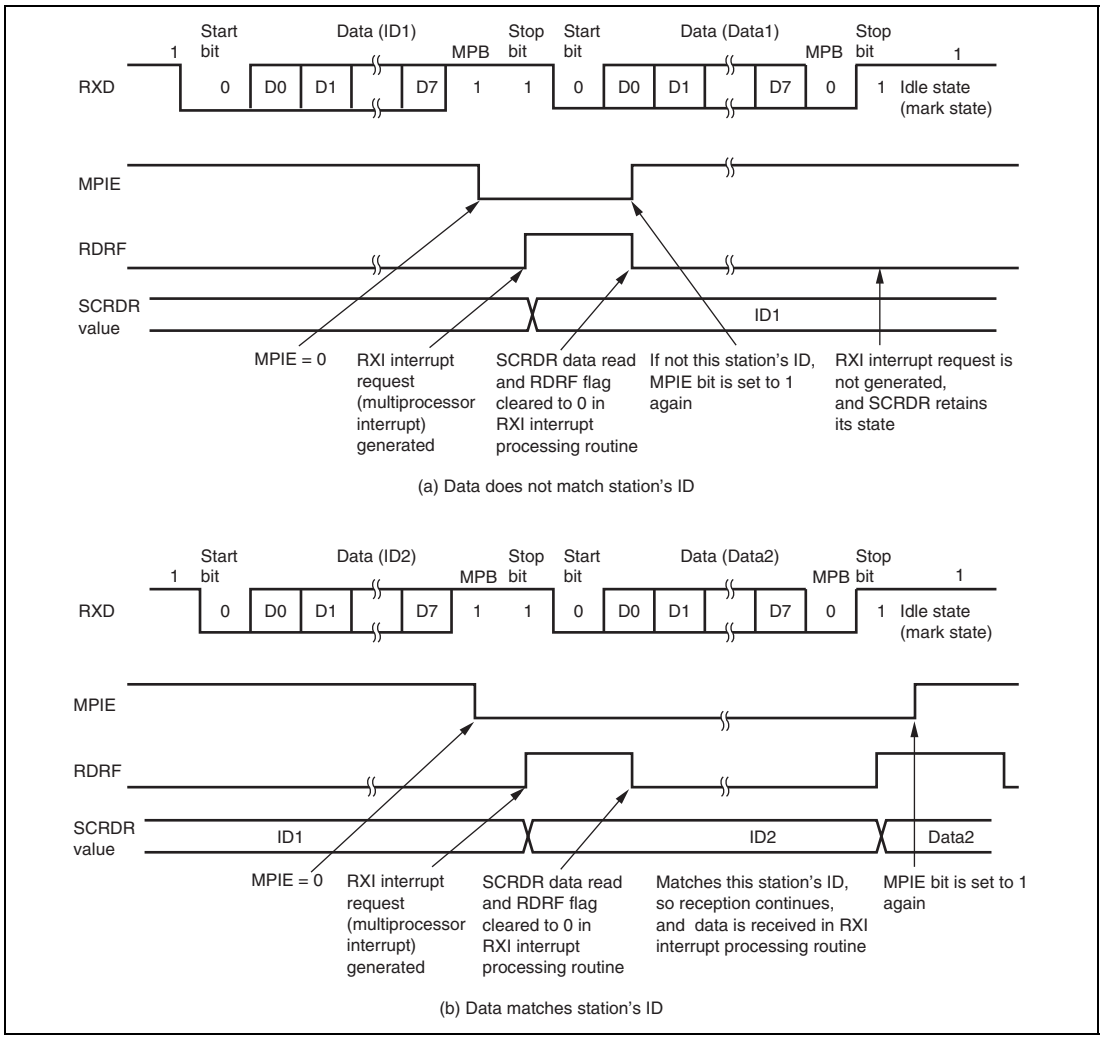
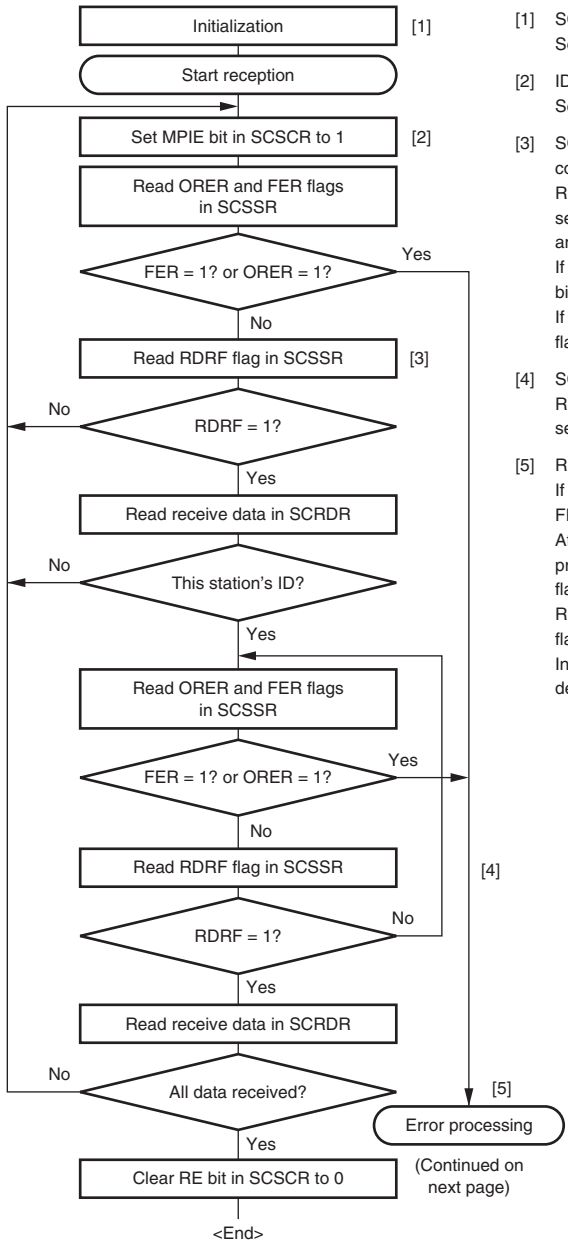


Figure 18.17 Example of SCI Receive Operation
(8-Bit Data with Multiprocessor Bit and One Stop Bit when LSB-First Transfer is Selected)



[1] SCI initialization:
Set the RXD pin using the PFC.

[2] ID reception cycle:
Set the MPIE bit in SCSCR to 1.

[3] SCI status check, ID reception and comparison:
Read SCSSR and check that the RDRF flag is set to 1, then read the receive data in SCRDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SCSSR and check that the RDRF flag is set to 1, then read the data in SCRDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SCSSR to identify the error.
After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared 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 RXD pin value.

Figure 18.18 Sample Multiprocessor Serial Reception Flowchart (1)

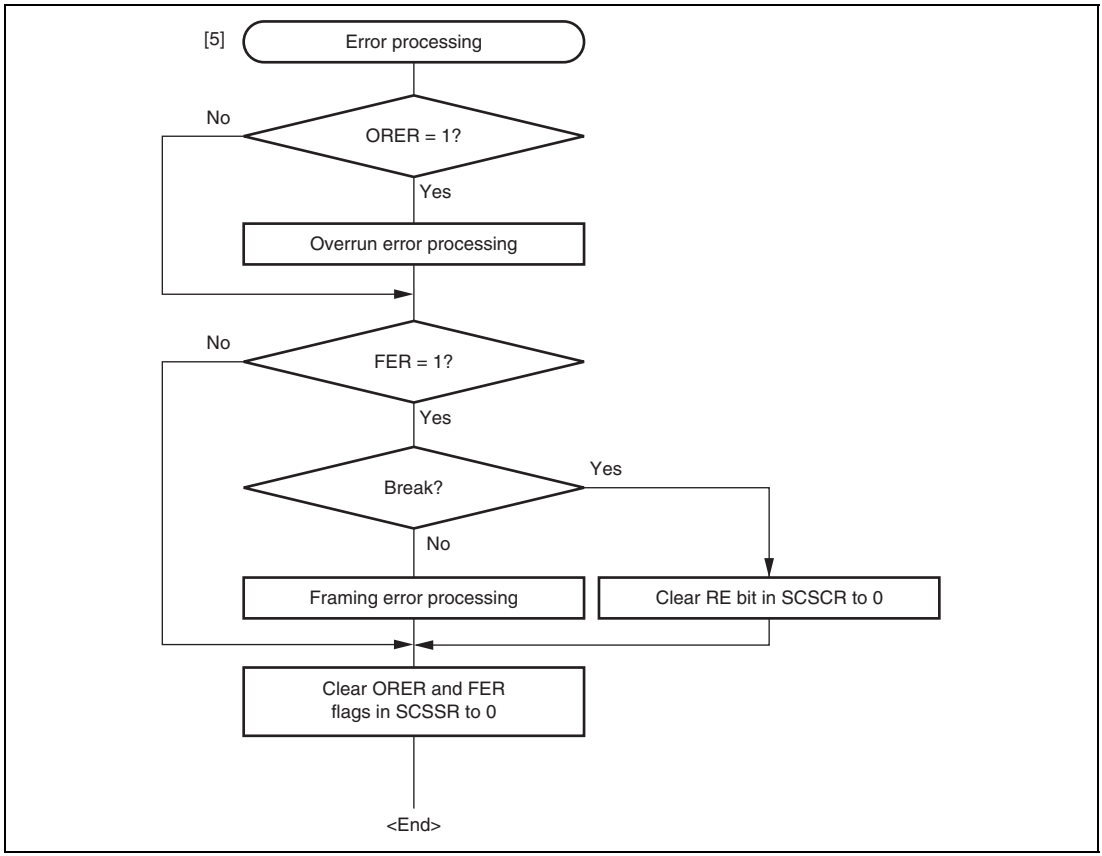


Figure 18.18 Sample Multiprocessor Serial Reception Flowchart (2)

18.5 Bit Rate Adjustment

The SCI has the bit rate adjustment function in asynchronous mode. Figure 18.19 shows an operation example when the bit rate adjustment function is enabled.

Setting the BAE bit in the serial mode register 2 (SCSMR2) to 1 allows fine adjustment of the bit rates and adjustment of the sampling timing with the bit rate adjust compare register (SCBACOR) and SPSEL[1:0] bits in SCSMR2.

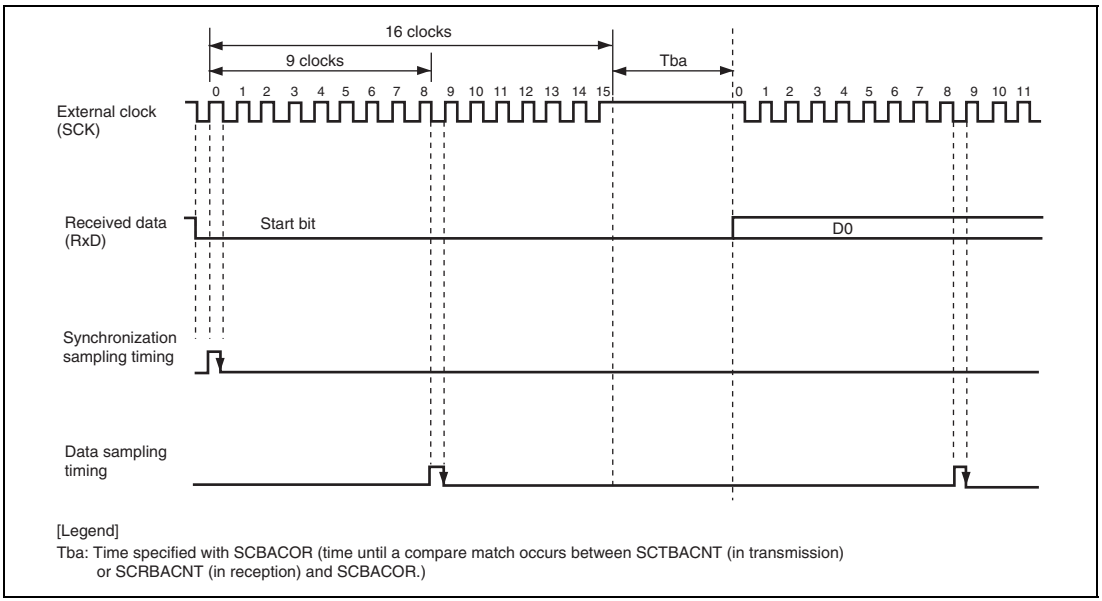


Figure 18.19 Operation Example when Bit Rate Adjustment Function is Enabled (when SPSEL[1:0] = 01 (sampling data on the ninth clock pulse))

When the bit rate adjustment function is used, the relationship among the bit rate, the bit rate register (SCBRR), and the bit rate adjust compare register (SCBACOR) is:

$$B = \frac{P\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1) + (C + 3)}$$

B: Bit rate (bits/s)

N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)

Pϕ: Operating frequency for peripheral modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 18.3)

C: SCBACOR setting ($1 \leq C \leq 255$)

The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{Bh(64 \times 2^{2n-1} \times (N + 1) + (C + 3))} - 1 \right\} \times 100$$

Bh: Bit rate to be set

Tables 18.14 shows examples of SPSEL[1:0] bit settings when the bit rate adjustment function is used, and table 18.15 shows examples of 500-kbps baud rate settings when the bit rate adjustment function is used.

Table 18.14 SPSEL[1:0] Setting when Bit Rate Adjustment Function is Used

Error when Bit Rate Adjustment Function not Used	SPSEL[1:0]	Setting
0% to 10%	00	Sampling on the eighth clock pulse
10% to 20%	01	Sampling on the ninth clock pulse
20% to 30%	10	Sampling on the tenth clock pulse
30% or more	11	Sampling on the eleventh clock pulse

Table 18.15 500-kbps Baud Rate Setting when Bit Rate Adjustment Function is Used

Pϕ Setting	SCSMR. CKS[1:0] Setting (n)	SCBRR Setting (N)	SCBACOR Setting (C)	Bit Rate before Adjustment	Bit Rate after Adjustment
Pϕ = 50 MHz	H'00	H'02	H'01	520833	500000

18.6 Interrupt Sources and DMAC/DTC

The SCI has four interrupt sources: transmit end (TEI), receive error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI) interrupt requests.

Table 18.16 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the TIE, RIE, and TEIE bits in SCSCR and the EIO bit in SCSPTR, and are separately issued to the interrupt controller.

When the TDRE flag in the serial status register (SCSSR) is set to 1, a TXI interrupt request is generated. This request can be used to activate the direct memory controller (DMAC) or data transfer controller (DTC) to transfer data. When the DMAC is used for data transfer and data is written to the transmit data register (SCTDR), the TDRE flag is automatically cleared to 0, and thus a TXI interrupt request is not issued to the CPU. When the DTC is used for data transfer and data is written to SCTDR, the TDRE flag is automatically cleared to 0, and thus a TXI interrupt request is not issued to the CPU if the DISEL bit of the DTC is 0 and the transfer counter value is not 0. However, if both the DISEL bit and the transfer counter value are 0 or if the DISEL bit is 1, the TDRE flag is not cleared to 0 even though data is written to SCTDR, and thus a TXI interrupt request is issued to the CPU after data write to SCTDR.

When the RDRF flag in SCSSR is set to 1, an RXI interrupt request is generated. This request can be used to activate the DMAC or DTC to transfer data. When the DMAC is used for data transfer and data is read from the receive data register (SCRDR), the RDRF flag is automatically cleared to 0, and thus an RXI interrupt request is not issued to the CPU. When the DTC is used for data transfer and data is read from SCRDR, the RDRF flag is automatically cleared to 0, and thus an RXI interrupt request is not issued to the CPU if the DISEL bit of the DTC is 0 and the transfer counter value is not 0. However, if both the DISEL bit and the transfer counter value are 0 or if the DISEL bit is 1, the RDRF flag is not cleared to 0 even though data is read from SCRDR, and thus an RXI interrupt request is issued to the CPU after data read from SCRDR.

When the ORER, FER, or PER flag in SCSSR is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the DMAC or DTC. When the DMAC or DTC is to be used for processing received data and an interrupt request to the CPU is to be used for processing a receive error, be sure to set the RIE bit to 1, and the EIO bit in SCSPTR to 1 so as to cause an interrupt error only by a receive error. Setting the EIO bit in SCSPTR is to 0 causes an interrupt to the CPU even in normal data reception.

When the TEND flag in SCSSR is set to 1, a TEI interrupt request is generated. This request cannot be used to activate the DMAC or DTC.

The TXI interrupt indicates that transmit data can be written, and the TEI interrupt indicates that transmission has been completed.

Table 18.16 SCI Interrupt Sources

Interrupt Source	Description	Interrupt Enable Bit	DMAC/DTC Activation	Priority
ERI	Interrupt caused by receive error (ORER, FER, or PER)	RIE = 1	Not possible	High
RXI	Interrupt caused by receive data full (RDRF)	RIE = 1 and EIO = 0	Possible	
TXI	Interrupt caused by transmit data empty (TDRE)	TIE = 1	Possible	
TEI	Interrupt caused by transmit end (TENT)	TEIE = 1	Not possible	Low

18.7 Serial Port Register (SCSPTR) and SCI Pins

The relationship between SCSPTR and the SCI pins is shown in figures 18.20 and 18.21.

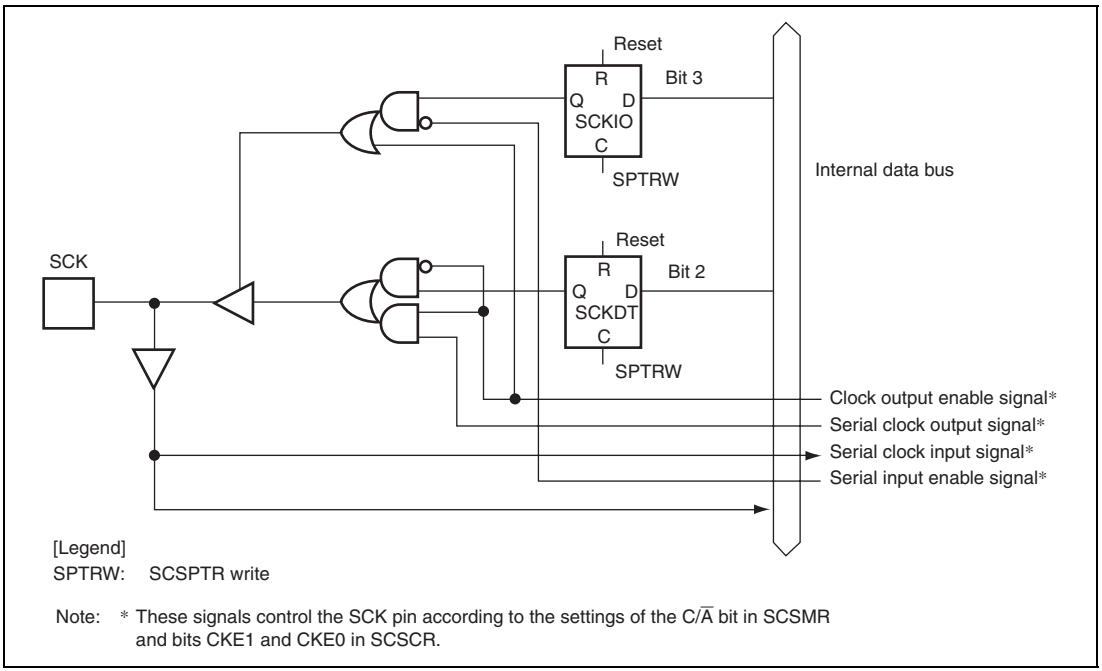


Figure 18.20 SCKIO Bit, SCKDT Bit, and SCK Pin

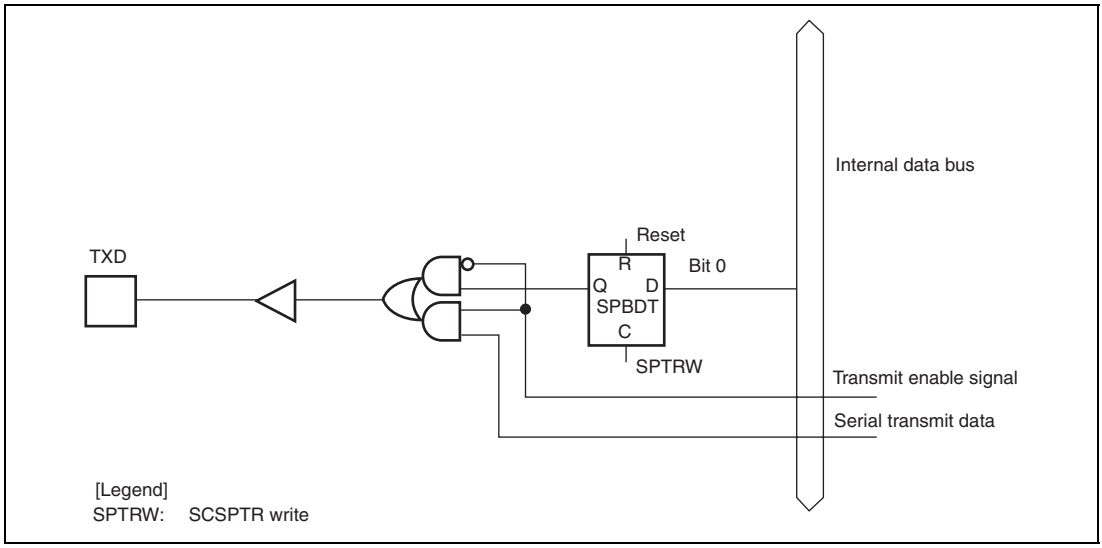


Figure 18.21 SPBDT Bit and TXD Pin

18.8 Usage Notes

18.8.1 SCTDR Writing and TDRE Flag

The TDRE flag in the serial status register (SCSSR) is a status flag indicating transferring of transmit data from SCTDR into SCTSR. The SCI sets the TDRE flag to 1 when it transfers data from SCTDR to SCTSR.

Data can be written to SCTDR regardless of the TDRE bit status.

If new data is written to SCTDR when TDRE is 0, however, the old data stored in SCTDR will be lost because the data has not yet been transferred to SCTSR. Before writing transmit data to SCTDR, be sure to check that the TDRE flag is set to 1.

18.8.2 Multiple Receive Error Occurrence

If multiple receive errors occur at the same time, the status flags in SCSSR are set as shown in table 18.17. When an overrun error occurs, data is not transferred from the receive shift register (SCRSR) to the receive data register (SCRDR) and the received data will be lost.

Table 18.17 SCSSR Status Flag Values and Transfer of Received Data

	SCSSR Status Flags				Receive Data Transfer from SCRSR to SCRDR
Receive Errors Generated	**RDRF**	**ORER**	**FER**	**PER**	
Overrun error	1	1	0	0	Not transferred
Framing error	0	0	1	0	Transferred
Parity error	0	0	0	1	Transferred
Overrun error + framing error	1	1	1	0	Not transferred
Overrun error + parity error	1	1	0	1	Not transferred
Framing error + parity error	0	0	1	1	Transferred
Overrun error + framing error + parity error	1	1	1	1	Not transferred

18.8.3 Break Detection and Processing

Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCRDR is halted in the break state, the SCI receiver continues to operate.

18.8.4 Sending a Break Signal

The I/O condition and level of the TXD pin are determined by the SPB0DT bit in the serial port register (SCSPTR). This feature can be used to send a break signal.

Until the TE bit is set to 1 (enabling transmission) after initializing, TXD pin does not work. During the period, mark state is performed by SPB0DT bit. Therefore, the SPB0DT bit should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB0DT bit to 0 (low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TXD pin.

18.8.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCI operates on a base clock with a frequency of 16 times the transfer rate in asynchronous mode. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. For setting the BAE, SPSEL[1:0] bits in the serial mode register 2 (SCSMR2), and the bit rate adjust compare register (SCBACOR), refer to section 18.5, Bit Rate Adjustment. The timing is shown in figure 18.22.

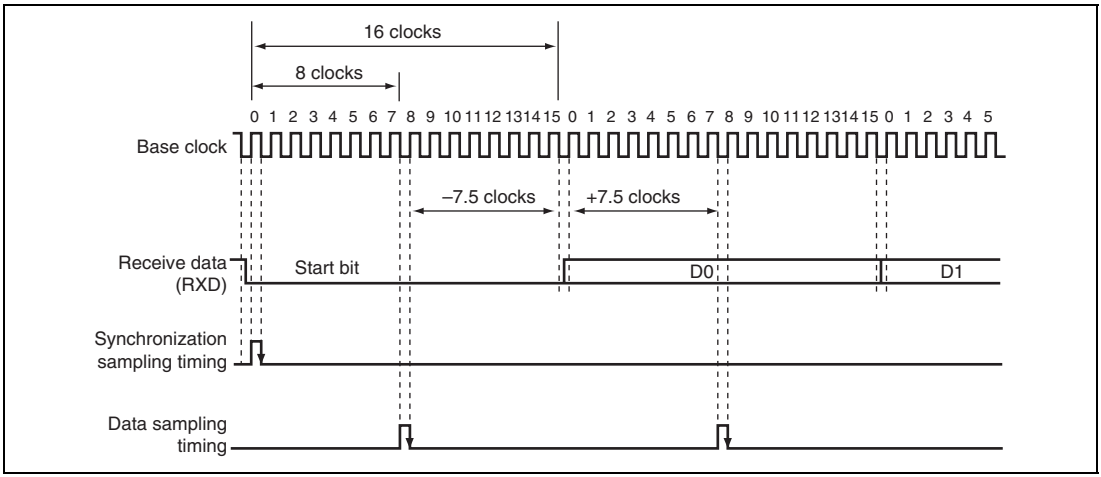


Figure 18.22 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1+F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

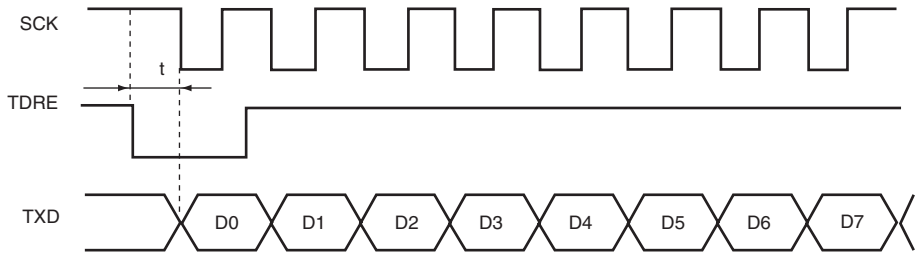
When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

18.8.6 Note on Using DMAC/DTC

When the external clock source is used for the clock for synchronization, input the external clock after waiting for five or more cycles of the peripheral operating clock after SCTDR is modified through the DMAC or DTC. If a transmit clock is input within four cycles after SCTDR is modified, a malfunction may occur (figure 18.23).



Note: When using the external clock, t must be set to larger than 4 cycles.

Figure 18.23 Example of Clock Synchronous Transfer Using DMAC or DTC (when LSB-First Transfer is Selected)

When data is written to SCTDR by activating the DMAC or DTC by a TXI interrupt, the TEND flag value becomes undefined. In this case, do not use the TEND flag as the transmit end flag.

18.8.7 Note on Using External Clock in Clock Synchronous Mode

TE and RE must be set to 1 after waiting for four or more cycles of the peripheral operating clock after the external clock (SCK) is changed from 0 (low) to 1 (high).

TE and RE must be set to 1 only while the external clock (SCK) is 1 (high).

18.8.8 Module Standby Mode Setting

SCI operation can be disabled or enabled using the standby control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 32, Power-Down Modes.

Section 19 Serial Communication Interface with FIFO (SCIF)

This LSI has four channels of serial communication interface with FIFO (SCIF) that supports both asynchronous and clocked synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

19.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RXD level directly from the port register when a framing error occurs.
- Clocked synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clocked synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)

- Four types of interrupts: Transmit-FIFO-data-empty interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupts are requested independently. The direct memory access controller (DMAC) or data transfer controller (DTC) can be activated by the transmit-FIFO-data-empty interrupt request or receive-FIFO-data-full interrupt request to transfer data.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- The number of data units in the transmit and receive FIFO data registers and the number of receive errors of the received data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.
- LSB-first or MSB-first transfer selectable (except for 7-bit data in asynchronous mode).

Figure 19.1 shows a block diagram of the SCIF.

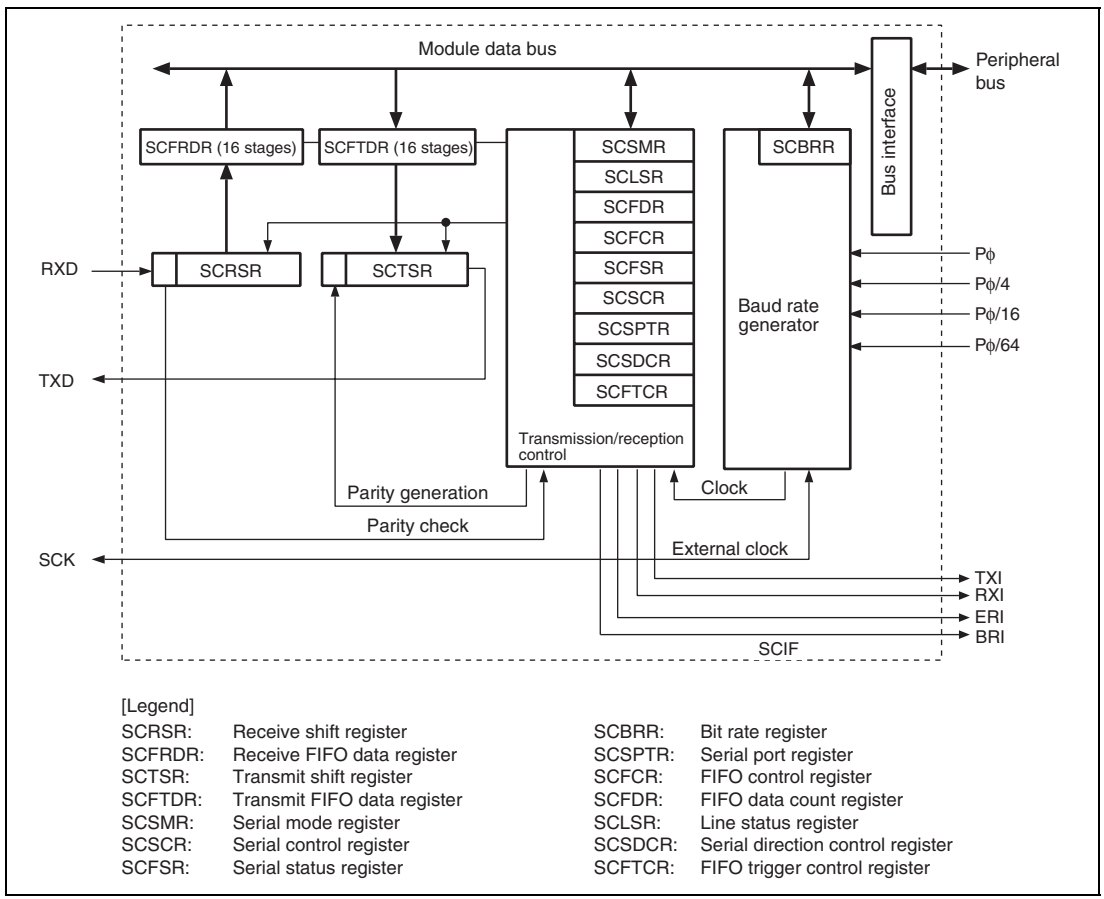


Figure 19.1 Block Diagram of SCIF

19.2 Input/Output Pins

Table 19.1 shows the pin configuration of the SCIF.

Table 19.1 Pin Configuration

Channel	Name	Pin Name*	I/O	Function
4 to 7	Serial clock pins	SCK4 to SCK7	I/O	Clock I/O
	Receive data pins	RXD4 to RXD7	Input	Receive data input
	Transmit data pins	TXD4 to TXD7	Output	Transmit data output

Note: * Pin names SCK, RXD, and TXD are used in the description for all channels, omitting the channel designation.

19.3 Register Descriptions

The SCIF has the following registers. For details on register addresses and register states during each processing, refer to section 34, List of Registers.

Table 19.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	Serial mode register_4	SCSMR_4	R/W	H'0000	H'FFFE8000	16
	Bit rate register_4	SCBRR_4	R/W	H'FF	H'FFFE8004	8
	Serial control register_4	SCSCR_4	R/W	H'0000	H'FFFE8008	16
	Transmit FIFO data register_4	SCFTDR_4	W	Undefined	H'FFFE800C	8
	Serial status register_4	SCFSR_4	R/W	H'0060	H'FFFE8010	16
	Receive FIFO data register_4	SCFRDR_4	R	Undefined	H'FFFE8014	8
	FIFO control register_4	SCFCR_4	R/W	H'0000	H'FFFE8018	16
	FIFO data count register_4	SCFDR_4	R	H'0000	H'FFFE801C	16
	Serial port register_4	SCSPTR_4	R/W	H'0050	H'FFFE8020	16
	Line status register_4	SCLSR_4	R/W	H'0000	H'FFFE8024	16
	Serial direction control register_4	SCSDCR_4	R/W	H'F2	H'FFFE8102	8
	FIFO trigger control register_4	SCFTCR_4	R/W	H'1F1F	H'FFFE8104	16
5	Serial mode register_5	SCSMR_5	R/W	H'0000	H'FFFE8800	16
	Bit rate register_5	SCBRR_5	R/W	H'FF	H'FFFE8804	8
	Serial control register_5	SCSCR_5	R/W	H'0000	H'FFFE8808	16
	Transmit FIFO data register_5	SCFTDR_5	W	Undefined	H'FFFE880C	8
	Serial status register_5	SCFSR_5	R/W	H'0060	H'FFFE8810	16
	Receive FIFO data register_5	SCFRDR_5	R	Undefined	H'FFFE8814	8
	FIFO control register_5	SCFCR_5	R/W	H'0000	H'FFFE8818	16
	FIFO data count register_5	SCFDR_5	R	H'0000	H'FFFE881C	16
	Serial port register_5	SCSPTR_5	R/W	H'0050	H'FFFE8820	16
	Line status register_5	SCLSR_5	R/W	H'0000	H'FFFE8824	16
	Serial direction control register_5	SCSDCR_5	R/W	H'F2	H'FFFE8902	8
	FIFO trigger control register_5	SCFTCR_5	R/W	H'1F1F	H'FFFE8904	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	Serial mode register_6	SCSMR_6	R/W	H'0000	H'FFFE9000	16
	Bit rate register_6	SCBRR_6	R/W	H'FF	H'FFFE9004	8
	Serial control register_6	SCSCR_6	R/W	H'0000	H'FFFE9008	16
	Transmit FIFO data register_6	SCFTDR_6	W	Undefined	H'FFFE900C	8
	Serial status register_6	SCFSR_6	R/W	H'0060	H'FFFE9010	16
	Receive FIFO data register_6	SCFRDR_6	R	Undefined	H'FFFE9014	8
	FIFO control register_6	SCFCR_6	R/W	H'0000	H'FFFE9018	16
	FIFO data count register_6	SCFDR_6	R	H'0000	H'FFFE901C	16
	Serial port register_6	SCSPTR_6	R/W	H'0050	H'FFFE9020	16
	Line status register_6	SCLSR_6	R/W	H'0000	H'FFFE9024	16
	Serial direction control register_6	SCSDCR_6	R/W	H'F2	H'FFFE9102	8
	FIFO trigger control register_6	SCFTCR_6	R/W	H'1F1F	H'FFFE9104	16
7	Serial mode register_7	SCSMR_7	R/W	H'0000	H'FFFE9800	16
	Bit rate register_7	SCBRR_7	R/W	H'FF	H'FFFE9804	8
	Serial control register_7	SCSCR_7	R/W	H'0000	H'FFFE9808	16
	Transmit FIFO data register_7	SCFTDR_7	W	Undefined	H'FFFE980C	8
	Serial status register_7	SCFSR_7	R/W	H'0060	H'FFFE9810	16
	Receive FIFO data register_7	SCFRDR_7	R	Undefined	H'FFFE9814	8
	FIFO control register_7	SCFCR_7	R/W	H'0000	H'FFFE9818	16
	FIFO data count register_7	SCFDR_7	R	H'0000	H'FFFE981C	16
	Serial port register_7	SCSPTR_7	R/W	H'0050	H'FFFE9820	16
	Line status register_7	SCLSR_7	R/W	H'0000	H'FFFE9824	16
	Serial direction control register_7	SCSDCR_7	R/W	H'F2	H'FFFE9902	8
	FIFO trigger control register_7	SCFTCR_7	R/W	H'1F1F	H'FFFE9904	16

19.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. The SCIF converts data input at the RXD pin and loaded into SCRSR to parallel form. When one byte has been received, it is automatically transferred to the receive FIFO data register (SCFRDR).

The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

19.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-byte FIFO register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to an undefined value by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

19.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCIF loads transmit data from the transmit FIFO data register (SCFTDR) into SCTSR, then transmits the data serially from the TXD pin. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmission again.

The CPU cannot read or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

19.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-byte FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into SCTSR and starts serial transmission. Continuous serial transmission is performed until there is no transmit data left in SCFTDR. Writing to SCFTDR is possible when the TDFE bit in SCFSR is 1.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

SCFTDR is initialized to an undefined value by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

19.3.5 Serial Mode Register (SCSMR)

SCSMR specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or clocked synchronous mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In clocked synchronous mode, the data length is always 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clocked synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in clocked synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clocked synchronous mode because no stop bits are added. When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit When transmitting, a single 1-bit is added at the end of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at the end of each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select Select the internal clock source of the on-chip baud rate generator. For further information on the clock source, bit rate register settings, and baud rate, see section 19.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

19.3.6 Serial Control Register (SCSCR)

SCSCR operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in the transmit FIFO data register becomes less than the specified number of transmit triggers, and when the TDFE flag in the serial status register (SCFSR) is set to 1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled* Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmit trigger number to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-FIFO-data-full (RXI) interrupts requested when the RDF flag or DR flag in the serial status register (SCFSR) is set to 1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to 1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in the line status register (SCLSR) is set to 1. 0: Receive-FIFO-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled 1: Receive-FIFO-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled* Note: * RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BRK or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing of transmit data into SCFTDR. Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the serial receiver of the SCIF. 0: Receiver disabled*[1] 1: Receiver enabled*[2] Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in clocked synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of the REIE bit is valid only when the RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled* Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BRK or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled. Set so if SCIF wants to inform INTC of ERI or BRI interrupt requests during DTC transfer.

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKE[1:0]	00	R/W	Clock Enable Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on CKE[1:0], the SCK pin can be used for serial clock output or serial clock input. If synchronous clock output is set in clocked synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1, and then set CKE[1:0]. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (input signal is ignored). The SCK pin state depends on the SCKIO and SCKDT bits in SCSPTR. 01: Internal clock, SCK pin used for clock output (The output clock frequency is 16 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate.) 11: Setting prohibited • Clocked synchronous mode 00: Internal clock, SCK pin used for synchronous clock output 01: Internal clock, SCK pin used for synchronous clock output 10: External clock, SCK pin used for synchronous clock input 11: Setting prohibited

19.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO data register, and the lower 8 bits indicate the status flag indicating SCIF operating state.

The CPU can always read and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER[3:0]	0000	R	Number of Parity Errors Indicate the quantity of data including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). The value indicated by bits 15 to 12 after the ER bit in SCFSR is set, represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER[3:0] shows 0000.
11 to 8	FER[3:0]	0000	R	Number of Framing Errors Indicate the quantity of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 after the ER bit in SCFSR is set, represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER[3:0] shows 0000.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/(W)*	Receive Error Indicates the occurrence of a framing error, or of a parity error when receiving data that includes parity.*[1] 0: Receiving is in progress or has ended normally [Clearing conditions] • ER is cleared to 0 by a power-on reset • ER is cleared to 0 when 0 is written after 1 is read from ER 1: A framing error or parity error has occurred. [Setting conditions] • ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one data receive operation*[2] • ER is set to 1 when the total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the O/$\overline{E}$ bit in SCSMR Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCFRDR includes a receive error can be detected by the FER and PER bits in SCFSR. 2. In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/(W)*	Transmit End Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] • TEND is cleared to 0 when 0 is written after 1 is read from TEND after transmit data is written in SCFTDR* 1: End of transmission [Setting conditions] • TEND is set to 1 by a power-on reset • TEND is set to 1 when TE is cleared to 0 in SCSCR • TEND is set to 1 when SCFTDR does not contain receive data when the last bit of a one-byte serial character is transmitted Note: * Do not use this bit as a transmit end flag when the DMAC or DTC writes data to SCFTDR due to a TXI interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/(W)*	Transmit FIFO Data Empty Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in SCFTDR has become less than the specified transmit trigger number, and writing of transmit data to SCFTDR is enabled. 0: The quantity of transmit data written to SCFTDR is greater than the specified transmit trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the specified transmit trigger number is written to SCFTDR after 1 is read from TDFE and then 0 is written • TDFE is cleared to 0 when data exceeding the specified transmit trigger number is written to SCFTDR by the DMAC. • TDFE is cleared to 0 when data exceeding the specified transmit trigger number is written to SCFTDR by the DTC. (Except the transfer counter value of DTC has become H'0000) 1: The quantity of transmit data in SCFTDR is less than the specified transmit trigger number* [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the quantity of transmit data in SCFTDR becomes less than the specified transmit trigger number as a result of transmission. Note: * Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the specified transmit trigger number". If an attempt is made to write additional data, the data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFDR.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal has been detected in receive data. 0: No break signal received [Clearing conditions] • BRK is cleared to 0 by a power-on reset • BRK is cleared to 0 when software reads BRK after it has been set to 1, then writes 0 to BRK 1: Break signal received* [Setting condition] • BRK is set to 1 when data including a framing error is received, and a framing error occurs with space 0 in the subsequent receive data Note: * When a break is detected, transfer of the receive data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of receive data resumes.
3	FER	0	R	Framing Error Indication Indicates a framing error in the data read from the receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive framing error occurred in the next data read from SCFRDR [Clearing conditions] • FER is cleared to 0 by a power-on reset • FER is cleared to 0 when no framing error is present in the next data read from SCFRDR 1: A receive framing error occurred in the next data read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is present in the next data read from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indication Indicates a parity error in the data read from the receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive parity error occurred in the next data read from SCFRDR [Clearing conditions] • PER is cleared to 0 by a power-on reset • PER is cleared to 0 when no parity error is present in the next data read from SCFRDR 1: A receive parity error occurred in the next data read from SCFRDR [Setting condition] • PER is set to 1 when a parity error is present in the next data read from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that receive data has been transferred to the receive FIFO data register (SCFRDR), and the quantity of data in SCFRDR has become more than the specified receive trigger number. 0: The quantity of receive data written to SCFRDR is less than the specified receive trigger number [Clearing conditions] • RDF is cleared to 0 by a power-on reset • RDF is cleared to 0 when SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF and then 0 is written • RDF is cleared to 0 when SCFRDR is read by the DMAC until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number. • RDF is cleared to 0 when SCFRDR is read by the DTC until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number. (Except the transfer counter value of DTC has become H'0000) 1: The quantity of receive data in SCFRDR is more than the specified receive trigger number [Setting condition] • RDF is set to 1 when a quantity of receive data more than the specified receive trigger number is stored in SCFRDR* Note: * As SCFRDR is a 16-byte FIFO register, the maximum quantity of data that can be read when RDF is 1 becomes the specified receive trigger number. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the quantity of data in the receive FIFO data register (SCFRDR) is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clocked synchronous mode, this bit is not set to 1. 0: Receiving is in progress, or no receive data remains in SCFRDR after receiving ended normally [Clearing conditions] • DR is cleared to 0 by a power-on reset • DR is cleared to 0 when all receive data are read after 1 is read from DR and then 0 is written. • DR is cleared to 0 when all receive data in SCFRDR are read by the DMAC/DTC. 1: Next receive data has not been received [Setting condition] • DR is set to 1 when SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit.* Note: * This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)

Note: * Only 0 can be written to clear the flag after 1 is read.

19.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS[1:0] bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SCBRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting must satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 19.3.)

Table 19.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous mode is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 19.4 lists examples of SCBRR settings in asynchronous mode, and table 19.5 lists examples of SCBRR settings in clocked synchronous mode.

Table 19.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1)

Bit Rate (bits/s)	Pϕ (MHz)																	
	10			12			14			16			18			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	-0.25	2	212	0.03	2	248	-0.17	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	32	-1.36	0	38	0.16	0	45	-0.93	0	51	0.16	0	58	-0.69	0	64	0.16
14400	0	21	-1.36	0	25	0.16	0	29	1.27	0	34	-0.79	0	38	0.16	0	42	0.94
19200	0	15	1.73	0	19	-2.34	0	22	-0.93	0	25	0.16	0	28	1.02	0	32	-1.36
28800	0	10	-1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	-2.34	0	21	-1.36
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	7	1.73	0	9	-2.34	0	10	3.57	0	12	0.16	0	14	-2.34	0	15	1.73
115200	0	2	-9.58	0	2	8.51	0	3	-5.06	0	3	8.51	0	4	-2.34	0	4	8.51
500000	0	0*	-37.5	0	0*	-25.0	0	0*	-12.5	0	0*	0.00	0	0*	12.5	0	0*	25.0

Table 19.4 Bit Rates and SCBRR Settings in Asynchronous Mode (2)

Bit Rate (bits/s)	Pφ (MHz)																	
	22			24			26			28			30			32		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	97	-0.35	3	106	-0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	-0.54	3	77	0.16	3	84	-0.43	3	90	0.16	3	97	-0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	-0.54	2	77	0.16	2	84	-0.43	2	90	0.16	2	97	-0.35	2	103	0.16
1200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2400	1	71	-0.54	1	77	0.16	1	84	-0.43	1	90	0.16	1	97	-0.35	1	103	0.16
4800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9600	0	71	-0.54	0	77	0.16	0	84	-0.43	0	90	0.16	0	97	-0.35	0	103	0.16
14400	0	47	-0.54	0	51	0.16	0	55	0.76	0	60	-0.39	0	64	0.16	0	68	0.64
19200	0	35	-0.54	0	38	0.16	0	41	0.76	0	45	-0.93	0	48	-0.35	0	51	0.16
28800	0	23	-0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	-1.36	0	34	-0.79
31250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38400	0	17	-0.54	0	19	-2.34	0	20	0.76	0	22	-0.93	0	23	1.73	0	25	0.16
115200	0	5	-0.54	0	6	-6.99	0	6	0.76	0	7	-5.06	0	7	1.73	0	8	-3.55
500000	0	0*	37.5	0	1	-25.0	0	1	-18.8	0	1	-12.5	0	1	-6.25	0	1	0.00

Table 19.4 Bit Rates and SCBRR Settings in Asynchronous Mode (3)

	Pϕ (MHz)														
Bit Rate	34			36			38			40			42		
(bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	150	-0.05	3	159	-0.12	3	168	-0.19	3	177	-0.25	3	185	0.23
150	3	110	-0.29	3	116	0.16	3	123	-0.24	3	129	0.16	3	136	-0.21
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16	3	67	0.53
600	2	110	-0.29	2	116	0.16	2	123	-0.24	2	129	0.16	2	136	-0.21
1200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16	2	67	0.53
2400	1	110	-0.29	1	116	0.16	1	123	-0.24	1	129	0.16	1	136	-0.21
4800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16	1	67	0.53
9600	0	110	-0.29	0	116	0.16	0	123	-0.24	0	129	0.16	0	136	-0.21
14400	0	73	-0.29	0	77	0.16	0	81	0.57	0	86	-0.22	0	90	0.16
19200	0	54	0.62	0	58	-0.69	0	61	-0.24	0	64	0.16	0	67	0.53
28800	0	36	-0.29	0	38	0.16	0	40	0.57	0	42	0.94	0	45	-0.93
31250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00	0	41	0.00
38400	0	27	-1.18	0	28	1.02	0	30	-0.24	0	32	-1.36	0	33	0.53
115200	0	8	2.48	0	9	-2.34	0	9	3.08	0	10	-1.36	0	10	3.57
500000	0	1	6.25	0	1	12.5	0	1	18.8	0	2	-16.7	0	2	-12.5

Table 19.4 Bit Rates and SCBRR Settings in Asynchronous Mode (4)

Bit Rate (bits/s)	Pϕ (MHz) 44 n	N	Error (%)	46 n	N	Error (%)	48 n	N	Error (%)	50 n	N	Error (%)
110	3	194	0.16	3	203	0.09	3	212	0.03	3	221	-0.02
150	3	142	0.16	3	149	-0.17	3	155	0.16	3	162	-0.15
300	3	71	-0.54	3	74	-0.17	3	77	0.16	3	80	0.47
600	2	142	0.16	2	149	-0.17	2	155	0.16	2	162	-0.15
1200	2	71	-0.54	2	74	-0.17	2	77	0.16	2	80	0.47
2400	1	142	0.16	1	149	-0.17	1	155	0.16	1	162	-0.15
4800	1	71	-0.54	1	74	-0.17	1	77	0.16	1	80	0.47
9600	0	142	0.16	0	149	-0.17	0	155	0.16	0	162	-0.15
14400	0	94	0.51	0	99	-0.17	0	103	0.16	0	108	-0.45
19200	0	71	-0.54	0	74	-0.17	0	77	0.16	0	80	0.47
28800	0	47	-0.54	0	49	-0.17	0	51	0.16	0	53	0.47
31250	0	43	0.00	0	45	0.00	0	47	0.00	0	49	0
38400	0	35	-0.54	0	36	1.18	0	38	0.16	0	40	-0.76
115200	0	11	-0.54	0	11	3.99	0	12	0.16	0	13	-3.12
500000	0	2	-8.33	0	2	-4.17	0	2	0.00	0	2	4.17

Table 19.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (1)

Bit Rate (bits/s)	Pϕ (MHz)											
	10		12		14		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1000	2	155	2	187	2	218	2	249	3	69	3	77
2500	1	249	2	74	2	87	2	99	2	112	2	124
5000	1	124	1	149	1	174	1	199	1	224	1	249
10000	0	249	1	74	1	87	1	99	1	112	1	124
25000	0	99	0	119	0	139	0	159	0	179	0	199
50000	0	49	0	59	0	69	0	79	0	89	0	99
100000	0	24	0	29	0	34	0	39	0	44	0	49
250000	0	9	0	11	0	13	0	15	0	17	0	19
500000	0	4	0	5	0	6	0	7	0	8	0	9
1000000	—	—	0	2	—	—	0	3	—	—	0	4
2500000	0	0*	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*

Table 19.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (2)

Bit Rate (bits/s)	Pϕ (MHz)											
	22		24		26		28		30		32	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1000	3	85	3	93	3	101	3	108	3	116	3	124
2500	2	137	2	149	2	162	2	174	2	187	2	199
5000	2	68	2	74	2	80	2	87	2	93	2	99
10000	1	137	1	149	1	162	1	174	1	187	1	199
25000	0	219	0	239	1	64	1	69	1	74	1	79
50000	0	109	0	119	0	129	0	139	0	149	0	159
100000	0	54	0	59	0	64	0	69	0	74	0	79
250000	0	21	0	23	0	25	0	27	0	29	0	31
500000	0	10	0	11	0	12	0	13	0	14	0	15
1000000	—	—	0	5	—	—	0	6	—	—	0	7
2500000	—	—	—	—	—	—	—	—	0	2	—	—
5000000	—	—	—	—	—	—	—	—	—	—	—	—

Table 19.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (3)

Bit Rate (bits/s)	Pϕ (MHz)									
	34		36		38		40		42	
	n	N	n	N	n	N	n	N	n	N
250										
500										
1000	3	132	3	140	3	147	3	155	3	163
2500	2	212	2	224	2	237	2	249	3	65
5000	2	105	2	112	2	118	2	124	2	130
10000	1	212	1	224	1	237	1	249	2	65
25000	1	84	1	89	1	94	1	99	1	104
50000	0	169	0	179	0	189	0	199	0	209
100000	0	84	0	89	0	94	0	99	0	104
250000	0	33	0	35	0	37	0	39	0	41
500000	0	16	0	17	0	18	0	19	0	20
1000000	—	—	0	8	—	—	0	9	—	—
2500000	—	—	—	—	—	—	0	3	—	—
5000000	—	—	—	—	—	—	0	1	—	—

Table 19.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode (4)

Bit Rate (bits/s)	Pϕ (MHz)							
	44		46		48		50	
	n	N	n	N	n	N	n	N
250								
500								
1000	3	171	3	179	3	187	3	194
2500	3	68	3	71	3	74	3	77
5000	2	137	2	143	2	149	2	155
10000	2	68	2	71	2	74	2	77
25000	1	109	1	114	1	119	1	124
50000	0	219	0	229	0	239	0	249
100000	0	109	0	114	0	119	0	124
250000	0	43	0	45	0	47	0	49
500000	0	21	0	22	0	23	0	24
1000000	0	10	—	—	0	11	—	—
2500000	—	—	—	—	—	—	0	4
5000000	—	—	—	—	—	—	—	—

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmission/reception is disabled.

Note: Settings with an error of 1% or less are recommended.

Table 19.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 19.7 indicates the maximum bit rates in clock synchronous mode when the baud rate generator is used. Tables 19.8 and 19.9 list the maximum rates for external clock input.

Table 19.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	Discontinuous Transmission/Reception			Continuous Transmission/Reception		
	Maximum Bit Rate	Settings		Maximum Bit Rate	Settings	
Pϕ (MHz)	(bits/s)	n	N	(bits/s)	n	N
10	312500	0	0	156250	0	1
12	375000	0	0	187500	0	1
14	437500	0	0	218750	0	1
16	500000	0	0	250000	0	1
18	562500	0	0	281250	0	1
20	625000	0	0	312500	0	1
22	687500	0	0	343750	0	1
24	750000	0	0	375000	0	1
26	812500	0	0	406250	0	1
28	875000	0	0	437500	0	1
30	937500	0	0	468750	0	1
32	1000000	0	0	500000	0	1
34	1062500	0	0	531250	0	1
36	1125000	0	0	562500	0	1
38	1187500	0	0	593750	0	1
40	1250000	0	0	625000	0	1
42	1312500	0	0	656250	0	1
44	1375000	0	0	687500	0	1
46	1437500	0	0	718750	0	1
48	1500000	0	0	750000	0	1
50	1562500	0	0	781250	0	1

Table 19.7 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Clock Synchronous Mode)

	Discontinuous Transmission/Reception			Continuous Transmission/Reception		
		Settings			Settings	
Pϕ (MHz)	Maximum Bit Rate (bits/s)	n	N	Maximum Bit Rate (bits/s)	n	N
10	2500000	0	0	1250000	0	1
12	3000000	0	0	1500000	0	1
14	3500000	0	0	1750000	0	1
16	4000000	0	0	2000000	0	1
18	4500000	0	0	2250000	0	1
20	5000000	0	0	2500000	0	1
22	5500000	0	0	2750000	0	1
24	6000000	0	0	3000000	0	1
26	6500000	0	0	3250000	0	1
28	7000000	0	0	3500000	0	1
30	7500000	0	0	3750000	0	1
32	8000000	0	0	4000000	0	1
34	8500000	0	0	4250000	0	1
36	9000000	0	0	4500000	0	1
38	9500000	0	0	4750000	0	1
40	10000000	0	0	5000000	0	1
42	10500000	0	0	5250000	0	1
44	11000000	0	0	5500000	0	1
46	11500000	0	0	5750000	0	1
48	12000000	0	0	6000000	0	1
50	12500000	0	0	6250000	0	1

Table 19.8 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10	2.5	156250
12	3.0	187500
14	3.5	218750
16	4.0	250000
18	4.5	281250
20	5.0	312500
22	5.5	343750
24	6.0	375000
26	6.5	406250
28	7.0	437500
30	7.5	468750
32	8.0	500000
34	8.5	531250
36	9.0	562500
38	9.5	593750
40	10.0	625000
42	10.5	656250
44	11.0	687500
46	11.5	718750
48	12.0	750000
50	12.5	781250

Table 19.9 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10	1.6667	1666666
12	2.0000	2000000
14	2.3333	2333333
16	2.6667	2666666
18	3.0000	3000000
20	3.3333	3333333
22	3.6667	3666666
24	4.0000	4000000
26	4.3333	4333333
28	4.6667	4666666
30	5.0000	5000000
32	5.3333	5333333
34	5.6667	5666666
36	6.0000	6000000
38	6.3333	6333333
40	6.6667	6666666
42	7.0000	7000000
44	7.3333	7333333
46	7.6667	7666666
48	8.0000	8000000
50	8.3333	8333333

19.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive FIFO data registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU. It is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTRG[1:0]		TTRG[1:0]		-	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger Number Set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register (SCFSR) when the RTRGS bit in the FIFO trigger control register (SCFTCR) is 0. The RDF flag is set to 1 when the quantity of receive data stored in the receive FIFO data register (SCFRDR) is increased more than the specified trigger number shown below. • Asynchronous mode: 00: 1; 01: 4; 10: 8; 11: 14 • Clocked synchronous mode: 00: 1; 01: 2; 10: 8; 11: 14 Setting this bit is valid when the RTRGS bit in SCFTCR is 0. When the RTRGS bit in SCFTCR is 1, the RFTC[4:0] bit setting in SCFTCR is valid.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger Number Set the quantity of remaining transmit data which sets the TDFE flag in the serial status register (SCFSR) when the TTRGS bit in the FIFO trigger control register (SCFTCR) is 0. The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the specified trigger number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of empty bytes in SCFTDR when the TDFE flag is set to 1. Setting this bit is valid when the TTRGS bit in SCFTCR is 0. When the TTRGS bit in SCFTCR is 1, the TFTC[4:0] bit setting in SCFTCR is valid.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TXD) and receive input pin (RXD) and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

19.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR).

It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU. SCFDR is initialized to H'0000 by a power on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]					-	-	-	R[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	T[4:0]	00000	R	Indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	R[4:0]	00000	R	Indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR is full of receive data.

19.3.11 Serial Port Register (SCSPTR)

SCSPTR controls input/output and data of pins multiplexed to SCIF function. Bits 3 and 2 can control input/output data of SCK pin. Bits 1 and 0 can input data from RXD pin and output data to TXD pin, so they control break of serial transmitting/receiving. To read the SCIF pin value, use the port register. For details on the port register, refer to section 23, I/O Ports.

The CPU can always read and write to SCSPTR. SCSPTR is initialized to H'0050 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SCKIO	SCKDT	SPB2IO	SPB2DT
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	SCKIO	0	R/W	SCK Port Input/Output Indicates input or output of the serial port SCK pin. When the SCK pin is actually used as a port outputting the SCKDT bit value, the CKE[1:0] bits in SCSCR should be cleared to 0. 0: SCKDT bit value not output to SCK pin 1: SCKDT bit value output to SCK pin

Bit	Bit Name	Initial Value	R/W	Description
2	SCKDT	0	R/W	SCK Port Data Indicates the input/output data of the serial port SCK pin. Input/output is specified by the SCKIO bit. For output, the SCKDT bit value is output to the SCK pin. The SCK pin status is read from the SCKDT bit regardless of the SCKIO bit setting. However, SCK input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level
1	SPB2IO	0	R/W	Serial Port Break Input/Output Indicates input or output of the serial port TXD pin. When the TXD pin is actually used as a port outputting the SPB2DT bit value, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value not output to TXD pin 1: SPB2DT bit value output to TXD pin
0	SPB2DT	0	R/W	Serial Port Break Data Indicates the input data of the RXD pin and the output data of the TXD pin used as serial ports. Input/output is specified by the SPB2IO bit. When the TXD pin is set to output, the SPB2DT bit value is output to the TXD pin. The RXD pin status is read from the SPB2DT bit regardless of the SPB2IO bit setting. However, RXD input and TXD output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

19.3.12 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1).

SCLSR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/(W)*	Overrun Error Indicates the occurrence of an overrun error. 0: Receiving is in progress or has ended normally*[1] [Clearing conditions] • ORER is cleared to 0 by a power-on reset • ORER is cleared to 0 when 0 is written after 1 is read from ORER. 1: An overrun error has occurred*[2] [Setting condition] • ORER is set to 1 when the next serial reception is completed while the receive FIFO is full of 16-byte receive data. Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains its previous value. 2. The receive FIFO data register (SCFRDR) retains the data before an overrun error has occurred, and the next received data is lost. When the ORER bit is set to 1, the SCIF cannot continue the next serial reception.

19.3.13 Serial Direction Control Register (SCSDCR)

The DIR bit in the serial direction control register (SCSDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	DIR	-	-	-
Initial value:	1	1	1	1	0	0	1	0
R/W:	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	DIR	0	R/W	Data Transfer Direction Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format. 0: SCFTDR contents are transmitted in LSB-first order. Received data is stored in SCFRDR in LSB-first. 1: SCFTDR contents are transmitted in MSB-first order. Received data is stored in SCFRDR in MSB-first.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

19.3.14 FIFO Trigger Control Register (SCFTCR)

SCFTCR is a 16-bit register that sets the FIFO trigger. SCFTCR can always be read and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTRGS	-	-	RFTC[4:0]					TTRGS	-	-	TFTC[4:0]				
Initial value:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RTRGS	0	R/W	Receive Trigger Select Selects a method to set the receive trigger number. 0: The RTRG[1:0] bits in the FIFO control register (SCFCR) set the receive trigger number. 1: The RFTC[4:0] bits in the FIFO trigger control register (SCFTCR) set the receive trigger number.
14, 13	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	RFTC[4:0]	All 1	R/W	Receive FIFO Data Trigger Number Specifies the number of received data units based on which the receive data full (RDF) flag in the serial status register (SCFSR) is set when the RTRGS bit in the FIFO trigger control register (SCFTCR) is 1. When the number of received data units in the receive FIFO data register (SCFRDR) reaches the specified trigger number, the RDF flag is set to 1. The number of received data units is 1 when these bits are H'01, and 16 when H'10. Do not set these bits to H'00 or H'11 to H'1F. Setting these bits is valid when the RTRGS bit in SCFTCR is 1. When the RTRGS bit in SCFTCR is 0, the RTRG[1:0] bit setting in SCFCR is valid.

Bit	Bit Name	Initial Value	R/W	Description
7	TTRGS	0	R/W	Transmit Trigger Select Selects a method to set the transmit trigger number. 0: The TTRG[1:0] bits in the FIFO control register (SCFCR) set the transmit trigger number. 1: The TFTC[4:0] bits in the FIFO trigger control register (SCFTCR) set the transmit trigger number.
6, 5	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	TFTC[4:0]	All 1	R/W	Transmit FIFO Data Trigger Number Specifies the number of data units to be transmitted based on which sets the TDFE flag in the serial status register (SCFSR) is set when the TTRGS bit in the FIFO trigger control register (SCFTCR) is 1. When the number of data units to be transmitted in the transmit FIFO data register (SCFTDR) reaches the specified trigger number, the TDFE flag is set to 1. The transmit data trigger number is 0 when these bits are H'00, and 15 when H'0F. Do not set these bits to H'10 to H'1F. Setting these bits is valid when the TTRGS bit in SCFTCR is 1. When the TTRGS bit in SCFTCR is 0, the TTRG[1:0] bit setting in SCFCR is valid.

19.4 Operation

19.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clocked synchronous mode in which communication is synchronized with clock pulses.

The SCIF has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 19.10. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 19.11.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of on-chip baud rate generator and can output the clock having a frequency 16 times the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clocked Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, the SCIF operates on the input synchronous clock not using the on-chip baud rate generator.

Table 19.10 SCSMR Settings and SCIF Communication Formats

SCSMR					SCIF Communication Format		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous	8 bits	Not set	None

[Legend]

x: Don't care

Table 19.11 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR				
Bit 7	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin. The SCK pin state depends on the SCKIO and SCKDT bits in SCSPTR.
		1			Outputs a clock with a frequency 16 times the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
		1		Setting prohibited	
1	0	x	Clocked synchronous	Internal	Outputs the synchronous clock
	1	0		External	Inputs the synchronous clock
		1		Setting prohibited	

[Legend]

x: Don't care

19.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCIF are independent, so full duplex communication is possible. The transmitter and receiver are 16-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 19.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIF monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first when LSB-first transfer is selected), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the start bit. The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

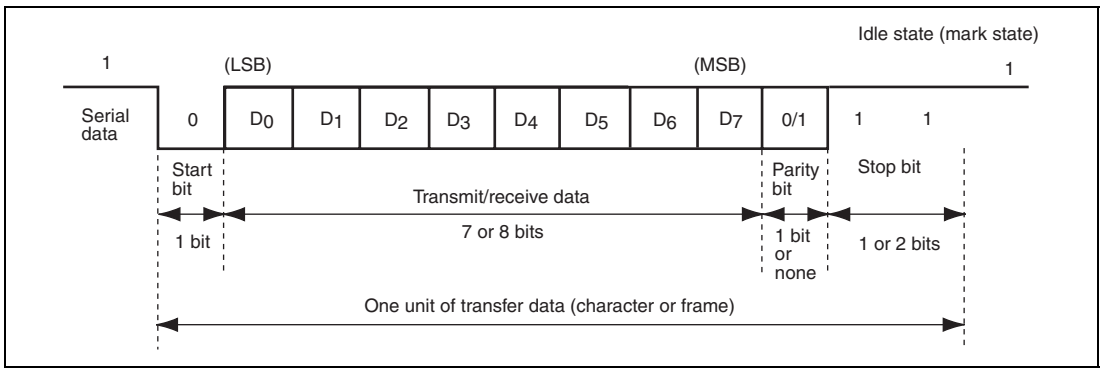


Figure 19.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits when LSB-First Transfer is Selected)

(1) Transmit/Receive Formats

Table 19.12 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 19.12 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data								STOP		
0	0	1	START	8-bit data								STOP	STOP	
0	1	0	START	8-bit data								P	STOP	
0	1	1	START	8-bit data								P	STOP	STOP
1	0	0	START	7-bit data							STOP			
1	0	1	START	7-bit data							STOP	STOP		
1	1	0	START	7-bit data							P	STOP		
1	1	1	START	7-bit data							P	STOP	STOP	

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE[1:0] in the serial control register (SCSCR). For clock source selection, refer to table 19.11.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16 times the desired bit rate.

(3) Transmitting and Receiving Data

- SCIF Initialization (Asynchronous Mode)

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the operating mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCFSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 19.3 shows a sample flowchart for initializing the SCIF.

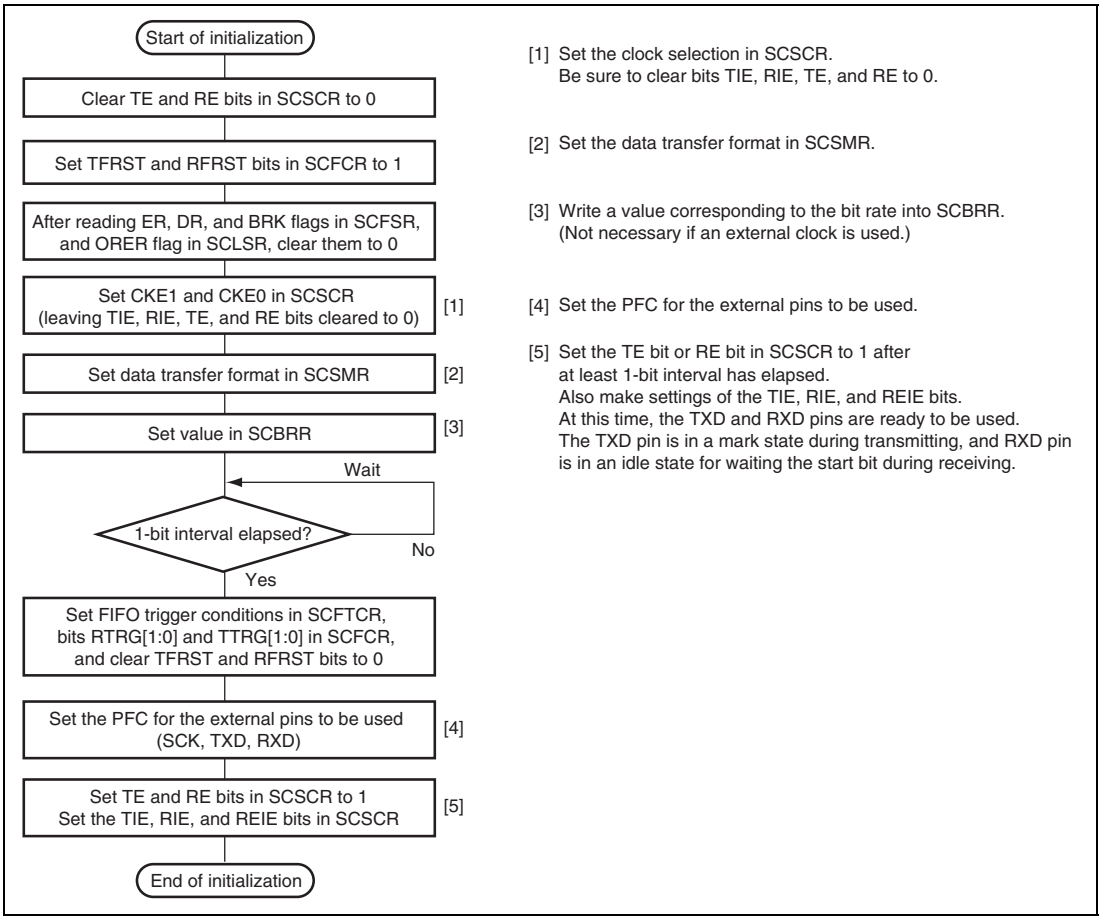


Figure 19.3 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Asynchronous Mode)

Figure 19.4 shows a sample flowchart for serial transmission. Use the following procedure for serial data transmission after enabling the SCIF for transmission.

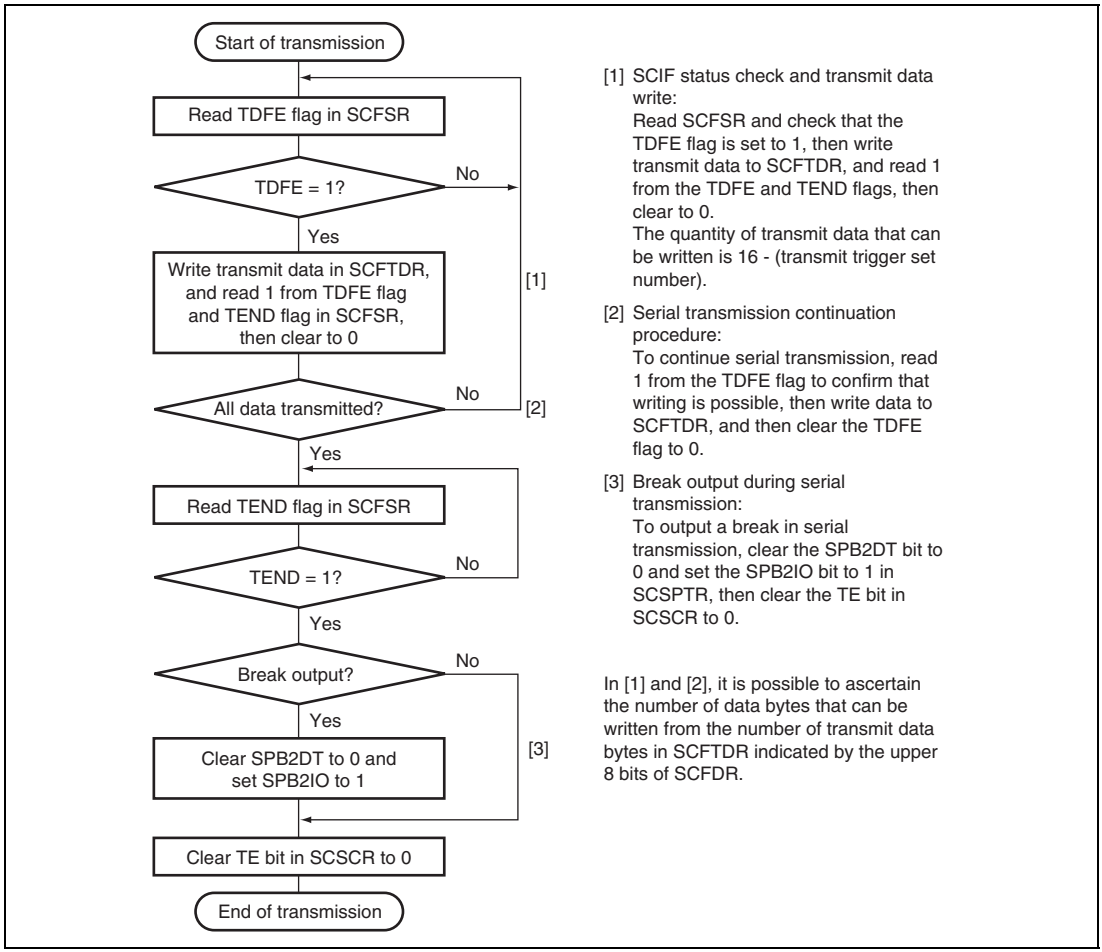


Figure 19.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmission. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – specified transmit trigger number).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR)or FIFO trigger control register (SCFTCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TXD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order (when LSB-first transfer is selected).
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started. If there is no data to be transmitted, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then mark state is entered in which 1 is output continuously.

Figure 19.5 shows an example of the operation for transmission in asynchronous mode.

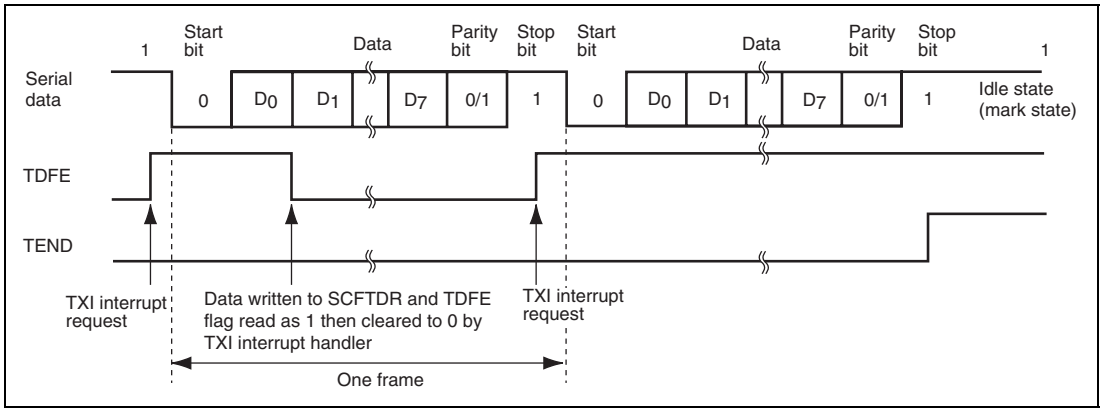
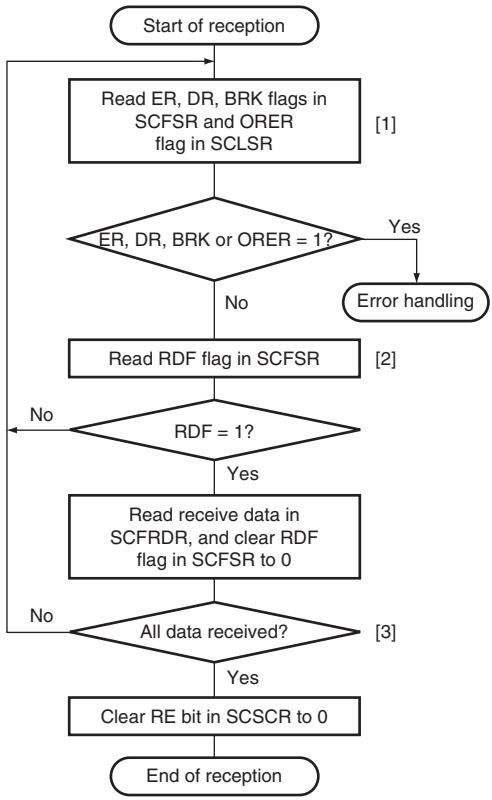


Figure 19.5 Example of Transmit Operation
(8-Bit Data with Parity and One Stop Bit when LSB-First Transfer is Selected)

- Receiving Serial Data (Asynchronous Mode)

Figures 19.6 and 19.7 show sample flowcharts for serial reception. Use the following procedure for serial data reception after enabling the SCIF for reception.



[1] Receive error handling and break detection:

Read the DR, ER, and BRK flags in SCFSR, and the ORER flag in SCLSR, to identify any error, perform the appropriate error handling, then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

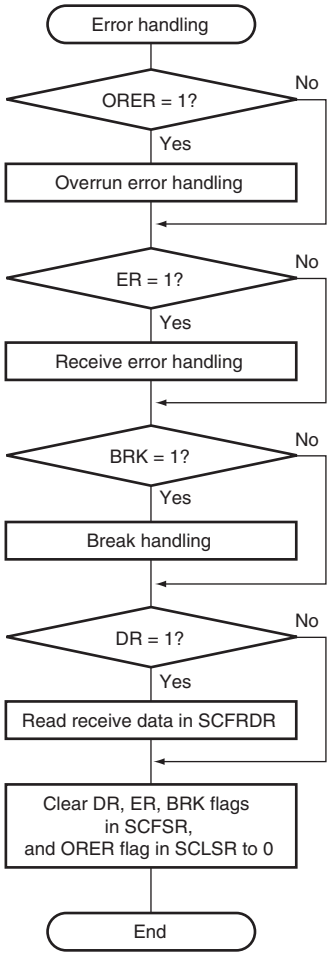
[2] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading lower bits of SCFDR.

Figure 19.6 Sample Flowchart for Receiving Serial Data (1)



- Whether a framing error or parity error has occurred in the receive data that is to be read from the receive FIFO data register (SCFRDR) can be ascertained from the FER and PER bits in the serial status register (SCFSR).
- When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 19.7 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the communication line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order (when LSB-first transfer is selected).
3. The parity bit and stop bit are received.
   After receiving these bits, the SCIF carries out the following checks.
   A. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   B. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
   C. Overrun error check: The SCIF checks that the ORER flag is 0, indicating that the overrun error has not occurred.
   D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCFRDR.

   Note: When a parity error or a framing error occurs, reception is not suspended.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 19.8 shows an example of the operation for reception in asynchronous mode.

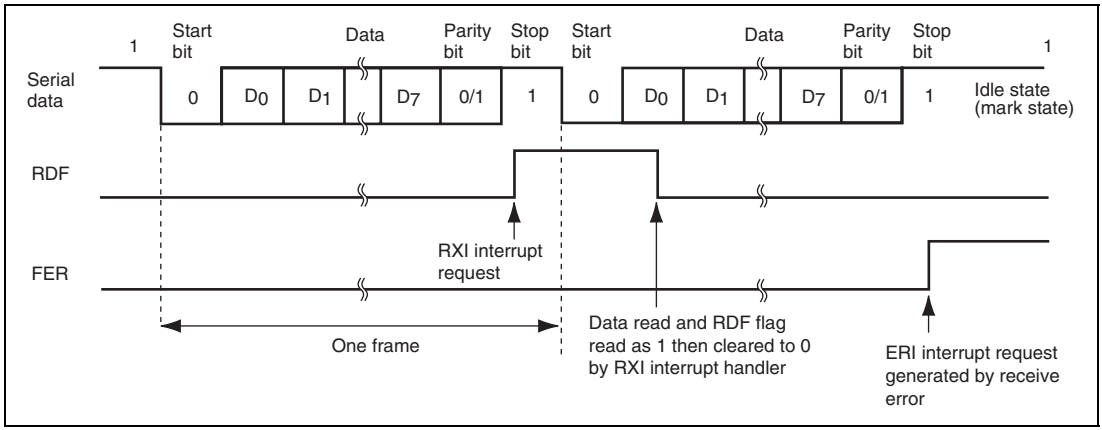


Figure 19.8 Example of SCIF Receive Operation (8-Bit Data with Parity and One Stop Bit when LSB-First Transfer is Selected)

19.4.3 Operation in Clocked Synchronous Mode

In clocked synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 19.9 shows the general format in clocked synchronous serial communication.

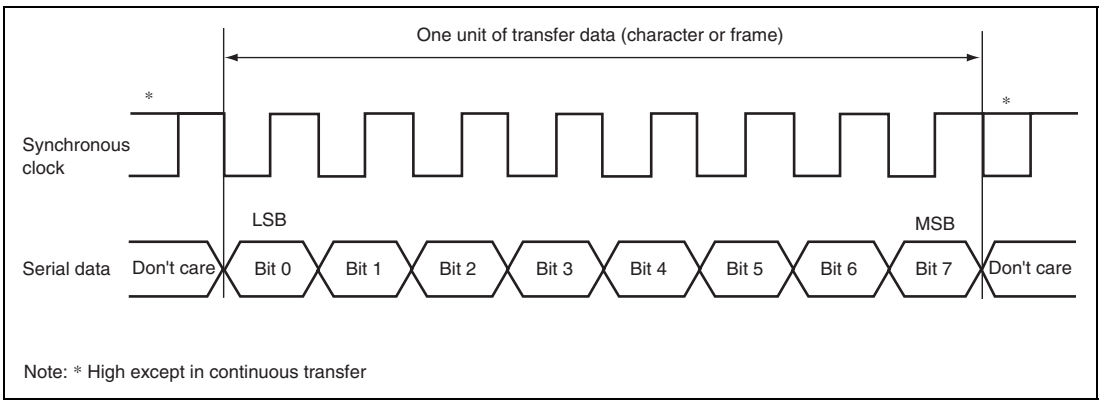


Figure 19.9 Data Format in Clocked Synchronous Communication (when LSB-First Transfer is Selected)

In clocked synchronous serial communication, each data bit is output on the communication line from one falling edge of the synchronous clock to the next. Data is guaranteed valid at the rising edge of the synchronous clock.

In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB (when LSB-first transfer is selected).

In clocked synchronous mode, the SCIF receives data by synchronizing with the rising edge of the synchronous clock.

(1) Transmit/Receive Formats

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator by the setting of the C/$\overline{A}$ bit in SCSMR and CKE[1:0] in SCSCR, or an external synchronous clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

When the SCIF operates on an internal clock, it outputs the synchronous clock signal at the SCK pin. Eight synchronous clock pulses are output per transmitted or received character. When the SCIF is not transmitting or receiving, the synchronous clock signal remains in the high state. When only receiving, the internal clock signal outputs while the RE bit in SCSCR is 1 until the number of data units in the receive FIFO reaches the receive FIFO data trigger number. In this case, 136 synchronous clock pulses ($8 \times (16 + 1)$) are output. To receive n characters, use the external clock as the clock source. When the internal clock is used, RE and TE should be set to 1, and dummy data transmission of n characters and reception of n characters should be performed simultaneously.

(3) Transmitting and Receiving Data

- SCIF Initialization (Clocked Synchronous Mode)

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and the receive FIFO data register (SCFRDR), which retain their previous contents.

Figure 19.10 shows a sample flowchart for initializing the SCIF.

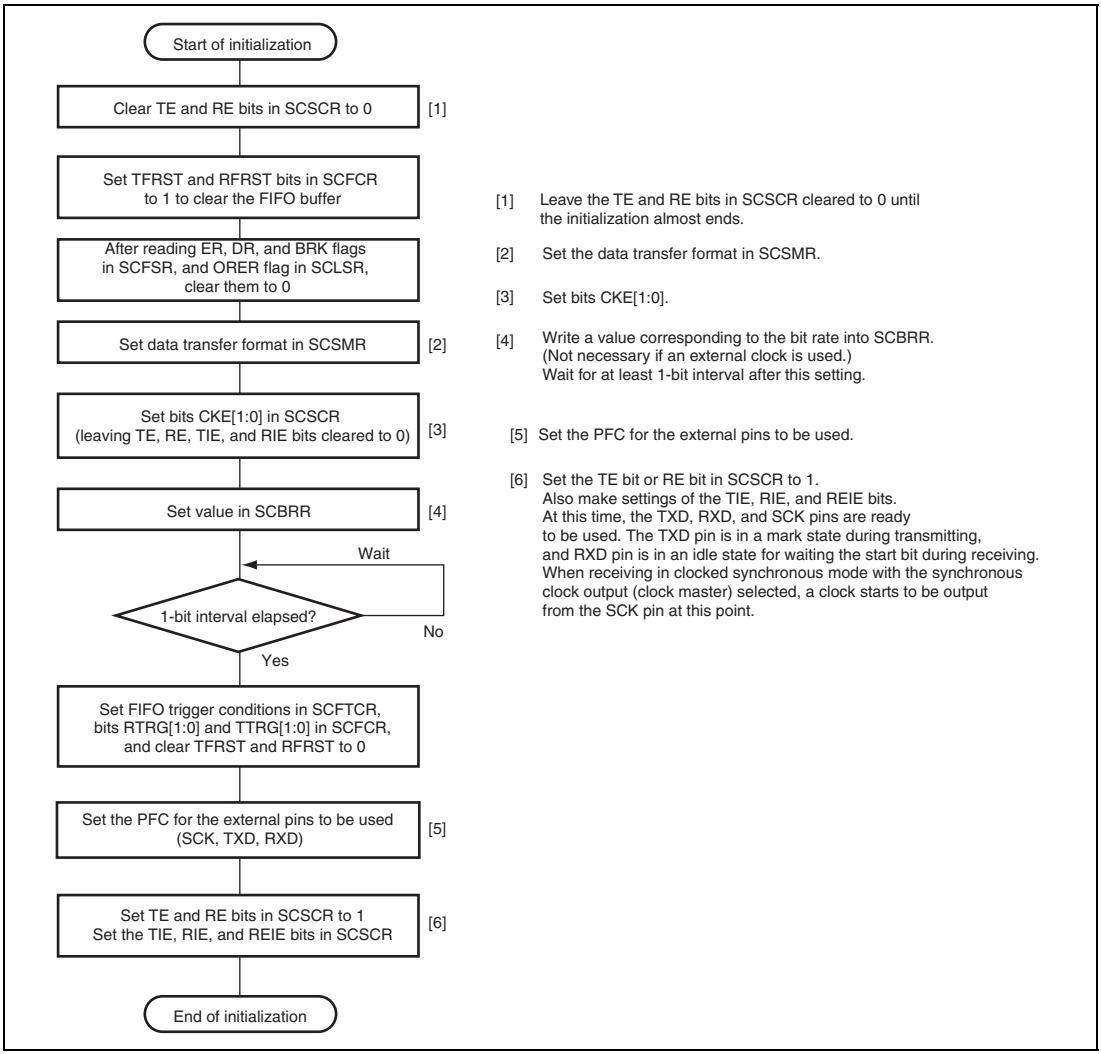


Figure 19.10 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Clocked Synchronous Mode)

Figure 19.11 shows a sample flowchart for transmitting serial data. Use the following procedure for serial data transmission after enabling the SCIF for transmission.

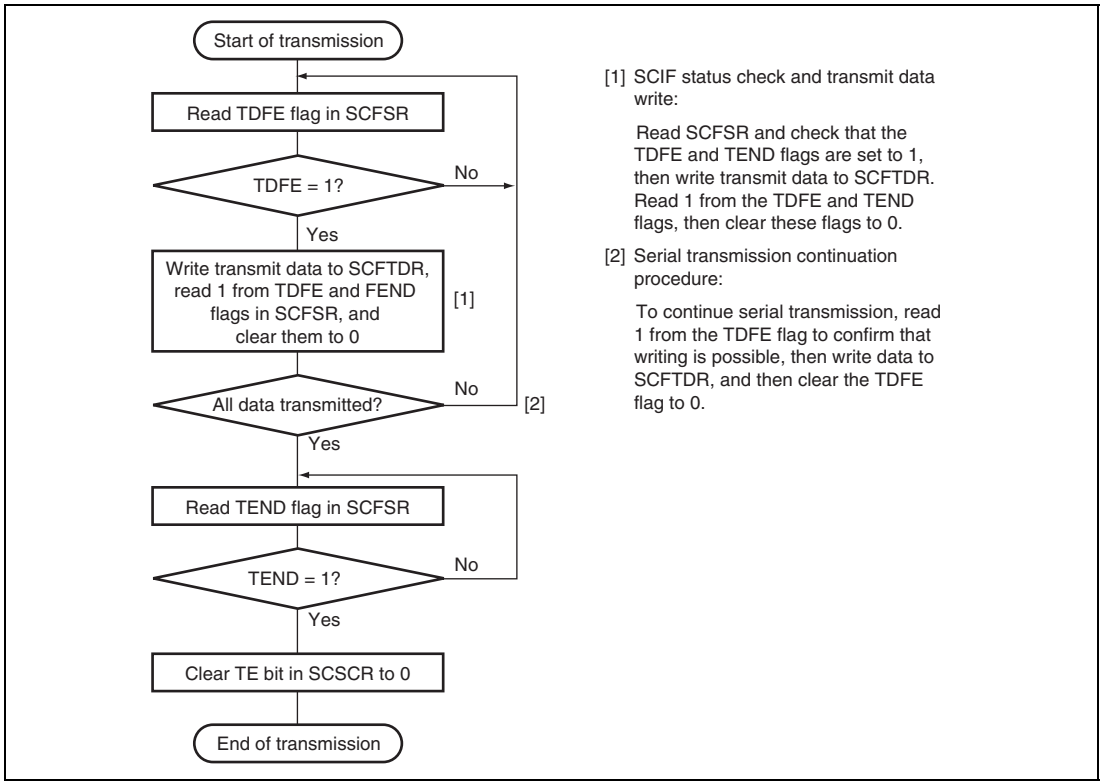


Figure 19.11 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmission. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – specified transmit trigger number).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR) or FIFO trigger control register (SCFTCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If an external clock source is selected, the SCIF outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7) (when LSB-first transfer is selected).
3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no data, the TXD pin holds the state after the TEND flag in SCFSR is set to 1 and the MSB (bit 7) is sent.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 19.12 shows an example of SCIF transmit operation.

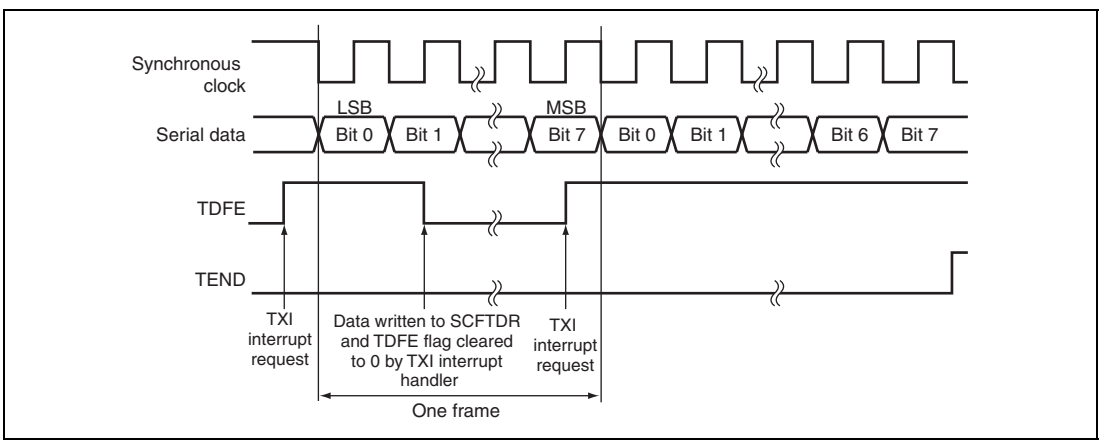


Figure 19.12 Example of SCIF Transmit Operation (when LSB-First Transfer is Selected)

- Receiving Serial Data (Clocked Synchronous Mode)

Figures 19.13 and 19.14 show sample flowcharts for receiving serial data. Use the following procedure for serial data reception after enabling the SCIF for reception. When switching from asynchronous mode to clocked synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.

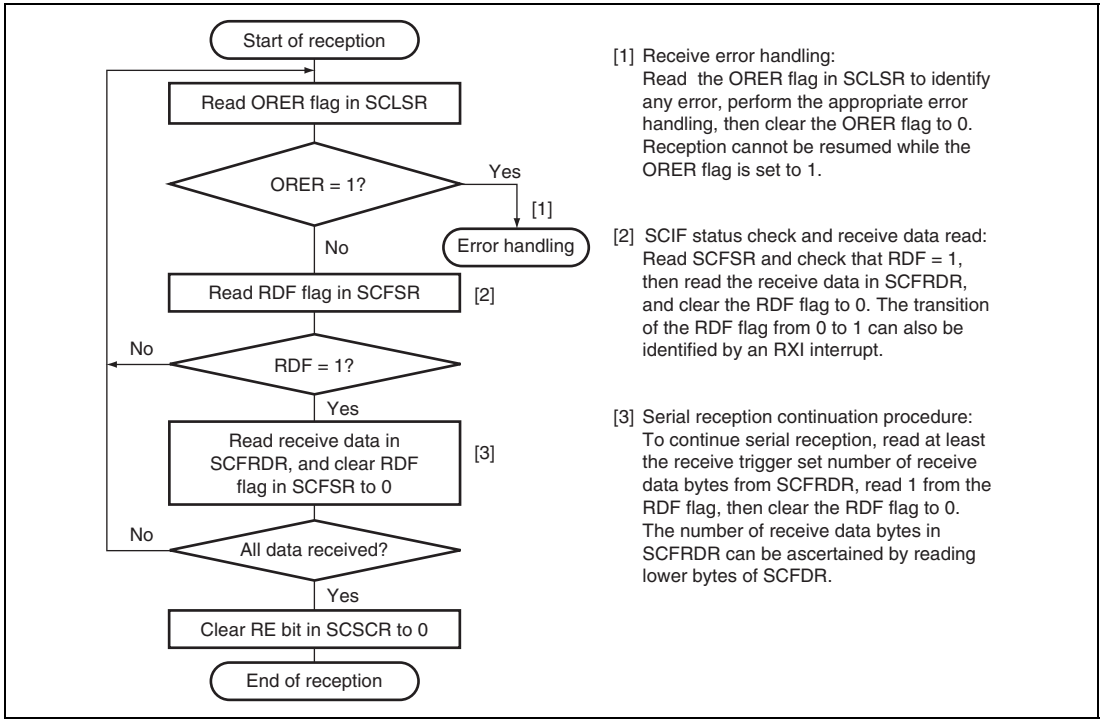


Figure 19.13 Sample Flowchart for Receiving Serial Data (1)

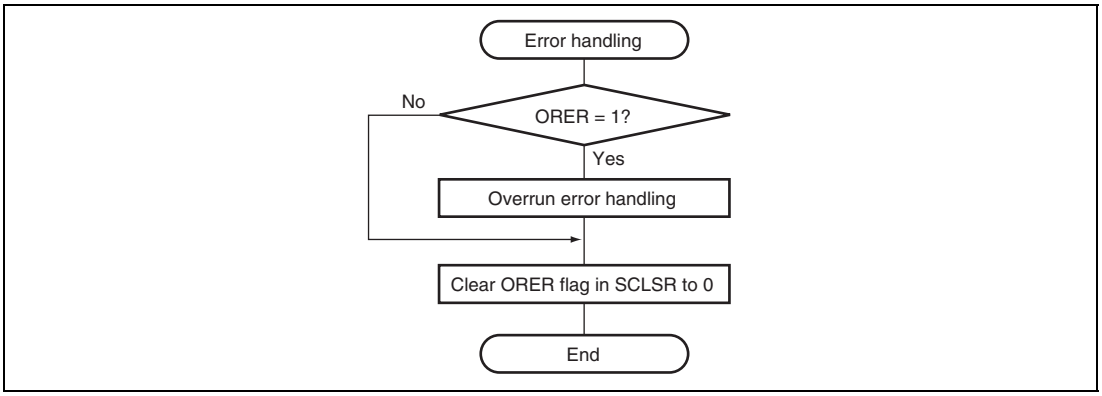


Figure 19.14 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF synchronizes with synchronous clock input or output and starts reception.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB (when LSB-first transfer is selected). After receiving the data, the SCIF checks whether the receive data can be loaded from SCRSR into SCFRDR or not. If this check is passed, the RDF flag is set to 1 and the SCIF stores the received data in SCFRDR. If the check is not passed (an overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if RIE in the serial control register (SCSCR) is set to 1, the SCIF requests a receive-data-full interrupt (RXI). If the ORER flag is set to 1 and RIE or REIE in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 19.15 shows an example of SCIF receive operation.

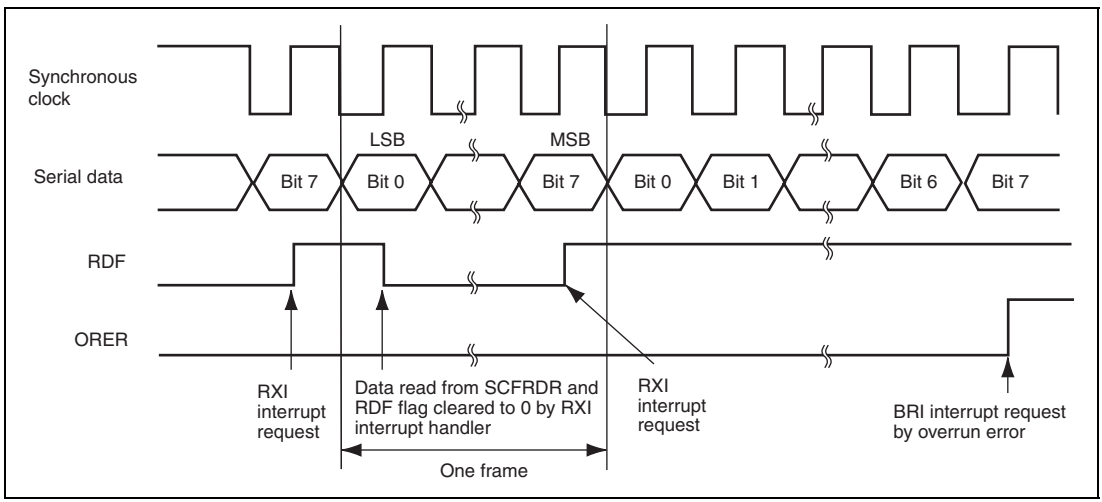
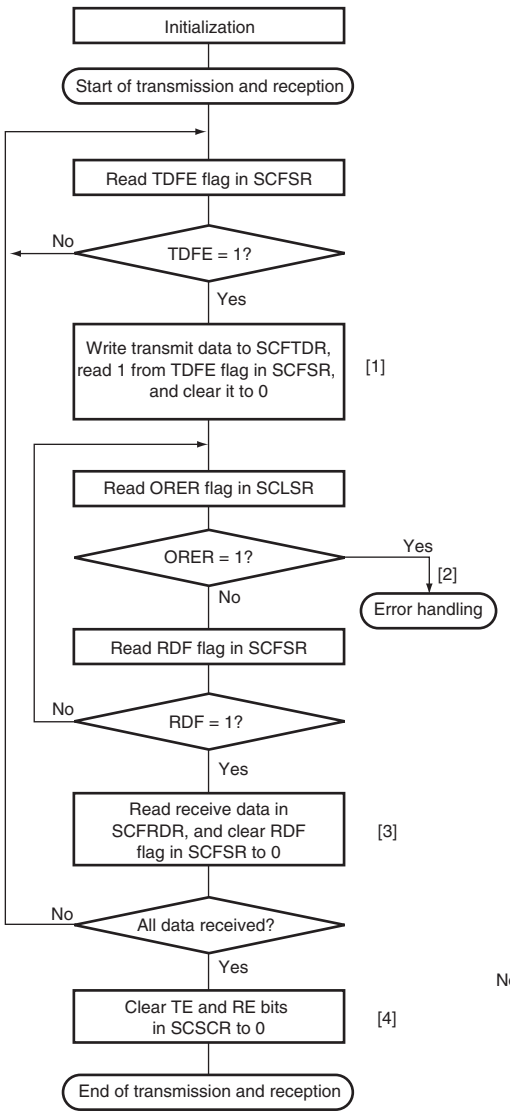


Figure 19.15 Example of SCIF Receive Operation (when LSB-First Transfer is Selected)

- Transmitting and Receiving Serial Data Simultaneously (Clocked Synchronous Mode)

Figure 19.16 shows a sample flowchart for transmitting and receiving serial data simultaneously. Use the following procedure for the simultaneous transmission/reception of serial data, after enabling the SCIF for transmission/reception.



[1] SCIF status check and transmit data write:

Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR. Read 1 from the TDFE flag, then clear this flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission and reception continuation procedure:

To continue serial transmission and reception, read 1 from the RDF flag and the receive data in SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 19.16 Sample Flowchart for Transmitting/Receiving Serial Data

19.5 Interrupt Sources and DMAC/DTC

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-FIFO-data-full (RXI), and break (BRI).

Table 19.13 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the TIE, RIE, and REIE bits in SCSCR and are separately issued to the interrupt controller.

When the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated. This request can be used to activate the direct memory controller (DMAC) or data transfer controller (DTC) to transfer data. When the DMAC is used for data transfer and data is written to the transmit FIFO data register (SCFTDR), the TDFE flag is automatically cleared to 0, and thus a TXI interrupt request is not issued to the CPU. When the DTC is used for data transfer and data is written to SCFTDR, the TDFE flag is automatically cleared to 0, and thus a TXI interrupt request is not issued to the CPU if the DISEL bit of the DTC is 0 and the transfer counter value is not 0. However, if both the DISEL bit and the transfer counter value are 0 or if the DISEL bit is 1, the TDFE flag is not cleared to 0 even though data is written to SCFTDR, and thus a TXI interrupt request is issued to the CPU after data write to SCFTDR.

When the RDF or DR flag in SCFSR is set to 1, an RXI interrupt request is generated. This request can be used to activate the DMAC or DTC to transfer data. When the DMAC is used for data transfer and data is read from the receive FIFO data register (SCFRDR), the RDF flag is automatically cleared to 0, and thus an RXI interrupt request is not issued to the CPU. When the DTC is used for data transfer and data is read from SCFRDR, the RDF flag is automatically cleared to 0, and thus an RXI interrupt request is not issued to the CPU if the DISEL bit of the DTC is 0 and the transfer counter value is not 0. However, if both the DISEL bit and the transfer counter value are 0 or if the DISEL bit is 1, the RDF flag is not cleared to 0 even though data is read from SCFRDR, and thus an RXI interrupt request is issued to the CPU after data read from SCFRDR. In addition, an RXI interrupt request by setting the DR flag to 1 is generated only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated. This request cannot be used to activate the DMAC or DTC.

When the ER flag in SCSCR is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the DMAC or DTC.

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests only an ERI and BRI interrupts without requesting an RXI interrupt.

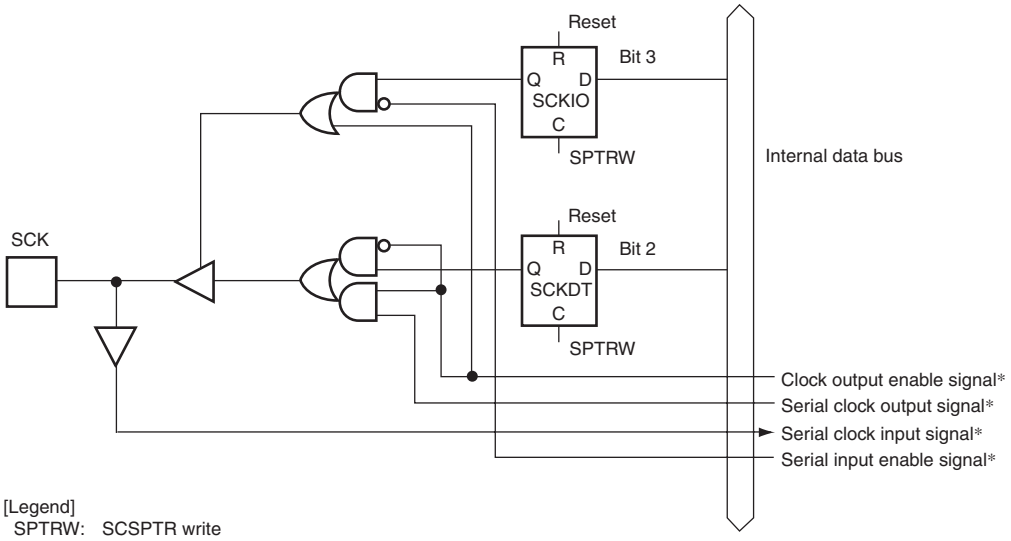
The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR.

Table 19.13 SCIF Interrupt Sources

Interrupt Source	Description	Interrupt Enable Bit	DMAC/DTC Activation	Priority
BRI	Interrupt caused by break (BRK) or overrun (ORER)	RIE or REIE	Not possible	High
ERI	Interrupt caused by receive error (ER)	RIE or REIE	Not possible	
RXI	Interrupt caused by receive FIFO data full (RDF) or data ready (DR)	RIE	Possible	
TXI	Interrupt caused by transmit FIFO data empty (TDFE)	TIE	Possible	Low

19.6 Serial Port Register (SCSPTR) and SCIF Pins

The relationship between SCSPTR and the SCIF pins is shown in figures 19.17 and 19.18.



Note: * These signals control the SCK pin according to the settings of the C/$\overline{A}$ bit in SCSMR and bits CKE1 and CKE0 in SCSCR.

Figure 19.17 SCKIO Bit, SCKDT Bit, and SCK Pin

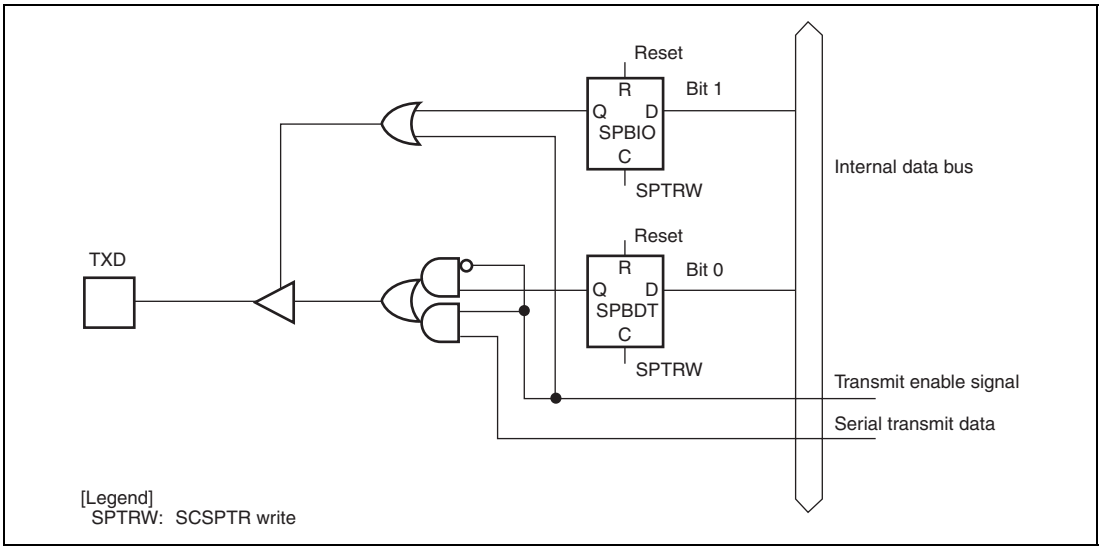


Figure 19.18 SPB2IO Bit, SPB2DT Bit, and TXD Pin

19.7 Usage Notes

Note the following when using the SCIF.

19.7.1 SCFTDR Writing and TDFE Flag

The TDFE flag in the serial status register (SCFSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG[1:0] in the FIFO control register (SCFCR) or bits TFTC[4:0] in the FIFO trigger control register (SCFTCR). After the TDFE flag is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE flag clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the FIFO data count register (SCFDR).

19.7.2 SCFRDR Reading and RDF Flag

The RDF flag in the serial status register (SCFSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG[1:0] in the FIFO control register (SCFCR)or bits RFTC[4:0] bits in the FIFO trigger control register (SCFTCR). After RDF flag is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR exceeds the receive trigger number, the RDF flag will be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after reading the number of the received data in the receive FIFO data register (SCFRDR) which is less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

19.7.3 Restriction on DMAC and DTC Usage

When the DMAC or DTC writes data to SCFTDR due to a TXI interrupt request, the state of the TEND flag becomes undefined. Therefore, the TEND flag should not be used as the transfer end flag in such a case.

19.7.4 Break Detection and Processing

Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate.

19.7.5 Sending a Break Signal

The I/O condition and level of the TXD pin are determined by the SPB2IO and SPB2DT bits in the serial port register (SCSPTR). This feature can be used to send a break signal.

Until the TE bit is set to 1 (enabling transmission) after initializing, the TXD pin does not work. During the period, mark state is performed by the SPB2DT bit. Therefore, the SPB2IO and SPB2DT bits should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TXD pin.

19.7.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 19.19.

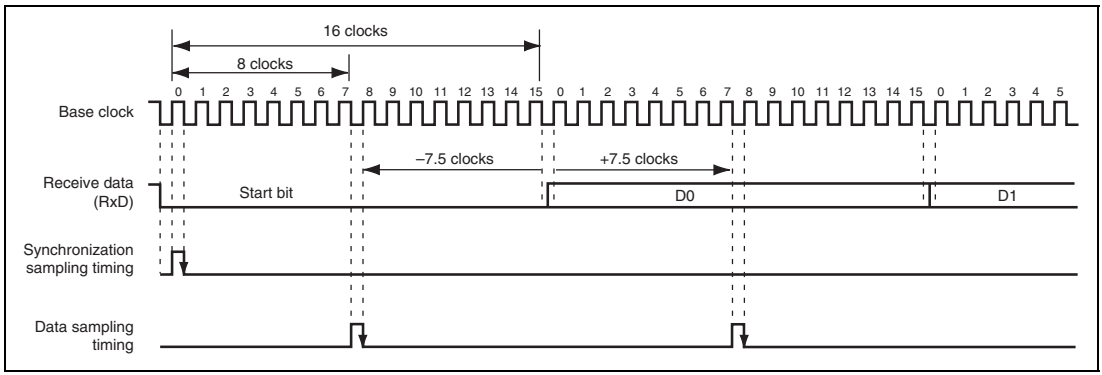


Figure 19.19 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ \%$$

Where:
- M: Receive margin (%)
- N: Ratio of clock frequency to bit rate (N = 16)
- D: Clock duty (D = 0 to 1.0)
- L: Frame length (L = 9 to 12)
- F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$M = (0.5 - 1/(2 \times 16)) \times 100\%$

$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

19.7.7 FER Flag and PER Flag in Serial Status Register (SCFSR)

The FER flag and PER flag in the serial status register (SCFSR) are status flags that apply to next entry to be read from the receive FIFO data register (SCFRDR). After the CPU or DMAC/DTC reads the receive FIFO data register, the flags of framing errors and parity errors will disappear.

To check the received data for the states of framing errors and parity errors, only read the receive FIFO data register after reading the serial status register.

19.7.8 Note on Using External Clock in Clocked Synchronous Mode

TE and RE must be set to 1 after waiting for four or more cycles of the peripheral operating clock after the external clock (SCK) is changed from 0 (low) to 1 (high). TE and RE must be set to 1 only while the external clock (SCK) is 1 (high).

19.7.9 Module Standby Mode Setting

SCIF operation can be disabled or enabled using the standby control register. The initial setting is for SCIF operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 32, Power-Down Modes.

Section 20 I²C Bus Interface 3 (IIC3)

The I²C bus interface 3 conforms to and provides a subset of the Philips I²C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I²C bus differs partly from the Philips register configuration.

20.1 Features

- Selection of I²C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Module standby mode settable

I²C bus format:

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- The direct memory access controller (DMAC) or data transfer controller (DTC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Clocked synchronous serial format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error
- The direct memory access controller (DMAC)) or data transfer controller (DTC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

Figure 20.1 shows a block diagram of the I²C bus interface 3.

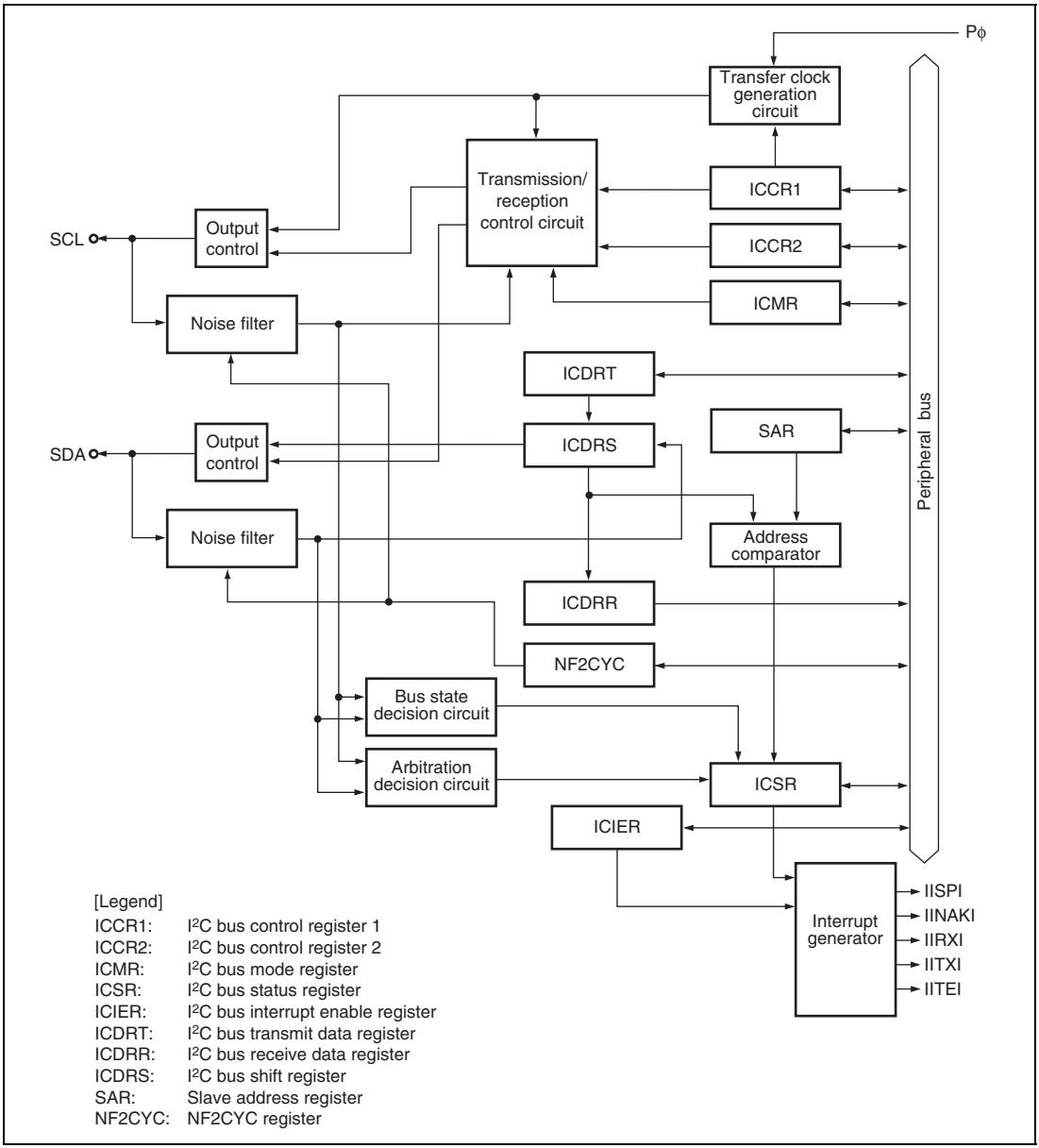


Figure 20.1 Block Diagram of I²C Bus Interface 3

20.2 Input/Output Pins

Table 20.1 shows the pin configuration of the I²C bus interface 3.

Table 20.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock	SCL	I/O	I²C serial clock input/output
Serial data	SDA	I/O	I²C serial data input/output

Figure 20.2 shows an example of I/O pin connections to external circuits.

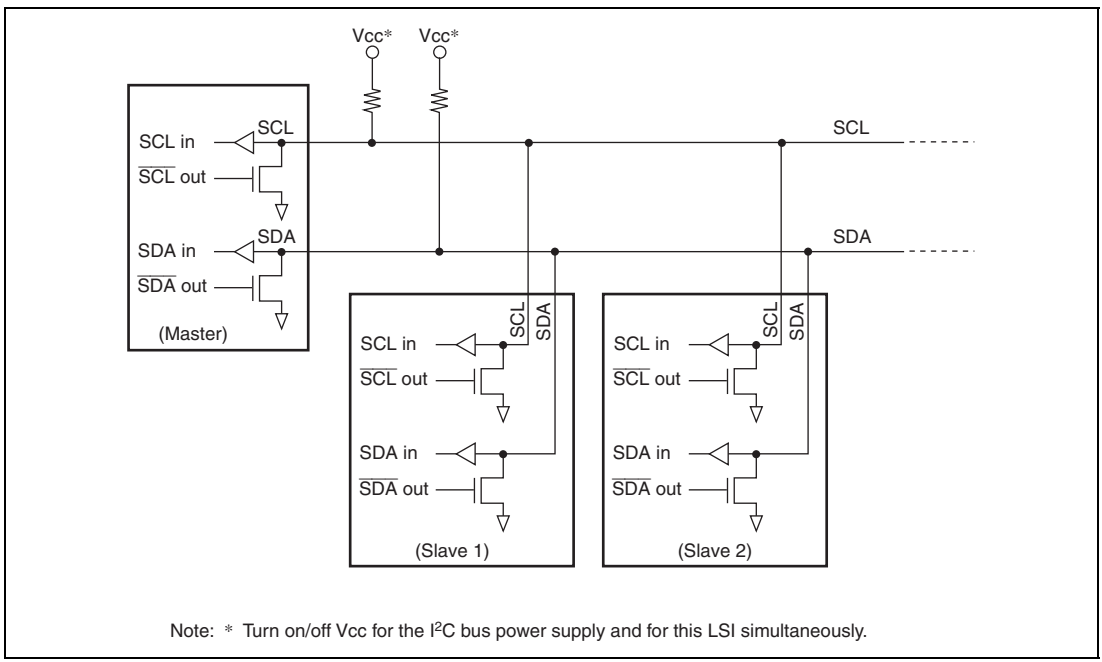


Figure 20.2 External Circuit Connections of I/O Pins

20.3 Register Descriptions

The I²C bus interface 3 has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 34, List of Registers.

Table 20.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
I²C bus control register 1	ICCR1	R/W	H'00	H'FFFEE000	8
I²C bus control register 2	ICCR2	R/W	H'7D	H'FFFEE001	8
I²C bus mode register	ICMR	R/W	H'38	H'FFFEE002	8
I²C bus interrupt enable register	ICIER	R/W	H'00	H'FFFEE003	8
I²C bus status register	ICSR	R/W	H'00	H'FFFEE004	8
Slave address register	SAR	R/W	H'00	H'FFFEE005	8
I²C bus transmit data register	ICDRT	R/W	H'FF	H'FFFEE006	8
I²C bus receive data register	ICDRR	R	H'FF	H'FFFEE007	8
NF2CYC register	NF2CYC	R/W	H'00	H'FFFEE008	8

20.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 3, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

ICCR1 is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface 3 Enable 0: Output from SCL and SDA is disabled. (Input to SCL and SDA enabled.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable Enables or disables the consecutive receive operation without reading ICDRR while TRS = 0. In master receive mode, when ICDRR cannot be read before the rising edge of the 8th clock of SCL, set RCVD to 1 so that data is received in byte units. 0: Enables next reception 1: Disables next reception

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the 8th bit is set to 1, TRS is automatically set to 1. If an overrun error occurs in master receive mode with the clocked synchronous serial format, MST is cleared and the mode changes to slave receive mode. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST = 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 20.3) in master mode.

Table 20.3 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate					
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 10 MHz	Pϕ = 12.5 MHz	Pϕ = 20 MHz	Pϕ = 25 MHz	Pϕ = 40 MHz	Pϕ = 50 MHz
0	0	0	0	Pϕ/64	156 kHz	195 kHz	313 kHz	391 kHz	625 kHz	781 kHz
0	0	0	1	Pϕ/72	139 kHz	174 kHz	278 kHz	347 kHz	556 kHz	694 kHz
0	0	1	0	Pϕ/84	119 kHz	149 kHz	238 kHz	298 kHz	476 kHz	595 kHz
0	0	1	1	Pϕ/92	109 kHz	136 kHz	217 kHz	272 kHz	435 kHz	543 kHz
0	1	0	0	Pϕ/100	100 kHz	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz
0	1	0	1	Pϕ/108	92.6 kHz	116 kHz	185 kHz	231 kHz	370 kHz	463 kHz
0	1	1	0	Pϕ/120	83.3 kHz	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz
0	1	1	1	Pϕ/124	80.6 kHz	101 kHz	161 kHz	202 kHz	323 kHz	403 kHz
1	0	0	0	Pϕ/256	39.1 kHz	48.8 kHz	78.1 kHz	97.7 kHz	156 kHz	195 kHz
1	0	0	1	Pϕ/288	34.7 kHz	43.4 kHz	69.4 kHz	86.8 kHz	139 kHz	174 kHz
1	0	1	0	Pϕ/336	29.8 kHz	37.2 kHz	59.5 kHz	74.4 kHz	119 kHz	149 kHz
1	0	1	1	Pϕ/368	27.2 kHz	34.0 kHz	54.3 kHz	67.9 kHz	109 kHz	136 kHz
1	1	0	0	Pϕ/400	25.0 kHz	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
1	1	0	1	Pϕ/432	23.1 kHz	28.9 kHz	46.3 kHz	57.9 kHz	92.6 kHz	116 kHz
1	1	1	0	Pϕ/480	20.8 kHz	26.0 kHz	41.7 kHz	52.1 kHz	83.3 kHz	104 kHz
1	1	1	1	Pϕ/496	20.2 kHz	25.2 kHz	40.3 kHz	50.4 kHz	80.6 kHz	101 kHz

Note: The settings should satisfy external specifications.

20.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus.

ICCR2 is initialized to H'7D by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	RX TIMES	IICRST	-
Initial value:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clocked synchronous serial format, this bit is always read as 0. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	R/W	Start/Stop Issue Condition Disable Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. Even if 1 is written to this bit, the data will not be stored.

Bit	Bit Name	Initial Value	R/W	Description
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).
4	SDAOP	1	R/W	SDAO Write Protect Controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0. This bit is always read as 1.
3	SCLO	1	R	SCL Output Level Monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	RXTIMES	1	R/W	IIRXI Interrupt Timing Select Selects the period at high level or the period at low level of the ninth cycle of SCL as the timing for generation of the IIRXI interrupt. 0: The period at low level of the ninth cycle of SCL 1: The period at high level of the ninth cycle of SCL
1	IICRST	0	R/W	IIC Control Part Reset IICRST resets the BC[2:0] bits in ICMR and the internal circuits of the IIC3 module. If the I²C bus hangs during operations due to a failure in transfer, etc., the BC[2:0] bits in ICMR and the internal circuits of the IIC3 module can be reset by setting the IICRST bit to 1.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Note: When 1 is written to the IICRST bit in ICCR2, the state becomes as follows.

- The SDAO and SCLO bits in ICCR2 are set to 1.
- If the module is in master transmit mode or slave transmit mode, the TDRE bit in ICSR is set to 1.
- Writing to the BBSY, SCP, and SDAO bits in ICCR2 is invalid while a reset is being applied by writing 1 to IICRST.
- Writing 1 to IICRST does not clear the BBSY bit in ICCR2 to 0. However, if the states of the SCL and SDA pins lead to the generation of a stop condition, (rising edge on SDA while SCL is at the high level), the BBSY bit may be cleared to 0 as a result. This can also affect other bits in the same way.
- Data transfer stops while a reset is being applied by writing 1 to IICRST. However, functions for detecting start conditions, stop conditions, and failure in bus contention continue to operate. Signals input to the SCL and SDA pins may alter the states of ICCR1, ICCR2, and ICSR.

20.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, and selects the transfer bit count.

ICMR is initialized to H'38 by a power-on reset. Bits BC[2:0] are initialized to H'0 by the IICRST bit in ICCR2.

Bit:	7	6	5	4	3	2	1	0
	MLS	-	-	-	BCWP	BC[2:0]		
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the BC[2:0] modifications. When modifying the BC[2:0] bits, this bit should be cleared to 0. In clocked synchronous serial mode, the BC[2:0] bits should not be modified. 0: When writing, values of the BC[2:0] bits are set. 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. Should be made between transfer frames. If these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. After the stop condition is detected, the value of these bits returns automatically to B'111. These bits are cleared by a power-on reset and by setting the IICRST bit in ICCR2 to 1. With the clocked synchronous serial format, these bits should not be modified.

I²C Bus Format	Clocked Synchronous Serial Format
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bit
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

20.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits received.

ICIER is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (IITXI). 0: Transmit data empty interrupt request (IITXI) is disabled. 1: Transmit data empty interrupt request (IITXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (IITEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. IITEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (IITEI) is disabled. 1: Transmit end interrupt request (IITEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (IIRXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. IIRXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (IIRXI) are disabled. 1: Receive data full interrupt request (IIRXI) are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection interrupt request (IINAKI) when the NACKF or AL/OVE bit in ICSR is set. IINAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: Disables the NACK receive interrupt request (IINAKI). 1: Enables the NACK receive interrupt request (IINAKI).
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (IISTPI) when the STOP bit in ICSR is set. 0: Stop condition detection interrupt request (IISTPI) is disabled. 1: Stop condition detection interrupt request (IISTPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. This bit can be canceled by setting the BBSY bit in ICCR2 to 1. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

20.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

ICSR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/(W)*1	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT • When data for transmission is transferred to ICDRT when the DMAC is activated by a IITXI interrupt • When data for transmission is transferred to ICDRT when the DTC is activated by a IITXI interrupt while the DISEL bit in MRB of the DTC is 0 [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	0	R/(W)*1	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT • When data for transmission is transferred to ICDRT when the DMAC is activated by a IITXI interrupt • When data for transmission is transferred to ICDRT when the DTC is activated by a IITXI interrupt while the DISEL bit in MRB of the DTC is 0 [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clocked synchronous serial format
5	RDRF	0	R/(W)*1	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read • When data for reception is transferred from ICDRR when the DMAC is activated by a IIRXI interrupt • When data for reception is transferred from ICDRR when the DTC is activated by a IIRXI interrupt while the DISEL bit in MRB of the DTC is 0 [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/(W)*1	No Acknowledge Detection Flag*2 [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1 [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/(W)*1	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting conditions] • In master mode, when a stop condition is detected after frame transfer • In slave mode, when the slave address in the first byte after detecting a start condition matches the address set in SAR, and then the stop condition is detected
2	AL/OVE	0	R/(W)*1	Arbitration Lost Flag/Overrun Error Flag Indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface 3 detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master. [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1 [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected • When the final bit is received with the clocked synchronous format while RDRF = 1

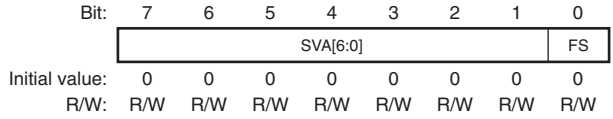
Bit	Bit Name	Initial Value	R/W	Description
1	AAS	0	R/(W)*1	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA[6:0] in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS = 1 [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode.
0	ADZ	0	R/(W)*1	General Call Address Recognition Flag This bit is valid in slave receive mode with the I²C bus format. [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1 [Setting condition] • When the general call address is detected in slave receive mode

Notes: 1. Only 0 can be written after reading 1, to clear the flag.
2. When NACKF = 1 has been detected, be sure to clear NACKF in processing on completion of transfer. Further reception and transmission are not possible until the bit is cleared.

20.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.

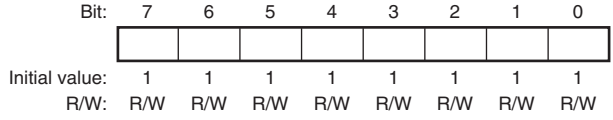
SAR is initialized to H'00 by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is selected

20.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores data for transmission. On detecting that the shift register (ICDRS) is empty, data for transmission that have been written to ICDRT are transferred to ICDRS and transfer starts. If the next byte for transfer is written to ICDRT while the previous byte is being transferred from ICDRS, continuous transfer becomes possible. Note that an MSB- or LSB-first value written to ICDRT is read in reverse order when ICDRT is read in the LSB-first order (the MLS bit in ICMR is 1). ICDRT is initialized to H'FF.



20.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.

ICDRR is initialized to H'FF by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

20.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

20.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 20.4.7, Noise Filter.

NF2CYC is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	NF2CYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the peripheral clock can be filtered out 1: The noise less than two cycles of the peripheral clock can be filtered out

20.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

20.4.1 I²C Bus Format

Figure 20.3 shows the I²C bus formats. Figure 20.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

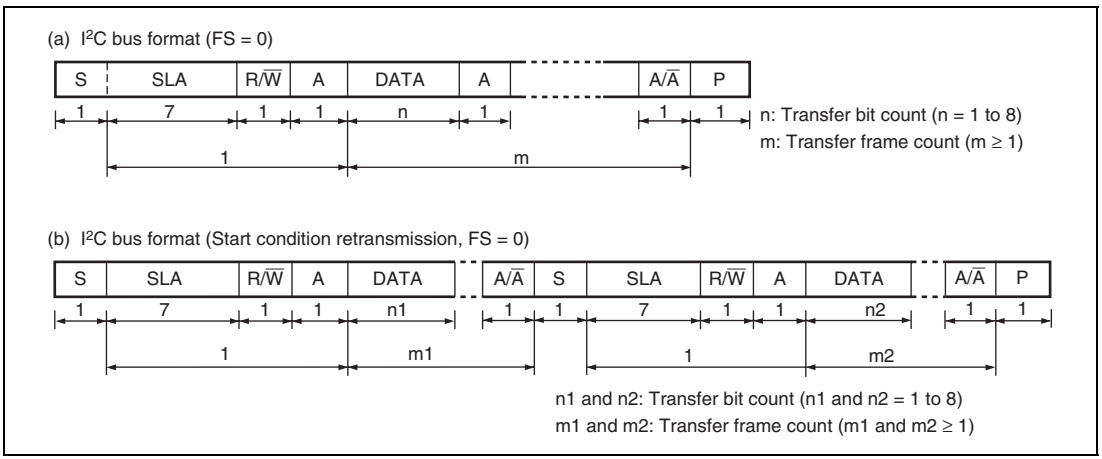


Figure 20.3 I²C Bus Formats

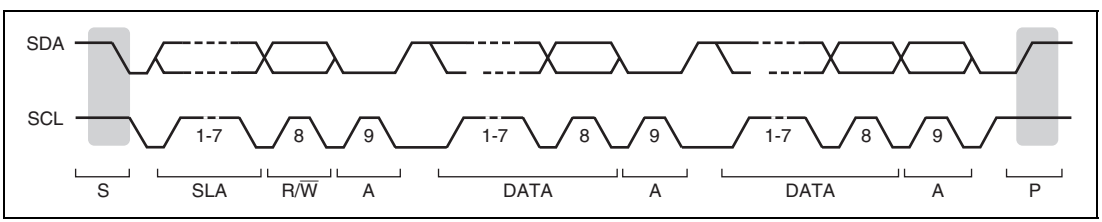


Figure 20.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

20.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 20.5 and 20.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set bit MLS in ICMR and bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1. Once TEND has been set, wait until 0 is read from SCLO in ICCR2.
7. Wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
8. When the STOP bit in ICSR is set to 1, the operation returns to slave receive mode.

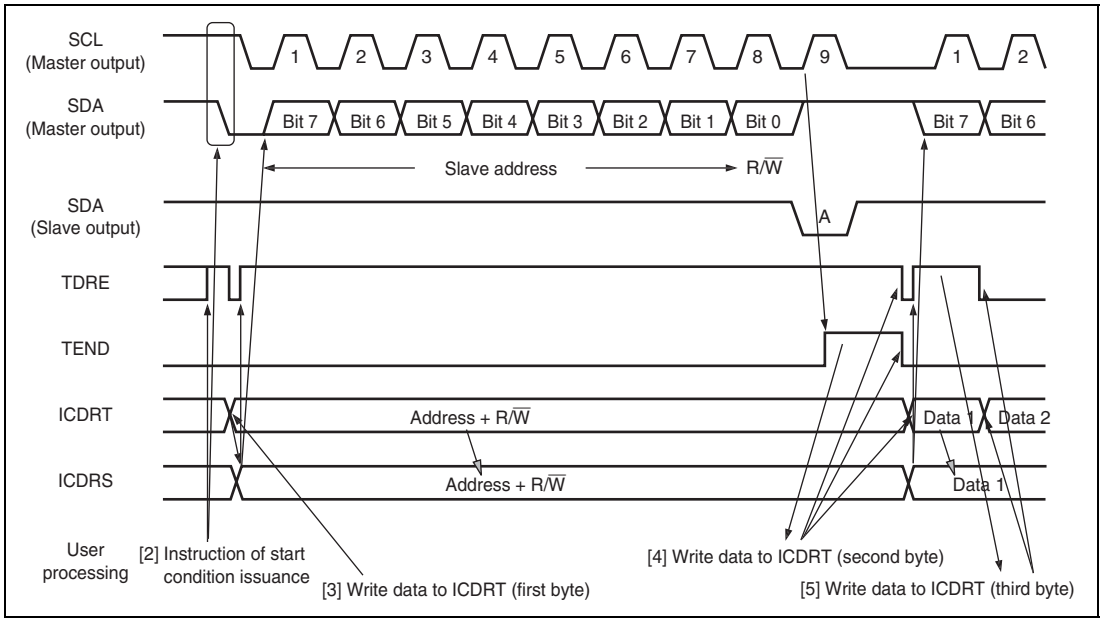


Figure 20.5 Master Transmit Mode Operation Timing (1)

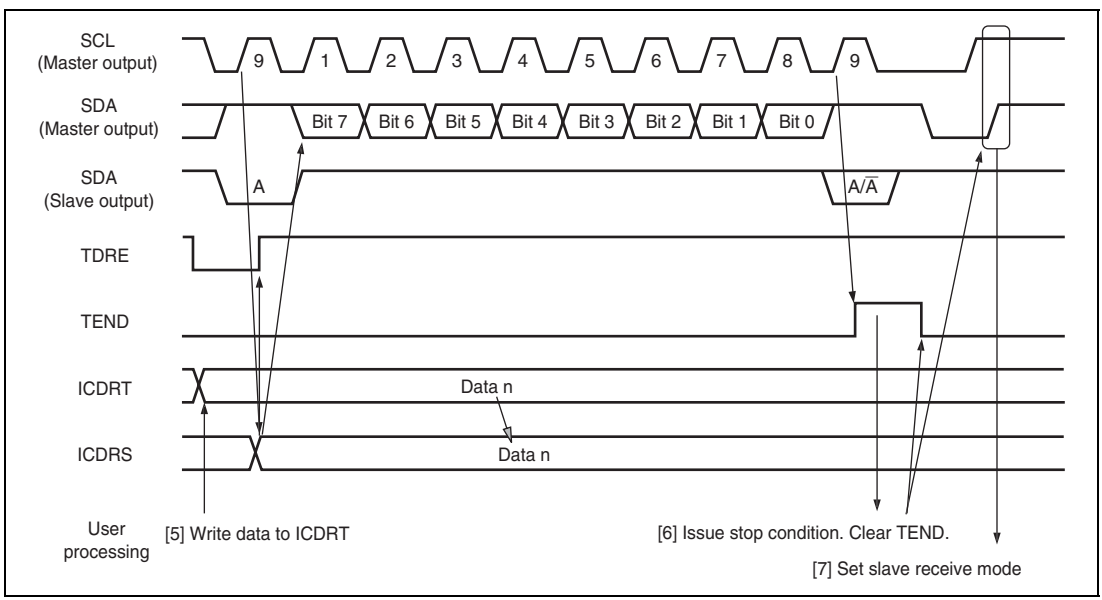


Figure 20.6 Master Transmit Mode Operation Timing (2)

20.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 20.7 and 20.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. After reading to confirm that the SCLO in ICCR2 is 0, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1. This enables the issuance of the stop condition after the next reception. After reading to confirm the SCLO in ICCR2 is 0, read ICDRR.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

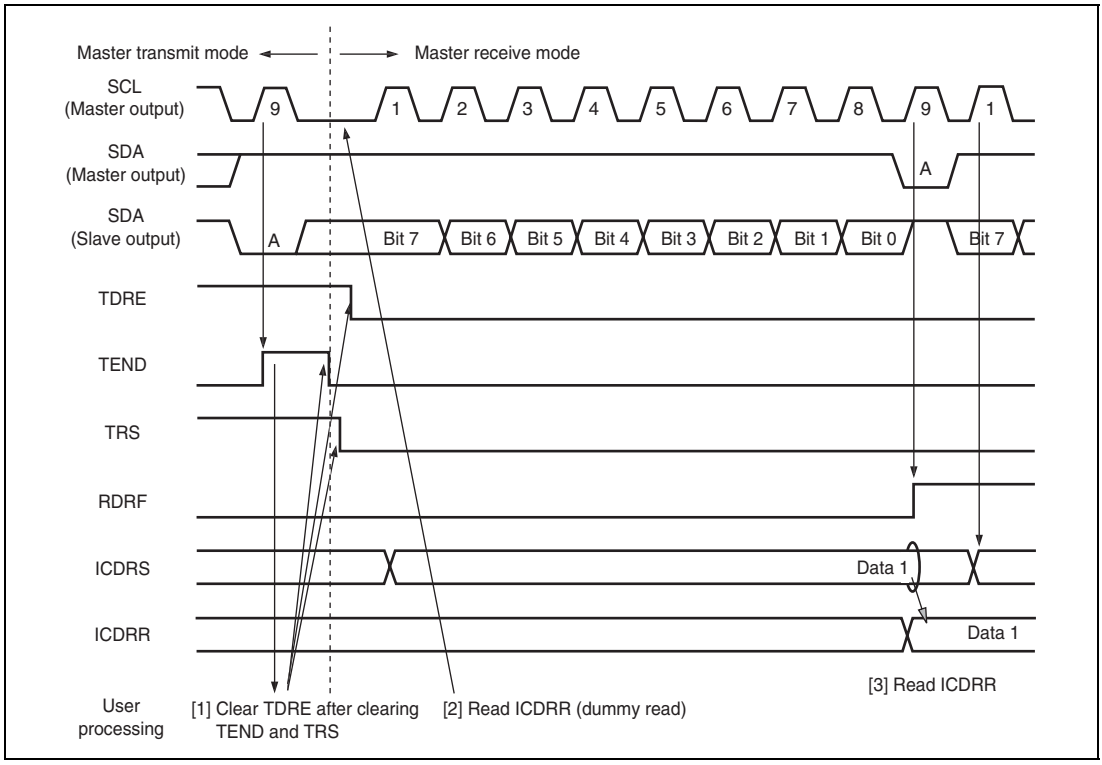


Figure 20.7 Master Receive Mode Operation Timing (1)

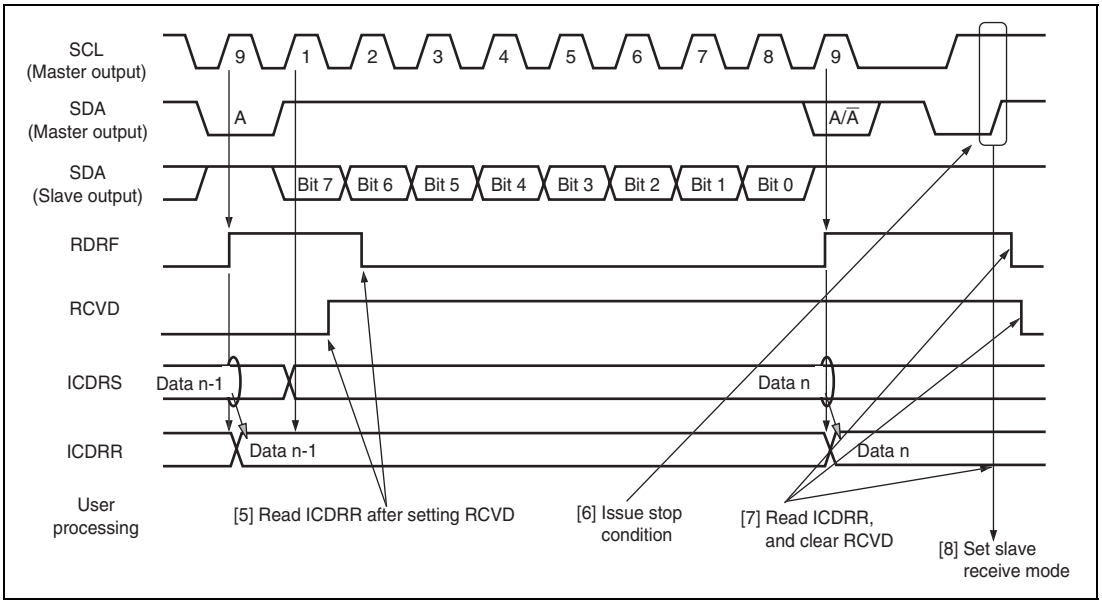


Figure 20.8 Master Receive Mode Operation Timing (2)

20.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 20.9 and 20.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS bit in ICCR1 and the TDRE bit in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is opened.
5. Clear TDRE.

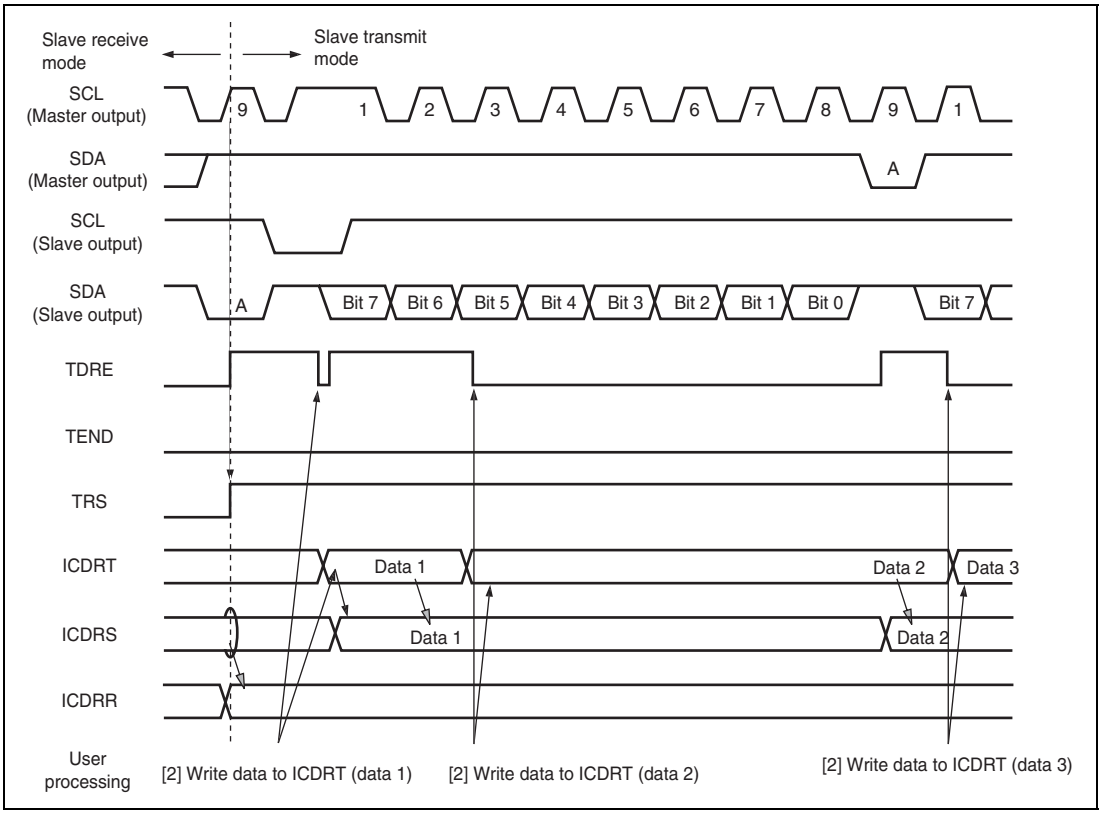


Figure 20.9 Slave Transmit Mode Operation Timing (1)

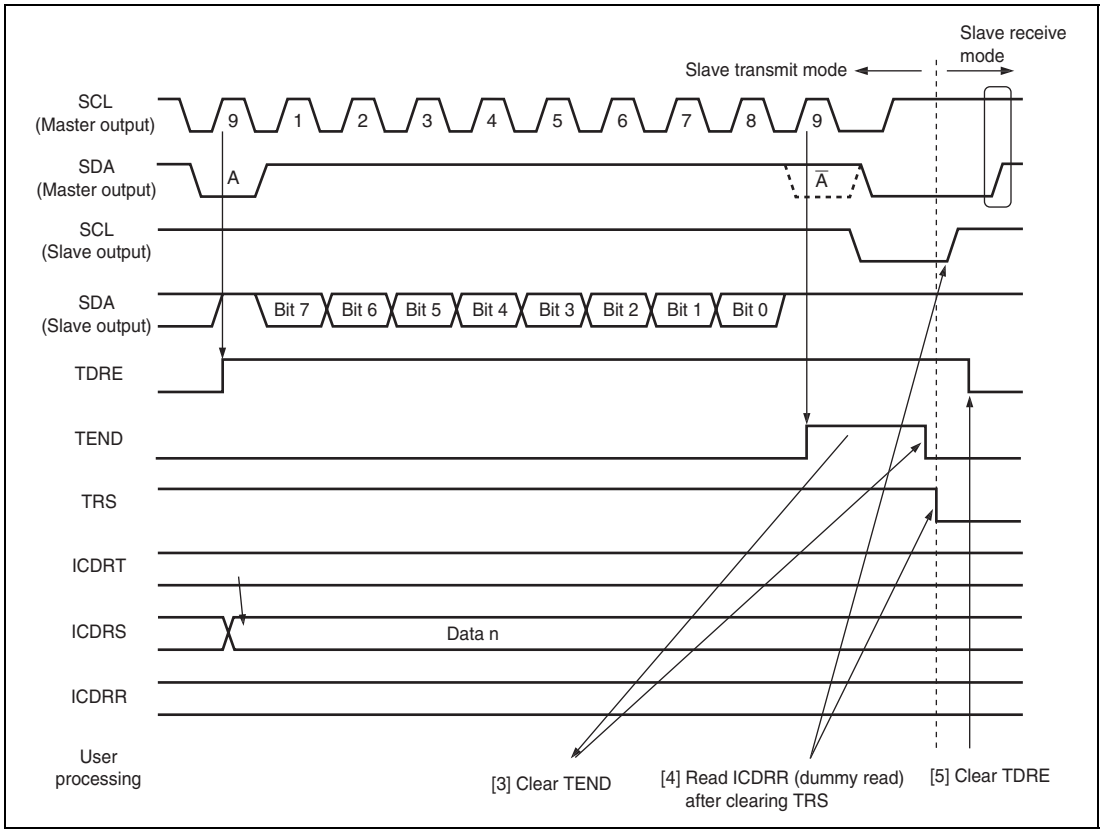


Figure 20.10 Slave Transmit Mode Operation Timing (2)

20.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 20.11 and 20.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

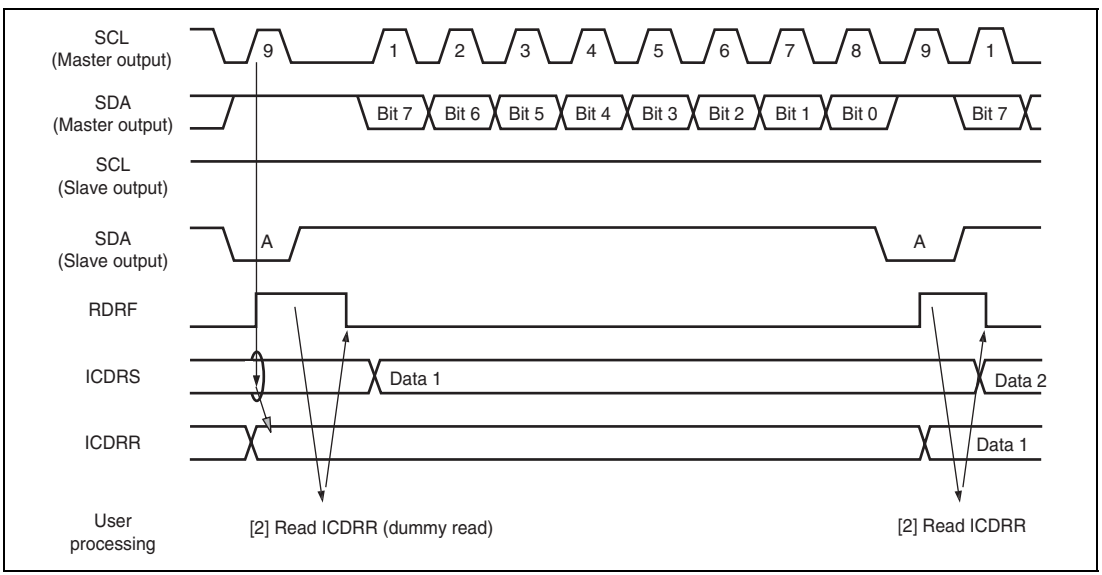


Figure 20.11 Slave Receive Mode Operation Timing (1)

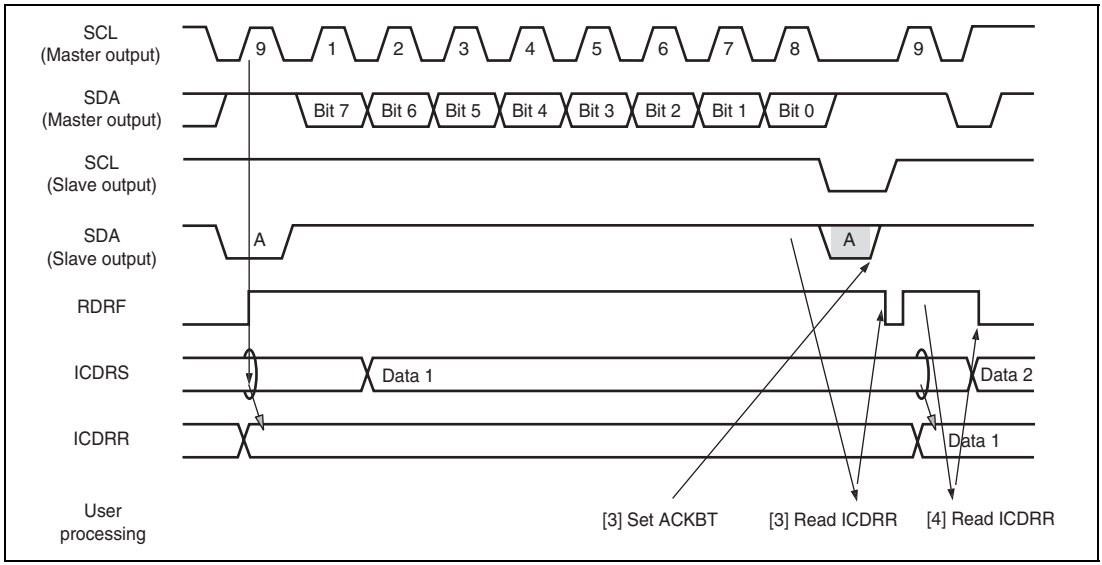


Figure 20.12 Slave Receive Mode Operation Timing (2)

20.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 20.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the fall to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

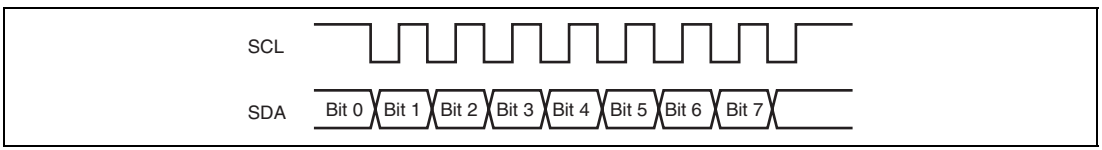


Figure 20.13 Clocked Synchronous Serial Transfer Format

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 20.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS[3:0] bits in ICCR1. (Initial setting)
2. Set the TRS bit in ICCR1 to select transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

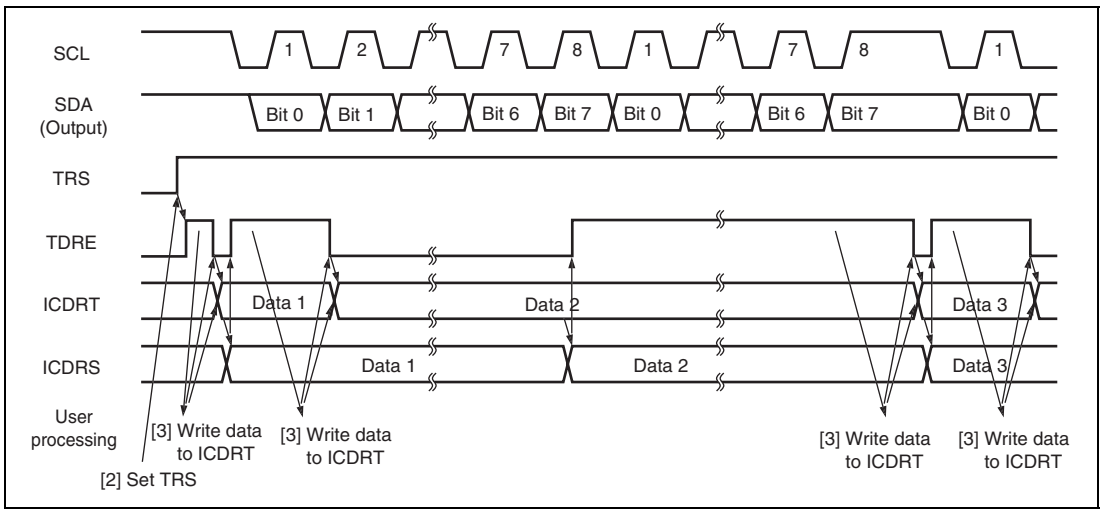


Figure 20.14 Transmit Mode Operation Timing

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 20.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock rises while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST = 1 specified. See figure 20.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. Set MST = 1 while the RCVD bit in ICCR1 is 0. This causes the receive clock to be output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 to 1. This causes the SCL to be fixed to the high level after outputting one byte of the receive clock.

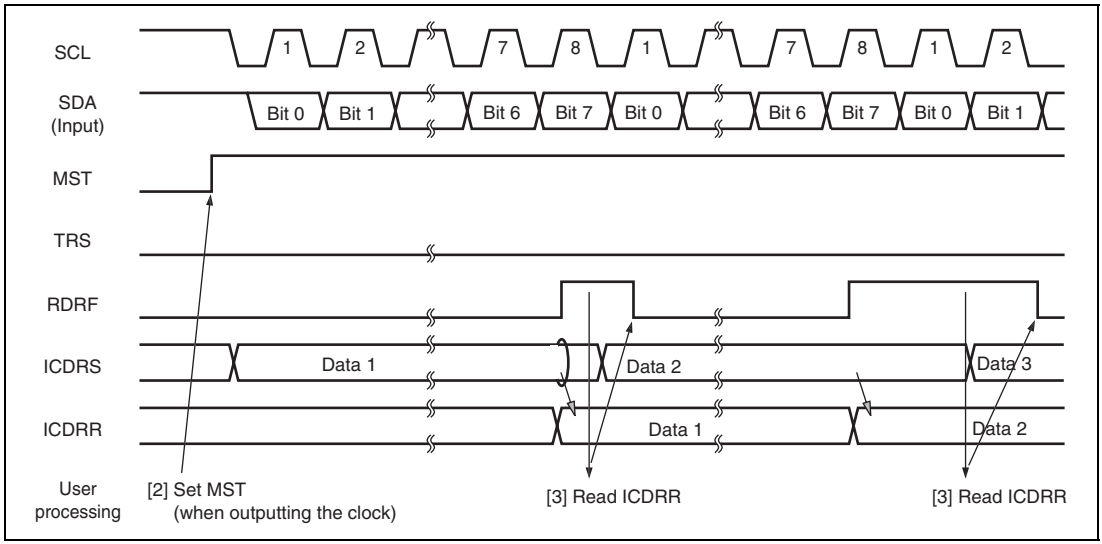


Figure 20.15 Receive Mode Operation Timing

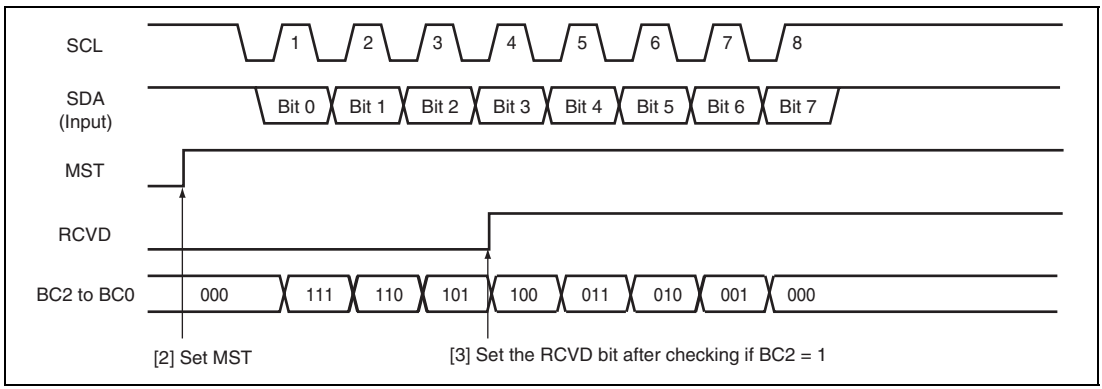


Figure 20.16 Operation Timing for Receiving One Byte (MST = 1)

20.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 20.17 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

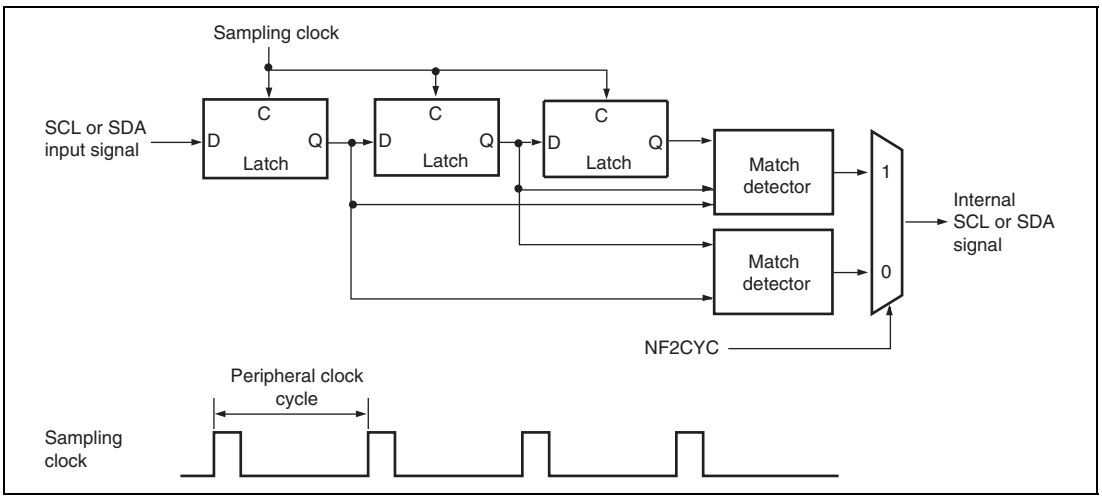


Figure 20.17 Block Diagram of Noise Filter

20.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 20.18 to 20.21.

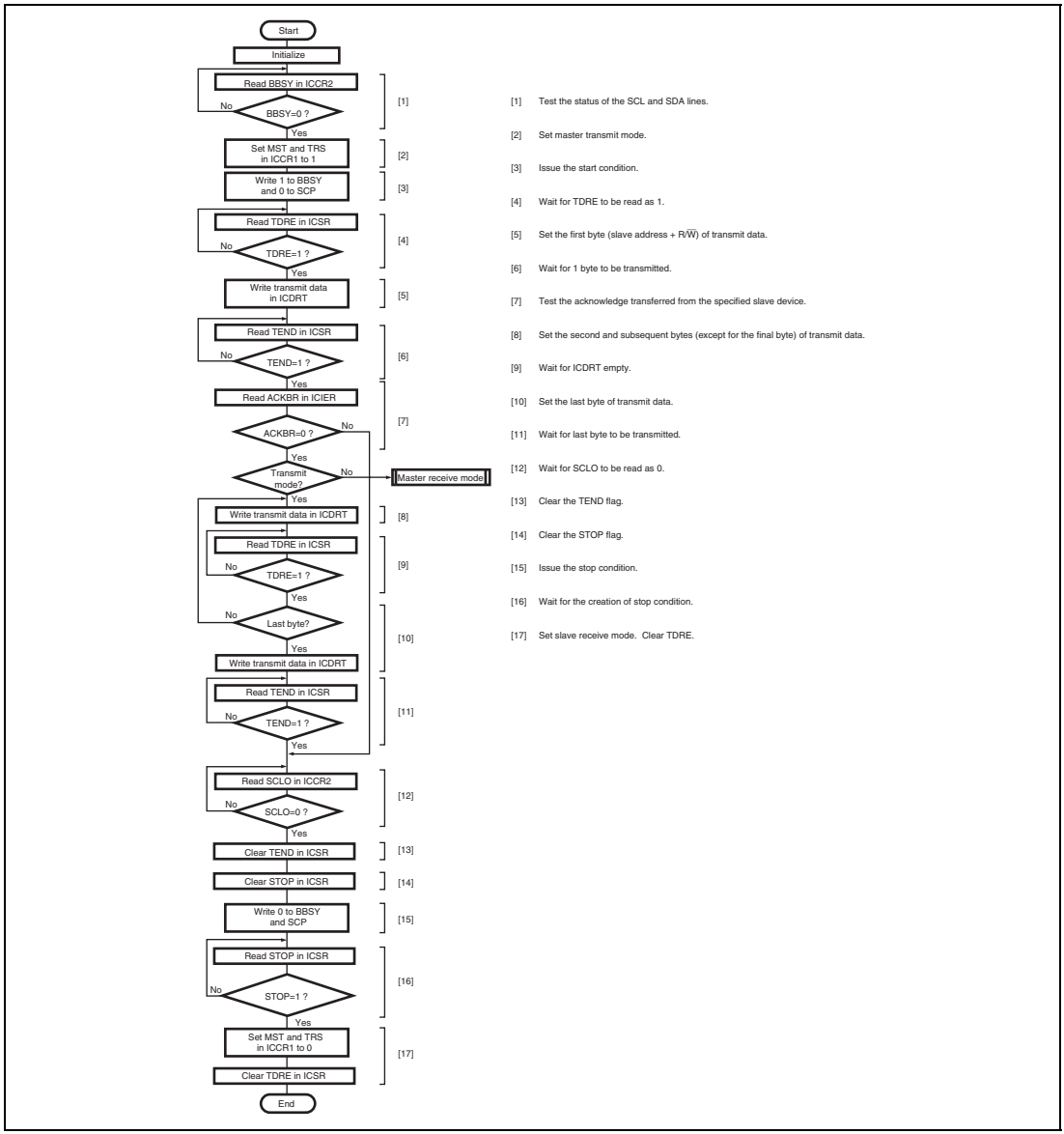
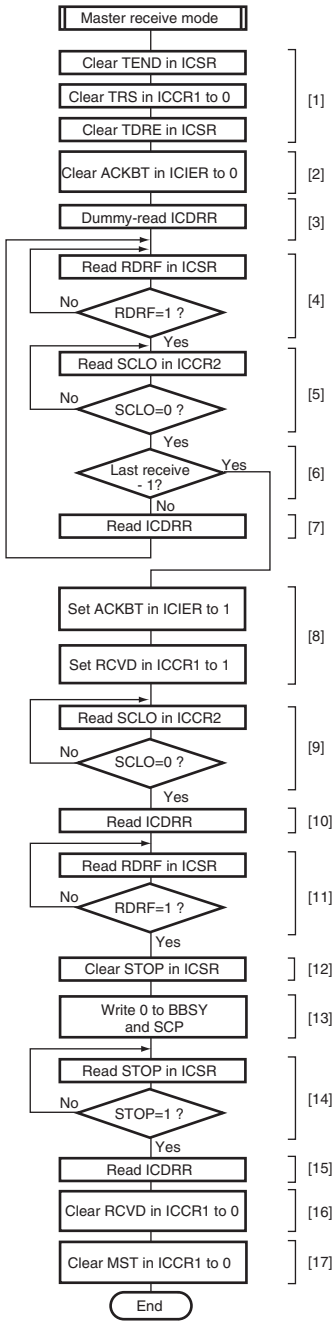


Figure 20.18 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE. *

[2] Set acknowledge to the transmit device. *

[3] Dummy-read ICDDR. *

[4] Wait for 1 byte to be received

[5] Wait for SCLO to be read as 0.

[6] Check whether it is the (last receive - 1).

[7] Read the receive data.

[8] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[9] Wait for SCLO to be read as 0.

[10] Read the (final byte - 1) of received data.

[11] Wait for the last byte to be receive.

[12] Clear the STOP flag.

[13] Issue the stop condition.

[14] Wait for the creation of stop condition.

[15] Read the last byte of receive data.

[16] Clear RCVD.

[17] Set slave receive mode.

Note: * Make sure that no interrupt will be generated during steps [1] to [3].

Supplementaly note:
When the size of receive data is only one byte in reception, steps [2] to [7] are skipped after step [1], before jumping to step [8]. The step [10] is dummy-read in ICDRR.

Figure 20.19 Sample Flowchart for Master Receive Mode

20.4.9 Using the IICRST Bit to Reset I²C Bus Interface 3

For I²C bus interface 3 (IIC3), writing 1 to the IICRST bit in register IICR2 causes resetting of the BC[2:0] bits in the ICMR register and the internal circuits of IIC3. Figure 20.20 shows an example of the sequence for resetting the I²C bus interface 3 by using the IICRST bit.

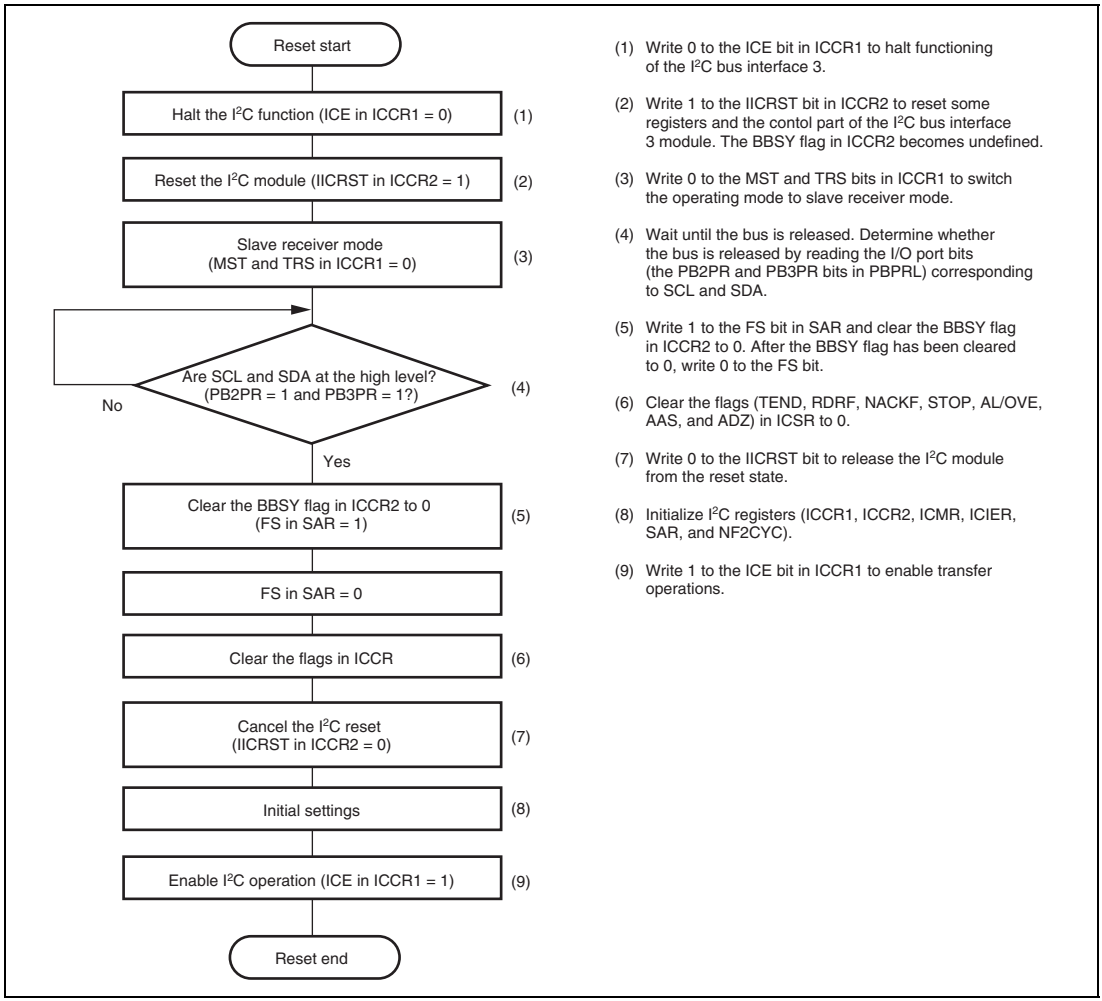


Figure 20.20 Sequence for Using the IICRST Bit to Reset I²C Bus Interface 3

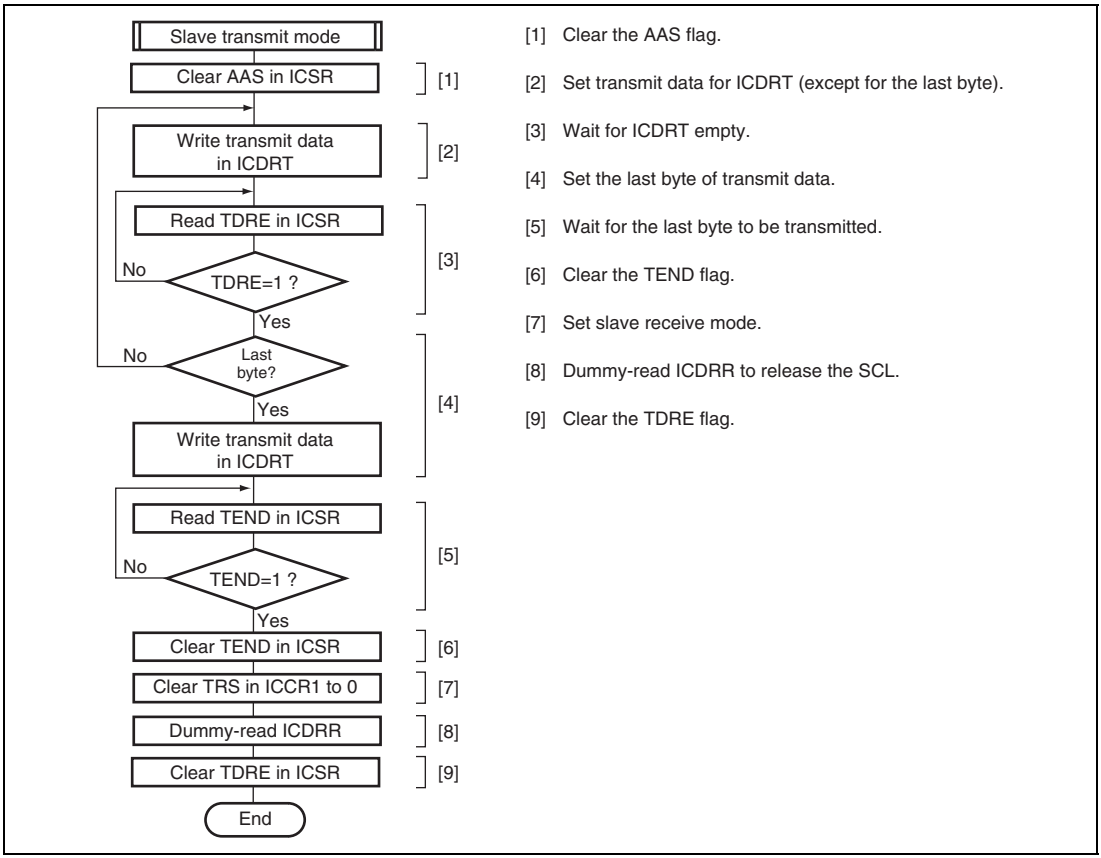


Figure 20.21 Sample Flowchart for Slave Transmit Mode

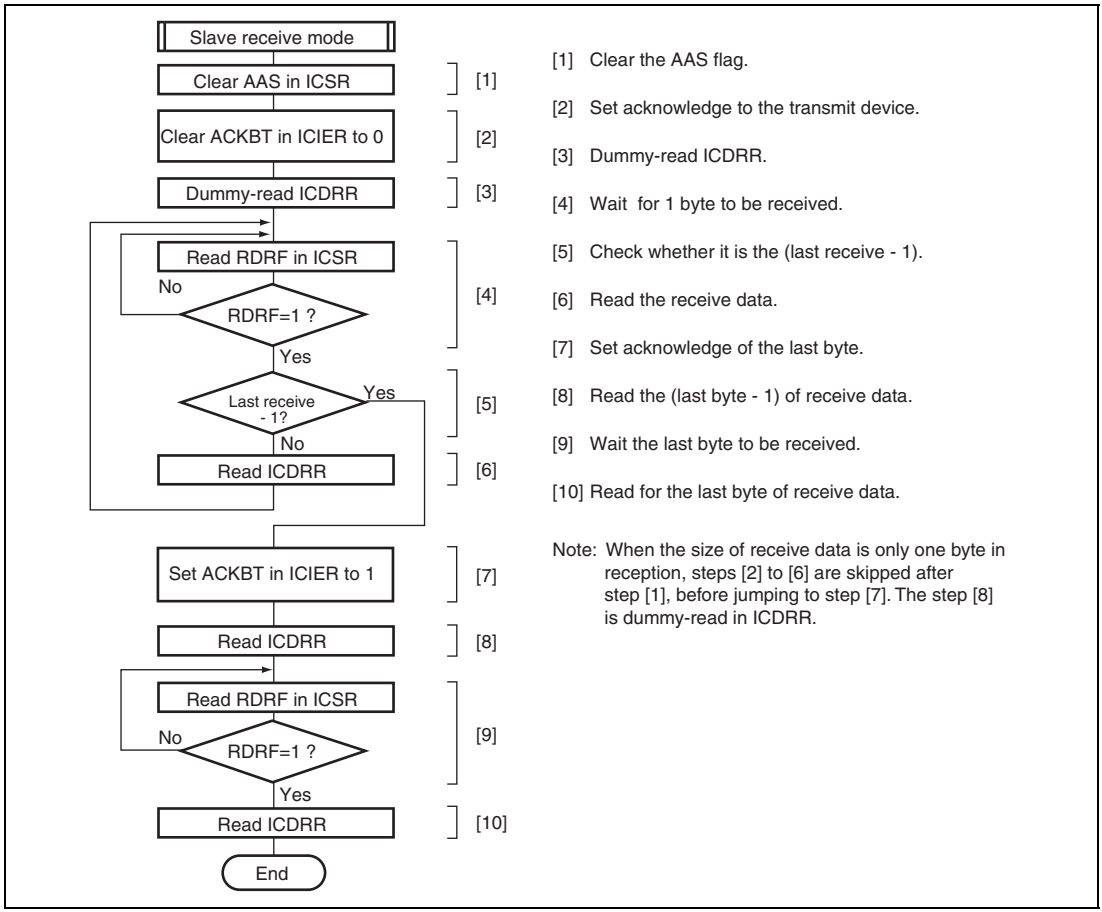


Figure 20.22 Sample Flowchart for Slave Receive Mode

20.5 Interrupt Sources and DMAC/DTC

The IIC3 module has six interrupt sources; these are for the transmit data empty interrupt (IITXI), transmit end interrupt (IITEI), receive data full interrupt (IIRXI), stop condition detection interrupt (IISTPI), and NACK detection, arbitration lost or overrun error interrupt (IINAKI) requests.

The interrupt sources and their order of priority are listed in table 20.4. The TIE, RIE, TEIE, NAKIE, and STIE bits in the I²C bus interrupt enable register (ICIER) enable or disable the various interrupt sources. Furthermore, each of the corresponding interrupt requests is independently conveyed to the interrupt controller.

A IITXI interrupt request is generated when the TDRE flag in the I²C bus status register (ICSR) is set to 1. A IITXI interrupt request can activate the direct memory access controller (DMAC) or data transfer controller (DTC) to handle data transfer. When the DMAC is activated to handle transfer, the TDRE flag is automatically cleared to 0 once the DMAC has written to the I²C bus transmit data register (ICDRT), so a IITXI interrupt request is not sent to the CPU. When the DTC is activated to handle transfer, if the DISEL bit of the DTC is 0 and the value of the transfer counter is non-zero, the TDRE flag is automatically cleared to 0 once the DTC has written to the ICDRT, so a IITXI interrupt request is not sent to the CPU. If the DISEL bit of the DTC is 0 and the value of the transfer counter is 0, or if the DISEL bit is 1, writing to ICDRT does not lead to automatic clearing of the TDRE flag, so a IITXI interrupt request is subsequently generated for the CPU.

A IIRXI interrupt request is generated when the RDRF flag in ICSR is set to 1. A IIRXI interrupt request can activate the DMAC/DTC to handle data transfer. When the DMAC is activated to handle transfer, the RDRF flag is automatically cleared to 0 once the DMAC has read from the I²C bus receive data register (ICDRR), so a IIRXI interrupt request is not sent to the CPU. When the DTC is activated to handle transfer, if the DISEL bit of the DTC is 0 and the value of the transfer counter is non-zero, the RDRF flag is automatically cleared to 0 once the DTC has read from ICDRR, so a IIRXI interrupt request is not sent to the CPU. If the DISEL bit of the DTC is 0 and the value of the transfer counter is 0, or if the DISEL bit is 1, Reading from ICDRR does not lead to automatic clearing of the RDRF flag, so a IIRXI interrupt request is subsequently generated for the CPU.

A IINAKI interrupt request is generated when the NACKF or AL/OVE flag in ICSR is set to 1. A IINAKI interrupt request is not capable of activating the DMAC or the DTC. Setting of the NACKF to 1 only leads to a IINAKI interrupt request when communications are in I²C format.

A IISTPI interrupt request is generated when the STOP flag in ICSR is set to 1. A IISTPI interrupt request is not capable of activating the DMAC or the DTC. Setting of the STOP flag to 1 only leads to a IISTPI interrupt request when communications are in I²C format.

A IITEI interrupt request is generated when the TEND flag in ICSR is set to 1. A IITEI interrupt request is not capable of activating the DMAC or the DTC.

Caution is required because writing data for transmission to ICDRT automatically clears TDRE and TEND and reading from ICDRR automatically clears RDRF. In particular, if TDRE is again set at the same time as data for transmission are written to ICDRT, an extra byte may be transmitted when TDRE is then cleared.

Table 20.4 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Bus Format	Clocked Synchronous Serial Format	DMAC/DTC Activation	Priority
Stop condition detection	IISTPI	(STOP = 1) • (STIE = 1)	√	—	—	High
NACK detection	IINAKI	{(NACKF = 1) + (AL/OVE = 1)} • (NAKIE = 1)	√	—	—	
Arbitration lost/ overrun error	IIRXI	(RDRF = 1) • (RIE = 1)	√	√	—	
Receive data full	IITXI	(TDRE = 1) • (TIE = 1)	√	√	√	
Transmit data empty	IITEI	(TEND = 1) • (TEIE = 1)	√	√	√	
Transmit end			√	√	—	Low

20.6 Data Transfer Using DMAC/DTC

In the I²C bus format, the slave device and transfer direction are selected through the slave address and R/$\overline{W}$ bit, and data reception is confirmed and the last frame is indicated through the acknowledge bit. Therefore, when the DMAC/DTC is used to transfer data continuously, the DMAC/DTC processing should be done in combination with the CPU processing activated by interrupts.

Table 20.5 shows an example of I²C data transfer using the DTC. This example assumes that the transfer data count is determined in advance in slave mode.

Table 20.5 Example of Data Transfer Using DMAC/DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/$\overline{W}$ bit transmit/receive	Transmitted by DMAC/DTC (ICDRT writing)	Transmitted by CPU (ICDRT writing)	Received by CPU (ICDRR reading)	Received by CPU (ICDRR reading)
Dummy data read	—	Processed by CPU (ICDRR reading)	—	Processed by CPU (ICDRR reading)
Main data transmit/receive	Transmitted by DMAC/DTC (ICDRT writing)	Received by DMAC/DTC (ICDRR reading)	Transmitted by DMAC/DTC (ICDRT writing)	Received by DMAC/DTC (ICDRR reading)
Last frame processing	Not necessary	Received by CPU (ICDRR reading)	Not necessary	Received by CPU (ICDRR reading)
DMAC/DTC transfer data frame count setting	Transmission: Actual data count + 1 (+1 is required for the slave address + R/$\overline{W}$ bit transfer)	Reception: Actual data count – 1 (–1 is required for processing of the last frame)	Transmission; Actual data count	Reception: Actual data count – 1 (–1 is required for processing of the last frame)

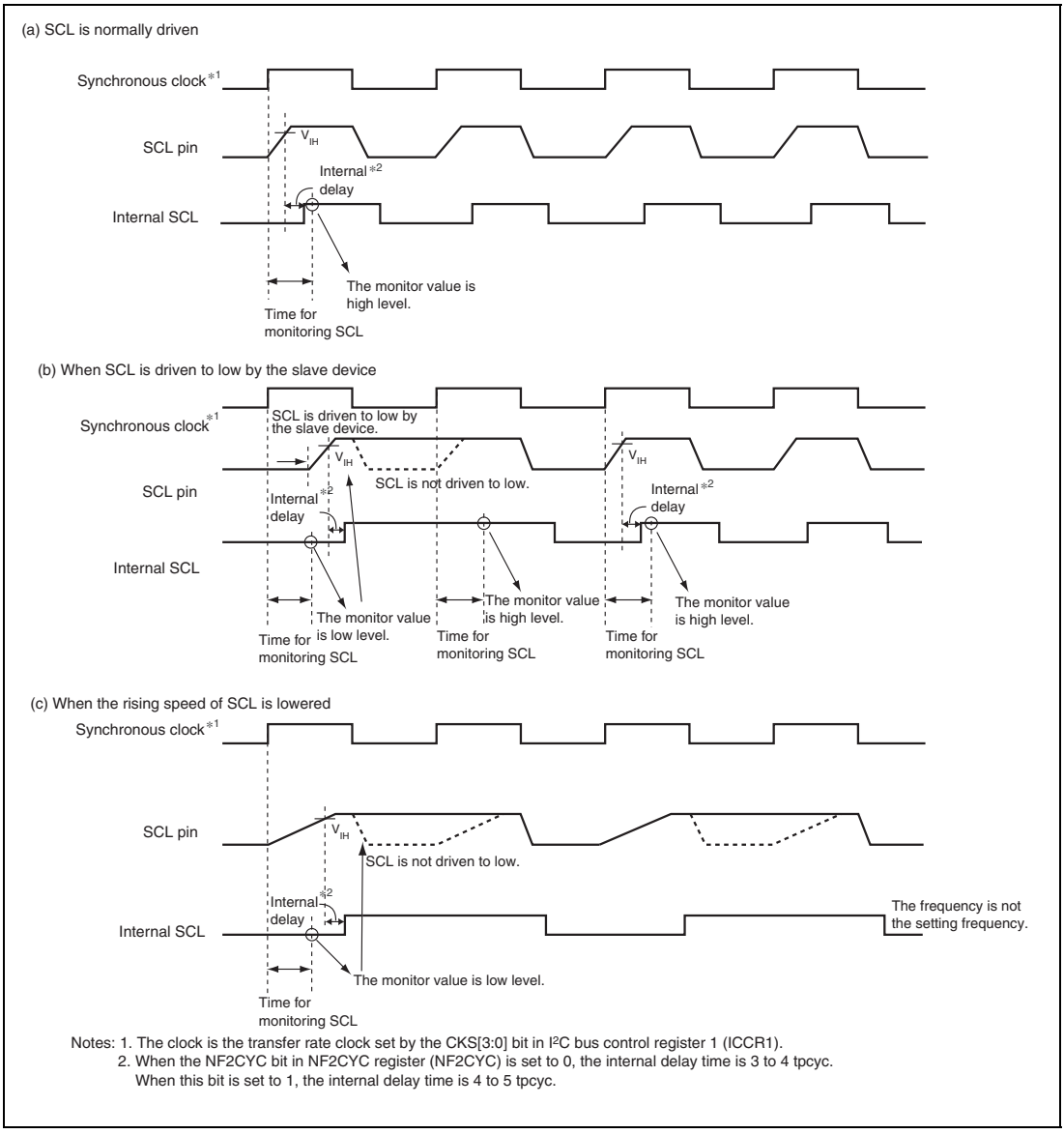
20.7 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 20.23 shows the timing of the bit synchronous circuit and table 20.6 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.



Notes: 1. The clock is the transfer rate clock set by the CKS[3:0] bit in I²C bus control register 1 (ICCR1).
2. When the NF2CYC bit in NF2CYC register (NF2CYC) is set to 0, the internal delay time is 3 to 4 tpcyc. When this bit is set to 1, the internal delay time is 4 to 5 tpcyc.

Figure 20.23 Bit Synchronous Circuit Timing

Table 20.6 Time for Monitoring SCL

CKS[3]	CKS[2]	Time for Monitoring SCL
0	0	9 tpcyc*
	1	21 tpcyc*
1	0	39 tpcyc*
	1	87 tpcyc*

Note: * tpcyc indicates peripheral clock (Pϕ) cycle.

20.8 Usage Notes

20.8.1 Setting for Module Standby Mode

The standby control register can disable or enable operation of the IIC3 module. Operation of the IIC3 module is stopped by the initial setting. The registers of the module become accessible when it is released from module standby mode. For details, refer to section 32, Power-Down Modes.

20.8.2 Note on Multi-Master Operation

In multi-master operation, when the setting for IIC transfer rate (ICCR1.CKS[3:0]) makes this LSI slower than the other masters, pulse cycles with an unexpected length will infrequently be output on SCL.

Be sure to specify a transfer rate that is at least 1/1.8 of the fastest transfer rate among the other masters.

20.8.3 Note on Master Receive Mode

Reading ICDRR around the falling edge of the 8th clock might fail to fetch the receive data.

In addition, when RCVD is set to 1 around the falling edge of the 8th clock and the receive buffer is full, a stop condition may not be issued.

In that case, use either of the following methods (1 or 2) to avoid this situation.

1. In master receive mode, read ICDRR before the rising edge of the 8th clock cycle.
2. In master receive mode, set the RCVD bit to 1 so that transfer is in byte units.

20.8.4 Note on Setting ACKBT in Master Receive Mode

In master receive mode operation, set ACKBT before the falling edge of the 8th SCL cycle of the last data being continuously transferred. Not doing so can lead to an overrun for the slave transmission device.

20.8.5 Point for Caution When Setting ACKE in Master Transmitter Mode

When the setting of the ACKE bit in the ICIER is 1, only issue a stop condition after confirming the falling edge of the ninth cycle of SCL. Attempting to issue a stop condition while the SCL signal is at the high level during its ninth cycle may lead to fixing of SCL to the low level.

20.8.6 Note on the States of Bits MST and TRS when Arbitration is Lost

When sequential bit-manipulation instructions are used to set the MST and TRS bits to select master transmission in multi-master operation, a conflicting situation where AL in ICSR = 1 but the mode is master transmit mode (MST = 1 and TRS = 1) may arise; this depends on the timing of the loss of arbitration when the bit manipulation instruction for TRS is executed.

This can be avoided in either of the following ways.

- In multi-master operation, use the MOV instruction to set the MST and TRS bits.
- When arbitration is lost, check whether the MST and TRS bits are 0. If the MST and TRS bits have been set to a value other than 0, clear the bits to 0.

20.8.7 Access to ICE and IICRST Bits during I²C Bus Operations

Writing 0 to the ICE bit in ICCR1 or 1 to the IICRST bit in ICCR2 while this LSI is in any of the following states (1 to 4) causes the BBSY flag in ICCR2 and the STOP flag in ICSR to become undefined.

1. This module is the I²C bus master in master transmit mode (MST = 1 and TRS = 1 in ICCR1).
2. This module is the I²C bus master in master receive mode (MST = 1 and TRS = 0 in ICCR1).
3. This module is transmitting data in slave transmit mode (MST = 0 and TRS = 1 in ICCR1).
4. This module is transmitting acknowledge signals in slave receive mode (MST = 0 and TRS = 0 in ICCR1).

Executing any of the following procedures releases the BBSY flag in ICCR2 from the undefined state.

- Input a start condition (falling edge of SDA while SCL is at the high level) to set the BBSY flag to 1.
- Input a stop condition (rising edge of SDA while SCL is at the high level) to clear the BBSY flag to 0.
- If the module is in master transmit mode, issue a start condition by writing 1 and 0 to the BBSY flag and the SCP bit in ICCR2, respectively, while SCL and SDA are at the high level. The BBSY flag is set to 1 on output of the start condition (falling edge of SDA while SCL is at the high level).
- With the module in master transmit or master receive mode, SDA at the low level, and no other device holding SCL at the low level, issue a stop condition by writing 0 to the BBSY flag and the SCP bit in ICCR2. The BBSY flag is cleared to 0 on output of the stop condition (rising edge of SDA while SCL is at the high level).
- Writing 1 to the FS bit in SAR clears the BBST flag to 0.

Section 21 A/D Converter (ADC)

This LSI includes a successive approximation type 10-bit A/D converter.

21.1 Features

- 10-bit resolution
- Input channels: 16 channels (in two independent A/D conversion modules)
- Conversion time:
  When Aϕ = 50 MHz: Minimum 1.0 µs per channel
- Eight operating modes
  - Single mode: A/D conversion on one channel
  - 2-channel scan mode: Continuous A/D conversion on up to two channels
  - 4-channel scan mode: Continuous A/D conversion on up to four channels
  - 8-channel scan mode: Continuous A/D conversion on up to eight channels

  For each of the above modes, the following modes can be set:
  - Continuous scan mode: Repetitive A/D conversion on the specified channel
  - Single-cycle scan mode: One A/D conversion on the specified channels each
- A/D data registers
  A/D conversion results are stored in 16-bit A/D data registers (ADDR) that correspond to the input channels.
- Sample-and-hold function
- Three methods for starting A/D conversion
  - Software
  - Conversion start trigger by multi function timer pulse unit 2 (MTU2) or multi function timer pulse unit 2S (MTU2S)
  - External trigger signal (can activate two modules at the same time)
- Interrupt source: A/D conversion end interrupts (ADI) are generated.
- Module standby mode can be set.
- A/D conversion results of two modules are also stored in the one-stage 32-bit buffer register (shadow register).
  - Upper 16 bits: AD_0 conversion results, lower 16 bits: AD_1 conversion results
  - A/D conversion results of two modules can be read by a single long word access.

Figure 21.1 is a block diagram of a single module of the A/D converter, while figure 21.2 shows the overall structure of the A/D converter.

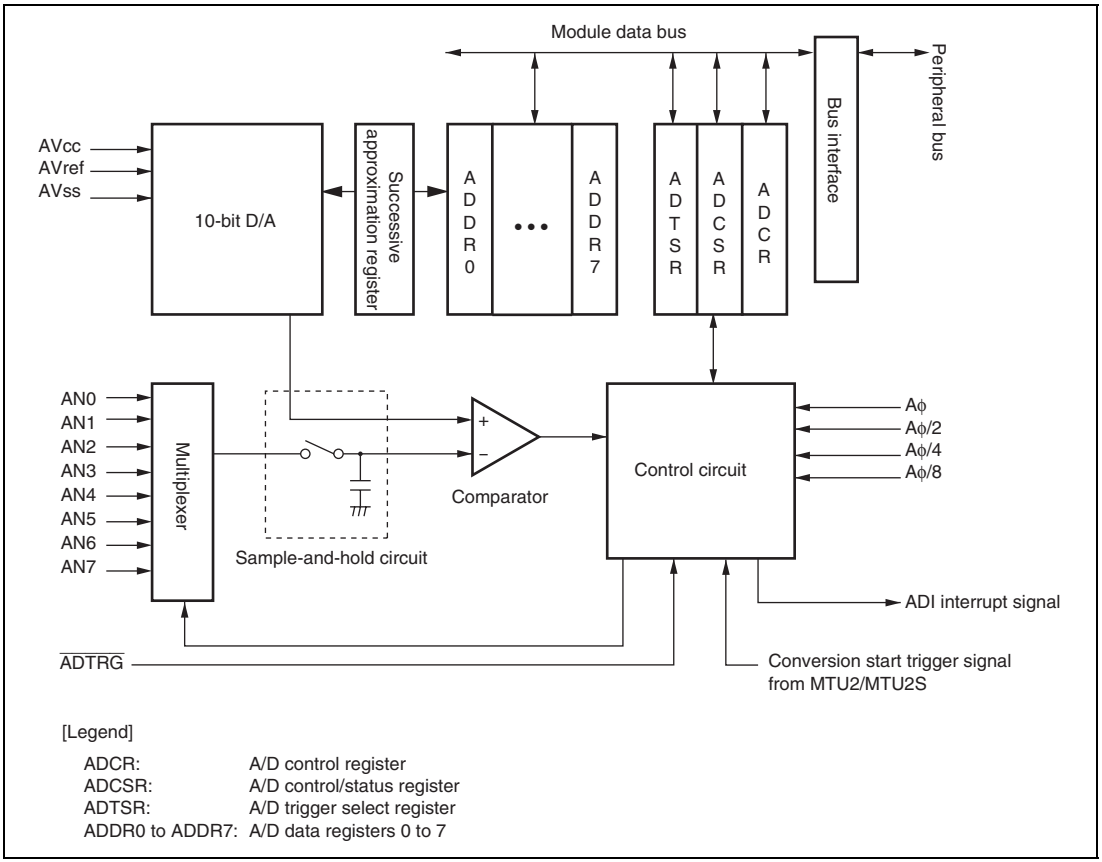


Figure 21.1 Block Diagram of A/D Converter (Figure of One Module)

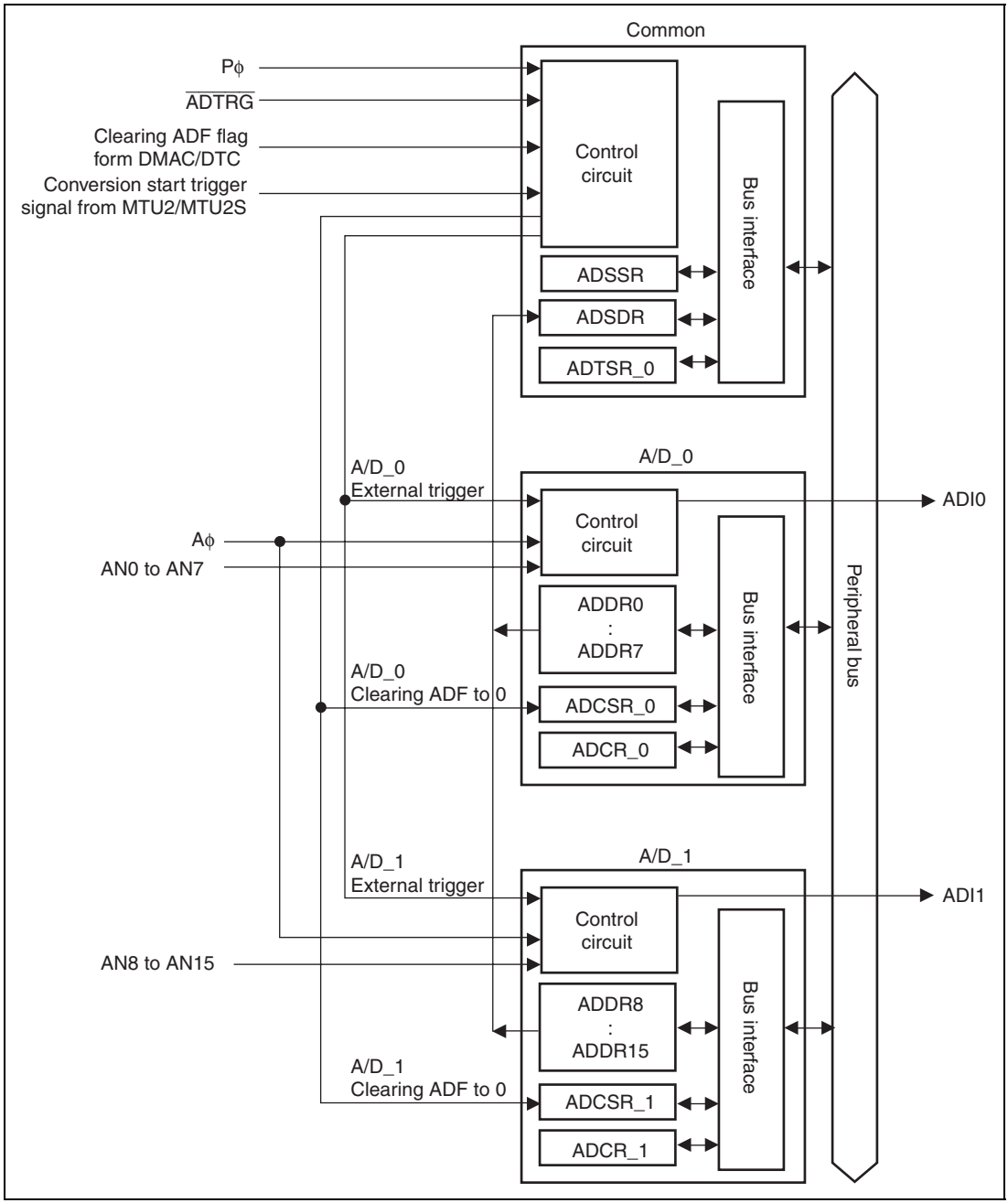


Figure 21.2 the Structure of the A/D Converter

21.2 Input/Output Pins

Table 21.1 shows the configuration of the pins used by the A/D converter. Two A/D conversion modules can perform A/D conversion independently. Input channels of A/D modules 0 and 1 (AN0 to AN3 and AN8 to AN11) can be divided into groups, each of which consists of two channels.

Table 21.1 Pin Configuration

Module	Pin Name	I/O	Function	
Common	AV_{CC}	Input	Analog block power supply and reference voltage	
	AVref	Input	A/D conversion reference voltage	
	AV_{SS}	Input	Analog block ground and reference voltage	
	$\overline{ADTRG}$	Input	A/D external trigger input pin	
A/D module 0 (A/D_0)	AN0	Input	Analog input pin 0	Group 0 in 2-channel scan mode
	AN1	Input	Analog input pin 1	
	AN2	Input	Analog input pin 2	Group 1 in 2-channel scan mode
	AN3	Input	Analog input pin 3	
	AN4	Input	Analog input pin 4	—
	AN5	Input	Analog input pin 5	—
	AN6	Input	Analog input pin 6	—
	AN7	Input	Analog input pin 7	—
A/D module 1 (A/D_1)	AN8	Input	Analog input pin 8	Group 0 in 2-channel scan mode
	AN9	Input	Analog input pin 9	
	AN10	Input	Analog input pin 10	Group 1 in 2-channel scan mode
	AN11	Input	Analog input pin 11	
	AN12	Input	Analog input pin 12	—
	AN13	Input	Analog input pin 13	—
	AN14	Input	Analog input pin 14	—
	AN15	Input	Analog input pin 15	—

Note: AN pins are connected to module 0 or 1. Set the control register of the appropriate module for the pins.

21.3 Register Descriptions

The A/D converter has the following registers. For the addresses and states of these registers in each processing status, refer to section 34, List of Registers.

Table 21.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register 0	ADDR0	R	H'0000	H'FFFE5800	16
A/D data register 1	ADDR1	R	H'0000	H'FFFE5802	16
A/D data register 2	ADDR2	R	H'0000	H'FFFE5804	16
A/D data register 3	ADDR3	R	H'0000	H'FFFE5806	16
A/D data register 4	ADDR4	R	H'0000	H'FFFE5808	16
A/D data register 5	ADDR5	R	H'0000	H'FFFE580A	16
A/D data register 6	ADDR6	R	H'0000	H'FFFE580C	16
A/D data register 7	ADDR7	R	H'0000	H'FFFE580E	16
A/D control/status register_0	ADCSR_0	R/W	H'0000	H'FFFE5810	16
A/D control register_0	ADCR_0	R/W	H'0000	H'FFFE5812	16
A/D data register 8	ADDR8	R	H'0000	H'FFFE5900	16
A/D data register 9	ADDR9	R	H'0000	H'FFFE5902	16
A/D data register 10	ADDR10	R	H'0000	H'FFFE5904	16
A/D data register 11	ADDR11	R	H'0000	H'FFFE5906	16
A/D data register 12	ADDR12	R	H'0000	H'FFFE5908	16
A/D data register 13	ADDR13	R	H'0000	H'FFFE590A	16
A/D data register 14	ADDR14	R	H'0000	H'FFFE590C	16
A/D data register 15	ADDR15	R	H'0000	H'FFFE590E	16
A/D control/status register_1	ADCSR_1	R/W	H'0000	H'FFFE5910	16
A/D control register_1	ADCR_1	R/W	H'0000	H'FFFE5912	16
A/D shadow data register	ADSDR	R	H'00000000	H'FFFE5B00	32
A/D shadow select register	ADSSR	R/W	H'3100	H'FFFE5B04	16
A/D trigger select register_0	ADTSR_0	R/W	H'0000	H'FFFE5B10	16

21.3.1 A/D Data Registers 0 to 15 (ADDR0 to ADDR15)

ADDRs are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. For example, the A/D conversion result of AN4 is stored in the A/D data register 4 (ADDR4).

The converted 10-bit data is stored in bits 15 to 6 in ADDR. The lower 6 bits of ADDR are always read as 0.

The initial value of ADDR is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADD[9:0]										-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	ADD[9:0]	All 0	R	10-bit data
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

21.3.2 A/D Control/Status Registers_0 and _1 (ADCSR_0 and ADCSR_1)

ADCSRs are 16-bit readable/writable registers that set various parameters for A/D conversion including the interrupt.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	TRGE	-	CON ADF	STC[2:0]			CKS[1:0]		ADM[1:0]		ADCS	CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the completion of A/D conversion. [Setting condition] • When A/D conversion is completed in single mode • When A/D conversion on all specified channels is completed in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DMAC is activated by an ADI interrupt and ADDR or ADCSR is read • When the DTC is activated by an ADI interrupt and ADDR or ADCSR is read with the DISEL bit in MRB of the DTC being 1.
14	ADIE	0	R/W	A/D Interrupt (ADI) Enable Enables the ADI interrupt generation by the ADF flag when this bit is 1.
13	TRGE	0	R/W	Trigger Enable Enables or disables A/D conversion start by the ADTRG, MTU2, or MTU2S triggers. 0: A/D conversion start by the triggers is disabled. 1: A/D conversion start by the triggers is enabled.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11	CONADF	0	R/W	ADF Control Controls ADF operation in 2-channel scan mode. This bit is valid in 2-channel scan mode when A/D conversion start by the triggers is set (TRGE = 1). This bit is ignored in single mode, 4-channel scan mode, and 8-channel scan mode. 0: ADF is set when either the group-0 or group-1 trigger conversion is completed. 1: ADF is set when both of the group-0 and group-1 trigger conversions are completed.
10 to 8	STC[2:0]	000	R/W	Sampling State Control Sets sampling time in combination with CKS[1:0]. 000: 200 states 001: 100 states 010: 86 states 011: 75 states 100: 63 states 101: 50 states 110: 38 states 111: 25 states
7, 6	CKS[1:0]	00	R/W	A/D Operation Clock Select Sets sampling time in combination with STC[2:0]. 00: Aϕ/8 01: Aϕ/4 10: Aϕ/2 11: Aϕ
5, 4	ADM[1:0]	00	R/W	A/D Mode 1, 0 Selects A/D conversion mode. 00: Single mode 01: 4-channel scan mode 10: 8-channel scan mode 11: 2-channel scan mode

Bit	Bit Name	Initial Value	R/W	Description
3	ADCS	0	R/W	A/D Continuous Scan Selects either a single-cycle or a continuous scan in scan mode. This bit is valid only when scan mode is selected. 0: Single-cycle scan 1: Continuous scan
2 to 0	CH[2:0]	000	R/W	Channel Select 2 to 0 Selects analog input channels for A/D conversion.

Note: * Only 0 can be written to the flag bit after 1 is read out to clear the flag bit.

Table 21.3 Channel Select List

- Single Mode

Bit 2	Bit 1	Bit 0	Analog Input Channels	
			Single Mode	
CH2	CH1	CH0	A/D_0	A/D_1
0	0	0	AN0	AN8
		1	AN1	AN9
	1	0	AN2	AN10
		1	AN3	AN11
1	0	0	AN4	AN12
		1	AN5	AN13
	1	0	AN6	AN14
		1	AN7	AN15

- 2-Channel Scan Mode

Bit 2	Bit 1	Bit 0	Analog Input Channels					
			Software Activation		Others			
					A/D_0		A/D_1	
CH2	CH1	CH0	A/D_0	A/D_1	Group 0	Group 1	Group 0	Group 1
0	0	0	AN0	AN8	AN0	AN2	AN8	AN9
		1	AN0, AN1	AN8, AN9	AN0, AN1	AN2, AN3	AN8, AN9	AN10, AN11
	1	0	AN2	An10	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
		1	AN2, AN3	AN10, AN11				
1	0	0	Setting prohibited	Setting prohibited				
		1						
	1	0						
		1						

Note: Even in 2-channel scan mode, A/D conversion is performed only on the channels specified by CH[2:0]. For example, when 2-channel scan mode is set with continuous scan mode and CH[2:0] = 000, A/D_0 performs repetitive conversions of AN0 or AN2.

- 4-Channel Scan Mode

Bit 2	Bit 1	Bit 0	Analog Input Channels	
			Single Mode	
CH2	CH1	CH0	A/D_0	A/D_1
0	0	0	AN0	AN8
		1	AN0, AN1	AN8, AN9
	1	0	AN0 to AN2	AN8 to AN10
		1	AN0 to AN3	AN8 to AN11
1	0	0	AN4	AN12
		1	AN4, AN5	AN12, AN13
	1	0	AN4 to AN6	AN12 to AN14
		1	AN4 to AN7	AN12 to AN15

Note: Even in 4-channel scan mode, A/D conversion is performed only on the channels specified by CH[2:0]. For example, when 4-channel scan mode is set with continuous scan mode and CH[2:0] = 000, A/D_0 performs repetitive conversions of AN0.

- 8-Channel Scan Mode

Bit 2	Bit 1	Bit 0	Analog Input Channels	
			Single Mode	
CH2	CH1	CH0	A/D_0	A/D_1
0	0	0	AN0	AN8
		1	AN0, AN1	AN8, AN9
	1	0	AN0 to AN2	AN8 to AN10
		1	AN0 to AN3	AN8 to AN11
1	0	0	AN0 to AN4	AN8 to AN12
		1	AN0 to AN5	AN8 to AN13
	1	0	AN0 to AN6	AN8 to AN14
		1	AN0 to AN7	AN8 to AN15

Note: Even in 8-channel scan mode, A/D conversion is performed only on the channels specified by CH[2:0]. For example, when 8-channel scan mode is set with continuous scan mode and CH[2:0] = 000, A/D_0 performs repetitive conversions of AN0.

21.3.3 A/D Control Registers_0 and _1 (ADCR_0 and ADCR_1)

ADCRs are 16-bit readable/writable registers that start or stop the A/D conversion.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ADST	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	ADST	0	R/W	A/D Start When this bit is cleared to 0, A/D conversion is stopped and the A/D converter enters the idle state. When this bit is set to 1, A/D conversion is started. In single mode or single-cycle scan mode, this bit is automatically cleared to 0 when A/D conversion ends on the selected single channel. In continuous scan mode, A/D conversion is continuously performed for the selected channels in sequence until this bit is cleared by software, a reset, or in module standby mode.
12 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

21.3.4 A/D Shadow Data Register (ADSDR)

ADSDR is a 32-bit read-only register where the value of AD_0 ADDR or AD_1 ADDR specified by ADSSR is aligned and stored in the upper or lower bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADSD0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADSD1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	ADSD0 [15:0]	0	R/W	A/D-0 ADDR Read (16 Bits) A/D_0 ADDR value is aligned and stored in the upper or lower 10 bits according to the ADDALS bit setting in ADSSR.
7 to 0	ADSD1 [15:0]	0	R/W	A/D-1 ADDR Read (16 Bits) A/D_1 ADDR value is aligned and stored in the upper or lower 10 bits according to the ADDALS bit setting in ADSSR.

21.3.5 A/D Shadow Select Register (ADSSR)

ADSSR is a 16-bit readable/writable register that enables or disables auto clearance of the ADF bit when the DMAC/DTC reads ADSDR and that selects ADDR of the A/D_0 and A/D_1 that can be read from ADSDR and data alignment.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	AD0FCE	AD1FCE	-	-	-	ADDALS	AD0SE	AD0SS[2:0]			AD1SE	AD1SS[2:0]		
Initial value:	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	AD0FCE	0	R/W	AD0 Flag Clear Enable Enables or disables auto clearance of the ADF bit in ADCSR_0 when the DMAC/DTC reads ADSDR. 0: Auto clearance of ADF in ADCSR_0 is disabled. 1: Auto clearance of ADF in ADCSR_0 is enabled.
12	AD1FCE	0	R/W	AD1 Flag Clear Enable Enables or disables auto clearance of the ADF bit in ADCSR_1 when the DMAC/DTC reads ADSDR. 0: Auto clearance of ADF in ADCSR_1 is disabled. 1: Auto clearance of ADF in ADCSR_1 is enabled.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ADDALS	1	R/W	AD Data Alignment Select Sets the position for storing the selected ADDR data of the A/D_0 or A/D_1 in ADSDR. 0: Data is stored in the upper 10 bits. 1: Data is stored in the lower 10 bits.

Bit	Bit Name	Initial Value	R/W	Description
7	AD0SE	0	R/W	AD0 Shadow Enable Enables or disables reading the A/D_0 ADDR, which can be read from ADSDR[31:16]. 0: Reading A/D_0 ADDR from ADSDR[31:16] is disabled. 1: Reading A/D_0 ADDR from ADSDR[31:16] is enabled.
6 to 4	AD0SS[2:0]	All 0	R/W	AD0 Shadow Select Selects an ADDR of A/D_0, which can be read from ADSDR[31:16]. 000: ADDR0 of A/D_0 001: ADDR1 of A/D_0 010: ADDR2 of A/D_0 011: ADDR3 of A/D_0 100: ADDR4 of A/D_0 101: ADDR5 of A/D_0 110: ADDR6 of A/D_0 111: ADDR7 of A/D_0
3	AD1SE	0	R/W	AD1 Shadow Enable Enables or disables reading the A/D_1 ADDR, which can be read from ADSDR[31:16]. 0: Reading A/D_1 ADDR from ADSDR[31:16] is disabled. 1: Reading A/D_1 ADDR from ADSDR[31:16] is enabled.
2 to 0	AD1SS[2:0]	All 0	R/W	AD1 Shadow Select Selects an ADDR of A/D_1, which can be read from ADSDR[31:16]. 000: ADDR0 of A/D_1 001: ADDR1 of A/D_1 010: ADDR2 of A/D_1 011: ADDR3 of A/D_1 100: ADDR4 of A/D_1 101: ADDR5 of A/D_1 110: ADDR6 of A/D_1 111: ADDR7 of A/D_1

21.3.6 A/D Trigger Select Register_0 (ADTSR_0)

ADTSR_0 is a 16-bit readable/writable register that selects an A/D conversion start trigger from among the external trigger, MTU2 triggers, and MTU2S triggers. In 2-channel scan mode, channels 0 to 4 of A/D module 0 and channels 8 to 11 of A/D module 1 are divided into two groups, for each of which an A/D trigger can be independently specified.

When 2-channel scan mode is not used, the A/D conversion start trigger should be selected with the TRG0S[3:0] and TRG1S[3:0] bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG11S[3:0]				TRG01S[3:0]				TRG1S[3:0]				TRG0S[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	TRG11S [3:0]	0000	R/W	A/D Trigger-1 Group-1 Select Selects an A/D conversion start trigger from among the external trigger, MTU2 triggers, and MTU2S triggers for group 1 of the A/D module 1 in 2-channel scan mode. 0000: Input from the external trigger pin ($\overline{\text{ADTRG}}$) 0001: TGRA input capture/compare match on the channels of the MTU2, TCNT_4 troughs in complementary PWM mode (TRGAN) 0010: CH0 compare match of the MTU2 (TRG0N) 0011: Delayed A/D conversion start request from the MTU2 (TRG4AN) 0100: Delayed A/D conversion start request from the MTU2 (TRG4BN) 0101: Delayed A/D conversion start requests from the MTU2 (TRG4AN and TRG4BN) 0110: TGRA input capture/compare match on the channels of the MTU2S, TCNT_4 troughs in complementary PWM mode (TRGAN) 0111: Delayed A/D conversion start request from the MTU2S (TRG4AN) 1000: Delayed A/D conversion start request from the MTU2S (TRG4BN) 1001: Delayed A/D conversion start requests from the MTU2S (TRG4AN and TRG4BN) 101x: Setting prohibited 11xx: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	TRG01S [3:0]	0000	R/W	A/D Trigger-0 Group-1 Select Selects an A/D conversion start trigger from among the external trigger, MTU2 triggers, and MTU2S triggers for group 1 of the A/D module 0 in 2-channel scan mode. 0000: Input from the external trigger pin ($\overline{\text{ADTRG}}$) 0001: TGRA input capture/compare match on the channels of the MTU2, TCNT_4 troughs in complementary PWM mode (TRGAN) 0010: CH0 compare match of the MTU2 (TRG0N) 0011: Delayed A/D conversion start request from the MTU2 (TRG4AN) 0100: Delayed A/D conversion start request from the MTU2 (TRG4BN) 0101: Delayed A/D conversion start requests from the MTU2 (TRG4AN and TRG4BN) 0110: TGRA input capture/compare match on the channels of the MTU2S, TCNT_4 troughs in complementary PWM mode (TRGAN) 0111: Delayed A/D conversion start request from the MTU2S (TRG4AN) 1000: Delayed A/D conversion start request from the MTU2S (TRG4BN) 1001: Delayed A/D conversion start requests from the MTU2S (TRG4AN and TRG4BN) 101x: Setting prohibited 11xx: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	TRG1S [3:0]	0000	R/W	A/D Trigger-1 Select Selects an A/D conversion start trigger from among the external trigger, MTU2 triggers, and MTU2S triggers for the A/D module 1 (only group 0 of the A/D module 1 in 2-channel scan mode) 0000: Input from the external trigger pin ($\overline{\text{ADTRG}}$) 0001: TGRA input capture/compare match on the channels of the MTU2, TCNT_4 troughs in complementary PWM mode (TRGAN) 0010: CH0 compare match of the MTU2 (TRG0N) 0011: Delayed A/D conversion start request from the MTU2 (TRG4AN) 0100: Delayed A/D conversion start request from the MTU2 (TRG4BN) 0101: Delayed A/D conversion start requests from the MTU2 (TRG4AN and TRG4BN) 0110: TGRA input capture/compare match on the channels of the MTU2S, TCNT_4 troughs in complementary PWM mode (TRGAN) 0111: Delayed A/D conversion start request from the MTU2S (TRG4AN) 1000: Delayed A/D conversion start request from the MTU2S (TRG4BN) 1001: Delayed A/D conversion start requests from the MTU2S (TRG4AN and TRG4BN) 101x: Setting prohibited 11xx: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	TRG0S [3:0]	0000	R/W	A/D Trigger-0 Select Selects an A/D conversion start trigger from among the external trigger, MTU2 triggers, and MTU2S triggers for the A/D module 0 (only group 0 of the A/D module 0 in 2-channel scan mode). 0000: Input from the external trigger pin ($\overline{\text{ADTRG}}$) 0001: TGRA input capture/compare match on the channels of the MTU2, TCNT_4 troughs in complementary PWM mode (TRGAN) 0010: CH0 compare match of the MTU2 (TRG0N) 0011: Delayed A/D conversion start request from the MTU2 (TRG4AN) 0100: Delayed A/D conversion start request from the MTU2 (TRG4BN) 0101: Delayed A/D conversion start requests from the MTU2 (TRG4AN and TRG4BN) 0110: TGRA input capture/compare match on the channels of the MTU2S, TCNT_4 troughs in complementary PWM mode (TRGAN) 0111: Delayed A/D conversion start request from the MTU2S (TRG4AN) 1000: Delayed A/D conversion start request from the MTU2S (TRG4BN) 1001: Delayed A/D conversion start requests from the MTU2S (TRG4AN and TRG4BN) 101x: Setting prohibited 11xx: Setting prohibited

[Legend]

x : Don't care

21.4 Operation

The A/D converter uses the successive approximation method with 10-bit resolution and supports the following operating modes:

- Single mode
- 2-channel scan mode
- 4-channel scan mode
- 8-channel scan mode

For each of the above four modes, single-cycle scan mode and continuous scan mode can be specified. To prevent erroneous operation, the ADST bit in ADCR should be 0 when operating modes or analog input channels are changed.

21.4.1 Single Mode

In single mode, the A/D converter converts analog inputs on the specified channel in the following procedure:

1. When the ADST bit in ADCR is set to 1 by the software, MTU2, MTU2S, or external trigger input, A/D conversion of the specified channel is started.
2. On completion of the A/D conversion, the conversion result is transferred to the A/D data register corresponding to the channel.
3. After the A/D conversion is completed, the ADF bit in ADCSR is set to 1. At this time, if the ADIE bit is 1, the ADI interrupt request is generated.
4. The ADST bit holds 1 during A/D conversion. In single-cycle scan mode, the ADST bit is automatically cleared when the conversion ends, and the A/D converter enters the wait state. In continuous scan mode, the ADST bit is not cleared automatically and the A/D conversion is started again. As long as the ADST bit is 1, steps 2 and 3 are repeated. If the ADST bit is cleared during the A/D conversion, the conversion is stopped and the A/D converter enters the wait state.

Figure 21.3 shows an example of A/D conversion in single-cycle scan mode with AN1 channel selected.

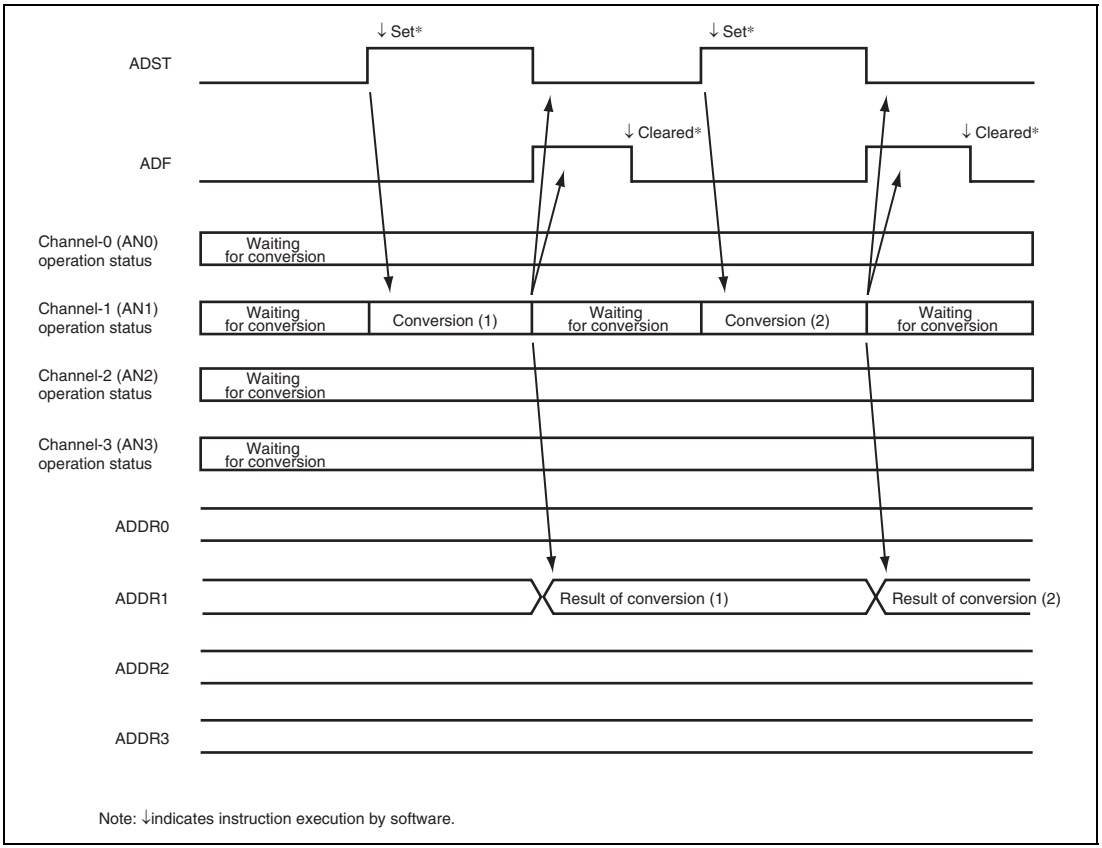


Figure 21.3 Example of A/D Converter Operation in Single Mode (Single-Cycle Scan Mode with AN1 Channel Selected)

21.4.2 2-Channel Scan Mode

In 2-channel scan mode, the A/D converter converts analog inputs on the specified one or two channels in the following procedure:

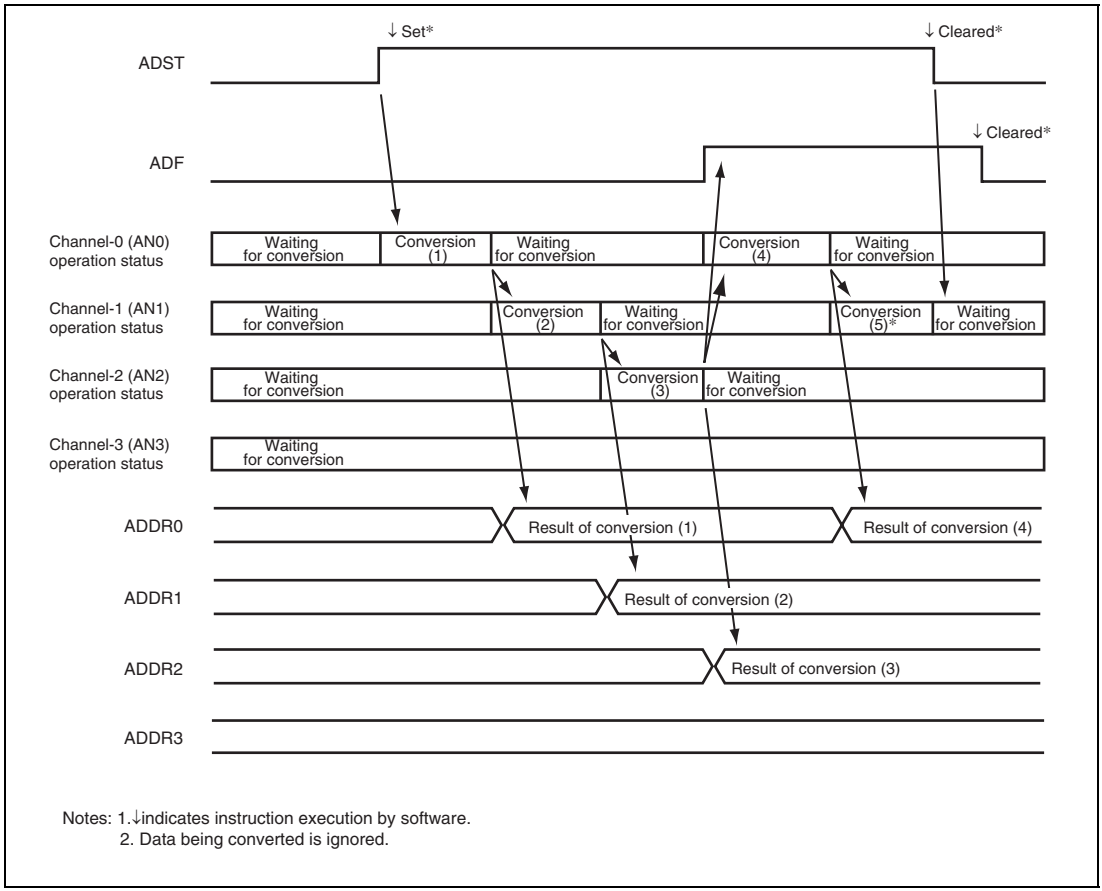
1. When the ADST bit in ADCR is set to 1 by the software, MTU2, MTU2S, or external trigger input, A/D conversion is performed on the channels in numerical order (for example, AN0 and then AN1).
2. On completion of the A/D conversion of each channel, the conversion result is transferred to the A/D data register corresponding to the channel.
3. After the A/D conversion on all the specified channels is completed, the ADF bit in ADCSR is set to 1. At this time, if the ADIE bit is 1, the ADI interrupt request is generated.
4. The ADST bit retains 1 during A/D conversion. In single-cycle scan mode, the ADST bit is automatically cleared when the conversion ends, and the A/D converter enters the wait state. In continuous scan mode, the ADST bit is not cleared automatically and A/D conversion is started again in numerical order (for example, AN0 and then AN1). As long as the ADST bit is 1, steps 2 and 3 are repeated. If the ADST bit is cleared to 0 during A/D conversion, the conversion is stopped and the A/D converter enters the wait state.

21.4.3 4-Channel Scan Mode

In 4-channel scan mode, the A/D converter converts analog inputs on the specified one to four channels in the following procedure:

1. When the ADST bit in ADCR is set to 1 by the software, MTU2, MTU2S, or external trigger input, A/D conversion is performed on the channels in numerical order (for example, AN0, AN1, AN2, and then AN3).
2. On completion of the A/D conversion of each channel, the conversion result is transferred to the A/D data register corresponding to the channel.
3. After the A/D conversion on all the specified channels is completed, the ADF bit in ADCSR is set to 1. At this time, if the ADIE bit is 1, the ADI interrupt request is generated.
4. The ADST bit retains 1 during A/D conversion. In single-cycle scan mode, the ADST bit is automatically cleared when the conversion ends, and the A/D converter enters the wait state. In continuous scan mode, the ADST bit is not cleared automatically and A/D conversion is started again in numerical order (for example, AN0, AN1, AN2, and then AN3). As long as the ADST bit is 1, steps 2 and 3 are repeated. If the ADST bit is cleared to 0 during A/D conversion, the conversion is stopped and the A/D converter enters the wait state.

Figure 21.4 shows an example of A/D conversion in continuous scan mode with three channels AN0 to AN2 selected.



Notes: 1.↓indicates instruction execution by software.
2. Data being converted is ignored.

Figure 21.4 Example of A/D Converter Operation in 4-Channel Scan Mode (Continuous Scan Mode with Three Channels AN0 to AN2 Selected)

21.4.4 8-Channel Scan Mode

In 8-channel scan mode, the A/D converter converts analog inputs on the specified one to eight channels in the following procedure:

1. When the ADST bit in ADCR is set to 1 by the software, MTU2, MTU2S, or external trigger input, A/D conversion is performed on the channels in numerical order (for example, AN0, AN1, AN2, AN3, AN4, AN5, AN6, AN7, and then AN8).
2. On completion of the A/D conversion of each channel, the conversion result is transferred to the A/D data register corresponding to the channel.
3. After the A/D conversion on all the specified channels is completed, the ADF bit in ADCSR is set to 1. At this time, if the ADIE bit is 1, the ADI interrupt request is generated.
4. The ADST bit retains 1 during A/D conversion. In single-cycle scan mode, the ADST bit is automatically cleared when the conversion ends, and the A/D converter enters the wait state. In continuous scan mode, the ADST bit is not cleared automatically and A/D conversion is started again in numerical order (for example, AN0, AN1, AN2, AN3, AN4, AN5, AN6, AN7, and then AN8). As long as the ADST bit is 1, steps 2 and 3 are repeated. If the ADST bit is cleared to 0 during A/D conversion, the conversion is stopped and the A/D converter enters the wait state.

21.4.5 A/D Conversion Activation Source in 2-Channel Scan Mode

In 2-channel scan mode, four channels of analog inputs are divided into two groups: group 0 and group 1. A different trigger can be separately specified for each group as an activation source. The 2-channel scan mode conversion end interrupt can be generated after the conversion on either of group 0 or group 1 is completed or after the conversion on both of group 0 and group 1 is completed. If a group-0 conversion request is generated during group-1 conversion, the group-0 conversion request is ignored. Figure 21.5 shows an example of A/D conversion with TRG4AN of the MTU2 specified as a group-0 conversion start request and TRG4BN of the MTU2 specified as a group-1 conversion start request.

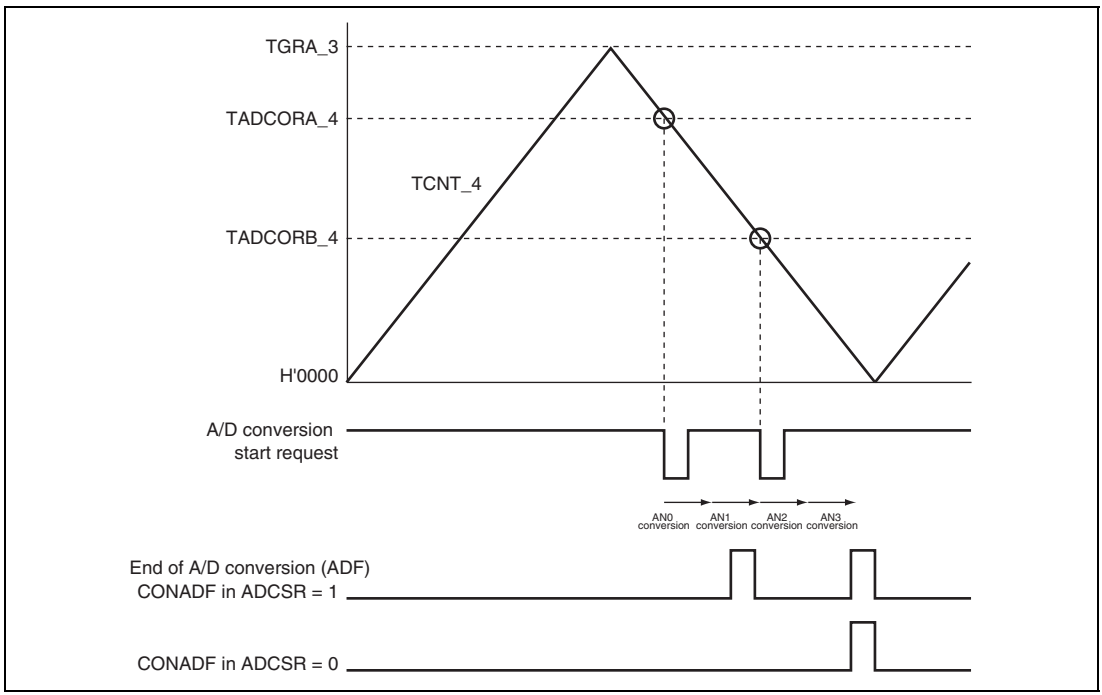


Figure 21.5 Example of 2-Channel Scan Mode Operation

21.4.6 Input Sampling and A/D Conversion Time

Each A/D converter module has a built-in sample-and-hold circuit. When the ADST bit is set to 1, inputs are sampled after the A/D conversion start delay time (t_D) has elapsed, and then A/D conversion is started.

Figure 21.6 shows the timing of A/D conversion and table 21.4 shows the A/D conversion time.

The A/D conversion time (t_{CONV}) includes the A/D conversion start delay time (t_D), sampling time (t_{SPL}), and input voltage comparison and result output time (t_{NC}), as shown in figure 21.6. t_D is determined by the ADCR write timing and is not a fixed value. The A/D conversion time varies within the range shown in table 21.4.

When scan mode is used, the value shown in table 21.4 applies to the first conversion. The second and subsequent conversion time is shown in table 21.5.

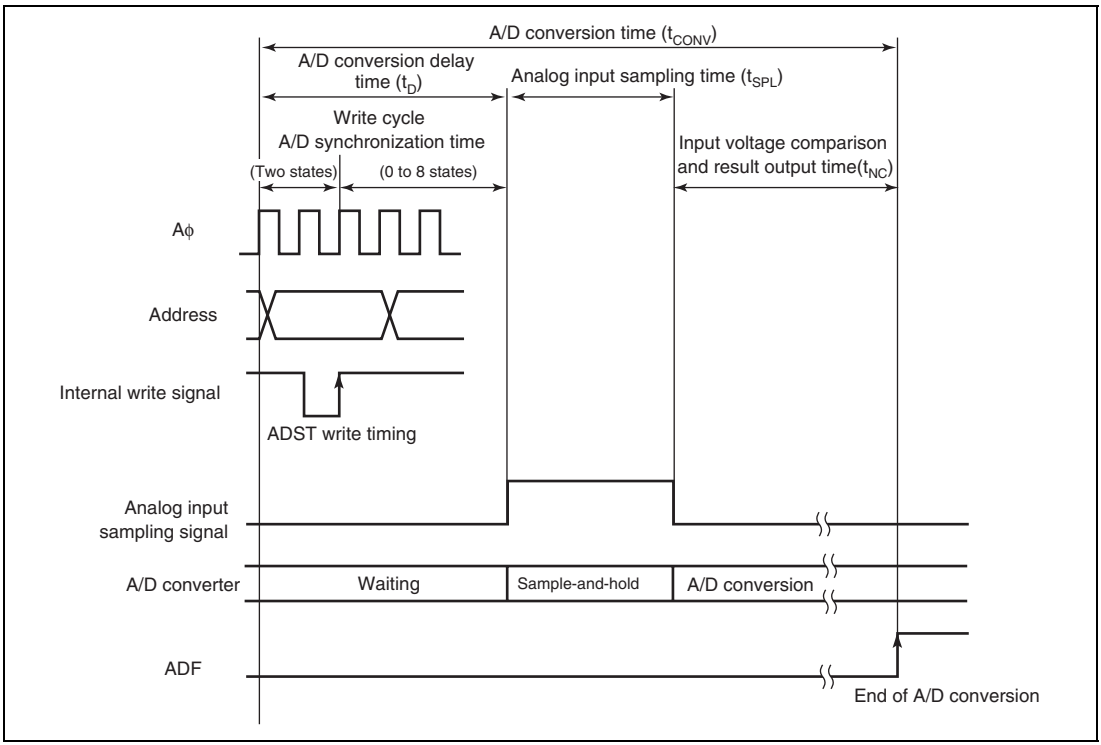


Figure 21.6 A/D Conversion Timing

Table 21.4 A/D Conversion Time (Single Mode)

						Conversion Time (states)			Example of Conversion Time Calculation Result (μs)	
CKS1	CKS0	STC2	STC1	STC0	Conversion Time [t_{CONV}] (states)	Conversion Start Delay [t_D]	Input Sampling [t_{SPL}]	Input Voltage Comparison and Result Output [t_{NC}]	Aϕ = 25 MHz [t_{CONV}]	Aϕ = 50 MHz [t_{CONV}]
0	0	0	0	0	1802 to 1810	2 to 10	1600	200	72.08 to 72.4	36.04 to 36.2
				1	1002 to 1010		800		40.08 to 40.4	20.04 to 20.2
			1	0	906 t0 914		704		36.24 to 36.56	18.12 to 18.28
				1	802 to 810		600		32.08 to 32.4	16.04 to 16.2
		1	0	0	706 to 714		504		28.24 to 28.56	14.12 to 14.28
				1	602 to 610		400		24.08 to 24.4	12.04 to 12.2
			1	0	506 to 514		304		20.24 to 20.56	10.12 to 10.28
				1	402 to 410		200		16.08 to 16.4	8.04 to 8.2
	1	0	0	0	902 to 906	2 to 6	800	100	36.08 to 36.24	18.04 to 18.12
				1	502 to 506		400		20.08 to 20.24	10.04 to 10.12
			1	0	454 to 458		352		18.16 to 18.32	9.08 to 9.16
				1	402 to 406		300		16.08 to 16.24	8.04 to 8.12
		1	0	0	354 to 358		252		14.16 to 14.32	7.08 to 7.16
				1	302 to 306		200		12.08 to 12.24	6.04 to 6.12
			1	0	254 to 258		152		10.16 to 10.32	5.08 to 5.16
				1	202 to 206		100		8.08 to 8.24	4.04 to 4.12

						Conversion Time (states)			Example of Conversion Time Calculation Result (μs)	
CKS1	CKS0	STC2	STC1	STC0	Conversion Time [t_{CONV}] (states)	Conversion Start Delay [t_D]	Input Sampling [t_{SPL}]	Input Voltage Comparison and Result Output [t_{NC}]	Aϕ = 25 MHz [t_{CONV}]	Aϕ = 50 MHz [t_{CONV}]
1	0	0	0	0	452 to 454	2 to 4	400	50	18.08 to 18.16	9.04 to 9.08
				1	252 go 254		200		10.08 to 10.16	5.04 to 5.08
			1	0	228 to 230		176		9.12 to 9.2	4.56 to 4.6
				1	202 to 204		150		8.08 to 8.16	4.04 to 4.08
		1	0	0	178 to 180		126		7.12 to 7.2	3.56 to 3.6
				1	152 to 154		100		6.08 to 6.16	3.04 to 3.08
			1	0	128 to 130		76		5.12 to 5.2	2.56 to 2.6
				1	102 to 104		50		4.08 to 4.16	2.04 to 2.08
	1	0	0	0	227 to 228	2 to 3	200	25	9.08 to 9.12	4.54 to 4.56
				1	127 to 128		100		5.08 to 5.12	2.54 to 2.56
			1	0	115 to 116		88		4.6 to 4.64	2.3 to 2.32
				1	102 to 103		75		4.08 to 4.12	2.04 to 2.06
		1	0	0	90 to 91		63		3.6 to 3.64	1.8 to 1.82
				1	77 to 76		50		3.08 to 3.12	1.54 to 1.56
			1	0	65 to 66		38		2.6 to 2.64	1.3 to 1.32
				1	52 to 53		25		2.08 to 2.12	1.04 to 1.06

Table 21.5 A/D Conversion Time (Scan Mode)

						Conversion Time (states)		Example of Conversion Time Calculation Result (μs)	
CKS1	CKS0	STC2	STC1	STC0	Conversion Time [t_{CONV}] (states)	Input Sampling [t_{SPL}]	Input Voltage Comparison and Result Output [t_{NC}]	Aϕ = 25 MHz [t_{CONV}]	Aϕ = 50 MHz [t_{CONV}]
0	0	0	0	0	1800	1600	200	72	36
				1	1000	800		40	20
			1	0	904	704		36.16	18.08
				1	800	600		32	16
		1	0	0	704	504		28.16	14.08
				1	600	400		24	12
			1	0	504	304		20.16	10.08
				1	400	200		16	8
	1	0	0	0	900	800	100	36	18
				1	500	400		20	10
			1	0	452	352		18.08	9.04
				1	400	300		16	8
		1	0	0	352	252		14.08	7.04
				1	300	200		12.	6
			1	0	252	152		10.08	5.04
				1	200	100		8	4

						Conversion Time (states)		Example of Conversion Time Calculation Result (μs)	
CKS1	CKS0	STC2	STC1	STC0	Conversion Time [t_{CONV}] (states)	Input Sampling [t_{SPL}]	Input Voltage Comparison and Result Output [t_{NC}]	Aϕ = 25 MHz [t_{CONV}]	Aϕ = 50 MHz [t_{CONV}]
1	0	0	0	0	450	400	50	18	9
				1	250	200		10	5
			1	0	226	176		9.04	4.52
				1	200	150		8	4
		1	0	0	176	126		7.04	3.52
				1	150	100		6	3
			1	0	126	76		5.04	2.52
				1	100	50		4	2
	1	0	0	0	225	200	25	9	4.5
				1	125	100		5	2.5
			1	0	113	88		4.52	2.26
				1	100	75		4	2
		1	0	0	88	63		3.52	1.76
				1	75	50		3	1.5
			1	0	63	38		2.52	1.26
				1	50	25		2	1

21.4.7 A/D Converter Activation by MTU2 and MTU2S

Each A/D converter module can be activated independently by an A/D conversion request from the interval timer of the MTU2 or MTU2S.

To activate the A/D converter by an A/D conversion start trigger from the MTU2 or MTU2S, set the TRGE bit in ADCSR to 1 and set ADTSR. After this setting is made, if an A/D conversion start trigger from the MTU2 or MTU2S interval timer is generated, the ADST bit is set to 1. The time between the setting of the ADST bit to 1 and the start of the A/D conversion is the same as when A/D conversion is activated by writing 1 to the ADST bit by software.

21.4.8 External Trigger Input Timing

The A/D conversion can also be externally triggered. An external trigger is input from the $\overline{\text{ADTRG}}$ pin when the TRGE bit in ADCSR is set to 1 and the external trigger pin input is selected with ADTSR_0. A falling edge of the $\overline{\text{ADTRG}}$ signal sets the ADST bit to 1 in ADCR, starting the A/D conversion. Other operations are conducted in the same way as when A/D conversion is activated by writing 1 to the ADST bit by software, regardless whether single mode or scan mode is used. Figure 21.7 shows the timing.

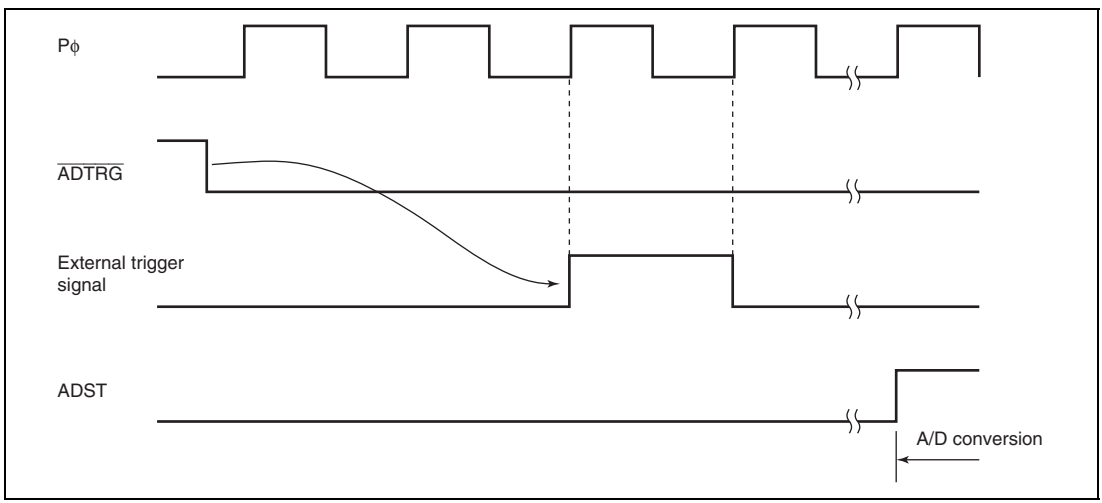


Figure 21.7 External Trigger Input Timing

21.4.9 Accessing A/D Shadow Register (ADSDR)

ADSDR contains the A/D_0 and A/D_1 conversion results selected by ADSSR and the conversion results of A/D_0 and A/D_1 can be read out by a single longword access.

When the AD0FCE and AD1FCE bits in ADSSR are 1, the ADF bits in ADCSR_0 and ADCSR_1 are cleared when ADSDR is accessed by the DMAC/DTC

Figure 21.8 shows the timing of ADSDR access by the DMAC.

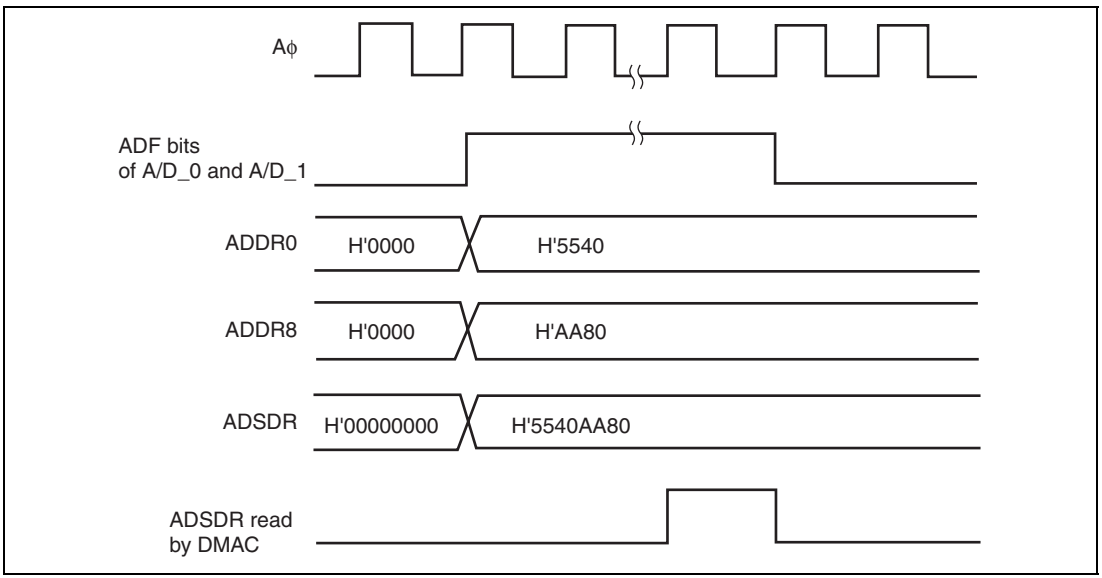


Figure 21.8 Example of ADSDR Access by DMAC (When AD0FCE and AD1FCE in ADSSR are 1)

21.5 Interrupt Sources and DMAC/DTC Transfer Requests

The A/D converter generates A/D conversion end interrupts (ADI). An ADI interrupt generation is enabled when the ADIE bit in ADCSR is set to 1 and disabled when ADIE is cleared to 0.

The DMAC and DTC can be activated to transfer data on generation of an ADI interrupt. When the DMAC is used for data transfer, the ADF bit in ADSR is automatically cleared at reading ADDR or ADSDR, resulting in that an ADI interrupt request is not generated to the CPU. In the same way, when the DTC is used for data transfer and if the DTC's DISEL bit is 0 and transfer counter value is not 0, the ADF bit in ADSR is automatically cleared at reading ADDR or ADSDR, resulting in that an ADI interrupt request is not generated to the CPU. However, if both the DISEL bit and transfer counter values are 0 or if the DISEL bit is 1, the ADF bit in ADSR is not automatically cleared at reading ADDR or ADSDR, resulting in that an ADI interrupt request is generated to the CPU.

By transferring converted data with the DMAC or DTC, continuous conversion can be performed without applying load to the software.

Table 21.6 A/D Converter Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DMAC Activation Request	DTC Activation Request
ADI0	A/D_0 conversion end	ADF in ADCSR_0	Available	Available
ADI1	A/D_1 conversion end	ADF in ADCSR_1	Available	Available

21.6 Definitions of A/D Conversion Accuracy

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 21.9).
- Offset error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic when the digital output value changes from the minimum voltage value B'000000000000 (H'00) to B'000000000001 (H'01). Does not include a quantization error (see figure 21.10).
- Full-scale error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic when the digital output value changes from B'111111111110 (H'3FE) to B'111111111111 (H'3FF) (see figure 21.10).
- Nonlinearity error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 21.10).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

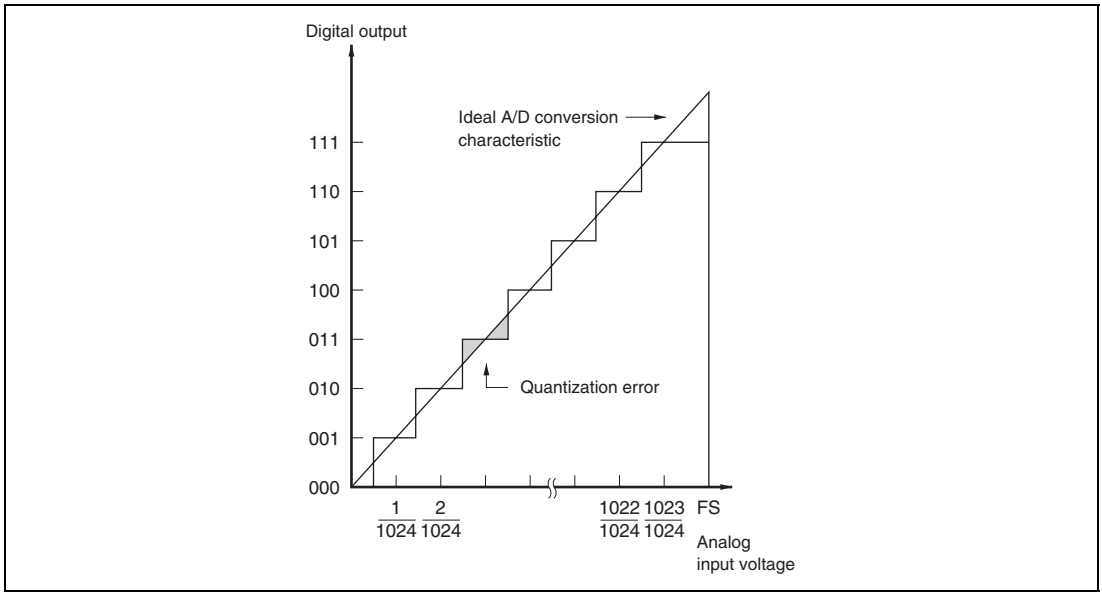


Figure 21.9 Definitions of A/D Conversion Accuracy

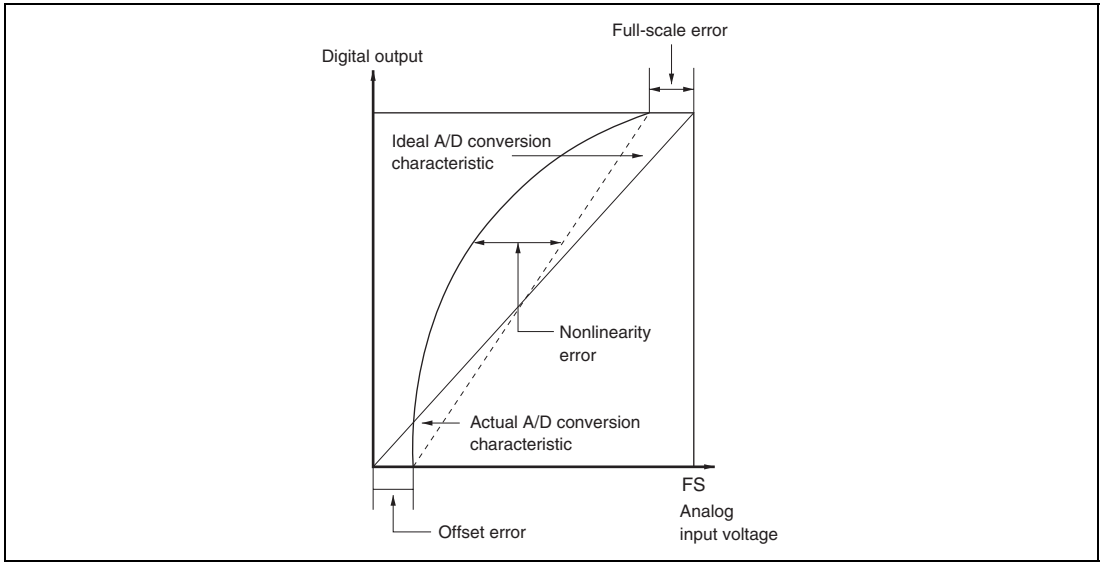


Figure 21.10 Definitions of A/D Conversion Accuracy

21.7 Usage Notes

21.7.1 Module Standby Mode Setting

The operation of the A/D converter modules can be enabled or disabled by the standby control register. In the initial state, the A/D converter operation is disabled. Releasing module standby mode enables accessing registers. For details, see section 32, Power-Down Modes.

21.7.2 A/D Conversion Requests in 2-Channel Scan Mode

If a group-1 conversion request is generated during group-0 conversion, the gourp-1 conversion request is ignored. In the same way, if a group-0 conversion request is generated during group-1 conversion, the gourp-0 conversion request is ignored.

21.7.3 Delayed A/D Conversion Request from MTU2 or MTU2S

When the TRG4AN or TRG4BN from the MTU2 has been selected as the A/D conversion start request and the MTU2 TRG4BN is input during A/D conversion by TRG4AN, the TRG4BN request is ignored. In the same way, if the MTU2 TRG4AN is input during A/D conversion by TRG4BN, the TRG4AN request is ignored.

When the TRG4AN or TRG4BN from the MTU2S has been selected as the A/D conversion start request and the MTU2S TRG4BN is input during A/D conversion by TRG4AN, the TRG4BN request is ignored. In the same way, if the MTU2S TRG4AN is input during A/D conversion by TRG4BN, the TRG4AN request is ignored.

21.7.4 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision at the minimum conversion time is guaranteed for an input signal for which the signal source impedance is 1.3 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 1.3 kΩ or when an analog signal with a large differential coefficient (e.g. greater than 5 mV/µs) within the sampling time is input, charging may be insufficient and the guaranteed precision of A/D conversion may not be possible. However, for A/D conversion in single mode with a large capacitance provided externally for A/D conversion in single mode, the input load will essentially comprise only the internal input resistance of 7 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater) as shown in figure 21.11. When converting a high-speed analog signal or in scan mode, a low-impedance buffer should be inserted.

21.7.5 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

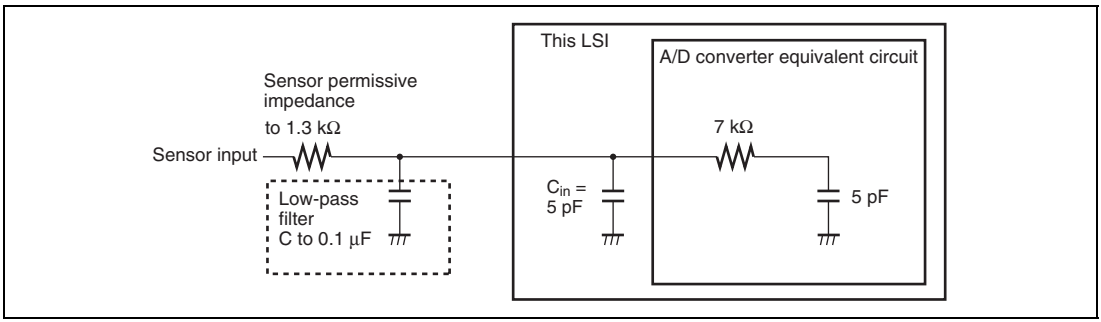


Figure 21.11 Example of Analog Input Circuit

21.7.6 Analog Pin Voltage Ranges

If the following conditions are not met, the reliability of the LSI may be adversely affected.

- Analog Input Voltage Range

  The voltage applied to analog input pin (ANn) during A/D conversion should be in the range AVss ≤ VAN ≤ AVref.
- Relationship between AVcc, AVss and Vcc, Vss

  When using the A/D converter, set AVss = Vss. When the A/D converter is not used, do not leave the AVcc and AVss pins open.
- AVref Input Voltage Range

  The AVref input voltage should be equal to or less than the AVcc. Set AVref = AVcc when the A/D converter is not used.

21.7.7 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values. In addition, digital circuitry must be isolated from the analog input pins (AN0 to AN15) and analog power supply (AVcc) by the analog ground (AVss). AVss should be connected at one point to a stable digital ground (Vss) on the board.

21.7.8 Notes on Noise Countermeasures

To prevent damage of the analog input pins (AN0 to AN15) due to an abnormal voltage, such as an excessive surge, a protection circuit should be connected between the AVcc and AVss, as shown in figure 21.12. The bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN15 should be connected to the AVss.

When a filter capacitor is connected, the input currents at the analog input pin (ANn) are averaged, and an error may occur. If A/D conversion is performed frequently, for example, in scan mode, an error may be generated in the voltage values at the analog input pins when the current charged or discharged to/from the A/D converter built-in sample and hold circuit capacitor exceeds the current level input via the input impedance (Rin)

Careful consideration is therefore required when deciding the circuit constants.

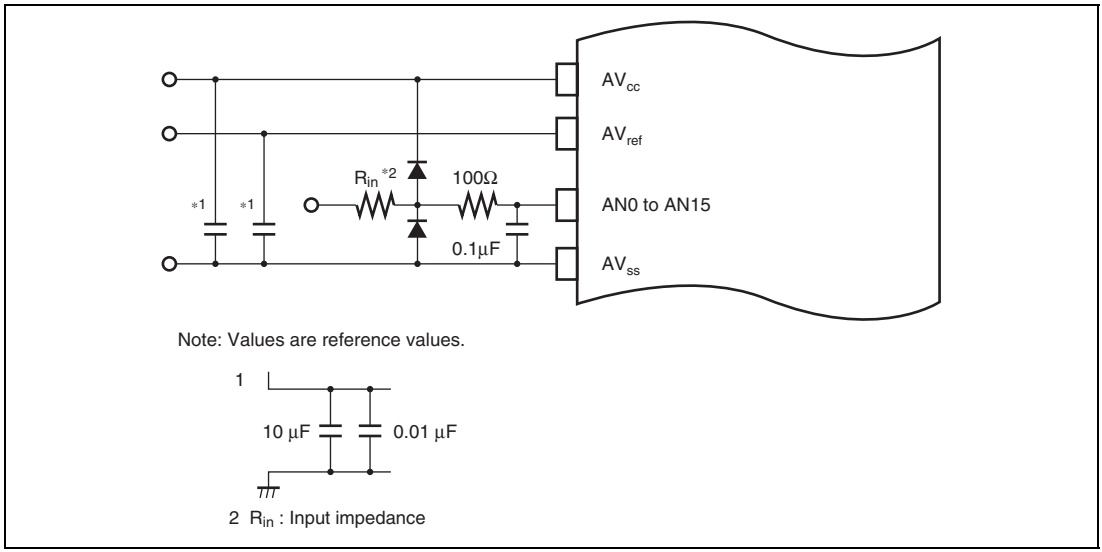


Figure 21.12 Example of Analog Input Pin Protection Circuit

Table 21.7 Analog Input Pin Specifications

Item	Min.	Max.	Unit	Condition
Analog input capacitance	—	5	pF	—
Permissible signal source impedance	—	1.3	kΩ	When the minimum conversion time is set.

Section 22 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of registers that are used to select the functions of multiplexed pins and assign pins to be inputs or outputs. Tables 22.1 to 22.11 list the multiplexed pins of this LSI. Table 22.12 lists the pin functions in each operating mode.

Table 22.1 Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	PA0 I/O (Port)	$\overline{\text{CS0}}$ output (BSC)	IRQ0 input (INTC)	RXD0 input (SCI_0)
	PA1 I/O (Port)	$\overline{\text{CS1}}$ output (BSC)	IRQ1 input (INTC)	TXD0 input (SCI_0)
	PA2 I/O (Port)	$\overline{\text{CS2}}$ output (BSC)	CTx0 output (RCAN-ET)	SCK0 I/O (SCI_0)
	PA3 I/O (Port)	$\overline{\text{CS3}}$ output (BSC)	CTx0 output (RCAN-ET)	SCK1 I/O (SCI_1)
	PA4 I/O (Port)	$\overline{\text{CS4}}$ output (BSC)	CRx0 input (RCAN-ET)	TXD1 output (SCI_1)
	PA5 I/O (Port)	$\overline{\text{CS5}}$ output (BSC)	—	RXD1 input (SCI_1)
	PA6 I/O (Port)	$\overline{\text{CS6}}$ output (BSC)	IRQ2 input (INTC)	TCLKA input (MTU2)
	PA7 I/O (Port)	$\overline{\text{CS7}}$ output (BSC)	IRQ3 input (INTC)	TCLKB input (MTU2)
	PA8 I/O (Port)	RD$\overline{\text{WR}}$ output (BSC)	IRQ4 input (INTC)	TCLKC input (MTU2)
	PA9 I/O (Port)	$\overline{\text{BS}}$ output (BSC)	IRQ5 input (INTC)	TCLKD input (MTU2)
	PA10 I/O (Port)	$\overline{\text{WRHH}}$/DQMUU output (BSC)	IRQ6 input (INTC)	RXD4 input (SCIF_4)
	PA11 I/O (Port)	$\overline{\text{WRHL}}$/DQMUL output (BSC)	IRQ7 input (INTC)	TXD4 output (SCIF_4)
	PA12 I/O (Port)	$\overline{\text{WRH}}$/DQMLU output (BSC)	IRQ8 input (INTC)	SCK4 I/O (SCIF_4)
	PA13 I/O (Port)	$\overline{\text{WRL}}$/DQMLL output (BSC)	IRQ9 input (INTC)	RXD5 input (SCIF_5)
	PA14 I/O (Port)	$\overline{\text{RD}}$ output (BSC)	IRQ10 input (INTC)	TXD5 output (SCIF_5)
	PA15 I/O (Port)	CK output (CPG)	IRQ11 input (INTC)	SCK5 I/O (SCIF_5)
	PA16 I/O (Port)	$\overline{\text{BACK}}$ output (BSC)	IRQ12 input (INTC)	$\overline{\text{ADTRG}}$ input (A/D)
	PA17 I/O (Port)	$\overline{\text{BREQ}}$ input (BSC)	IRQ13 input (INTC)	$\overline{\text{POE8}}$ input (POE2)
	PA18 I/O (Port)	$\overline{\text{WAIT}}$ input (BSC)	IRQ14 input (INTC)	$\overline{\text{POE4}}$ input (POE2)
	PA19 I/O (Port)	$\overline{\text{AH}}$ output (BSC)	IRQ15 input (INTC)	$\overline{\text{POE0}}$ input (POE2)

Table 22.2 Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
B	PB0 I/O (Port)	A16 output (BSC)	$\overline{\text{POE0}}$ input (POE2)	—
	PB1 I/O (Port)	A17 output (BSC)	$\overline{\text{ADTRG}}$ input (A/D)	—
	PB2 I/O (Port)	$\overline{\text{MRES}}$ input (INTC)	$\overline{\text{POE7}}$ input (POE2)	SCL I/O (IIC3)
	PB3 I/O (Port)	$\overline{\text{IRQOUT}}$ output (INTC)	$\overline{\text{POE6}}$ input (POE2)	SDA I/O (IIC3)
	PB4 I/O (Port)	A18 output (BSC)	$\overline{\text{POE5}}$ input (POE2)	RXD0 input (SCI_0)
	PB5 I/O (Port)	A19 output (BSC)	$\overline{\text{POE4}}$ input (POE2)	TXD0 output (SCI_0)
	PB6 I/O (Port)	A20 output (BSC)	$\overline{\text{POE8}}$ input (POE2)	SCK0 I/O (SCI_0)
	PB7 I/O (Port)	$\overline{\text{CS0}}$ output (BSC)	$\overline{\text{CS4}}$ output (BSC)	$\overline{\text{REFOUT}}$ output (BSC)
	PB8 I/O (Port)	$\overline{\text{CS1}}$ output (BSC)	$\overline{\text{CS5}}$ output (BSC)	RXD2 input (SCI_2)
	PB9 I/O (Port)	A21 output (BSC)	CKE output (BSC)	TXD2 output (SCI_2)
	PB10 I/O (Port)	A22 output (BSC)	$\overline{\text{CASL}}$ output (BSC)	SCK2 I/O (SCI_2)
	PB11 I/O (Port)	A23 output (BSC)	$\overline{\text{CASU}}$ output (BSC)	RXD6 input (SCIF_6)
	PB12 I/O (Port)	A24 output (BSC)	$\overline{\text{RASL}}$ output (BSC)	TXD6 output (SCIF_6)
	PB13 I/O (Port)	A25 output (BSC)	$\overline{\text{RASU}}$ output (BSC)	SCK6 I/O (SCIF_6)

Table 22.3 Multiplexed Pins (Port C)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
C	PC0 I/O (Port)	A0 output (BSC)	TIC5U input (MTU2)	$\overline{\text{POE0}}$ input (POE2)
	PC1 I/O (Port)	A1 output (BSC)	TIC5V input (MTU2)	$\overline{\text{POE1}}$ input (POE2)
	PC2 I/O (Port)	A2 output (BSC)	TIC5W input (MTU2)	$\overline{\text{POE2}}$ input (POE2)
	PC3 I/O (Port)	A3 output (BSC)	TIC5WS input (MTU2S)	$\overline{\text{POE3}}$ input (POE2)
	PC4 I/O (Port)	A4 output (BSC)	TIC5VS input (MTU2S)	SCK3 I/O (SCI_3)
	PC5 I/O (Port)	A5 output (BSC)	TIC5US input (MTU2S)	TXD3 output (SCI_3)
	PC6 I/O (Port)	A6 output (BSC)	$\overline{\text{UBCTRG}}$ output (UBC)	RXD3 input (SCI_3)
	PC7 I/O (Port)	A7 output (BSC)	$\overline{\text{IRQOUT}}$ output (INTC)	SCK6 I/O (SCIF_6)
	PC8 I/O (Port)	A8 output (BSC)	IRQ16 input (INTC)	TXD6 output (SCIF_6)
	PC9 I/O (Port)	A9 output (BSC)	IRQ17 input (INTC)	RXD6 input (SCIF_6)
	PC10 I/O (Port)	A10 output (BSC)	IRQ18 input (INTC)	SCK4 I/O (SCIF_4)
	PC11 I/O (Port)	A11 output (BSC)	IRQ19 input (INTC)	TXD4 output (SCIF_4)
	PC12 I/O (Port)	A12 output (BSC)	IRQ20 input (INTC)	RXD4 input (SCIF_4)
	PC13 I/O (Port)	A13 output (BSC)	IRQ21 input (INTC)	SCK7 I/O (SCIF_7)
	PC14 I/O (Port)	A14 output (BSC)	IRQ22 input (INTC)	TXD7 output (SCIF_7)
	PC15 I/O (Port)	A15 output (BSC)	IRQ23 input (INTC)	RXD7 input (SCIF_7)

Table 22.4 Multiplexed Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD0 I/O (Port)	D0 I/O (BSC)	SCK1 I/O (SCI_1)	KEY0 input (KEYC)
	PD1 I/O (Port)	D1 I/O (BSC)	TXD1 output (SCI_1)	KEY1 input (KEYC)
	PD2 I/O (Port)	D2 I/O (BSC)	RXD1 input (SCI_1)	KEY2 input (KEYC)
	PD3 I/O (Port)	D3 I/O (BSC)	SCK7 I/O (SCIF_7)	KEY3 input (KEYC)
	PD4 I/O (Port)	D4 I/O (BSC)	TXD7 output (SCIF_7)	KEY4 input (KEYC)
	PD5 I/O (Port)	D5 I/O (BSC)	RXD7 input (SCIF_7)	KEY5 input (KEYC)
	PD6 I/O (Port)	D6 I/O (BSC)	SCK2 I/O (SCI_2)	KEY6 input (KEYC)
	PD7 I/O (Port)	D7 I/O (BSC)	$\overline{\text{POE8}}$ input (POE2)	KEY7 input (KEYC)
	PD8 I/O (Port)	D8 I/O (BSC)	TXD2 output (SCI_2)	KEY8 input (KEYC)
	PD9 I/O (Port)	D9 I/O (BSC)	RXD2 input (SCI_2)	KEY9 input (KEYC)
	PD10 I/O (Port)	D10 I/O (BSC)	SCK5 I/O (SCIF_5)	KEY10 input (KEYC)
	PD11 I/O (Port)	D11 I/O (BSC)	TXD5 output (SCIF_5)	KEY11 input (KEYC)
	PD12 I/O (Port)	D12 I/O (BSC)	RXD5 input (SCIF_5)	KEY12 input (KEYC)
	PD13 I/O (Port)	D13 I/O (BSC)	SCK3 I/O (SCI_3)	KEY13 input (KEYC)
	PD14 I/O (Port)	D14 I/O (BSC)	TXD3 output (SCI_3)	KEY14 input (KEYC)
	PD15 I/O (Port)	D15 I/O (BSC)	RXD3 input (SCI_3)	KEY15 input (KEYC)
	PD16 I/O (Port)	D16 I/O (BSC)	TCLKA input (MTU2)	KEY16 input/ $\overline{\text{COM0}}$ output (KEYC)
	PD17 I/O (Port)	D17 I/O (BSC)	TCLKB input (MTU2)	KEY17 input/ $\overline{\text{COM1}}$ output (KEYC)
	PD18 I/O (Port)	D18 I/O (BSC)	TCLKC input (MTU2)	KEY18 input/ $\overline{\text{COM2}}$ output (KEYC)
	PD19 I/O (Port)	D19 I/O (BSC)	TCLKD input (MTU2)	KEY19 input/ $\overline{\text{COM3}}$ output (KEYC)
	PD20 I/O (Port)	D20 I/O (BSC)	TIC5WS input (MTU2S)	KEY20 input/ $\overline{\text{COM4}}$ output (KEYC)
	PD21 I/O (Port)	D21 I/O (BSC)	TIC5VS input (MTU2S)	KEY21 input/ $\overline{\text{COM5}}$ output (KEYC)
	PD22 I/O (Port)	D22 I/O (BSC)	TIC5US input (MTU2S)	KEY22 input/ $\overline{\text{COM6}}$ output (KEYC)
	PD23 I/O (Port)	D23 I/O (BSC)	TEND0 output (DMAC)	KEY23 input/ $\overline{\text{COM7}}$ output (KEYC)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD24 I/O (Port)	D24 I/O (BSC)	DREQ0 input (DMAC)	KEY24 input/ $\overline{\text{COM0}}$ output/ P0 input (KEYC)
	PD25 I/O (Port)	D25 I/O (BSC)	DREQ1 input (DMAC)	KEY25 input/ $\overline{\text{COM1}}$ output/ P1 input (KEYC)
	PD26 I/O (Port)	D26 I/O (BSC)	DACK1 output (DMAC)	KEY26 input/ $\overline{\text{COM2}}$ output/ P2 input (KEYC)
	PD27 I/O (Port)	D27 I/O (BSC)	DACK0 output (DMAC)	KEY27 input/ $\overline{\text{COM3}}$ output/ P3 input (KEYC)
	PD28 I/O (Port)	D28 I/O (BSC)	TEND1 output (DMAC)	KEY28 input/P0 input/ P4 input (KEYC)
	PD29 I/O (Port)	D29 I/O (BSC)	TIC5U input (MTU2)	KEY29 input/P1 input/ P5 input (KEYC)
	PD30 I/O (Port)	D30 I/O (BSC)	TIC5V input (MTU2)	KEY30 input/P2 input/ P6 input (KEYC)
	PD31 I/O (Port)	D31 I/O (BSC)	TIC5W input (MTU2)	KEY31 input/P3 input/ P7 input (KEYC)

Table 22.5 Multiplexed Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
E	PE0 I/O (Port)	IRQ0 input (INTC)	TIOC0A I/O (MTU2)	SCK4 I/O (SCIF_4)
	PE1 I/O (Port)	IRQ1 input (INTC)	TIOC0B I/O (MTU2)	TXD4 output (SCIF_4)
	PE2 I/O (Port)	IRQ2 input (INTC)	TIOC0C I/O (MTU2)	RXD4 input (SCIF_4)
	PE3 I/O (Port)	IRQ3 input (INTC)	TIOC0D I/O (MTU2)	SCK3 I/O (SCI_3)
	PE4 I/O (Port)	IRQ4 input (INTC)	TIOC1A I/O (MTU2)	RXD3 input (SCI_3)
	PE5 I/O (Port)	IRQ5 input (INTC)	TIOC1B I/O (MTU2)	TXD3 output (SCI_3)
	PE6 I/O (Port)	IRQ6 input (INTC)	TIOC2A I/O (MTU2)	TXD2 output (SCI_2)
	PE7 I/O (Port)	IRQ7 input (INTC)	TIOC2B I/O (MTU2)	RXD2 input (SCI_2)
	PE8 I/O (Port)	IRQ8 input (INTC)	TIOC3A I/O (MTU2)	SCK2 I/O (SCI_2)
	PE9 I/O (Port)	IRQ9 input (INTC)	TIOC3B I/O (MTU2)	SCK1 I/O (SCI_1)
	PE10 I/O (Port)	IRQ10 input (INTC)	TIOC3C I/O (MTU2)	TXD1 output (SCI_1)
	PE11 I/O (Port)	IRQ11 input (INTC)	TIOC3D I/O (MTU2)	RXD1 input (SCI_1)
	PE12 I/O (Port)	IRQ12 input (INTC)	TIOC4A I/O (MTU2)	SCK5 I/O (SCIF_5)
	PE13 I/O (Port)	IRQ13 input (INTC)	TIOC4B I/O (MTU2)	TXD5 output (SCIF_5)
	PE14 I/O (Port)	IRQ14 input (INTC)	TIOC4C I/O (MTU2)	RXD5 input (SCIF_5)
	PE15 I/O (Port)	IRQ15 input (INTC)	TIOC4D I/O (MTU2)	SCK0 I/O (SCI_0)
	PE16 I/O (Port)	$\overline{\text{UBCTRG}}$ output (UBC)	TIOC3AS I/O (MTU2S)	TXD0 output (SCI_0)
	PE17 I/O (Port)	$\overline{\text{MRES}}$ input (INTC)	TIOC3BS I/O (MTU2S)	RXD0 input (SCI_0)
	PE18 I/O (Port)	DREQ0 input (DMAC)	TIOC3CS I/O (MTU2S)	SCK6 I/O (SCIF_6)
	PE19 I/O (Port)	DACK0 output (DMAC)	TIOC3DS I/O (MTU2S)	TXD6 output (SCIF_6)
	PE20 I/O (Port)	TEND0 output (DMAC)	TIOC4AS I/O (MTU2S)	RXD6 input (SCIF_6)
	PE21 I/O (Port)	DREQ1 input (DMAC)	TIOC4BS I/O (MTU2S)	SCK7 I/O (SCIF_7)
	PE22 I/O (Port)	DACK1 output (DMAC)	TIOC4CS I/O (MTU2S)	TXD7 output (SCIF_7)
	PE23 I/O (Port)	TEND1 output (DMAC)	TIOC4DS I/O (MTU2S)	RXD7 input (SCIF_7)

Table 22.6 Multiplexed Pins (Port F)

Port	Function 1 (Related Module)	Function 2 (Related Module)
F	PF0 input (Port)	AN0 input (A/D_0)
	PF1 input (Port)	AN1 input (A/D_0)
	PF2 input (Port)	AN2 input (A/D_0)
	PF3 input (Port)	AN3 input (A/D_0)
	PF4 input (Port)	AN4 input (A/D_0)
	PF5 input (Port)	AN5 input (A/D_0)
	PF6 input (Port)	AN6 input (A/D_0)
	PF7 input (Port)	AN7 input (A/D_0)
	PF8 input (Port)	AN8 input (A/D_1)
	PF9 input (Port)	AN9 input (A/D_1)
	PF10 input (Port)	AN10 input (A/D_1)
	PF11 input (Port)	AN11 input (A/D_1)
	PF12 input (Port)	AN12 input (A/D_1)
	PF13 input (Port)	AN13 input (A/D_1)
	PF14 input (Port)	AN14 input (A/D_1)
	PF15 input (Port)	AN15 input (A/D_1)

Note: AN input function is valid during A/D sampling.

Table 22.7 Multiplexed Pins (Port G)

Port	Function 1 (Related Module)	Function 2 (Related Module)
G	PG0 I/O (Port)	IRQ0 input (INTC)
	PG1 I/O (Port)	IRQ1 input (INTC)
	PG2 I/O (Port)	IRQ2 input (INTC)
	PG3 I/O (Port)	IRQ3 input (INTC)
	PG4 I/O (Port)	IRQ4 input (INTC)
	PG5 I/O (Port)	IRQ5 input (INTC)
	PG6 I/O (Port)	IRQ6 input (INTC)
	PG7 I/O (Port)	IRQ7 input (INTC)
	PG8 I/O (Port)	IRQ8 input (INTC)
	PG9 I/O (Port)	IRQ9 input (INTC)
	PG10 I/O (Port)	TI32I0A input (TIM32C)
	PG11 I/O (Port)	TI32I0B input (TIM32C)
	PG12 I/O (Port)	TI32I1A input (TIM32C)
	PG13 I/O (Port)	TI32I1B input (TIM32C)
	PG14 I/O (Port)	CK32 output (32-kHz clock)
	PG15 I/O (Port)	—

Table 22.8 Multiplexed Pins (Port H)

Port	Function 1 (Related Module)	Function 2 (Related Module)
H	PH0 I/O (Port)	TIC5U input (MTU2)
	PH1 I/O (Port)	TIC5V input (MTU2)
	PH2 I/O (Port)	TIC5W input (MTU2)
	PH3 I/O (Port)	TIC5US input (MTU2S)
	PH4 I/O (Port)	TIC5VS input (MTU2S)
	PH5 I/O (Port)	TIC5WS input (MTU2S)
	PH6 I/O (Port)	SCK7 I/O (SCIF_7)
	PH7 I/O (Port)	TXD7 output (SCIF_7)
	PH8 I/O (Port)	RXD7 input (SCIF_7)
	PH9 I/O (Port)	—
	PH10 I/O (Port)	—
	PH11 I/O (Port)	—
	PH12 I/O (Port)	TIC0 input (CMT2)
	PH13 I/O (Port)	TIC1 input (CMT2)
	PH14 I/O (Port)	TOC0 output (CMT2)
	PH15 I/O (Port)	TOC1 output (CMT2)

Table 22.9 Multiplexed Pins (Port J)

Port	Function 1 (Related Module)	Function 2 (Related Module)
J	PJ0 I/O (Port)	IRQ10 input (INTC)
	PJ1 I/O (Port)	IRQ11 input (INTC)
	PJ2 I/O (Port)	IRQ12 input (INTC)
	PJ3 I/O (Port)	SCK4 I/O (SCIF_4)
	PJ4 I/O (Port)	TXD4 output (SCIF_4)
	PJ5 I/O (Port)	RXD4 input (SCIF_4)
	PJ6 I/O (Port)	SCK5 I/O (SCIF_5)
	PJ7 I/O (Port)	TXD5 output (SCIF_5)
	PJ8 I/O (Port)	RXD5 input (SCIF_5)
	PJ9 I/O (Port)	RSPCK0 I/O (RSPI_0)
	PJ10 I/O (Port)	MOSI0 I/O (RSPI_0)
	PJ11 I/O (Port)	MISO0 I/O (RSPI_0)
	PJ12 I/O (Port)	SSL0 I/O (RSPI_0)
	PJ13 I/O (Port)	SSL1 output (RSPI_0)
	PJ14 I/O (Port)	SSL2 output (RSPI_0)
	PJ15 I/O (Port)	SSL3 output (RSPI_0)

Table 22.10 Multiplexed Pins (Port K)

Port	Function 1 (Related Module)	Function 2 (Related Module)
K	PK0 I/O (Port)	IRQ13 input (INTC)
	PK1 I/O (Port)	IRQ14 input (INTC)
	PK2 I/O (Port)	IRQ15 input (INTC)
	PK3 I/O (Port)	SCK6 I/O (SCIF_6)
	PK4 I/O (Port)	TXD6 output (SCIF_6)
	PK5 I/O (Port)	RXD6 input (SCIF_6)
	PK6 I/O (Port)	—
	PK7 I/O (Port)	—

Table 22.11 Multiplexed Pins (Port L)

Port	Function 1 (Related Module)	Function 2 (Related Module)
L	PL0 input (Port)	RXCLKINP input (LVDS) (SH72315A only)
	PL1 input (Port)	RXCLKINM input (LVDS) (SH72315A only)
	PL2 input (Port)	RXIN0P input (LVDS) (SH72315A only)
	PL3 input (Port)	RXIN0M input (LVDS) (SH72315A only)
	PL4 input (Port)	RXIN1P input (LVDS) (SH72315A only)
	PL5 input (Port)	RXIN1M input (LVDS) (SH72315A only)

Note: The LVDS input function is valid when LVDS module standby mode is canceled (the MSTP67 bit in the standby control register 6 (STBCR6) is 0). (SH72315A only)

Table 22.12 Pin Functions in Each Operating Mode

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
A16, B4, B7, C20, F21, J17, M20, P21, R2, U12, U20, V2, Y7, Y17, AA4, AA10, AA14, AA19	A2, A6, C19, D14, D20, G20, M20, P17, R1, U7, U9, V2, W16, W20, Y4, Y11, Y14, Y19	Vcc				—
A5, A7, B18, B21, H17, L20, R1, R21, U11, U21, Y2, Y6, Y16, AA5, AA9, AA13, AA20	A3, A5, B19, D15, H20, K10, K11, L10, L11, L19, R17, T1, U6, V20, W2, W19, Y3, Y7, Y10, Y13, Y16	Vss				—
D1	E4	PVcc1				—
E1	F4	PVss1				—
H1	F1	PVcc2				—
G1	G1	PVss2				—
A20	A19	PLLVcc				—
A19	A18	PLLVss				—
M4	L3	LVDSVcc (SH72315A), Vcc (SH72315L/SH72314L)				—
J2, M2	J2, M2	LVDSVss (SH72315A), Vss (SH72315L/SH72314L)				—
M17, T1, U7	N17, R2, Y5	V_{CL}				—
A12	A13	AVcc				—
A13	A14	AVss				—
E12	A12	AVref				—
C21	B20	EXTAL				—
D21	C20	XTAL				—

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name				
		Default Function				
		On-Chip ROM Disabled		On-Chip ROM Enabled	Single Chip	Function Specified with PFC
		MCU Mode 0	MCU Mode 1	MCU Mode 2	MCU Mode 3	
Y1	W1	EXTAL32				—
W1	V1	XTAL32				—
E15	A16	MD0				—
E10	C10	MD1				—
G21	E20	$\overline{\text{RES}}$				—
B10	B10	$\overline{\text{WDTOVF}}$				—
J21	J18	NMI				—
F18	F19	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$				—
H20	H19	$\overline{\text{ASEMD0}}$				—
G20	G18	TCK				—
G17	F20	TMS				—
H18	H18	TDI				—
G18	G19	TDO				—
H21	H17	$\overline{\text{TRST}}$				—
A18	D17	AUDATA3				—
E17	D16	AUDATA2				—
A17	B18	AUDATA1				—
D17	C16	AUDATA0				—
B17	C17	AUDCK				—
B19	E17	$\overline{\text{AUDSYNC}}$				—
AA17	V16	PA0	PA0	PA0	PA0	PA0/$\overline{\text{CS0}}$*1/ IRQ0/RXD0
Y18	U16	PA1	PA1	PA1	PA1	PA1/$\overline{\text{CS1}}$*1/ IRQ1/TXD0
V18	V17	PA2	PA2	PA2	PA2	PA2/$\overline{\text{CS2}}$*1/ CTx0/SCK0
W20	U18	PA3	PA3	PA3	PA3	PA3/$\overline{\text{CS3}}$*1/ CTx0/SCK1

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
U17	T17	PA4	PA4	PA4	PA4	PA4/$\overline{CS4}$*1/ CRx0/TXD1
U18	T18	PA5	PA5	PA5	PA5	PA5/$\overline{CS5}$*1/ RXD1
T17	U20	PA6	PA6	PA6	PA6	PA6/$\overline{CS6}$*1/ IRQ2/ TCLKA
R20	P18	PA7	PA7	PA7	PA7	PA7/$\overline{CS7}$*1/ IRQ3/ TCLKB
T18	T19	PA8	PA8	PA8	PA8	PA8/ RD$\overline{WR}$*1/ IRQ4/ TCLKC
T20	T20	PA9	PA9	PA9	PA9	PA9/$\overline{BS}$*1/ IRQ5/ TCLKD
AA18	W17	PA10	$\overline{WRHH}$	PA10	PA10	PA10/ $\overline{WRHH}$*1/ DQMUU*1/ IRQ6/RXD4
Y19	Y18	PA11	$\overline{WRHL}$	PA11	PA11	PA11/ $\overline{WRHL}$*1/ DQMUL*1/ IRQ7/TXD4
AA21	W18	$\overline{WRH}$	$\overline{WRH}$	PA12	PA12	PA12/ $\overline{WRH}$*1/ DQMLU*1/ IRQ8/SCK4
Y21	Y20	$\overline{WRL}$	$\overline{WRL}$	PA13	PA13	PA13/ $\overline{WRL}$*1/ DQMLL*1/ IRQ9/RXD5
Y20	V18	$\overline{RD}$	$\overline{RD}$	PA14	PA14	PA14/$\overline{RD}$*1/ IRQ10/TXD5

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
L21	L20	CK	CK	CK	PA15	PA15/CK/ IRQ11/SCK5
T21	R18	PA16	PA16	PA16	PA16	PA16/ $\overline{\text{BACK}}$*1/ IRQ12/ $\overline{\text{ADTRG}}$
W21	U17	PA17	PA17	PA17	PA17	PA17/ $\overline{\text{BREQ}}$*1/ IRQ13/ $\overline{\text{POE8}}$
V20	U19	PA18	PA18	PA18	PA18	PA18/ $\overline{\text{WAIT}}$*1/ IRQ14/ $\overline{\text{POE4}}$
V21	V19	PA19	PA19	PA19	PA19	PA19/$\overline{\text{AH}}$*1/ IRQ15/ $\overline{\text{POE0}}$
Y4	V4	A16	A16	PB0	PB0	PB0/A16*1/ $\overline{\text{POE0}}$
AA6	W5	A17	A17	PB1	PB1	PB1/A17*1/ $\overline{\text{ADTRG}}$
Y5	V3	PB2	PB2	PB2	PB2	PB2/$\overline{\text{MRES}}$/ $\overline{\text{POE7}}$/SCL
V5	V5	PB3	PB3	PB3	PB3	PB3/ $\overline{\text{IRQOUT}}$/ $\overline{\text{POE6}}$/SDA
V1	U2	A18	A18	PB4	PB4	PB4/A18*1/ $\overline{\text{POE5}}$/RXD0
W2	T4	A19	A19	PB5	PB5	PB5/A19*1/ $\overline{\text{POE4}}$/TXD0
V4	R4	A20	A20	PB6	PB6	PB6/A20*1/ $\overline{\text{POE8}}$/SCK0

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
U6	W4	$\overline{CS0}$	$\overline{CS0}$	PB7	PB7	PB7/$\overline{CS0}$*1/ $\overline{CS4}$*1/ $\overline{REFOUT}$*1
U5	Y2	PB8	PB8	PB8	PB8	PB8/$\overline{CS1}$*1/ $\overline{CS5}$*1/RXD2
V6	V6	PB9	PB9	PB9	PB9	PB9/A21*1/ CKE*1/TXD2
AA2	Y1	PB10	PB10	PB10	PB10	PB10/A22*1/ $\overline{CASL}$*1/ SCK2
AA1	U4	PB11	PB11	PB11	PB11	PB11/A23*1/ $\overline{CASU}$*1/ RXD6
AA3	U5	PB12	PB12	PB12	PB12	PB12/A24*1/ $\overline{RASL}$*1/ TXD6
Y3	W3	PB13	PB13	PB13	PB13	PB13/A25*1/ $\overline{RASU}$*1/ SCK6
P20	P20	A0	A0	PC0	PC0	PC0/A0*1/ TIC5U/ $\overline{POE0}$
N21	L17	A1	A1	PC1	PC1	PC1/A1*1/ TIC5V/$\overline{POE1}$
N20	N20	A2	A2	PC2	PC2	PC2/A2*1/ TIC5W/ $\overline{POE2}$
R18	R20	A3	A3	PC3	PC3	PC3/A3*1/ TIC5WS/ $\overline{POE3}$
P18	P19	A4	A4	PC4	PC4	PC4/A4*1/ TIC5VS/ SCK3

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
N18	N19	A5	A5	PC5	PC5	PC5/A5*1/ TIC5US/ TXD3
M21	M18	A6	A6	PC6	PC6	PC6/A6*1/ $\overline{\text{UBCTRG}}$/ RXD3
R17	R19	A7	A7	PC7	PC7	PC7/A7*1/ $\overline{\text{IRQOUT}}$/ SCK6
P17	M17	A8	A8	PC8	PC8	PC8/A8*1/ IRQ16/TXD6
N17	N18	A9	A9	PC9	PC9	PC9/A9*1/ IRQ17/ RXD6
M18	M19	A10	A10	PC10	PC10	PC10/A10*1/ IRQ18/SCK4
L18	K17	A11	A11	PC11	PC11	PC11/A11*1/ IRQ19/TXD4
L17	L18	A12	A12	PC12	PC12	PC12/A12*1/ IRQ20/ RXD4
K17	K20	A13	A13	PC13	PC13	PC13/A13*1/ IRQ21/SCK7
K18	K19	A14	A14	PC14	PC14	PC14/A14*1/ IRQ22/TXD7
K20	J20	A15	A15	PC15	PC15	PC15/A15*1/ IRQ23/ RXD7
AA7	U8	D0	D0	PD0	PD0	PD0/D0*1/ SCK1/KEY0
AA8	Y6	D1	D1	PD1	PD1	PD1/D1*1/ TXD1/KEY1
Y8	W7	D2	D2	PD2	PD2	PD2/D2*1/ RXD1/KEY2

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
Y9	Y8	D3	D3	PD3	PD3	PD3/D3*1/ SCK7/KEY3
Y10	Y9	D4	D4	PD4	PD4	PD4/D4*1/ TXD7/KEY4
AA11	V10	D5	D5	PD5	PD5	PD5/D5*1/ RXD7/KEY5
Y11	W10	D6	D6	PD6	PD6	PD6/D6*1/ SCK2/KEY6
Y12	V11	D7	D7	PD7	PD7	PD7/D7*1/ $\overline{\text{POE8}}$/KEY7
AA12	Y12	D8	D8	PD8	PD8	PD8/D8*1/ TXD2/KEY8
Y13	U12	D9	D9	PD9	PD9	PD9/D9*1/ RXD2/KEY9
Y14	W13	D10	D10	PD10	PD10	PD10/D10*1/ SCK5/ KEY10
U14	V13	D11	D11	PD11	PD11	PD11/D11*1/ TXD5/ KEY11
V15	Y15	D12	D12	PD12	PD12	PD12/D12*1/ RXD5/ KEY12
AA15	W15	D13	D13	PD13	PD13	PD13/D13*1/ SCK3/ KEY13
V16	Y17	D14	D14	PD14	PD14	PD14/D14*1/ TXD3/ KEY14
AA16	U14	D15	D15	PD15	PD15	PD15/D15*1/ RXD3/ KEY15

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
V7	W6	PD16	D16	PD16	PD16	PD16/D16*1/ TCLKA/ KEY16/ $\overline{\text{COM0}}$
U8	V7	PD17	D17	PD17	PD17	PD17/D17*1/ TCLKB/ KEY17/ $\overline{\text{COM1}}$
V8	V8	PD18	D18	PD18	PD18	PD18/D18*1/ TCLKC/ KEY18/ $\overline{\text{COM2}}$
U9	W8	PD19	D19	PD19	PD19	PD19/D19*1/ TCLKD/ KEY19/ $\overline{\text{COM3}}$
V9	V9	PD20	D20	PD20	PD20	PD20/D20*1/ TIC5WS/ KEY20/ $\overline{\text{COM4}}$
U10	W9	PD21	D21	PD21	PD21	PD21/D21*1/ TIC5VS/ KEY21/ $\overline{\text{COM5}}$
V10	U10	PD22	D22	PD22	PD22	PD22/D22*1/ TIC5US/ KEY22/ $\overline{\text{COM6}}$
V11	W11	PD23	D23	PD23	PD23	PD23/D23*1/ TEND0*1/ KEY23/ $\overline{\text{COM7}}$

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
V12	U11	PD24	D24	PD24	PD24	PD24/D24*1/ DREQ0/ KEY24/ $\overline{\text{COM0}}$/P0
V13	W12	PD25	D25	PD25	PD25	PD25/D25*1/ DREQ1/ KEY25/ $\overline{\text{COM1}}$/P1
U13	V12	PD26	D26	PD26	PD26	PD26/D26*1/ DACK1*1/ KEY26/ $\overline{\text{COM2}}$/P2
V14	U13	PD27	D27	PD27	PD27	PD27/D27*1/ DACK0*1/ KEY27/ $\overline{\text{COM3}}$/P3
Y15	W14	PD28	D28	PD28	PD28	PD28/D28*1/ TEND1*1/ KEY28/P0/ P4
U15	V14	PD29	D29	PD29	PD29	PD29/D29*1/ TIC5U/ KEY29/P1/ P5
U16	V15	PD30	D30	PD30	PD30	PD30/D30*1/ TIC5V/ KEY30/P2/ P6
V17	U15	PD31	D31	PD31	PD31	PD31/D31*1/ TIC5W/ KEY31/P3/ P7
A10	A9	PE0	PE0	PE0	PE0	PE0/IRQ0/ TIOC0A/ SCK4

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	On-Chip ROM Disabled: MCU Mode 1	On-Chip ROM Enabled: MCU Mode 2	Single Chip: MCU Mode 3	Function Specified with PFC
D9	D9	PE1	PE1	PE1	PE1	PE1/IRQ1/ TIOC0B/ TXD4
B9	B9	PE2	PE2	PE2	PE2	PE2/IRQ2/ TIOC0C/ RXD4
E9	C9	PE3	PE3	PE3	PE3	PE3/IRQ3/ TIOC0D/ SCK3
A9	A8	PE4	PE4	PE4	PE4	PE4/IRQ4/ TIOC1A/ RXD3
B8	B8	PE5	PE5	PE5	PE5	PE5/IRQ5/ TIOC1B/ TXD3
D8	A7	PE6	PE6	PE6	PE6	PE6/IRQ6/ TIOC2A/ TXD2
A8	D8	PE7	PE7	PE7	PE7	PE7/IRQ7/ TIOC2B/ RXD2
E8	C8	PE8	PE8	PE8	PE8	PE8/IRQ8/ TIOC3A/ SCK2
D7	B7	PE9	PE9	PE9	PE9	PE9/IRQ9/ TIOC3B/ SCK1
E7	C7	PE10	PE10	PE10	PE10	PE10/ IRQ10/ TIOC3C/ TXD1

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
B6	B6	PE11	PE11	PE11	PE11	PE11/ IRQ11/ TIOC3D/ RXD1
B5	D7	PE12	PE12	PE12	PE12	PE12/ IRQ12/ TIOC4A/ SCK5
D5	C5	PE13	PE13	PE13	PE13	PE13/ IRQ13/ TIOC4B/ TXD5
D4	B4	PE14	PE14	PE14	PE14	PE14/ IRQ14/ TIOC4C/ RXD5
D6	B5	PE15	PE15	PE15	PE15	PE15/ IRQ15/ TIOC4D/ SCK0
E6	A4	PE16	PE16	PE16	PE16	PE16/ $\overline{\text{UBCTRG}}$/ TIOC3AS/ TXD0
A6	C6	PE17	PE17	PE17	PE17	PE17/ $\overline{\text{MRES}}$/ TIOC3BS/ RXD0
B2	C3	PE18	PE18	PE18	PE18	PE18/ DREQ0/ TIOC3CS/ SCK6

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
A3	A1	PE19	PE19	PE19	PE19	PE19/ DACK0*1/ TIOC3DS/ TXD6
A2	D5	PE20	PE20	PE20	PE20	PE20/ TEND0*1/ TIOC4AS/ RXD6
A4	C4	PE21	PE21	PE21	PE21	PE21/ DREQ1/ TIOC4BS/ SCK7
B3	B3	PE22	PE22	PE22	PE22	PE22/ DACK1*1/ TIOC4CS/ TXD7
A1	D6	PE23	PE23	PE23	PE23	PE23/ TEND1*1/ TIOC4DS/ RXD7
B15	C14	PF0/AN0				—*2
D15	B15	PF1/AN1				—*2
A14	A15	PF2/AN2				—*2
E14	D13	PF3/AN3				—*2
B14	B14	PF4/AN4				—*2
D14	C13	PF5/AN5				—*2
E13	B13	PF6/AN6				—*2
B13	D12	PF7/AN7				—*2
D13	C12	PF8/AN8				—*2
B12	D11	PF9/AN9				—*2
A11	B12	PF10/AN10				—*2
D12	C11	PF11/AN11				—*2

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	MCU Mode 1	On-Chip ROM Enabled: MCU Mode 2	Single Chip: MCU Mode 3	Function Specified with PFC
E11	B11	PF12/AN12				—*2
B11	A11	PF13/AN13				—*2
D11	A10	PF14/AN14				—*2
D10	D10	PF15/AN15				—*2
N2	L4	PG0	PG0	PG0	PG0	PG0/IRQ0
N4	N1	PG1	PG1	PG1	PG1	PG1/IRQ1
N1	N2	PG2	PG2	PG2	PG2	PG2/IRQ2
N5	P1	PG3	PG3	PG3	PG3	PG3/IRQ3
P2	N3	PG4	PG4	PG4	PG4	PG4/IRQ4
P4	P2	PG5	PG5	PG5	PG5	PG5/IRQ5
P1	M3	PG6	PG6	PG6	PG6	PG6/IRQ6
P5	M4	PG7	PG7	PG7	PG7	PG7/IRQ7
R4	P3	PG8	PG8	PG8	PG8	PG8/IRQ8
R5	R3	PG9	PG9	PG9	PG9	PG9/IRQ9
T2	T3	PG10	PG10	PG10	PG10	PG10/ TI32I0A
T4	N4	PG11	PG11	PG11	PG11	PG11/ TI32I0B
T5	P4	PG12	PG12	PG12	PG12	PG12/ TI32I1A
U2	T2	PG13	PG13	PG13	PG13	PG13/ TI32I1B
U1	U3	PG14	PG14	PG14	PG14	PG14/CK32
U4	U1	PG15	PG15	PG15	PG15	PG15
J18	K18	PH0	PH0	PH0	PH0	PH0/TIC5U
K21	J19	PH1	PH1	PH1	PH1	PH1/TIC5V
J20	K17	PH2	PH2	PH2	PH2	PH2/TIC5W
F20	G17	PH3	PH3	PH3	PH3	PH3/TIC5US
F17	F18	PH4	PH4	PH4	PH4	PH4/TIC5VS

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name				
		Default Function				
		On-Chip ROM Disabled		On-Chip ROM Enabled	Single Chip	Function Specified with PFC
		MCU Mode 0	MCU Mode 1	MCU Mode 2	MCU Mode 3	
E18	E19	PH5	PH5	PH5	PH5	PH5/ TIC5WS
E20	E18	PH6	PH6	PH6	PH6	PH6/SCK7
D20	D19	PH7	PH7	PH7	PH7	PH7/TXD7
E21	D18	PH8	PH8	PH8	PH8	PH8/RXD7
D18	F17	PH9	PH9	PH9	PH9	PH9
A21	A20	PH10	PH10	PH10	PH10	PH10
B20	C18	PH11	PH11	PH11	PH11	PH11
E16	B17	PH12	PH12	PH12	PH12	PH12/TIC0
B16	A17	PH13	PH13	PH13	PH13	PH13/TIC1
D16	C15	PH14	PH14	PH14	PH14	PH14/TOC0
A15	B16	PH15	PH15	PH15	PH15	PH15/TOC1
B1	B2	PJ0	PJ0	PJ0	PJ0	PJ0/IRQ10
C2	D4	PJ1	PJ1	PJ1	PJ1	PJ1/IRQ11
E5	B1	PJ2	PJ2	PJ2	PJ2	PJ2/IRQ12
D2	C1	PJ3	PJ3	PJ3	PJ3	PJ3/SCK4
C1	D3	PJ4	PJ4	PJ4	PJ4	PJ4/TXD4
F5	E3	PJ5	PJ5	PJ5	PJ5	PJ5/RXD4
E4	C2	PJ6	PJ6	PJ6	PJ6	PJ6/SCK5
G5	F3	PJ7	PJ7	PJ7	PJ7	PJ7/TXD5
E2	D1	PJ8	PJ8	PJ8	PJ8	PJ8/RXD5
F4	D2	PJ9	PJ9	PJ9	PJ9	PJ9/ RSPCK0
H5	E1	PJ10	PJ10	PJ10	PJ10	PJ10/MOSI0
F2	G4	PJ11	PJ11	PJ11	PJ11	PJ11/MISO0
G4	E2	PJ12	PJ12	PJ12	PJ12	PJ12/SSL0
J5	F2	PJ13	PJ13	PJ13	PJ13	PJ13/SSL1
F1	G3	PJ14	PJ14	PJ14	PJ14	PJ14/SSL2

Pin No. (P-LFBGA 1111-256)	Pin No. (P-FBGA 1717-272)	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 0	Pin Name: Default Function: On-Chip ROM Disabled: MCU Mode 1	Pin Name: Default Function: On-Chip ROM Enabled: MCU Mode 2	Pin Name: Default Function: Single Chip: MCU Mode 3	Function Specified with PFC
G2	H4	PJ15	PJ15	PJ15	PJ15	PJ15/SSL3
H4	H3	PK0	PK0	PK0	PK0	PK0/IRQ13
J4	J4	PK1	PK1	PK1	PK1	PK1/IRQ14
H2	J3	PK2	PK2	PK2	PK2	PK2/IRQ15
K5	G2	PK3	PK3	PK3	PK3	PK3/SCK6
L5	H2	PK4	PK4	PK4	PK4	PK4/TXD6
M5	K3	PK5	PK5	PK5	PK5	PK5/RXD6
K4	H1	PK6	PK6	PK6	PK6	PK6
L4	K4	PK7	PK7	PK7	PK7	PK7
K2	K2	PL0/RXCLKINP (SH72315A), PL0 (SH72315L/SH72314L)				—*3
L2	L2	PL1/RXCLKINM (SH72315A), PL1 (SH72315L/SH72314L)				—*3
J1	J1	PL2/RXIN0P (SH72315A), PL2 (SH72315L/SH72314L)				—*3
K1	K1	PL3/RXIN0M (SH72315A), PL3 (SH72315L/SH72314L)				—*3
L1	L1	PL4/RXIN1P (SH72315A), PL4 (SH72315L/SH72314L)				—*3
M1	M1	PL5/RXIN1M (SH72315A), PL5 (SH72315L/SH72314L)				—*3

Notes: 1. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

2. AN input function is valid during A/D sampling.

3. For the SH72315A, the general input function is valid when the LVDS is in the module standby state, and the LVDS input function is valid when the LVDS is not in the module standby state. For the SH72315L/SH72314L, the general input function is always valid.

22.1 Register Descriptions

The PFC has the following registers. See section 34, List of Registers for register addresses and register states in each operating mode.

Table 22.13 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A I/O register H	PAIORH	R/W	H'0000	H'FFFE3804	8, 16
Port A I/O register L	PAIORL	R/W	H'0000	H'FFFE3806	8, 16
Port A control register H1	PACRH1	R/W	H'0000	H'FFFE380A	8, 16
Port A control register L2	PACRL2	R/W	H'0000*	H'FFFE380C	8, 16
Port A control register L1	PACRL1	R/W	H'0000	H'FFFE380E	8, 16
Port A pull-up MOS control register H	PAPCRH	R/W	H'0000	H'FFFE3820	8, 16
Port A pull-up MOS control register L	PAPCRL	R/W	H'0000	H'FFFE3822	8, 16
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE3886	8, 16
Port B control register L2	PBCRL2	R/W	H'0000	H'FFFE388C	8, 16
Port B control register L1	PBCRL1	R/W	H'0000*	H'FFFE388E	8, 16
Port B pull-up MOS control register L	PBPCRL	R/W	H'0000	H'FFFE38A2	8, 16
Port C I/O register L	PCIORL	R/W	H'0000	H'FFFE3906	8, 16
Port C control register L2	PCCRL2	R/W	H'0000*	H'FFFE390C	8, 16
Port C control register L1	PCCRL1	R/W	H'0000*	H'FFFE390E	8, 16
Port C pull-up MOS control register L	PCPCRL	R/W	H'0000	H'FFFE3922	8, 16
Port D I/O register H	PDIORH	R/W	H'0000	H'FFFE3984	8, 16
Port D I/O register L	PDIORL	R/W	H'0000	H'FFFE3986	8, 16
Port D control register H2	PDCRH2	R/W	H'0000*	H'FFFE3988	8, 16
Port D control register H1	PDCRH1	R/W	H'0000*	H'FFFE398A	8, 16
Port D control register L2	PDCRL2	R/W	H'0000*	H'FFFE398C	8, 16
Port D control register L1	PDCRL1	R/W	H'0000*	H'FFFE398E	8, 16
Port D pull-up MOS control register H	PDPCRH	R/W	H'0000	H'FFFE39A0	8, 16
Port D pull-up MOS control register L	PDPCRL	R/W	H'0000	H'FFFE39A2	8, 16
Port E I/O register H	PEIORH	R/W	H'0000	H'FFFE3A04	8, 16
Port E I/O register L	PEIORL	R/W	H'0000	H'FFFE3A06	8, 16
Port E control register H1	PECRH1	R/W	H'0000	H'FFFE3A0A	8, 16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E control register L2	PECRL2	R/W	H'0000	H'FFFE3A0C	8, 16
Port E control register L1	PECRL1	R/W	H'0000	H'FFFE3A0E	8, 16
Port E pull-up MOS control register H	PEPCRH	R/W	H'0000	H'FFFE3A20	8, 16
Port E pull-up MOS control register L	PEPCRL	R/W	H'0000	H'FFFE3A22	8, 16
Port G I/O register L	PGIORL	R/W	H'0000	H'FFFE3B06	8, 16
Port G control register L1	PGCRL1	R/W	H'0000	H'FFFE3B0E	8, 16
Port G pull-up MOS control register L	PGPCRL	R/W	H'0000	H'FFFE3B22	8, 16
Port H I/O register L	PHIORL	R/W	H'0000	H'FFFE3B86	8, 16
Port H control register L1	PHCRL1	R/W	H'0000	H'FFFE3B8E	8, 16
Port H pull-up MOS control register L	PHPCRL	R/W	H'0000	H'FFFE3BA2	8, 16
Port J I/O register L	PJIORL	R/W	H'0000	H'FFFE3C86	8, 16
Port J control register L1	PJCRL1	R/W	H'0000	H'FFFE3C8E	8, 16
Port J pull-up MOS control register L	PJPCRL	R/W	H'0000	H'FFFE3CA2	8, 16
Port K I/O register L	PKIORL	R/W	H'0000	H'FFFE3D06	8, 16
Port K control register L1	PKCRL1	R/W	H'0000	H'FFFE3D0E	8, 16
Port K pull-up MOS control register L	PKPCRL	R/W	H'0000	H'FFFE3D22	8, 16
Port L pull-up MOS control register L	PLPCRL	R/W	H'0000	H'FFFE3DA2	8, 16
Large current port control register	HCPCR	R/W	H'0003	H'FFFE3A14	8, 16
I/O buffer driver control register	DRVCR	R/W	H'A000	H'FFFE39A8	8, 16
Port function extension register	PFEXCR	R/W	H'0000	H'FFFE3BA8	8, 16

Note: * The initial values of registers vary according to the setting of the operating mode. See the description of each register in this section for details.

22.1.1 Port A I/O Registers L and H (PAIORL and PAIORH)

PAIORL and PAIORH are 16-bit readable/writable registers that are used to set the pins on port A as inputs or outputs. Bits PA19IOR to PA0IOR correspond to pins PA19 to PA0 (multiplexed port pin names except for the port names are abbreviated here). PAIORL and PAIORH are enabled when the port A pins are functioning as general-purpose inputs/outputs (PA19 to PA16 for PAIORH and PA15 to PA0 for PAIORL). In other states, they are disabled.

A given pin on port A will be an output pin if the corresponding bit in PAIORL or PAIORH is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

Bits 15 to 4 of PAIORH are reserved. These bits are always read as 0. The write value should always be 0. The initial values of PAIORL and PAIORH are both H'0000.

- Port A I/O Register H (PAIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PA19 IOR	PA18 IOR	PA17 IOR	PA16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

- Port A I/O Register L (PAIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.2 Port A Control Registers L1, L2, and H1 (PACRL1, PACRL2, and PACRH1)

PACRL1, PACRL2, and PACRH1 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port A.

- Port A Control Register H1 (PACRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PA19 MD1	PA19 MD0	PA18 MD1	PA18 MD0	PA17 MD1	PA17 MD0	PA16 MD1	PA16 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	PA19MD[1:0]	00	R/W	PA19 Mode Select the function of the PA19/$\overline{\text{AH}}$/IRQ15/$\overline{\text{POE0}}$ pin. 00: PA19 I/O (port) 01: $\overline{\text{AH}}$ output (BSC)* 10: IRQ15 input (INTC) 11: $\overline{\text{POE0}}$ input (POE2)
5, 4	PA18MD[1:0]	00	R/W	PA18 Mode Select the function of the PA18/$\overline{\text{WAIT}}$/IRQ14/$\overline{\text{POE4}}$ pin. 00: PA18 I/O (port) 01: $\overline{\text{WAIT}}$ input (BSC)* 10: IRQ14 input (INTC) 11: $\overline{\text{POE4}}$ input (POE2)
3, 2	PA17MD[1:0]	00	R/W	PA17 Mode Select the function of the PA17/$\overline{\text{BREQ}}$/IRQ13/$\overline{\text{POE8}}$ pin. 00: PA17 I/O (port) 01: $\overline{\text{BREQ}}$ input (BSC)* 10: IRQ13 input (INTC) 11: $\overline{\text{POE8}}$ input (POE2)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PA16MD[1:0]	00	R/W	PA16 Mode Select the function of the PA16/$\overline{\text{BACK}}$/IRQ12/$\overline{\text{ADTRG}}$ pin. 00: PA16 I/O (port) 01: $\overline{\text{BACK}}$ output (BSC)* 10: IRQ12 input (INTC) 11: $\overline{\text{ADTRG}}$ input (A/D)

Note: * This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port A Control Register L2 (PACRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 MD1	PA15 MD0	PA14 MD1	PA14 MD0	PA13 MD1	PA13 MD0	PA12 MD1	PA12 MD0	PA11 MD1	PA11 MD0	PA10 MD1	PA10 MD0	PA9 MD1	PA9 MD0	PA8 MD1	PA8 MD0
Initial value:	0	0*1	0	0*2	0	0*2	0	0*2	0	0*3	0	0*3	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. The initial value is 1 in on-chip ROM enabled/disabled external extension mode.
2. The initial value is 1 in on-chip ROM disabled external extension mode.
3. The initial value is 1 in on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PA15MD [1:0]	00*1	R/W	PA15 Mode Select the function of the PA15/CK/IRQ11/SCK5 pin. 00: PA15 I/O (port) 01: CK output (CPG) 10: IRQ11 input (INTC) 11: SCK5 I/O (SCIF_5)
13, 12	PA14MD [1:0]	00*2	R/W	PA14 Mode Select the function of the PA14/$\overline{RD}$/IRQ10/TXD5 pin. 00: PA14 I/O (port) 01: $\overline{RD}$ output (BSC)*4 10: IRQ10 input (INTC) 11: TXD5 output (SCIF_5)
11, 10	PA13MD [1:0]	00*2	R/W	PA13 Mode Select the function of the PA13/$\overline{WRL}$/DQMLL/IRQ9/RXD5 pin. 00: PA13 I/O (port) 01: $\overline{WRL}$/DQMLL output (BSC)*4 10: IRQ9 input (INTC) 11: RXD5 input (SCIF_5)
9, 8	PA12MD [1:0]	00*2	R/W	PA12 Mode Select the function of the PA12/$\overline{WRH}$/DQMLU/IRQ8/SCK4 pin. 00: PA12 I/O (port) 01: $\overline{WRH}$/DQMLU output (BSC)*4 10: IRQ8 input (INTC) 11: SCK4 I/O (SCIF_4)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PA11MD [1:0]	00*3	R/W	PA11 Mode Select the function of the PA11/$\overline{\text{WRHL}}$/DQMUL/IRQ7/TXD4 pin. 00: PA11 I/O (port) 01: $\overline{\text{WRHL}}$/DQMUL output (BSC)*4 10: IRQ7 input (INTC) 11: TXD4 output (SCIF_4)
5, 4	PA10MD [1:0]	00*3	R/W	PA10 Mode Select the function of the PA10/$\overline{\text{WRHH}}$/DQMUU/IRQ6/RXD4 pin. 00: PA10 I/O (port) 01: $\overline{\text{WRHH}}$/DQMUU output (BSC)*4 10: IRQ6 input (INTC) 11: RXD4 input (SCIF_4)
3, 2	PA9MD [1:0]	00	R/W	PA9 Mode Select the function of the PA9/$\overline{\text{BS}}$/IRQ5/TCLKD pin. 00: PA9 I/O (port) 01: $\overline{\text{BS}}$ output (BSC)*4 10: IRQ5 input (INTC) 11: TCLKD input (MTU2)
1, 0	PA8MD [1:0]	00	R/W	PA8 Mode Select the function of the PA8/RD$\overline{\text{WR}}$/IRQ4/TCLKC pin. 00: PA8 I/O (port) 01: RD$\overline{\text{WR}}$ output (BSC)*4 10: IRQ4 input (INTC) 11: TCLKC input (MTU2)

Notes: 1. The initial value is 01 in on-chip ROM enabled/disabled external extension mode.
2. The initial value is 01 in on-chip ROM disabled external extension mode.
3. The initial value is 01 in on-chip ROM disabled 32-bit external extension mode.
4. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port A Control Register L1 (PACRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA7 MD1	PA7 MD0	PA6 MD1	PA6 MD0	PA5 MD1	PA5 MD0	PA4 MD1	PA4 MD0	PA3 MD1	PA3 MD0	PA2 MD1	PA2 MD0	PA1 MD1	PA1 MD0	PA0 MD1	PA0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PA7MD[1:0]	00	R/W	PA7 Mode Select the function of the PA7/$\overline{CS7}$/IRQ3/TCLKB pin. 00: PA7 I/O (port) 01: $\overline{CS7}$ output (BSC)* 10: IRQ3 input (INTC) 11: TCLKB input (MTU2)
13, 12	PA6MD[1:0]	00	R/W	PA6 Mode Select the function of the PA6/$\overline{CS6}$/IRQ2/TCLKA pin. 00: PA6 I/O (port) 01: $\overline{CS6}$ output (BSC)* 10: IRQ2 input (INTC) 11: TCLKA input (MTU2)
11, 10	PA5MD[1:0]	00	R/W	PA5 Mode Select the function of the PA5/$\overline{CS5}$/RXD1 pin. 00: PA5 I/O (port) 01: $\overline{CS5}$ output (BSC)* 10: Setting prohibited 11: RXD1 input (SCI_1)
9, 8	PA4MD[1:0]	00	R/W	PA4 Mode Select the function of the PA4/$\overline{CS4}$/CRx0/TXD1 pin. 00: PA4 I/O (port) 01: $\overline{CS4}$ output (BSC)* 10: CRx0 input (RCAN-ET) 11: TXD1 output (SCI_1)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PA3MD[1:0]	00	R/W	PA3 Mode Select the function of the PA3/$\overline{CS3}$/CTx0/SCK1 pin. 00: PA3 I/O (port) 01: $\overline{CS3}$ output (BSC)* 10: CTx0 output (RCAN-ET) 11: SCK1 I/O (SCI_1)
5, 4	PA2MD[1:0]	00	R/W	PA2 Mode Select the function of the PA2/$\overline{CS2}$/CTx0/SCK0 pin. 00: PA2 I/O (port) 01: $\overline{CS2}$ output (BSC)* 10: CTx0 output (RCAN-ET) 11: SCK0 I/O (SCI_0)
3, 2	PA1MD[1:0]	00	R/W	PA1 Mode Select the function of the PA1/$\overline{CS1}$/IRQ1/TXD0 pin. 00: PA1 I/O (port) 01: $\overline{CS1}$ output (BSC)* 10: IRQ1 input (INTC) 11: TXD0 output (SCI_0)
1, 0	PA0MD[1:0]	00	R/W	PA0 Mode Select the function of the PA0/$\overline{CS0}$/IRQ0/RXD0 pin. 00: PA0 I/O (port) 01: $\overline{CS0}$ output (BSC)* 10: IRQ0 input (INTC) 11: RXD0 input (SCI_0)

Note: * This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

22.1.3 Port A Pull-Up MOS Control Registers L and H (PAPCRL and PAPCRH)

PAPCRL and PAPCRH control on and off of the input pull-up MOS of port A in bits.

- Port A Pull-Up MOS Control Register H (PAPCRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PA19 PCR	PA18 PCR	PA17 PCR	PA16 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	PA19PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
2	PA18PCR	0	R/W	
1	PA17PCR	0	R/W	
0	PA16PCR	0	R/W	

- Port A Pull-Up MOS Control Register L (PAPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PCR	PA14 PCR	PA13 PCR	PA12 PCR	PA11 PCR	PA10 PCR	PA9 PCR	PA8 PCR	PA7 PCR	PA6 PCR	PA5 PCR	PA4 PCR	PA3 PCR	PA2 PCR	PA1 PCR	PA0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PA14PCR	0	R/W	
13	PA13PCR	0	R/W	
12	PA12PCR	0	R/W	
11	PA11PCR	0	R/W	
10	PA10PCR	0	R/W	
9	PA9PCR	0	R/W	
8	PA8PCR	0	R/W	
7	PA7PCR	0	R/W	
6	PA6PCR	0	R/W	
5	PA5PCR	0	R/W	
4	PA4PCR	0	R/W	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

22.1.4 Port B I/O Register L (PBIORL)

PBIORL is a 16-bit readable/writable register that is used to set the pins on port B as inputs or outputs. Bits PB13IOR to PB0IOR correspond to pins PB13 to PB0, respectively (multiplexed port pin names except for the port names are abbreviated here). PBIORL is enabled when the port B pins are functioning as general-purpose inputs/outputs (PB13 to PB0). In other states, PBIORL is disabled.

A given pin on port B will be an output pin if the corresponding bit in PBIORL is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

Bits 15 and 14 of PAIORL are reserved. These bits are always read as 0. The write value should always be 0. The initial value of PBIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.5 Port B Control Registers L1 and L2 (PBCRL1 and PBCRL2)

PBCRL1 and PBCRL2 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port B.

- Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PB13 MD1	PB13 MD0	PB12 MD1	PB12 MD0	PB11 MD1	PB11 MD0	PB10 MD1	PB10 MD0	PB9 MD1	PB9 MD0	PB8 MD1	PB8 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are read as 0. The write value should always be 0.
11, 10	PB13MD [1:0]	00	R/W	PB13 Mode Select the function of the PB13/A25/$\overline{\text{RASU}}$/SCK6 pin. 00: PB13 I/O (port) 01: A25 output (BSC)* 10: $\overline{\text{RASU}}$ output (BSC)* 11: SCK6 I/O(SCIF_6)
9, 8	PB12MD [1:0]	00	R/W	PB12 Mode Select the function of the PB12/A24/$\overline{\text{RASL}}$/TXD6 pin. 00: PB12 I/O (port) 01: A24 output (BSC)* 10: $\overline{\text{RASL}}$ output (BSC)* 11: TXD6 output (SCIF_6)
7, 6	PB11MD [1:0]	00	R/W	PB11 Mode Select the function of the PB11/A23/$\overline{\text{CASU}}$/RXD6 pin. 00: PB11 I/O (port) 01: A23 output (BSC)* 10: $\overline{\text{CASU}}$ output (BSC)* 11: RXD6 input (SCIF_6)

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB10MD [1:0]	00	R/W	PB10 Mode Select the function of the PB10/A22/$\overline{\text{CASL}}$/SCK2 pin. 00: PB10 I/O (port) 01: A22 output (BSC)* 10: $\overline{\text{CASL}}$ output (BSC)* 11: SCK2 I/O (SCI_2)
3, 2	PB9MD [1:0]	00	R/W	PB9 Mode Select the function of the PB9/A21/CKE/TXD2 pin. 00: PB9 I/O (port) 01: A21 output (BSC)* 10: CKE output (BSC)* 11: TXD2 output (SCI_2)
1, 0	PB8MD [1:0]	00	R/W	PB8 Mode Select the function of the PB8/$\overline{\text{CS1}}$/$\overline{\text{CS5}}$/RXD2 pin. 00: PB8 I/O (port) 01: $\overline{\text{CS1}}$ output (BSC)* 10: $\overline{\text{CS5}}$ output (BSC)* 11: RXD2 input (SCI_2)

Note: * This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port B Control Register L1 (PBCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB7 MD1	PB7 MD0	PB6 MD1	PB6 MD0	PB5 MD1	PB5 MD0	PB4 MD1	PB4 MD0	PB3 MD1	PB3 MD0	PB2 MD1	PB2 MD0	PB1 MD1	PB1 MD0	PB0 MD1	PB0 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0	0	0	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PB7MD[1:0]	00*[1]	R/W	PB7 Mode Select the function of the PB7/$\overline{\text{CS0}}$/$\overline{\text{CS4}}$/$\overline{\text{REFOUT}}$ pin. 00: PB7 I/O (port) 01: $\overline{\text{CS0}}$ output (BSC)*[2] 10: $\overline{\text{CS4}}$ output (BSC)*[2] 11: $\overline{\text{REFOUT}}$ output (BSC)*[2]
13, 12	PB6MD[1:0]	00*[1]	R/W	PB6 Mode Select the function of the PB6/A20/$\overline{\text{POE8}}$/SCK0 pin. 00: PB6 I/O (port) 01: A20 output (BSC)*[2] 10: $\overline{\text{POE8}}$ input (POE2) 11: SCK0 I/O (SCI_0)
11, 10	PB5MD[1:0]	00*[1]	R/W	PB5 Mode Select the function of the PB5/A19/$\overline{\text{POE4}}$/TXD0 pin. 00: PB5 I/O (port) 01: A19 output (BSC)*[2] 10: $\overline{\text{POE4}}$ input (POE2) 11: TXD0 output (SCI_0)
9, 8	PB4MD[1:0]	00*[1]	R/W	PB4 Mode Select the function of the PB4/A18/$\overline{\text{POE5}}$/RXD0 pin. 00: PB4 I/O (port) 01: A18 output (BSC)*[2] 10: $\overline{\text{POE5}}$ input (POE2) 11: RXD0 input (SCI_0)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PB3MD[1:0]	00	R/W	PB3 Mode Select the function of the PB3/$\overline{\text{IRQOUT}}$/$\overline{\text{POE6}}$/SDA pin. 00: PB3 I/O (port) 01: $\overline{\text{IRQOUT}}$ output (INTC) 10: $\overline{\text{POE6}}$ input (POE2) 11: SDA I/O (IIC3)
5, 4	PB2MD[1:0]	00	R/W	PB2 Mode Select the function of the PB2/$\overline{\text{MRES}}$/$\overline{\text{POE7}}$/SCL pin. 00: PB2 I/O (port) 01: $\overline{\text{MRES}}$ input (INTC) 10: $\overline{\text{POE7}}$ input (POE2) 11: SCL I/O (IIC3)
3, 2	PB1MD[1:0]	00*[1]	R/W	PB1 Mode Select the function of the PB1/A17/$\overline{\text{ADTRG}}$ pin. 00: PB1 I/O (port) 01: A17 output (BSC)*[2] 10: $\overline{\text{ADTRG}}$ input (A/D) 11: Setting prohibited
1, 0	PB0MD[1:0]	00*[1]	R/W	PB0 Mode Select the function of the PB0/A16/$\overline{\text{POE0}}$ pin. 00: PB0 I/O (port) 01: A16 output (BSC)*[2] 10: $\overline{\text{POE0}}$ input (POE2) 11: Setting prohibited

Notes: 1. The initial value is 01 in on-chip ROM disabled external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

22.1.6 Port B Pull-Up MOS Control Register L (PBPCRL)

PBPCRL controls on/off of the input pull-up MOS of port B in bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB13 PCR	PB12 PCR	PB11 PCR	PB10 PCR	PB9 PCR	PB8 PCR	PB7 PCR	PB6 PCR	PB5 PCR	PB4 PCR	PB3 PCR	PB2 PCR	PB1 PCR	PB0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are read as 0. The write value should always be 0.
13	PB13PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
12	PB12PCR	0	R/W	
11	PB11PCR	0	R/W	
10	PB10PCR	0	R/W	
9	PB9PCR	0	R/W	
8	PB8PCR	0	R/W	
7	PB7PCR	0	R/W	
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

22.1.7 Port C I/O Register L (PCIORL)

PCIORL is a 16-bit readable/writable register that is used to set the pins on port C as inputs or outputs. Bits PC15IOR to PC0IOR correspond to pins PC15 to PC0, respectively (multiplexed port pin names except for the port names are abbreviated here). PCIORL is enabled when the port C pins are functioning as general-purpose inputs/outputs (PC15 to PC0). In other states, PCIORL is disabled.

A given pin on port C will be an output pin if the corresponding bit in PCIORL is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

The initial value of PCIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.8 Port C Control Registers L1 and L2 (PCCRL1 and PCCRL2)

PCCRL1 and PACRL2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port C.

- Port C Control Register L2 (PCCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 MD1	PC15 MD0	PC14 MD1	PC14 MD0	PC13 MD1	PC13 MD0	PC12 MD1	PC12 MD0	PC11 MD1	PC11 MD0	PC10 MD1	PC10 MD0	PC9 MD1	PC9 MD0	PC8 MD1	PC8 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PC15MD[1:0]	00*[1]	R/W	PC15 Mode Select the function of the PC15/A15/IRQ23/RXD7 pin. 00: PC15 I/O (port) 01: A15 output (BSC)*[2] 10: IRQ23 input (INTC) 11: RXD7 input (SCIF_7)
13, 12	PC14MD[1:0]	00*[1]	R/W	PC14 Mode Select the function of the PC14/A14/IRQ22/TXD7 pin. 00: PC14 I/O (port) 01: A14 output (BSC)*[2] 10: IRQ22 input (INTC) 11: TXD7 output (SCIF_7)
11, 10	PC13MD[1:0]	00*[1]	R/W	PC13 Mode Select the function of the PC13/A13/IRQ21/SCK7 pin. 00: PC13 I/O (port) 01: A13 output (BSC)*[2] 10: IRQ21 input (INTC) 11: SCK7 I/O (SCIF_7)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PC12MD[1:0]	00*[1]	R/W	PC12 Mode Select the function of the PC12/A12/IRQ20/RXD4 pin. 00: PC12 I/O (port) 01: A12 output (BSC)*[2] 10: IRQ20 input (INTC) 11: RXD4 input (SCIF_4)
7, 6	PC11MD[1:0]	00*[1]	R/W	PC11 Mode Select the function of the PC11/A11/IRQ19/TXD4 pin. 00: PC11 I/O (port) 01: A11 output (BSC)*[2] 10: IRQ19 input (INTC) 11: TXD4 output (SCIF_4)
5, 4	PC10MD[1:0]	00*[1]	R/W	PC10 Mode Select the function of the PC10/A10/IRQ18/SCK4 pin. 00: PC10 I/O (port) 01: A10 output (BSC)*[2] 10: IRQ18 input (INTC) 11: SCK4 I/O (SCIF_4)
3, 2	PC9MD[1:0]	00*[1]	R/W	PC9 Mode Select the function of the PC9/A9/IRQ17/RXD6 pin. 00: PC9 I/O (port) 01: A9 output (BSC)*[2] 10: IRQ17 input (INTC) 11: RXD6 input (SCIF_6)
1, 0	PC8MD[1:0]	00*[1]	R/W	PC8 Mode Select the function of the PC8/A8/IRQ17/TXD6 pin. 00: PC8 I/O (port) 01: A8 output (BSC)*[2] 10: IRQ17 input (INTC) 11: TXD6 output (SCIF_6)

Notes: 1. The initial value is 01 in on-chip ROM disabled external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port C Control Register L1 (PCCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7 MD1	PC7 MD0	PC6 MD1	PC6 MD0	PC5 MD1	PC5 MD0	PC4 MD1	PC4 MD0	PC3 MD1	PC3 MD0	PC2 MD1	PC2 MD0	PC1 MD1	PC1 MD0	PC0 MD1	PC0 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PC7MD[1:0]	00*[1]	R/W	PC7 Mode Select the function of the PC7/A7/$\overline{\text{IRQOUT}}$/SCK6 pin. 00: PC7 I/O (port) 01: A7 output (BSC)*[2] 10: $\overline{\text{IRQOUT}}$ output (INTC) 11: SCK6 I/O (SCIF_6)
13, 12	PC6MD[1:0]	00*[1]	R/W	PC6 Mode Select the function of the PC6/A6/$\overline{\text{UBCTRG}}$/RXD3 pin. 00: PC6 I/O (port) 01: A6 output (BSC)*[2] 10: $\overline{\text{UBCTRG}}$ output (UBC) 11: RXD3 input (SCI_3)
11, 10	PC5MD[1:0]	00*[1]	R/W	PC5 Mode Select the function of the PC5/A5/TIC5US/TXD3 pin. 00: PC5 I/O (port) 01: A5 output (BSC)*[2] 10: TIC5US input (MTU2S) 11: TXD3 output (SCI_3)
9, 8	PC4MD[1:0]	00*[1]	R/W	PC4 Mode Select the function of the PC4/A4/TIC5VS/SCK3 pin. 00: PC4 I/O (port) 01: A4 output (BSC)*[2] 10: TIC5VS input (MTU2S) 11: SCK3 I/O (SCI_3)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PC3MD[1:0]	00*[1]	R/W	PC3 Mode Select the function of the PC3/A3/TIC5WS/$\overline{\text{POE3}}$ pin. 00: PC3 I/O (port) 01: A3 output (BSC)*[2] 10: TIC5WS input (MTU2S) 11: $\overline{\text{POE3}}$ input (POE2)
5, 4	PC2MD[1:0]	00*[1]	R/W	PC2 Mode Select the function of the PC2/A2/TIC5W/$\overline{\text{POE2}}$ pin. 00: PC2 I/O (port) 01: A2 output (BSC)*[2] 10: TIC5W input (MTU2) 11: $\overline{\text{POE2}}$ input (POE2)
3, 2	PC1MD[1:0]	00*[1]	R/W	PC1 Mode Select the function of the PC1/A1/TIC5V/$\overline{\text{POE1}}$ pin. 00: PC1 I/O (port) 01: A1 output (BSC)*[2] 10: TIC5V input (MTU2) 11: $\overline{\text{POE1}}$ input (POE2)
1, 0	PC0MD[1:0]	00*[1]	R/W	PC0 Mode Select the function of the PC0/A0/TIC5U/$\overline{\text{POE0}}$ pin. 00: PC0 I/O (port) 01: A0 output (BSC)*[2] 10: TIC5U input (MTU2) 11: $\overline{\text{POE0}}$ input (POE2)

Notes: 1. The initial value is 01 in on-chip ROM disabled external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

22.1.9 Port C Pull-Up MOS Control Register L (PCPCRL)

PCPCRL controls on/off of the input pull-up MOS of port C in bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 PCR	PC14 PCR	PC13 PCR	PC12 PCR	PC11 PCR	PC10 PCR	PC9 PCR	PC8 PCR	PC7 PCR	PC6 PCR	PC5 PCR	PC4 PCR	PC3 PCR	PC2 PCR	PC1 PCR	PC0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PC15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PC14PCR	0	R/W	
13	PC13PCR	0	R/W	
12	PC12PCR	0	R/W	
11	PC11PCR	0	R/W	
10	PC10PCR	0	R/W	
9	PC9PCR	0	R/W	
8	PC8PCR	0	R/W	
7	PC7PCR	0	R/W	
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

22.1.10 Port D I/O Registers L and H (PDIORL and PDIORH)

PDIORL and PDIORH are 16-bit readable/writable registers that are used to set the pins on port D as inputs or outputs. Bits PD31IOR to PD0IOR correspond to pins PD31 to PD0, respectively (multiplexed port pin names except for the port names are abbreviated here). PDIORL and PDIORH are enabled when the port D pins are functioning as general-purpose inputs/outputs (PD31 to PD16 for PDIORH and PD15 to PD0 for PDIORL). In other states, they are disabled.

A given pin on port D will be an output pin if the corresponding bit in PDIORL and PDIORH is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

The initial values of PDIORL and PDIORH are both H'0000.

- Port D I/O Register H (PDIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 IOR	PD30 IOR	PD29 IOR	PD28 IOR	PD27 IOR	PD26 IOR	PD25 IOR	PD24 IOR	PD23 IOR	PD22 IOR	PD21 IOR	PD20 IOR	PD19 IOR	PD18 IOR	PD17 IOR	PD16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Port D I/O Register L (PDIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.11 Port D Control Registers L1, L2, H1, and H2 (PDCRL1, PDCRL2, PDCRH1, and PDCRH2)

PDCRL1, PDCRL2, PDCRH1, and PDCRH2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port D.

PDCRL1, PDCRL2, PDCRH1, and PDCRH2 are initialized by a power-on reset signal from the $\overline{\text{RES}}$ pin. They are not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

- Port D Control Register H2 (PDCRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 MD1	PD31 MD0	PD30 MD1	PD30 MD0	PD29 MD1	PD29 MD0	PD28 MD1	PD28 MD0	PD27 MD1	PD27 MD0	PD26 MD1	PD26 MD0	PD25 MD1	PD25 MD0	PD24 MD1	PD24 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PD31MD[1:0]	00*[1]	R/W	PD31 Mode Select the function of the PD31/D31/TIC5W/KEY31/P3/P7 pin. 00: PD31 I/O (port) 01: D31 I/O (BSC)*[2] 10: TIC5W input (MTU2) 11: KEY31 input/P3 input/P7 input (KEYC)
13, 12	PD30MD[1:0]	00*[1]	R/W	PD30 Mode Select the function of the PD30/D30/TIC5V/KEY30/P2/P6 pin. 00: PD30 I/O (port) 01: D30 I/O (BSC)*[2] 10: TIC5V input (MTU2) 11: KEY30 input/P2 input/P6 input (KEYC)

Bit	Bit Name	Initial Value	R/W	Description
11, 10	PD29MD[1:0]	00*1	R/W	PD29 Mode Select the function of the PD29/D29/TIC5U/KEY29/P1/P5 pin. 00: PD29 I/O (port) 01: D29 I/O (BSC)*2 10: TIC5U input (MTU2) 11: KEY29 input/P1 input/P5 input (KEYC)
9, 8	PD28MD[1:0]	00*1	R/W	PD28 Mode Select the function of the PD28/D28/TEND1/KEY28/P0/P4 pin. 00: PD28 I/O (port) 01: D28 I/O (BSC)*2 10: TEND1 output (DMAC)*2 11: KEY28 input/P0 input/P4 input (KEYC)
7, 6	PD27MD[1:0]	00*1	R/W	PD27 Mode Select the function of the PD27/D27/DACK0/KEY27/$\overline{\text{COM3}}$/P3 pin. 00: PD27 I/O (port) 01: D27 I/O (BSC)*2 10: DACK0 output (DMAC)*2 11: KEY27 input/$\overline{\text{COM3}}$ output/P3 input (KEYC)
5, 4	PD26MD[1:0]	00*1	R/W	PD26 Mode Select the function of the PD26/D26/DACK1/KEY26/$\overline{\text{COM2}}$/P2 pin. 00: PD26 I/O (port) 01: D26 I/O (BSC)*2 10: DACK1 output (DMAC)*2 11: KEY26 input/$\overline{\text{COM2}}$ output/P2 input (KEYC)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	PD25MD[1:0]	00*1	R/W	PD25 Mode Select the function of the PD25/D25/DREQ1/KEY25/$\overline{\text{COM1}}$/P1 pin. 00: PD25 I/O (port) 01: D25 I/O (BSC)*2 10: DREQ1 input (DMAC) 11: KEY25 input/$\overline{\text{COM1}}$ output/P1 input (KEYC)
1, 0	PD24MD[1:0]	00*1	R/W	PD24 Mode Select the function of the PD24/D24/DREQ0/KEY24/$\overline{\text{COM0}}$/P0 pin. 00: PD24 I/O (port) 01: D24 I/O (BSC)*2 10: DREQ0 input (DMAC) 11: KEY24 input/$\overline{\text{COM0}}$ output/P0 input (KEYC)

Notes: 1. The initial value is 01 in on-chip ROM disabled 32-bit external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port D Control Register H1 (PDCRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD23 MD1	PD23 MD0	PD22 MD1	PD22 MD0	PD21 MD1	PD21 MD0	PD20 MD1	PD20 MD0	PD19 MD1	PD19 MD0	PD18 MD1	PD18 MD0	PD17 MD1	PD17 MD0	PD16 MD1	PD16 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PD23MD[1:0]	00*[1]	R/W	PD23 Mode Select the function of the PD23/D23/TEND0/KEY23/$\overline{\text{COM7}}$ pin. 00: PD23 I/O (port) 01: D23 I/O (BSC)*[2] 10: TEND0 output (DMAC)*[2] 11: KEY23 input/$\overline{\text{COM7}}$ output (KEYC)
13, 12	PD22MD[1:0]	00*[1]	R/W	PD22 Mode Select the function of the PD22/D22/TIC5US/KEY22/$\overline{\text{COM6}}$ pin. 00: PD22 I/O (port) 01: D22 I/O (BSC)*[2] 10: TIC5US input (MTU2S) 11: KEY22 input/$\overline{\text{COM6}}$ output (KEYC)
11, 10	PD21MD[1:0]	00*[1]	R/W	PD21 Mode Select the function of the PD21/D21/TIC5VS/KEY21/$\overline{\text{COM5}}$ pin. 00: PD21 I/O (port) 01: D21 I/O (BSC)*[2] 10: TIC5VS input (MTU2S) 11: KEY21 input/$\overline{\text{COM5}}$ output (KEYC)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PD20MD[1:0]	00*[1]	R/W	PD20 Mode Select the function of the PD20/D20/TIC5WS/KEY20/$\overline{\text{COM4}}$ pin. 00: PD20 I/O (port) 01: D20 I/O (BSC)*[2] 10: TIC5WS input (MTU2S) 11: KEY20 input/$\overline{\text{COM4}}$ output (KEYC)
7, 6	PD19MD[1:0]	00*[1]	R/W	PD19 Mode Select the function of the PD19/D19/TCLKD/KEY19/$\overline{\text{COM3}}$ pin. 00: PD19 I/O (port) 01: D19 I/O (BSC)*[2] 10: TCLKD input (MTU2) 11: KEY19 input/$\overline{\text{COM3}}$ output (KEYC)
5, 4	PD18MD[1:0]	00*[1]	R/W	PD18 Mode Select the function of the PD18/D18/TCLKC/KEY18/$\overline{\text{COM2}}$ pin. 00: PD18 I/O (port) 01: D18 I/O (BSC)*[2] 10: TCLKC input (MTU2) 11: KEY18 input/$\overline{\text{COM2}}$ output (KEYC)
3, 2	PD17MD[1:0]	00*[1]	R/W	PD17 Mode Select the function of the PD17/D17/TCLKB/KEY17/$\overline{\text{COM1}}$ pin. 00: PD17 I/O (port) 01: D17 I/O (BSC)*[2] 10: TCLKB input (MTU2) 11: KEY17 input/$\overline{\text{COM1}}$ output (KEYC)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PD16MD[1:0]	00*1	R/W	PD16 Mode Select the function of the PD16/D16/TCLKA/KEY16/$\overline{\text{COM0}}$ pin. 00: PD16 I/O (port) 01: D16 I/O (BSC)*2 10: TCLKA input (MTU2) 11: KEY16 input/$\overline{\text{COM0}}$ output (KEYC)

Notes: 1. The initial value is 01 in on-chip ROM disabled 32-bit external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port D Control Register L2 (PDCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 MD1	PD15 MD0	PD14 MD1	PD14 MD0	PD13 MD1	PD13 MD0	PD12 MD1	PD12 MD0	PD11 MD1	PD11 MD0	PD10 MD1	PD10 MD0	PD9 MD1	PD9 MD0	PD8 MD1	PD8 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PD15MD[1:0]	00*[1]	R/W	PD15 Mode Select the function of the PD15/D15/RXD3/KEY15 pin. 00: PD15 I/O (port) 01: D15 I/O (BSC)*[2] 10: RXD3 input (SCI_3) 11: KEY15 input (KEYC)
13, 12	PD14MD[1:0]	00*[1]	R/W	PD14 Mode Select the function of the PD14/D14/TXD3/KEY14 pin. 00: PD14 I/O (port) 01: D14 I/O (BSC)*[2] 10: TXD3 output (SCI_3) 11: KEY14 input (KEYC)
11, 10	PD13MD[1:0]	00*[1]	R/W	PD13 Mode Select the function of the PD13/D13/SCK3/KEY13 pin. 00: PD13 I/O (port) 01: D13 I/O (BSC)*[2] 10: SCK3 I/O (SCI_3) 11: KEY13 input (KEYC)
9, 8	PD12MD[1:0]	00*[1]	R/W	PD12 Mode Select the function of the PD12/D12/RXD5/KEY12 pin. 00: PD12 I/O (port) 01: D12 I/O (BSC)*[2] 10: RXD5 input (SCIF_5) 11: KEY12 input (KEYC)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PD11MD[1:0]	00*[1]	R/W	PD11 Mode Select the function of the PD11/D11/TXD5/KEY11 pin. 00: PD11 I/O (port) 01: D11 I/O (BSC)*[2] 10: TXD5 output (SCIF_5) 11: KEY11 input (KEYC)
5, 4	PD10MD[1:0]	00*[1]	R/W	PD10 Mode Select the function of the PD10/D10/SCK5/KEY10 pin. 00: PD10 I/O (port) 01: D10 I/O (BSC)*[2] 10: SCK5 I/O (SCIF_5) 11: KEY10 input (KEYC)
3, 2	PD9MD[1:0]	00*[1]	R/W	PD9 Mode Select the function of the PD9/D9/RXD2/KEY9 pin. 00: PD9 I/O (port) 01: D9 I/O (BSC)*[2] 10: RXD2 input (SCI_2) 11: KEY9 input (KEYC)
1, 0	PD8MD[1:0]	00*[1]	R/W	PD8 Mode Select the function of the PD8/D8/TXD2/KEY8 pin. 00: PD8 I/O (port) 01: D8 I/O (BSC)*[2] 10: TXD2 output (SCI_2) 11: KEY8 input (KEYC)

Notes: 1. The initial value is 01 in on-chip ROM disabled external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port D Control Register L1 (PDCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD7 MD1	PD7 MD0	PD6 MD1	PD6 MD0	PD5 MD1	PD5 MD0	PD4 MD1	PD4 MD0	PD3 MD1	PD3 MD0	PD2 MD1	PD2 MD0	PD1 MD1	PD1 MD0	PD0 MD1	PD0 MD0
Initial value:	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*	0	0*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The initial value is 1 in on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PD7MD[1:0]	00*[1]	R/W	PD7 Mode Select the function of the PD7/D7/$\overline{\text{POE8}}$/KEY7 pin. 00: PD7 I/O (port) 01: D7 I/O (BSC)*[2] 10: $\overline{\text{POE8}}$ input (POE2) 11: KEY7 input (KEYC)
13, 12	PD6MD[1:0]	00*[1]	R/W	PD6 Mode Select the function of the PD6/D6/SCK2/KEY6 pin. 00: PD6 I/O (port) 01: D6 I/O (BSC)*[2] 10: SCK2 I/O (SCI_2) 11: KEY6 input (KEYC)
11, 10	PD5MD[1:0]	00*[1]	R/W	PD5 Mode Select the function of the PD5/D5/RXD7/KEY5 pin. 00: PD5 I/O (port) 01: D5 I/O (BSC)*[2] 10: RXD7 input (SCIF_7) 11: KEY5 input (KEYC)
9, 8	PD4MD[1:0]	00*[1]	R/W	PD4 Mode Select the function of the PD4/D4/TXD7/KEY4 pin. 00: PD4 I/O (port) 01: D4 I/O (BSC)*[2] 10: TXD7 output (SCIF_7) 11: KEY4 input (KEYC)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PD3MD[1:0]	00*1	R/W	PD3 Mode Select the function of the PD3/D3/SCK7/KEY3 pin. 00: PD3 I/O (port) 01: D3 I/O (BSC)*2 10: SCK7 I/O (SCIF_7) 11: KEY3 input (KEYC)
5, 4	PD2MD[1:0]	00*1	R/W	PD2 Mode Select the function of the PD2/D2/RXD1/KEY2 pin. 00: PD2 I/O (port) 01: D2 I/O (BSC)*2 10: RXD1 input (SCI_1) 11: KEY2 input (KEYC)
3, 2	PD1MD[1:0]	00*1	R/W	PD1 Mode Select the function of the PD1/D1/TXD1/KEY1 pin. 00: PD1 I/O (port) 01: D1 I/O (BSC)*2 10: TXD1 output (SCI_1) 11: KEY1 input (KEYC)
1, 0	PD0MD[1:0]	00*1	R/W	PD0 Mode Select the function of the PD0/D0/SCK1/KEY0 pin. 00: PD0 I/O (port) 01: D0 I/O (BSC)*2 10: SCK1 I/O (SCI_1) 11: KEY0 input (KEYC)

Notes: 1. The initial value is 01 in on-chip ROM disabled external extension mode.
2. This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

22.1.12 Port D Pull-Up MOS Control Registers L and H (PDPCRL and PDPCRH)

PDPCRL and PDPCRH control on/off of the input pull-up MOS of port D in bits.

- Port D Pull-Up MOS Control Register H (PDPCRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 PCR	PD30 PCR	PD29 PCR	PD28 PCR	PD27 PCR	PD26 PCR	PD25 PCR	PD24 PCR	PD23 PCR	PD22 PCR	PD21 PCR	PD20 PCR	PD19 PCR	PD18 PCR	PD17 PCR	PD16 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD31PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PD30PCR	0	R/W	
13	PD29PCR	0	R/W	
12	PD28PCR	0	R/W	
11	PD27PCR	0	R/W	
10	PD26PCR	0	R/W	
9	PD25PCR	0	R/W	
8	PD24PCR	0	R/W	
7	PD23PCR	0	R/W	
6	PD22PCR	0	R/W	
5	PD21PCR	0	R/W	
4	PD20PCR	0	R/W	
3	PD19PCR	0	R/W	
2	PD18PCR	0	R/W	
1	PD17PCR	0	R/W	
0	PD16PCR	0	R/W	

- Port D Pull-Up MOS Control Register L (PDPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 PCR	PD14 PCR	PD13 PCR	PD12 PCR	PD11 PCR	PD10 PCR	PD9 PCR	PD8 PCR	PD7 PCR	PD6 PCR	PD5 PCR	PD4 PCR	PD3 PCR	PD2 PCR	PD1 PCR	PD0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PD14PCR	0	R/W	
13	PD13PCR	0	R/W	
12	PD12PCR	0	R/W	
11	PD11PCR	0	R/W	
10	PD10PCR	0	R/W	
9	PD9PCR	0	R/W	
8	PD8PCR	0	R/W	
7	PD7PCR	0	R/W	
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

22.1.13 Port E I/O Registers L and H (PEIORL and PEIORH)

PEIORL and PEIORH are 16-bit readable/writable registers that are used to set the pins on port E as inputs or outputs. Bits PE23IOR to PE0IOR correspond to pins PE23 to PE0, respectively (multiplexed port pin names except for the port names are abbreviated here). PEIORL is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE15 to PE0) and TIOC inputs/outputs in MTU2. In other states, PEIORL is disabled. PEIORH is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE23 to PE16) and TIOC inputs/outputs in MTU2S. In other states, PEIORH is disabled.

A given pin on port E will be an output pin if the corresponding bit in PEIORL and PEIORH is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

Bits 15 to 8 of PEIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial values of PEIORL and PEIORH are both H'0000.

- Port E I/O Register H (PEIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PE23 IOR	PE22 IOR	PE21 IOR	PE20 IOR	PE19 IOR	PE18 IOR	PE17 IOR	PE16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Port E I/O Register L (PEIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.14 Port E Control Registers L1, L2, and H1 (PECRL1, PECRL2, and PECRH1)

PECRL1, PECRL2, and PECRH1 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port E.

- Port E Control Register H1 (PECRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE23 MD1	PE23 MD0	PE22 MD1	PE22 MD0	PE21 MD1	PE21 MD0	PE20 PE1	PE20 PE0	PE19 MD1	PE19 MD0	PE18 MD1	PE18 MD0	PE17 MD1	PE17 MD0	PE16 MD1	PE16 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PE23MD[1:0]	00	R/W	PE23 Mode Select the function of the PE23/TEND1/TIOC4DS/RXD7 pin. 00: PE23 I/O (port) 01: TEND1 output (DMAC)* 10: TIOC4DS I/O (MTU2S) 11: RXD7 input (SCIF_7)
13, 12	PE22MD[1:0]	00	R/W	PE22 Mode Select the function of the PE22/DACK1/TIOC4CS/TXD7 pin. 00: PE22 I/O (port) 01: DACK1 output (DMAC)* 10: TIOC4CS I/O (MTU2S) 11: TXD7 output (SCIF_7)
11, 10	PE21MD[1:0]	00	R/W	PE21 Mode Select the function of the PE21/DREQ1/TIOC4BS/SCK7 pin. 00: PE21 I/O (port) 01: DREQ1 input (DMAC) 10: TIOC4BS I/O (MTU2S) 11: SCK7 I/O (SCIF_7)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PE20MD[1:0]	00	R/W	PE20 Mode Select the function of the PE20/TEND0/TIOC4AS/RXD6 pin. 00: PE20 I/O (port) 01: TEND0 output (DMAC)* 10: TIOC4AS I/O (MTU2S) 11: RXD6 input (SCIF_6)
7, 6	PE19MD[1:0]	00	R/W	PE19 Mode Select the function of the PE19/DACK0/TIOC3DS/TXD6 pin. 00: PE19 I/O (port) 01: DACK0 output (DMAC)* 10: TIOC3DS I/O (MTU2S) 11: TXD6 output (SCIF_6)
5, 4	PE18MD[1:0]	00	R/W	PE18 Mode Select the function of the PE18/DREQ0/TIOC3CS/SCK6 pin. 00: PE18 I/O (port) 01: DREQ0 input (DMAC) 10: TIOC3CS I/O (MTU2S) 11: SCK6 I/O (SCIF_6)
3, 2	PE17MD[1:0]	00	R/W	PE17 Mode Select the function of the PE17/$\overline{\text{MRES}}$/TIOC3BS/RXD0 pin. 00: PE17 I/O (port) 01: $\overline{\text{MRES}}$ input (INTC) 10: TIOC3BS I/O (MTU2S) 11: RXD0 input (SCI_0)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PE16MD[1:0]	00	R/W	PE16 Mode Select the function of the PE16/$\overline{\text{UBCTRG}}$/TIOC3AS/TXD0 pin. 00: PE16 I/O (port) 01: $\overline{\text{UBCTRG}}$ output (UBC) 10: TIOC3AS I/O (MTU2S) 11: TXD0 output (SCI_0)

Note: * This function is valid only in on-chip ROM enabled/disabled external extension mode. Do not set this function in single chip mode.

- Port E Control Register L2 (PECRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 MD1	PE15 MD0	PE14 MD1	PE14 MD0	PE13 MD1	PE13 MD0	PE12 MD1	PE12 MD0	PE11 MD1	PE11 MD0	PE10 MD1	PE10 MD0	PE9 MD1	PE9 MD0	PE8 MD1	PE8 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PE15MD[1:0]	00	R/W	PE15 Mode Select the function of the PE15/IRQ15/TIOC4D/SCK0 pin. 00: PE15 I/O (port) 01: IRQ15 input (INTC) 10: TIOC4D I/O (MTU2) 11: SCK0 I/O (SCI_0)
13, 12	PE14MD[1:0]	00	R/W	PE14 Mode Select the function of the PE14/IRQ14/TIOC4C/RXD5 pin. 00: PE14 I/O (port) 01: IRQ14 input (INTC) 10: TIOC4C I/O (MTU2) 11: RXD5 input (SCIF_5)

Bit	Bit Name	Initial Value	R/W	Description
11, 10	PE13MD[1:0]	00	R/W	PE13 Mode Select the function of the PE13/IRQ13/TIOC4B/TXD5 pin. 00: PE13 I/O (port) 01: IRQ13 input (INTC) 10: TIOC4B I/O (MTU2) 11: TXD5 output (SCIF_5)
9, 8	PE12MD[1:0]	00	R/W	PE12 Mode Select the function of the PE12/IRQ12/TIOC4A/SCK5 pin. 00: PE12 I/O (port) 01: IRQ12 input (INTC) 10: TIOC4A I/O (MTU2) 11: SCK5 I/O (SCIF_5)
7, 6	PE11MD[1:0]	00	R/W	PE11 Mode Select the function of the PE11/IRQ11/TIOC3D/RXD1 pin. 00: PE11 I/O (port) 01: IRQ11 input (INTC) 10: TIOC3D I/O (MTU2) 11: RXD1 input (SCI_1)
5, 4	PE10MD[1:0]	00	R/W	PE10 Mode Select the function of the PE10/IRQ10/TIOC3C/TXD1 pin. 00: PE10 I/O (port) 01: IRQ10 input (INTC) 10: TIOC3C I/O (MTU2) 11: TXD1 output (SCI_1)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	PE9MD[1:0]	00	R/W	PE9 Mode Select the function of the PE9/IRQ9/TIOC3B/SCK1 pin. 00: PE9 I/O (port) 01: IRQ9 input (INTC) 10: TIOC3B I/O (MTU2) 11: SCK1 I/O (SCI_1)
1, 0	PE8MD[1:0]	00	R/W	PE8 Mode Select the function of the PE8/IRQ8/TIOC3A/SCK2 pin. 00: PE8 I/O (port) 01: IRQ8 input (INTC) 10: TIOC3A I/O (MTU2) 11: SCK2 I/O (SCI_2)

- Port E Control Register L1 (PECRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE7 MD1	PE7 MD0	PE6 MD1	PE6 MD0	PE5 MD1	PE5 MD0	PE4 MD1	PE4 MD0	PE3 MD1	PE3 MD0	PE2 MD1	PE2 MD0	PE1 MD1	PE1 MD0	PE0 MD1	PE0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PE7MD[1:0]	00	R/W	PE7 Mode Select the function of the PE7/IRQ7/TIOC2B/RXD2 pin. 00: PE7 I/O (port) 01: IRQ7 input (INTC) 10: TIOC2B I/O (MTU2) 11: RXD2 input (SCI_2)

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PE6MD[1:0]	00	R/W	PE6 Mode Select the function of the PE6/IRQ6/TIOC2A/TXD2 pin. 00: PE6 I/O (port) 01: IRQ6 input (INTC) 10: TIOC2A I/O (MTU2) 11: TXD2 output (SCI_2)
11, 10	PE5MD[1:0]	00	R/W	PE5 Mode Select the function of the PE5/IRQ5/TIOC1B/TXD3 pin. 00: PE5 I/O (port) 01: IRQ5 input (INTC) 10: TIOC1B I/O (MTU2) 11: TXD3 output (SCI_3)
9, 8	PE4MD[1:0]	00	R/W	PE4 Mode Select the function of the PE4/IRQ4/TIOC1A/RXD3 pin. 00: PE4 I/O (port) 01: IRQ4 input (INTC) 10: TIOC1A I/O (MTU2) 11: RXD3 input (SCI_3)
7, 6	PE3MD[1:0]	00	R/W	PE3 Mode Select the function of the PE3/IRQ3/TIOC0D/SCK3 pin. 00: PE3 I/O (port) 01: IRQ3 input (INTC) 10: TIOC0D I/O (MTU2) 11: SCK3 I/O (SCI_3)

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PE2MD[1:0]	00	R/W	PE2 Mode Select the function of the PE2/IRQ2/TIOC0C/RXD4 pin. 00: PE2 I/O (port) 01: IRQ2 input (INTC) 10: TIOC0C I/O (MTU2) 11: RXD4 input (SCIF_4)
3, 2	PE1MD[1:0]	00	R/W	PE1 Mode Select the function of the PE1/IRQ1/TIOC0B/TXD4 pin. 00: PE1 I/O (port) 01: IRQ1 input (INTC) 10: TIOC0B I/O (MTU2) 11: TXD4 output (SCIF_4)
1, 0	PE0MD[1:0]	00	R/W	PE0 Mode Select the function of the PE0/IRQ0/TIOC0A/SCK4 pin. 00: PE0 I/O (port) 01: IRQ0 input (INTC) 10: TIOC0A I/O (MTU2) 11: SCK4 I/O (SCIF_4)

22.1.15 Port E Pull-Up MOS Control Registers L and H (PEPCRL and PEPCRH)

PEPCRL and PEPCRH control the on/off of the input pull-up MOS of the port E in bits.

- Port E Pull-Up MOS Control Register H (PEPCRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PE23 PCR	PE22 PCR	PE21 PCR	PE20 PCR	PE19 PCR	PE18 PCR	PE17 PCR	PE16 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PE23PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
6	PE22PCR	0	R/W	
5	PE21PCR	0	R/W	
4	PE20PCR	0	R/W	
3	PE19PCR	0	R/W	
2	PE18PCR	0	R/W	
1	PE17PCR	0	R/W	
0	PE16PCR	0	R/W	

- Port E Pull-Up MOS Control Register L (PEPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 PCR	PE14 PCR	PE13 PCR	PE12 PCR	PE11 PCR	PE10 PCR	PE9 PCR	PE8 PCR	PE7 PCR	PE6 PCR	PE5 PCR	PE4 PCR	PE3 PCR	PE2 PCR	PE1 PCR	PE0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PE15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PE14PCR	0	R/W	
13	PE13PCR	0	R/W	
12	PE12PCR	0	R/W	
11	PE11PCR	0	R/W	
10	PE10PCR	0	R/W	
9	PE9PCR	0	R/W	
8	PE8PCR	0	R/W	
7	PE7PCR	0	R/W	
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

22.1.16 Port G I/O Register L (PGIORL)

PGIORL is a 16-bit readable/writable register that is used to set the pins on port G as inputs or outputs. Bits PG15IOR to PG0IOR correspond to pins PG15 to PG0, respectively (multiplexed port pin names except for the port names are abbreviated here). PGIORL is enabled when the port G pins are functioning as general-purpose inputs/outputs (PG15 to PG0). In other states, PGIORL is disabled.

A given pin on port G will be an output pin if the corresponding bit in PGIORL is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

The initial value of PGIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 IOR	PG14 IOR	PG13 IOR	PG12 IOR	PG11 IOR	PG10 IOR	PG9 IOR	PG8 IOR	PG7 IOR	PG6 IOR	PG5 IOR	PG4 IOR	PG3 IOR	PG2 IOR	PG1 IOR	PG0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.17 Port G Control Register L1 (PGCRL1)

PGCRL1 is a 16-bit readable/writable register that is used to select the functions of the multiplexed pins on port G.

PGCRL1 is initialized by a power-on reset signal from the $\overline{\text{RES}}$ pin. PGCRL1 is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15MD	PG14MD	PG13MD	PG12MD	PG11MD	PG10MD	PG9MD	PG8MD	PG7MD	PG6MD	PG5MD	PG4MD	PG3MD	PG2MD	PG1MD	PG0MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PG15MD	0	R/W	PG15 Mode Select the function of the PG15 pin. 0: PG15 I/O (port) 1: Setting prohibited
14	PG14MD	0	R/W	PG14 Mode Select the function of the PG14/CK32 pin. 0: PG14 I/O (port) 1: CK32 output (32-kHz clock)
13	PG13MD	0	R/W	PG13 Mode Select the function of the PG13/TI32I1B pin. 0: PG13 I/O (port) 1: TI32I1B input (TIM32C)
12	PG12MD	0	R/W	PG12 Mode Select the function of the PG12/TI32I1A pin. 0: PG12 I/O (port) 1: TI32I1A input (TIM32C)
11	PG11MD	0	R/W	PG11 Mode Select the function of the PG11/TI32I0B pin. 0: PG11 I/O (port) 1: TI32I0B input (TIM32C)

Bit	Bit Name	Initial Value	R/W	Description
10	PG10MD	0	R/W	PG10 Mode Select the function of the PG10/TI32I0A pin. 0: PG10 I/O (port) 1: TI32I0A input (TIM32C)
9	PG9MD	0	R/W	PG9 Mode Select the function of the PG9/IRQ9 pin. 0: PG9 I/O (port) 1: IRQ9 input (INTC)
8	PG8MD	0	R/W	PG8 Mode Select the function of the PG8/IRQ8 pin. 0: PG8 I/O (port) 1: IRQ8 input (INTC)
7	PG7MD	0	R/W	PG7 Mode Select the function of the PG7/IRQ7 pin. 0: PG7 I/O (port) 1: IRQ7 input (INTC)
6	PG6MD	0	R/W	PG6 Mode Select the function of the PG6/IRQ6 pin. 0: PG6 I/O (port) 1: IRQ6 input (INTC)
5	PG5MD	0	R/W	PG5 Mode Select the function of the PG5/IRQ5 pin. 0: PG5 I/O (port) 1: IRQ5 input (INTC)
4	PG4MD	0	R/W	PG4 Mode Select the function of the PG4/IRQ4 pin. 0: PG4 I/O (port) 1: IRQ4 input (INTC)
3	PG3MD	0	R/W	PG3 Mode Select the function of the PG3/IRQ3 pin. 0: PG3 I/O (port) 1: IRQ3 input (INTC)

Bit	Bit Name	Initial Value	R/W	Description
2	PG2MD	0	R/W	PG2 Mode Select the function of the PG2/IRQ2 pin. 0: PG2 I/O (port) 1: IRQ2 input (INTC)
1	PG1MD	0	R/W	PG1 Mode Select the function of the PG1/IRQ1 pin. 0: PG1 I/O (port) 1: IRQ1 input (INTC)
0	PG0MD	0	R/W	PG0 Mode Select the function of the PG0/IRQ0 pin. 0: PG0 I/O (port) 1: IRQ0 input (INTC)

22.1.18 Port G Pull-Up MOS Control Register L (PGPCRL)

PGPCRL controls the on/off of the input pull-up MOS of the port G in bits.

PGPCRL is initialized by a power-on reset signal from the $\overline{\text{RES}}$ pin. PGPCRL is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15PCR	PG14PCR	PG13PCR	PG12PCR	PG11PCR	PG10PCR	PG9PCR	PG8PCR	PG7PCR	PG6PCR	PG5PCR	PG4PCR	PG3PCR	PG2PCR	PG1PCR	PG0PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PG15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PG14PCR	0	R/W	
13	PG13PCR	0	R/W	
12	PG12PCR	0	R/W	
11	PG11PCR	0	R/W	
10	PG10PCR	0	R/W	
9	PG9PCR	0	R/W	
8	PG8PCR	0	R/W	
7	PG7PCR	0	R/W	
6	PG6PCR	0	R/W	
5	PG5PCR	0	R/W	
4	PG4PCR	0	R/W	
3	PG3PCR	0	R/W	
2	PG2PCR	0	R/W	
1	PG1PCR	0	R/W	
0	PG0PCR	0	R/W	

22.1.19 Port H I/O Register L (PHIORL)

PHIORL is a 16-bit readable/writable register that is used to set the pins on port H as inputs or outputs. Bits PH15IOR to PH0IOR correspond to pins PH15 to PH0, respectively (multiplexed port pin names except for the port names are abbreviated here). PHIORL is enabled when the port H pins are functioning as general-purpose inputs/outputs (PH15 to PH0). In other states, PHIORL is disabled.

A given pin on port H will be an output pin if the corresponding bit in PHIORL is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

The initial value of PHIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15 IOR	PH14 IOR	PH13 IOR	PH12 IOR	PH11 IOR	PH10 IOR	PH9 IOR	PH8 IOR	PH7 IOR	PH6 IOR	PH5 IOR	PH4 IOR	PH3 IOR	PH2 IOR	PH1 IOR	PH0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.20 Port H Control Register L1 (PHCRL1)

PHCRL1 is a 16-bit readable/writable register that is used to select the functions of the multiplexed pins on port H.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15 MD	PH14 MD	PH13 MD	PH12 MD	PH11 MD	PH10 MD	PH9 MD	PH8 MD	PH7 MD	PH6 MD	PH5 MD	PH4 MD	PH3 MD	PH2 MD	PH1 MD	PH0 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PH15MD	0	R/W	PH15 Mode Select the function of the PH15/TOC1 pin. 0: PH15 I/O (port) 1: TOC1 output (CMT2)
14	PH14MD	0	R/W	PH14 Mode Select the function of the PH14/TOC0 pin. 0: PH14 I/O (port) 1: TOC0 output (CMT2)
13	PH13MD	0	R/W	PH13 Mode Select the function of the PH13/TIC1 pin. 0: PH13 I/O (port) 1: TIC1 input (CMT2)
12	PH12MD	0	R/W	PH12 Mode Select the function of the PH12/TIC0 pin. 0: PH12 I/O (port) 1: TIC0 input (CMT2)
11	PH11MD	0	R/W	PH11 Mode Select the function of the PH11 pin. 0: PH11 I/O (port) 1: Setting prohibited
10	PH10MD	0	R/W	PH10 Mode Select the function of the PH10 pin. 0: PH10 I/O (port) 1: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
9	PH9MD	0	R/W	PH9 Mode Select the function of the PH9 pin. 0: PH9 I/O (port) 1: Setting prohibited
8	PH8MD	0	R/W	PH8 Mode Select the function of the PH8/RXD7 pin. 0: PH8 I/O (port) 1: RXD7 input (SCIF_7)
7	PH7MD	0	R/W	PH7 Mode Select the function of the PH7/TXD7 pin. 0: PH7 I/O (port) 1: TXD7 output (SCIF_7)
6	PH6MD	0	R/W	PH6 Mode Select the function of the PH6/SCK7 pin. 0: PH6 I/O (port) 1: SCK7 I/O (SCIF_7)
5	PH5MD	0	R/W	PH5 Mode Select the function of the PH5/TIC5WS pin. 0: PH5 I/O (port) 1: TIC5WS input (MTU2S)
4	PH4MD	0	R/W	PH4 Mode Select the function of the PH4/TIC5VS pin. 0: PH4 I/O (port) 1: TIC5VS input (MTU2S)
3	PH3MD	0	R/W	PH3 Mode Select the function of the PH3/TIC5US pin. 0: PH3 I/O (port) 1: TIC5US input (MTU2S)
2	PG2MD	0	R/W	PH2 Mode Select the function of the PH2/TIC5W pin. 0: PH2 I/O (port) 1: TIC5W input (MTU2)

Bit	Bit Name	Initial Value	R/W	Description
1	PH1MD	0	R/W	PH1 Mode Select the function of the PH1/TIC5V pin. 0: PH1 I/O (port) 1: TIC5V input (MTU2)
0	PH0MD	0	R/W	PH0 Mode Select the function of the PH0/TIC5U pin. 0: PH0 I/O (port) 1: TIC5U input (MTU2)

22.1.21 Port H Pull-Up MOS Control Register L (PHPCRL)

PHPCRL controls the on/off of the input pull-up MOS of the port H in bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15 PCR	PH14 PCR	PH13 PCR	PH12 PCR	PH11 PCR	PH10 PCR	PH9 PCR	PH8 PCR	PH7 PCR	PH6 PCR	PH5 PCR	PH4 PCR	PH3 PCR	PH2 PCR	PH1 PCR	PH0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PH15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PH14PCR	0	R/W	
13	PH13PCR	0	R/W	
12	PH12PCR	0	R/W	
11	PH11PCR	0	R/W	
10	PH10PCR	0	R/W	
9	PH9PCR	0	R/W	
8	PH8PCR	0	R/W	
7	PH7PCR	0	R/W	
6	PH6PCR	0	R/W	
5	PH5PCR	0	R/W	
4	PH4PCR	0	R/W	
3	PH3PCR	0	R/W	
2	PH2PCR	0	R/W	
1	PH1PCR	0	R/W	
0	PH0PCR	0	R/W	

22.1.22 Port J I/O Register L (PJIORL)

PJIORL is a 16-bit readable/writable register that is used to set the pins on port J as inputs or outputs. Bits PJ15IOR to PJ0IOR correspond to pins PJ15 to PJ0, respectively (multiplexed port pin names except for the port names are abbreviated here). PJIORL is enabled when the port J pins are functioning as general-purpose inputs/outputs (PJ15 to PJ0). In other states, PJIORL is disabled.

A given pin on port J will be an output pin if the corresponding bit in PJIORL is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

The initial value of PJIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ15 IOR	PJ14 IOR	PJ13 IOR	PJ12 IOR	PJ11 IOR	PJ10 IOR	PJ9 IOR	PJ8 IOR	PJ7 IOR	PJ6 IOR	PJ5 IOR	PJ4 IOR	PJ3 IOR	PJ2 IOR	PJ1 IOR	PJ0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.23 Port J Control Register L1 (PJCRL1)

PJCRL1 is a 16-bit readable/writable register that is used to select the functions of the multiplexed pins on port J.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ15 MD	PJ14 MD	PJ13 MD	PJ12 MD	PJ11 MD	PJ10 MD	PJ9 MD	PJ8 MD	PJ7 MD	PJ6 MD	PJ5 MD	PJ4 MD	PJ3 MD	PJ2 MD	PJ1 MD	PJ0 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PJ15MD	0	R/W	PJ15 Mode Select the function of the PJ15/SSL3 pin. 0: PJ15 I/O (port) 1: SSL3 output (RSPI)
14	PJ14MD	0	R/W	PJ14 Mode Select the function of the PJ14/SSL2 pin. 0: PJ14 I/O (port) 1: SSL2 output (RSPI)
13	PJ13MD	0	R/W	PJ13 Mode Select the function of the PJ13/SSL1 pin. 0: PJ13 I/O (port) 1: SSL1 output (RSPI)
12	PJ12MD	0	R/W	PJ12 Mode Select the function of the PJ12/SSL0 pin. 0: PJ12 I/O (port) 1: SSL0 I/O (RSPI)
11	PJ11MD	0	R/W	PJ11 Mode Select the function of the PJ11/MISO0 pin. 0: PJ11 I/O (port) 1: MISO0 I/O (RSPI)
10	PJ10MD	0	R/W	PJ10 Mode Select the function of the PJ10/MOSI0 pin. 0: PJ10 I/O (port) 1: MOSI0 I/O (RSPI)

Bit	Bit Name	Initial Value	R/W	Description
9	PJ9MD	0	R/W	PJ9 Mode Select the function of the PJ9/RSPCK0 pin. 0: PJ9 I/O (port) 1: RSPCK0 I/O (RSPI)
8	PJ8MD	0	R/W	PJ8 Mode Select the function of the PJ8/RXD5 pin. 0: PJ8 I/O (port) 1: RXD5 input (SCIF_5)
7	PJ7MD	0	R/W	PJ7 Mode Select the function of the PJ7/TXD5 pin. 0: PJ7 I/O (port) 1: TXD5 output (SCIF_5)
6	PJ6MD	0	R/W	PJ6 Mode Select the function of the PJ6/SCK5 pin. 0: PJ6 I/O (port) 1: SCK5 I/O (SCIF_5)
5	PJ5MD	0	R/W	PJ5 Mode Select the function of the PJ5/RXD4 pin. 0: PJ5 I/O (port) 1: RXD4 input (SCIF_4)
4	PJ4MD	0	R/W	PJ4 Mode Select the function of the PJ4/TXD4 pin. 0: PJ4 I/O (port) 1: TXD4 output (SCIF_4)
3	PJ3MD	0	R/W	PJ3 Mode Select the function of the PJ3/SCK4 pin. 0: PJ3 I/O (port) 1: SCK4 I/O (SCIF_4)
2	PJ2MD	0	R/W	PJ2 Mode Select the function of the PJ2/IRQ12 pin. 0: PJ2 I/O (port) 1: IRQ12 input (INTC)

Bit	Bit Name	Initial Value	R/W	Description
1	PJ1MD	0	R/W	PJ1 Mode Select the function of the PJ1/IRQ11 pin. 0: PJ1 I/O (port) 1: IRQ11 input (INTC)
0	PJ0MD	0	R/W	PJ0 Mode Select the function of the PJ0/IRQ10 pin. 0: PJ0 I/O (port) 1: IRQ10 input (INTC)

22.1.24 Port J Pull-Up MOS Control Register L (PJPCRL)

PJPCRL controls the on/off of the input pull-up MOS of the port J in bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ15 PCR	PJ14 PCR	PJ13 PCR	PJ12 PCR	PJ11 PCR	PJ10 PCR	PJ9 PCR	PJ8 PCR	PJ7 PCR	PJ6 PCR	PJ5 PCR	PJ4 PCR	PJ3 PCR	PJ2 PCR	PJ1 PCR	PJ0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PJ15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PJ14PCR	0	R/W	
13	PJ13PCR	0	R/W	
12	PJ12PCR	0	R/W	
11	PJ11PCR	0	R/W	
10	PJ10PCR	0	R/W	
9	PJ9PCR	0	R/W	
8	PJ8PCR	0	R/W	
7	PJ7PCR	0	R/W	
6	PJ6PCR	0	R/W	
5	PJ5PCR	0	R/W	
4	PJ4PCR	0	R/W	
3	PJ3PCR	0	R/W	
2	PJ2PCR	0	R/W	
1	PJ1PCR	0	R/W	
0	PJ0PCR	0	R/W	

22.1.25 Port K I/O Register L (PKIORL)

PKIORL is a 16-bit readable/writable register that is used to set the pins on port K as inputs or outputs. Bits PK7IOR to PK0IOR correspond to pins PK7 to PK0, respectively (multiplexed port pin names except for the port names are abbreviated here). PKIORL is enabled when the port K pins are functioning as general-purpose inputs/outputs (PK7 to PK0). In other states, PKIORL is disabled.

A given pin on port K will be an output pin if the corresponding bit in PKIORL is set to 1, and an input pin if the bit is cleared to 0. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).

The initial value of PKIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK7 IOR	PK6 IOR	PK5 IOR	PK4 IOR	PK3 IOR	PK2 IOR	PK1 IOR	PK0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.26 Port K Control Register L1 (PKCRL1)

PKCRL1 is a 16-bit readable/writable register that is used to select the functions of the multiplexed pins on port K.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK7MD	PK6MD	PK5MD	PK4MD	PK3MD	PK2MD	PK1MD	PK0MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PK7MD	0	R/W	PK7 Mode Select the function of the PK7 pin. 0: PK7 I/O (port) 1: Setting prohibited
6	PK6MD	0	R/W	PK6 Mode Select the function of the PK6 pin. 0: PK6 I/O (port) 1: Setting prohibited
5	PK5MD	0	R/W	PK5 Mode Select the function of the PK5/RXD6 pin. 0: PK5 I/O (port) 1: RXD6 input (SCIF_6)
4	PK4MD	0	R/W	PK4 Mode Select the function of the PK4/TXD6 pin. 0: PK4 I/O (port) 1: TXD6 output (SCIF_6)
3	PK3MD	0	R/W	PK3 Mode Select the function of the PK3/SCK6 pin. 0: PK3 I/O (port) 1: SCK6 I/O (SCIF_6)

Bit	Bit Name	Initial Value	R/W	Description
2	PK2MD	0	R/W	PK2 Mode Select the function of the PK2/IRQ15 pin. 0: PK2 I/O (port) 1: IRQ15 input (INTC)
1	PK1MD	0	R/W	PK1 Mode Select the function of the PK1/IRQ14 pin. 0: PK1 I/O (port) 1: IRQ14 input (INTC)
0	PK0MD	0	R/W	PK0 Mode Select the function of the PK0/IRQ13 pin. 0: PK0 I/O (port) 1: IRQ13 input (INTC)

22.1.27 Port K Pull-Up MOS Control Register L (PKPCRL)

PKPCRL controls the on/off of the input pull-up MOS of the port K in bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK7 PCR	PK6 PCR	PK5 PCR	PK4 PCR	PK3 PCR	PK2 PCR	PK1 PCR	PK0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PK7PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
6	PK6PCR	0	R/W	
5	PK5PCR	0	R/W	
4	PK4PCR	0	R/W	
3	PK3PCR	0	R/W	
2	PK2PCR	0	R/W	
1	PK1PCR	0	R/W	
0	PK0PCR	0	R/W	

22.1.28 Port L Pull-Up MOS Control Register L (PLPCRL)

PLPCRL controls the on/off of the input pull-up MOS of the port L in bits.

PLPCRL is enabled when the port L pins are functioning as general-purpose inputs/outputs (PL5 to PL0) (when the LVDS is in the module standby state, that is, when the MSTP67 bit in the standby control register 6 (STBCR6) is 1 for SH72315A; and is always enabled for SH72351L/SH72314L). In other states, PLIORL is disabled.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PL5PCR	PL4PCR	PL3PCR	PL2PCR	PL1PCR	PL0PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PL5PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
4	PL4PCR	0	R/W	
3	PL3PCR	0	R/W	
2	PL2PCR	0	R/W	
1	PL1PCR	0	R/W	
0	PL0PCR	0	R/W	

22.1.29 Large Current Port Control Register (HCPCR)

HCPCR is a 16-bit readable/writable register that is used to control the large current port. It controls twelve pins PE9, PE11 to PE15, PE17, and PE19 to PE 23.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MZI ZEH	MZI ZEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MZIZEH	1	R/W	Port E Large Current Port High Impedance H Selects whether to set the large current port of PE17, and PE19 to PE23 to the high-impedance state regardless of the setting of the PFC during the oscillation stop detection and software standby mode. 0: Set to the high-impedance state 1: Do not set to the high-impedance state The pin state is retained during the oscillation stop detection when this bit is set to 1. See appendix A, Pin States, for details on the software standby mode
0	MZIZEL	1	R/W	Port E Large Current Port High Impedance L Selects whether to set the large current port of PE9, and PE11 to PE15 to the high-impedance state regardless of the setting of the PFC during the oscillation stop detection and software standby mode. 0: Set to the high-impedance state 1: Do not set to the high-impedance state The pin state is retained during the oscillation stop detection when this bit is set to 1. See appendix A, Pin States, for details on the software standby mode.

22.1.30 I/O Buffer Driver Control Register (DRVCR)

DRVCR is a 16-bit readable/writable register that is used to control the drivability of the I/O buffer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ DRV	-	PK DRV	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PJDRV	1	R/W	Port J I/O Buffer Drivability Select Selects the drivability of the port J pin. 0: High drivability (This bit should be set to 0 when PVcc1 = 1.8 V) 1: Low drivability (This bit should be set to 1 when PVcc1 = 3.3V)
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13	PKDRV	1	R/W	Port K I/O Buffer Drivability Select Selects the drivability of the port K pin. 0: High drivability (This bit should be set to 0 when PVcc2 = 1.8 V) 1: Low drivability (This bit should be set to 1 when PVcc2 = 3.3V)
12 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

22.1.31 Port Function Extension Register (PFEXCR)

PFEXCR is a 16-bit readable/writable register that is used to enable or disable the general input function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	GPIE	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GPIE	0	R/W	General Input Function Enable Enables or disables general input function. To set the pin as a general input, set this bit to 1. Setting this bit is valid when the functions of the pins on ports A, B, C, D, E, G, H, J, and K is general inputs (PA19 to PA0, PB13 to PB0, PC15 to PC0, PD31 to PD0, PE23 to PE0, PG15 to PG0, PH15 to PH0, PJ15 to PJ0, and PK7 to PK0). In the other states, setting this bit is invalid. 0: Disables the general input function. 1: Enables the general input function. When this bit is 0, the pin can be open without causing penetration current in the input circuit of this LSI even if the pin is set as the general input.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

22.2 Usage Notes

22.2.1 Point for Caution on Pin Function Controller

1. In this LSI, the same function is available as a multiplexed function on multiple pins. This approach is intended to increase the number of selectable pin functions and to allow the easier design of boards. Note the following points when two or more pins are specified for one function.

- When the pin function is input

  Signals input to several pins are formed as one signal through OR or AND logic and the signal is transmitted into the LSI. Therefore, a signal that differs from the input signals may be transmitted to the LSI depending on the input signals in other pins that have the same functions. Table 22.14 shows the transmit forms of input functions allocated to several pins. When using one of the functions shown below in multiple pins, use it with care of signal polarity considering the transmit forms.

Table 22.14 Transmission Format of Input Function Allocated on Multiple Pins

OR Type	AND Type
SCK0 to SCK7, RXD0 to RXD7, TCLKA, TCLKB, TCLKC, TCLKD, TIC5U, TIC5V, TIC5W, TIC5US, TIC5VS, TIC5WS	IRQ0 to IRQ15, DREQ0, DREQ1, $\overline{\text{ADTRG}}$, $\overline{\text{POE0}}$, $\overline{\text{POE4}}$, $\overline{\text{POE8}}$, $\overline{\text{MRES}}$

OR Type: Signals input to several pins are formed as one signal through OR logic and the signal is transmitted into the LSI.

AND Type: Signals input to several pins are formed as one signal through AND logic and the signal is transmitted into the LSI.

- When the pin function is output

  Each selected pin can output the same function.

2. When the port input is switched from the low level to the DREQ edge or the IRQ edge for the pins that are multiplexed with I/O and DREQ or IRQ, the corresponding edge is detected.
3. Do not set functions other than settable functions in table 22.12. Otherwise, correct operation cannot be guaranteed.
4. To use the pins as general inputs, set the GPIE bit in the port function extension register (PFEXCR) to 1. For details, refer to section 22.1.31, Port Function Extension Register (PFEXCR).
5. To use IRQ9, and IRQ11 to IRQ15 which are multiplexed with PE9, and PE11 to PE15, do not set the MZIZEL bit in the large current port control register (HCPCR) to 0.

6. If the GPIE bit in the port function extension register (PFEXCR) is set to 1, read PFEXCR and confirm that the GPIE bit has been set, and then read the pin state.

22.2.2 Point for Caution on Changing the Pin Function of Port Pins Allocated by the [IRQ23:0] Bits

When the function of a port pin that was allocated to an IRQ signal which acts as a source of requests for release from standby or deep software standby is changed, switching the pin function in accord with the procedure below avoids the generation of an interrupt at the time the pin function is switched.

1. To avoid the generation of an interrupt due to edge detection at the time the pin function is switched, change the IPR setting for the given IRQ interrupt to 0 or change bit 1 in the status register of the CPU to a value above the interrupt priority level of the given IRQ interrupt.
2. Change the IRQ sense selection bits in the IRQ control register for the given IRQ interrupt to H'00 (low-level detection).
3. Change the pin function of the port pin to which the IRQ function was allocated.
4. Clear the flag for the given interrupt in IRQ interrupt request register 0 or 1 (IRQPR 0 or 1) to 0 if it was previously set to 1.
5. Restore the original value to whichever of the IPR or status register of the CPU was set in step 1.

Section 23 I/O Ports

This LSI has eleven ports: A, B, C, D, E, F, G, H, J, K, and L. Port A is a 20-bit, port B is a 14-bit, port C is a 16-bit, port D is a 32-bit, port E is a 24-bit, port G is a 16-bit, port H is a 16-bit, port J is a 16-bit, and port K is an 8-bit I/O port.

Port F is a 16-bit, and port L is a 6-bit input-only port.

All port pins are multiplexed with other pin functions. The functions of the multiplex pins are selected by means of the pin function controller (PFC).

Each port is provided with data registers for storing the pin data.

23.1 Port A

Port A is an I/O port with 20 pins shown in figure 23.1.

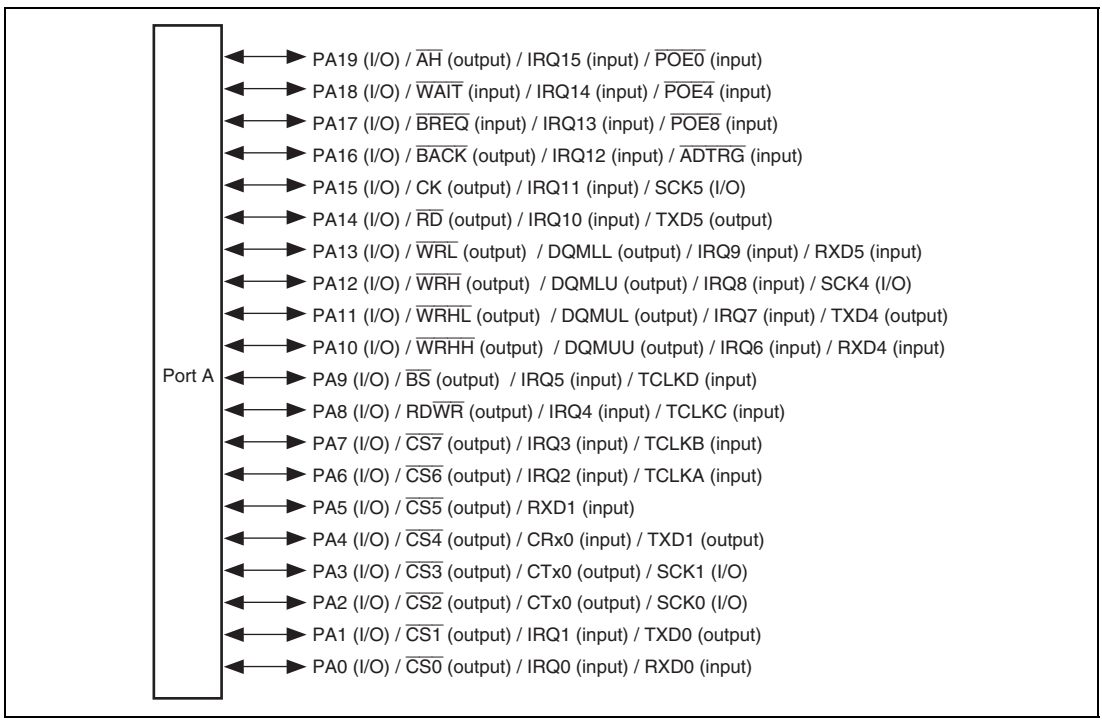


Figure 23.1 Port A

23.1.1 Register Descriptions

Port A has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register H	PADRH	R/W	H'0000	H'FFFE3800	8, 16
Port A data register L	PADRL	R/W	H'0000	H'FFFE3802	8, 16
Port A port register H	PAPRH	R	—	H'FFFE3810	8, 16
Port A port register L	PAPRL	R	—	H'FFFE3812	8, 16

23.1.2 Port A Data Registers H and L (PADRH and PADRL)

PADRH and PADRL are 16-bit readable/writable registers that store port A data. Bits PA19DR to PA0DR correspond to pins PA19 to PA0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PADRH or PADRL, the value is output directly from the pin, and if PADRH or PADRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PADRH or PADRL is read, the pin state, not the register value, is returned directly. If a value is written to PADRH or PADRL, although that value is written into PADRH or PADRL, it does not affect the pin state.

Table 23.2 summarizes read/write operations of port A data register.

- Port A data register H (PADRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PA19 DR	PA18 DR	PA17 DR	PA16 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	PA19DR	0	R/W	See table 23.2.
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

- Port A data register L (PADRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 23.2.
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

Table 23.2 Port A Data Registers H and L (PADRH and PADRL) Read/Write Operations

- PADRH bits 3 to 0 and PADRL bits 15 to 0

PAIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PADRH and PADRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PADRH and PADRL, but it has no effect on pin state.
1	General output	PADRH or PADRL value	The value written is output from the pin.
	Other than general output	PADRH or PADRL value	Can write to PADRH and PADRL, but it has no effect on pin state.

23.1.3 Port A Port Registers H and L (PAPRH and PAPRL)

PAPRH and PAPRL are 16-bit read-only registers, which return the states of the pins regardless of the PFC setting. In this LSI, bits PA19PR to PA0PR correspond to pins PA19 to PA0, respectively (description of multiplexed functions are abbreviated here).

- Port A port register H (PAPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PA19 PR	PA18 PR	PA17 PR	PA16 PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	PA19PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
2	PA18PR	Pin state	R	
1	PA17PR	Pin state	R	
0	PA16PR	Pin state	R	

- Port A port register L (PAPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PA15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PA14PR	Pin state	R	
13	PA13PR	Pin state	R	
12	PA12PR	Pin state	R	
11	PA11PR	Pin state	R	
10	PA10PR	Pin state	R	
9	PA9PR	Pin state	R	
8	PA8PR	Pin state	R	
7	PA7PR	Pin state	R	
6	PA6PR	Pin state	R	
5	PA5PR	Pin state	R	
4	PA4PR	Pin state	R	
3	PA3PR	Pin state	R	
2	PA2PR	Pin state	R	
1	PA1PR	Pin state	R	
0	PA0PR	Pin state	R	

23.2 Port B

Port B is an I/O port with 14 pins shown in figure 23.2.

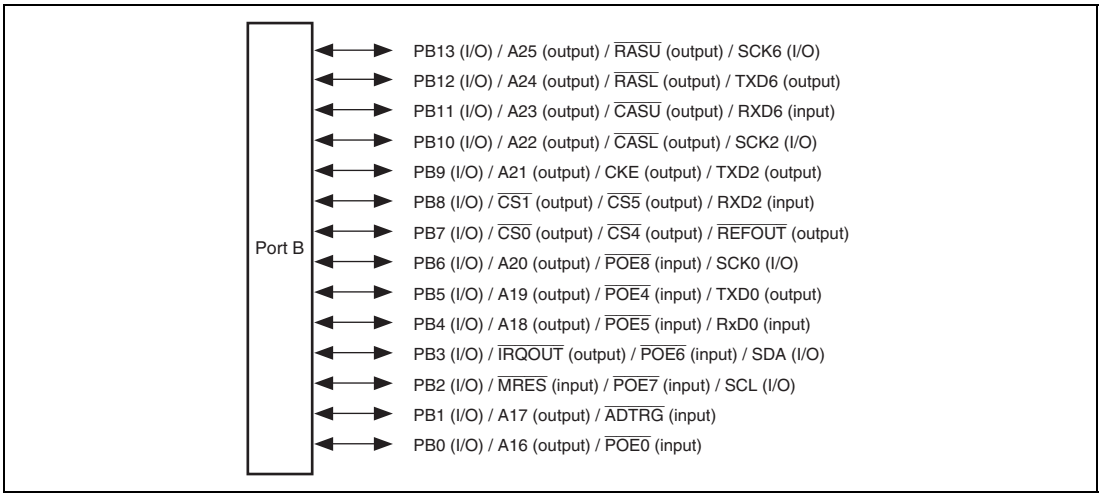


Figure 23.2 Port B

23.2.1 Register Descriptions

Port B has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register L	PBDRL	R/W	H'0000	H'FFFE3882	8, 16
Port B port register L	PBPRL	R	—	H'FFFE3892	8, 16

23.2.2 Port B Data Register L (PBDRL)

PBDRL is a 16-bit readable/writable register that stores port B data. Bits PB13DR to PB0DR correspond to pins PB13 to PB0, respectively (description of multiplexed functions are abbreviated here. When a pin function is general output, if a value is written to PBDRL, the value is output directly from the pin, and if PBDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PBDRL is read, the pin state, not the register value, is returned directly. If a value is written to PBDRL, although that value is written into PBDRL, it does not affect the pin state.

Table 23.4 summarizes read/write operations of port B data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PB13DR	0	R/W	See table 23.4.
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 23.4 Port B Data Register L (PBDRL) Read/Write Operations

- PBDRL bits 13 to 0

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PBDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PBDRL, but it has no effect on pin state.
1	General output	PBDRL value	The value written is output from the pin.
	Other than general output	PBDRL value	Can write to PBDRL, but it has no effect on pin state.

23.2.3 Port B Port Register L (PBPRL)

PBPRL is a 16-bit read-only register, which returns the states of the pins regardless of the PFC setting. In this LSI, bits PB13PR to PB0PR correspond to pins PB13 to PB0, respectively (description of multiplexed functions are abbreviated here).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PB13PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
12	PB12PR	Pin state	R	
11	PB11PR	Pin state	R	
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
6	PB6PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
5	PB5PR	Pin state	R	
4	PB4PR	Pin state	R	
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	PB0PR	Pin state	R	

23.3 Port C

Port C is an I/O port with 16 pins shown in figure 23.3.

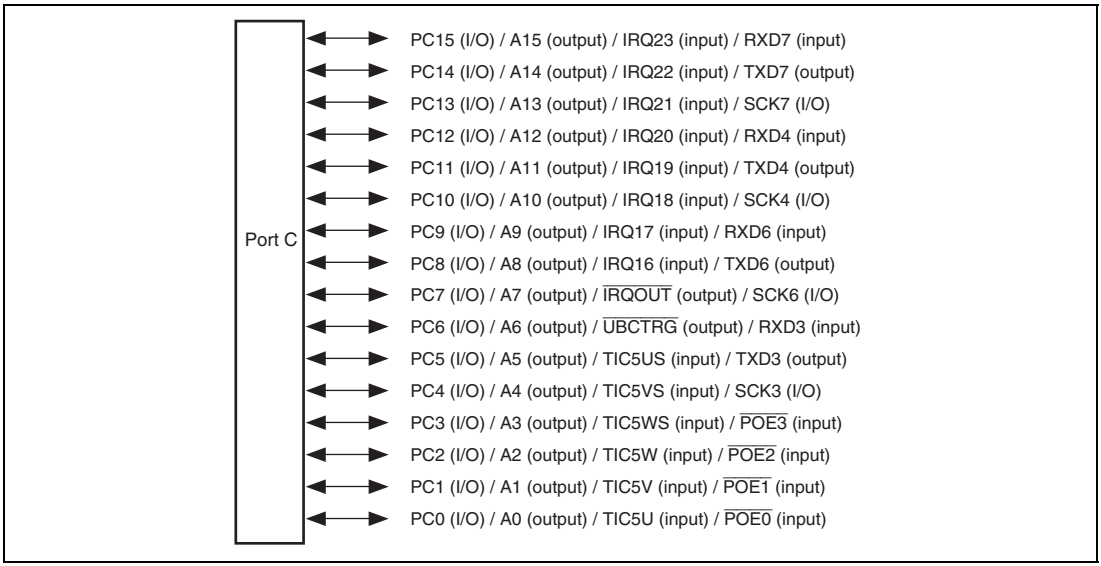


Figure 23.3 Port C

23.3.1 Register Descriptions

Port C has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register L	PCDRL	R/W	H'0000	H'FFFE3902	8, 16
Port C port register L	PCPRL	R	—	H'FFFE3912	8, 16

23.3.2 Port C Data Register L (PCDRL)

PCDRL is a 16-bit readable/writable register that store port C data. Bits PC15DR to PC0DR correspond to pins PC15 to PC0 (description of multiplexed functions are abbreviated) respectively. When a pin function is general output, if a value is written to PCDRL, the value is output directly from the pin, and if PCDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PCDRL is read, the pin state, not the register value, is returned directly. If a value is written to PCDRL, although that value is written into PCDRL, it does not affect the pin state.

Table 23.6 summarizes read/write operations of port C data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PC15DR	0	R/W	See table 23.6.
14	PC14DR	0	R/W	
13	PC13DR	0	R/W	
12	PC12DR	0	R/W	
11	PC11DR	0	R/W	
10	PC10DR	0	R/W	
9	PC9DR	0	R/W	
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

Table 23.6 Port C Data Register L (PCDRL) Read/Write Operations

- PCDRL bits 15 to 0

PCIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PCDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PCDRL, but it has no effect on pin state.
1	General output	PCDRL value	The value written is output from the pin.
	Other than general output	PCDRL value	Can write to PCDRL, but it has no effect on pin state.

23.3.3 Port C Port Register L (PCPRL)

PCPRL is a 16-bit read-only register, which always returns the states of the pins regardless of the PFC setting. In this LSI, bits PC15PR to PC0PR correspond to pins PC15 to PC0, respectively (description of multiplexed functions are abbreviated here).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PC15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PC14PR	Pin state	R	
13	PC13PR	Pin state	R	
12	PC12PR	Pin state	R	
11	PC11PR	Pin state	R	
10	PC10PR	Pin state	R	
9	PC9PR	Pin state	R	
8	PC8PR	Pin state	R	
7	PC7PR	Pin state	R	
6	PC6PR	Pin state	R	
5	PC5PR	Pin state	R	
4	PC4PR	Pin state	R	
3	PC3PR	Pin state	R	
2	PC2PR	Pin state	R	
1	PC1PR	Pin state	R	
0	PC0PR	Pin state	R	

23.4 Port D

Port D is an I/O port with 32 pins shown in figure 23.4.

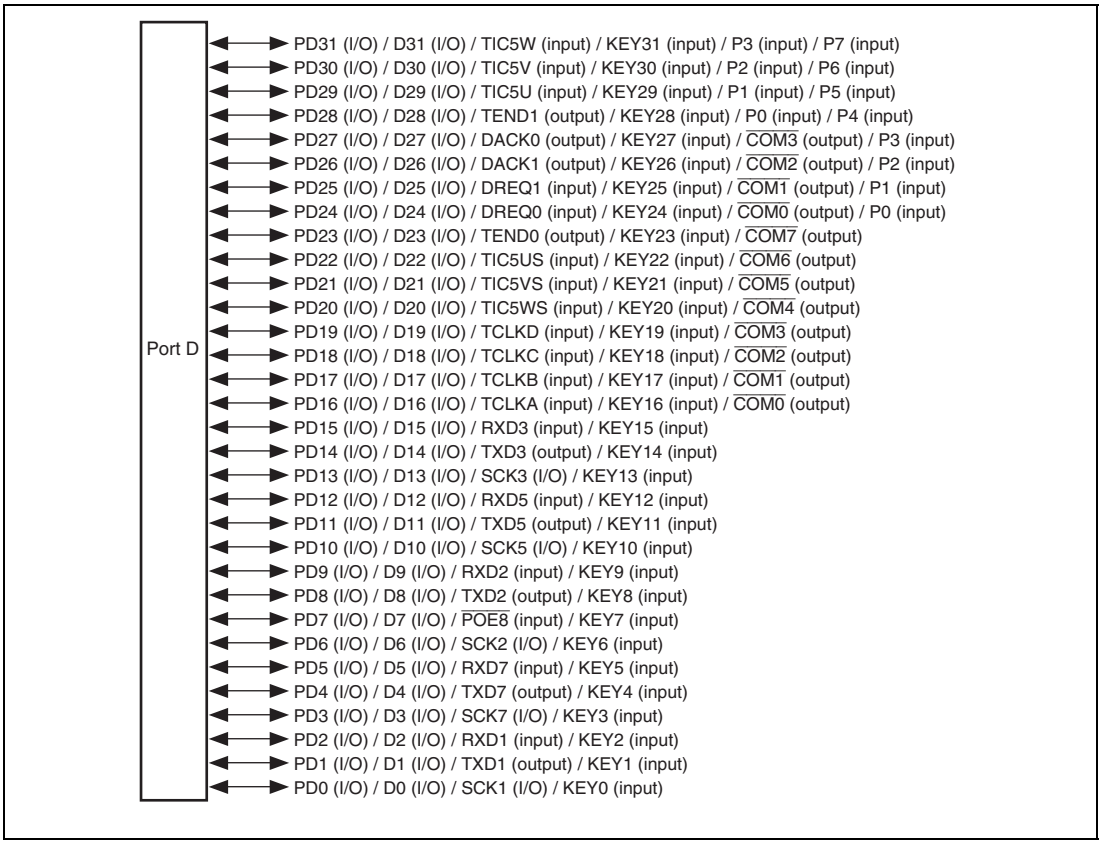


Figure 23.4 Port D

23.4.1 Register Descriptions

Port D has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register H	PDDRH	R/W	H'0000	H'FFFE3980	8, 16
Port D data register L	PDDRL	R/W	H'0000	H'FFFE3982	8, 16
Port D port register H	PDPRH	R	—	H'FFFE3990	8, 16
Port D port register L	PDPRL	R	—	H'FFFE3992	8, 16

23.4.2 Port D Data Registers H and L (PDDRH and PDDRL)

PDDRH and PDDRL are 16-bit readable/writable registers that store port D data. In this LSI, bits PD31DR to PD0DR correspond to pins PD31 to PD0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PDDRH or PDDRL, the value is output directly from the pin, and if PDDRH or PDDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PDDRH or PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRH or PDDRL, although that value is written into PDDRH or PDDRL, it does not affect the pin state.

Table 23.8 summarizes read/write operations of port D data register.

- Port D data register H (PDDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 DR	PD30 DR	PD29 DR	PD28 DR	PD27 DR	PD26 DR	PD25 DR	PD24 DR	PD23 DR	PD22 DR	PD21 DR	PD20 DR	PD19 DR	PD18 DR	PD17 DR	PD16 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD31DR	0	R/W	See table 23.8.
14	PD30DR	0	R/W	
13	PD29DR	0	R/W	
12	PD28DR	0	R/W	
11	PD27DR	0	R/W	
10	PD26DR	0	R/W	
9	PD25DR	0	R/W	
8	PD24DR	0	R/W	
7	PD23DR	0	R/W	
6	PD22DR	0	R/W	
5	PD21DR	0	R/W	
4	PD20DR	0	R/W	
3	PD19DR	0	R/W	
2	PD18DR	0	R/W	
1	PD17DR	0	R/W	
0	PD16DR	0	R/W	

- Port D data register L (PDDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD15DR	0	R/W	See table 23.8.
14	PD14DR	0	R/W	
13	PD13DR	0	R/W	
12	PD12DR	0	R/W	
11	PD11DR	0	R/W	
10	PD10DR	0	R/W	
9	PD9DR	0	R/W	
8	PD8DR	0	R/W	
7	PD7DR	0	R/W	
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

Table 23.8 Port D Data Registers H and L (PDDRH and PDDRL) Read/Write Operations

- PDDRH bits 15 to 0 and PDDRL bits 15 to 0

PDIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PDDRH and PDDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PDDRH and PDDRL, but it has no effect on pin state.
1	General output	PDDRH or PDDRL value	The value written is output from the pin.
	Other than general output	PDDRH or PDDRL value	Can write to PDDRH and PDDRL, but it has no effect on pin state.

23.4.3 Port D Port Registers H and L (PDPRH and PDPRL)

PDPRH and PDPRL are 16-bit read-only registers, which return the states of the pins regardless of the PFC setting. In this LSI, bits PD31PR to PD0PR correspond to pins PD31 to PD0, respectively (description of multiplexed functions are abbreviated here).

- Port D port register H (PDPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31PR	PD30PR	PD29PR	PD28PR	PD27PR	PD26PR	PD25PR	PD24PR	PD23PR	PD22PR	PD21PR	PD20PR	PD19PR	PD18PR	PD17PR	PD16PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PD31PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PD30PR	Pin state	R	
13	PD29PR	Pin state	R	
12	PD28PR	Pin state	R	
11	PD27PR	Pin state	R	
10	PD26PR	Pin state	R	
9	PD25PR	Pin state	R	
8	PD24PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
7	PD23PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
6	PD22PR	Pin state	R	
5	PD21PR	Pin state	R	
4	PD20PR	Pin state	R	
3	PD19PR	Pin state	R	
2	PD18PR	Pin state	R	
1	PD17PR	Pin state	R	
0	PD16PR	Pin state	R	

- Port D port register L (PDPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 PR	PD14 PR	PD13 PR	PD12 PR	PD11 PR	PD10 PR	PD9 PR	PD8 PR	PD7 PR	PD6 PR	PD5 PR	PD4 PR	PD3 PR	PD2 PR	PD1 PR	PD0 PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PD15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PD14PR	Pin state	R	
13	PD13PR	Pin state	R	
12	PD12PR	Pin state	R	
11	PD11PR	Pin state	R	
10	PD10PR	Pin state	R	
9	PD9PR	Pin state	R	
8	PD8PR	Pin state	R	
7	PD7PR	Pin state	R	
6	PD6PR	Pin state	R	
5	PD5PR	Pin state	R	
4	PD4PR	Pin state	R	
3	PD3PR	Pin state	R	
2	PD2PR	Pin state	R	
1	PD1PR	Pin state	R	
0	PD0PR	Pin state	R	

23.5 Port E

Port E is an I/O port with 24 pins shown in figure 23.5.

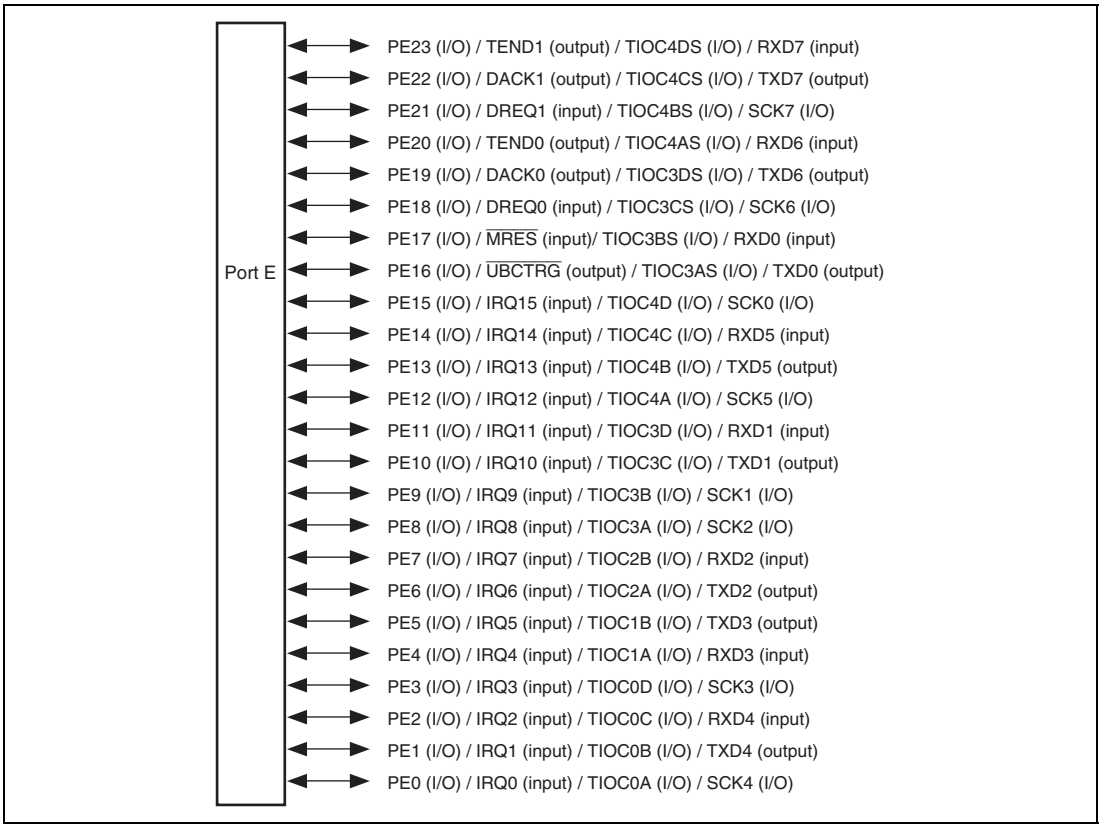


Figure 23.5 Port E

23.5.1 Register Descriptions

Port E has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.9 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register H	PEDRH	R/W	H'0000	H'FFFE3A00	8, 16
Port E data register L	PEDRL	R/W	H'0000	H'FFFE3A02	8, 16
Port E port register H	PEPRH	R	—	H'FFFE3A10	8, 16
Port E port register L	PEPRL	R	—	H'FFFE3A12	8, 16

23.5.2 Port E Data Registers H and L (PEDRH and PEDRL)

PEDRH and PEDRL are 16-bit readable/writable registers that store port E data. In this LSI, bits PE23DR to PE0DR correspond to pins PE23 to PE0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PEDRH or PEDRL, the value is output directly from the pin, and if PEDRH or PEDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PEDRH or PEDRL is read, the pin state, not the register value, is returned directly. If a value is written to PEDRH or PEDRL, although that value is written into PEDRH or PEDRL, it does not affect the pin state.

Table 23.10 summarizes read/write operations of port E data register.

- Port E data register H (PEDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PE23DR	0	R/W	See table 23.10.
6	PE22DR	0	R/W	
5	PE21DR	0	R/W	
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

- Port E data register L (PEDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PE15DR	0	R/W	See table 23.10.
14	PE14DR	0	R/W	
13	PE13DR	0	R/W	
12	PE12DR	0	R/W	
11	PE11DR	0	R/W	
10	PE10DR	0	R/W	
9	PE9DR	0	R/W	
8	PE8DR	0	R/W	
7	PE7DR	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
6	PE6DR	0	R/W	See table 23.10.
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 23.10 Port E Data Registers H and L (PEDRH and PEDRL) Read/Write Operations

- PEDRH bits 7 to 0 and PEDRL bits 15 to 0

PEIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PEDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PEDRL, but it has no effect on pin state.
1	General output	PEDRL value	The value written is output from the pin.
	Other than general output	PEDRL value	Can write to PEDRL, but it has no effect on pin state.

23.5.3 Port E Port Registers H and L (PEPRH and PEPRL)

PEPRH and PEPRL are 16-bit read-only registers, which returns the states of the pins regardless of the PFC setting. In this LSI, bits PE23PR to PE0PR correspond to pins PE23 to PE0, respectively (description of multiplexed functions are abbreviated here).

- Port E port register H (PEPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PE23PR	PE22PR	PE21PR	PE20PR	PE19PR	PE18PR	PE17PR	PE16PR
Initial value:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PE23PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
6	PE22PR	Pin state	R	
5	PE21PR	Pin state	R	
4	PE20PR	Pin state	R	
3	PE19PR	Pin state	R	
2	PE18PR	Pin state	R	
1	PE17PR	Pin state	R	
0	PE16PR	Pin state	R	

- Port E port register L (PEPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PE15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PE14PR	Pin state	R	
13	PE13PR	Pin state	R	
12	PE12PR	Pin state	R	
11	PE11PR	Pin state	R	
10	PE10PR	Pin state	R	
9	PE9PR	Pin state	R	
8	PE8PR	Pin state	R	
7	PE7PR	Pin state	R	
6	PE6PR	Pin state	R	
5	PE5PR	Pin state	R	
4	PE4PR	Pin state	R	
3	PE3PR	Pin state	R	
2	PE2PR	Pin state	R	
1	PE1PR	Pin state	R	
0	PE0PR	Pin state	R	

23.6 Port F

Port F is an I/O port with 16 pins shown in figure 23.6.

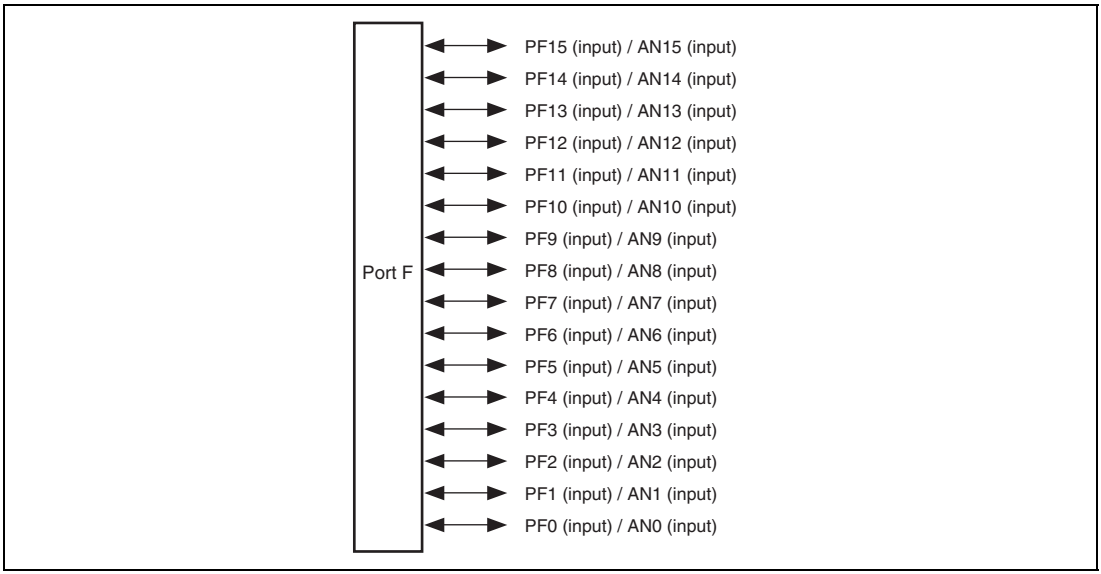


Figure 23.6 Port F

23.6.1 Register Descriptions

Port F has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.11 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register L	PFDRL	R	—	H'FFFE3A82	8, 16

23.6.2 Port F Data Register L (PFDRL)

PFDRL is a 16-bit read-only register that stores port F data. In this LSI, bits PF15DR to PF0DR correspond to pins PF15 to PF0, respectively (description of multiplexed functions are abbreviated here).

Even if a value is written to PFDR, the value is not written into PFDR, and it does not affect the pin state. If PFDR is read, the pin state, not the register value, is returned directly. However, when sampling the analog input of A/D converter, 1 is read. Table 23.12 summarizes read/write operations of port F data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PF15DR	Pin state	R	See table 23.12.
14	PF14DR	Pin state	R	
13	PF13DR	Pin state	R	
12	PF12DR	Pin state	R	
11	PF11DR	Pin state	R	
10	PF10DR	Pin state	R	
9	PF9DR	Pin state	R	
8	PF8DR	Pin state	R	
7	PF7DR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
6	PF6DR	Pin state	R	See table 23.12.
5	PF5DR	Pin state	R	
4	PF4DR	Pin state	R	
3	PF3DR	Pin state	R	
2	PF2DR	Pin state	R	
1	PF1DR	Pin state	R	
0	PF0DR	Pin state	R	

Table 23.12 Port F Data Register L (PFDRL) Read/Write Operations

- PFDRL bits 15 to 0

State	Read	Write
Other than analog input sampling	Pin state	Ignored (no effect on pin state)
Analog input sampling	1	Ignored (no effect on pin state)

23.7 Port G

Port G is an I/O port with 16 pins shown in figure 23.7.

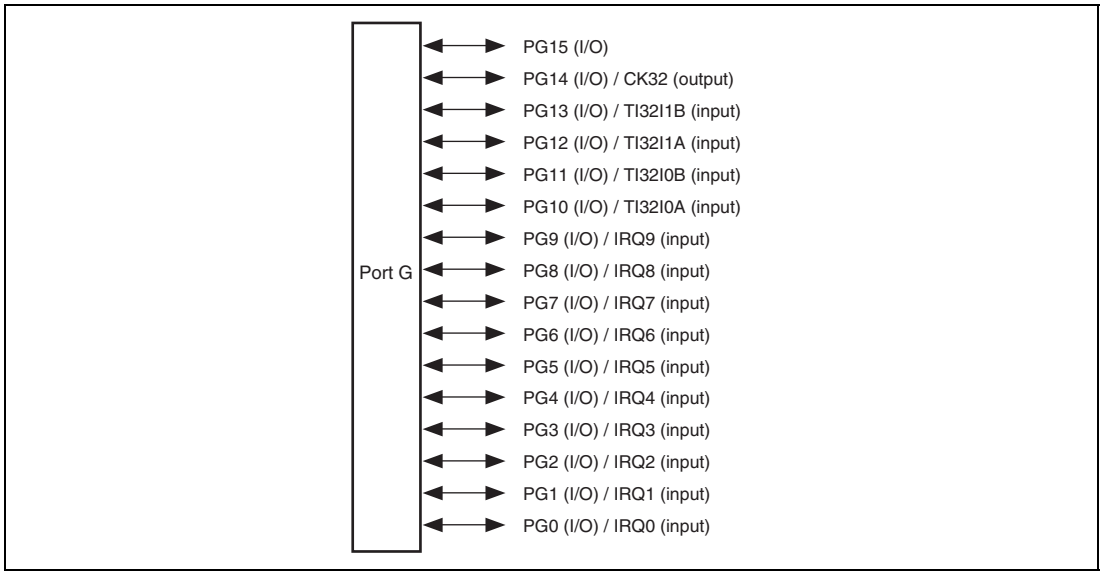


Figure 23.7 Port G

23.7.1 Register Descriptions

Port G has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.13 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port G data register L	PGDRL	R/W	H'0000	H'FFFE3B02	8, 16
Port G port register L	PGPRL	R	—	H'FFFE3B12	8, 16

23.7.2 Port G Data Register L (PGDRL)

PGDRL is a 16-bit readable/writable register that stores port G data. In this LSI, bits PG15DR to PG0DR correspond to pins PG15 to PG0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PGDRL, the value is output directly from the pin, and if PGDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PGDRL is read, the pin state, not the register value, is returned directly. If a value is written to PGDRL, although that value is written into PGDRL, it does not affect the pin state.

Table 23.14 summarizes read/write operations of port G data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15DR	PG14DR	PG13DR	PG12DR	PG11DR	PG10DR	PG9DR	PG8DR	PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PG15DR	0	R/W	See table 23.14.
14	PG14DR	0	R/W	
13	PG13DR	0	R/W	
12	PG12DR	0	R/W	
11	PG11DR	0	R/W	
10	PG10DR	0	R/W	
9	PG9DR	0	R/W	
8	PG8DR	0	R/W	
7	PG7DR	0	R/W	
6	PG6DR	0	R/W	
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

Table 23.14 Port G Data Register L (PGDRL) Read/Write Operations

- PGDRL bits 15 to 0

PGIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PGDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PGDRL, but it has no effect on pin state.
1	General output	PGDRL value	The value written is output from the pin.
	Other than general output	PGDRL value	Can write to PGDRL, but it has no effect on pin state.

23.7.3 Port G Port Register L (PGPRL)

PGPRL is a 16-bit read-only register, which returns the states of the pins regardless of the PFC setting. In this LSI, bits PG15PR to PG0PR correspond to pins PG15 to PG0, respectively (description of multiplexed functions are abbreviated here).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 PR	PG14 PR	PG13 PR	PG12 PR	PG11 PR	PG10 PR	PG9 PR	PG8 PR	PG7 PR	PG6 PR	PG5 PR	PG4 PR	PG3 PR	PG2 PR	PG1 PR	PG0 PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PG15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PG14PR	Pin state	R	
13	PG13PR	Pin state	R	
12	PG12PR	Pin state	R	
11	PG11PR	Pin state	R	
10	PG10PR	Pin state	R	
9	PG9PR	Pin state	R	
8	PG8PR	Pin state	R	
7	PG7PR	Pin state	R	
6	PG6PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
5	PG5PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
4	PG4PR	Pin state	R	
3	PG3PR	Pin state	R	
2	PG2PR	Pin state	R	
1	PG1PR	Pin state	R	
0	PG0PR	Pin state	R	

23.8 Port H

Port H is an I/O port with 16 pins shown in figure 23.8.

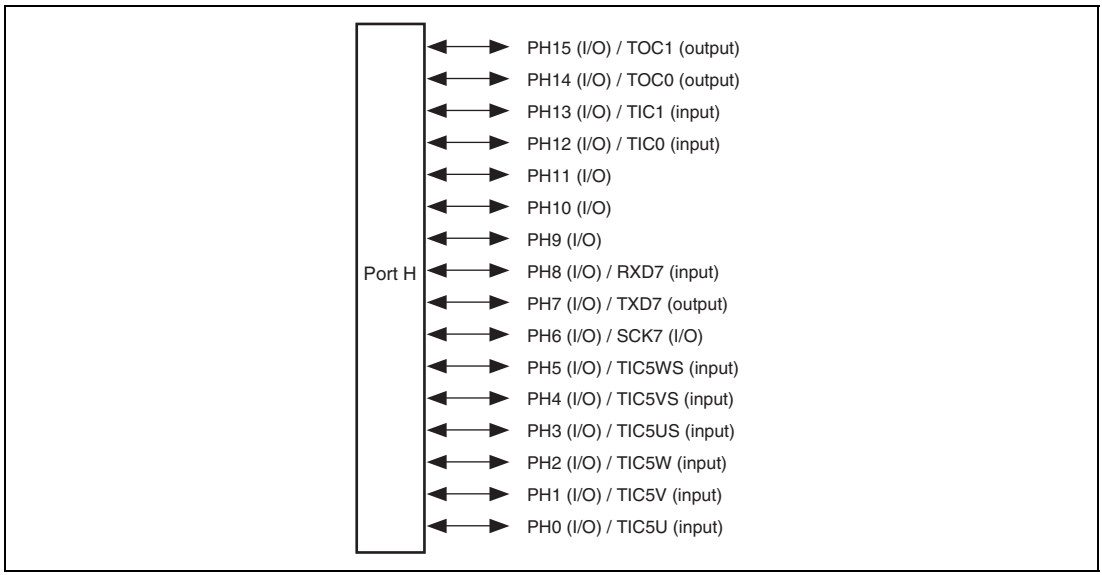


Figure 23.8 Port H

23.8.1 Register Descriptions

Port H has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.15 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port H data register L	PHDRL	R/W	H'0000	H'FFFE3B82	8, 16
Port H port register L	PHPRL	R	—	H'FFFE3B92	8, 16

23.8.2 Port H Data Register L (PHDRL)

PHDRL is a 16-bit readable/writable register that stores port H data. In this LSI, bits PH15DR to PH0DR correspond to pins PH15 to PH0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PHDRL, the value is output directly from the pin, and if PHDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PHDRL is read, the pin state, not the register value, is returned directly. If a value is written to PHDRL, although that value is written into PHDRL, it does not affect the pin state.

Table 23.16 summarizes read/write operations of port H data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15 DR	PH14 DR	PH13 DR	PH12 DR	PH11 DR	PH10 DR	PH9 DR	PH8 DR	PH7 DR	PH6 DR	PH5 DR	PH4 DR	PH3 DR	PH2 DR	PH1 DR	PH0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PH15DR	0	R/W	See table 23.16.
14	PH14DR	0	R/W	
13	PH13DR	0	R/W	
12	PH12DR	0	R/W	
11	PH11DR	0	R/W	
10	PH10DR	0	R/W	
9	PH9DR	0	R/W	
8	PH8DR	0	R/W	
7	PH7DR	0	R/W	
6	PH6DR	0	R/W	
5	PH5DR	0	R/W	
4	PH4DR	0	R/W	
3	PH3DR	0	R/W	
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

Table 23.16 Port H Data Register L (PHDRL) Read/Write Operations

- PHDRL bits 15 to 0

PHIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PHDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PHDRL, but it has no effect on pin state.
1	General output	PHDRL value	The value written is output from the pin.
	Other than general output	PHDRL value	Can write to PHDRL, but it has no effect on pin state.

23.8.3 Port H Port Register L (PHPRL)

PHPRL is a 16-bit read-only register, which returns the states of the pins regardless of the PFC setting. In this LSI, bits PH15PR to PH0PR correspond to pins PH15 to PH0, respectively (description of multiplexed functions are abbreviated here).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15PR	PH14PR	PH13PR	PH12PR	PH11PR	PH10PR	PH9PR	PH8PR	PH7PR	PH6PR	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PH15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PH14PR	Pin state	R	
13	PH13PR	Pin state	R	
12	PH12PR	Pin state	R	
11	PH11PR	Pin state	R	
10	PH10PR	Pin state	R	
9	PH9PR	Pin state	R	
8	PH8PR	Pin state	R	
7	PH7PR	Pin state	R	
6	PH6PR	Pin state	R	
5	PH5PR	Pin state	R	
4	PH4PR	Pin state	R	
3	PH3PR	Pin state	R	
2	PH2PR	Pin state	R	
1	PH1PR	Pin state	R	
0	PH0PR	Pin state	R	

23.9 Port J

Port J is an I/O port with 16 pins shown in figure 23.9.

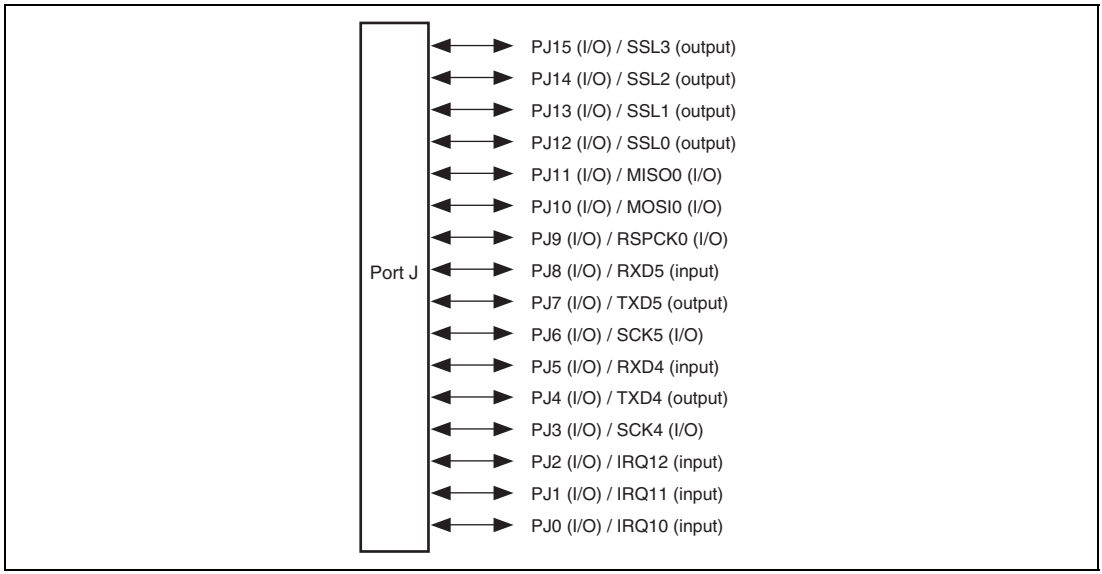


Figure 23.9 Port J

23.9.1 Register Descriptions

Port J has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.17 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port J data register L	PJDRL	R/W	H'0000	H'FFFE3C82	8, 16
Port J port register L	PJPRL	R	—	H'FFFE3C92	8, 16

23.9.2 Port J Data Register L (PJDRL)

PJDRL is a 16-bit readable/writable register that stores port J data. In this LSI, bits PJ15DR to PJ0DR correspond to pins PJ15 to PJ0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PJDRL, the value is output directly from the pin, and if PJDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PJDRL is read, the pin state, not the register value, is returned directly. If a value is written to PJDRL, although that value is written into PJDRL, it does not affect the pin state.

Table 23.18 summarizes read/write operations of port J data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ15DR	PJ14DR	PJ13DR	PJ12DR	PJ11DR	PJ10DR	PJ9DR	PJ8DR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PJ15DR	0	R/W	See table 23.18.
14	PJ14DR	0	R/W	
13	PJ13DR	0	R/W	
12	PJ12DR	0	R/W	
11	PJ11DR	0	R/W	
10	PJ10DR	0	R/W	
9	PJ9DR	0	R/W	
8	PJ8DR	0	R/W	
7	PJ7DR	0	R/W	
6	PJ6DR	0	R/W	
5	PJ5DR	0	R/W	
4	PJ4DR	0	R/W	
3	PJ3DR	0	R/W	
2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

Table 23.18 Port J Data Register L (PJDRL) Read/Write Operations

- PJDRL bits 15 to 0

PJIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PJDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PJDRL, but it has no effect on pin state.
1	General output	PJDRL value	The value written is output from the pin.
	Other than general output	PJDRL value	Can write to PJDRL, but it has no effect on pin state.

23.9.3 Port J Port Register L (PJPRL)

PJPRL is a 16-bit read-only register, which returns the states of the pins regardless of the PFC setting. In this LSI, bits PJ15PR to PJ0PR correspond to pins PJ15 to PJ0, respectively (description of multiplexed functions are abbreviated here).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ15PR	PJ14PR	PJ13PR	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR	PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PJ15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PJ14PR	Pin state	R	
13	PJ13PR	Pin state	R	
12	PJ12PR	Pin state	R	
11	PJ11PR	Pin state	R	
10	PJ10PR	Pin state	R	
9	PJ9PR	Pin state	R	
8	PJ8PR	Pin state	R	
7	PJ7PR	Pin state	R	
6	PJ6PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
5	PJ5PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
4	PJ4PR	Pin state	R	
3	PJ3PR	Pin state	R	
2	PJ2PR	Pin state	R	
1	PJ1PR	Pin state	R	
0	PJ0PR	Pin state	R	

23.10 Port K

Port K is an I/O port with 8 pins shown in figure 23.10.

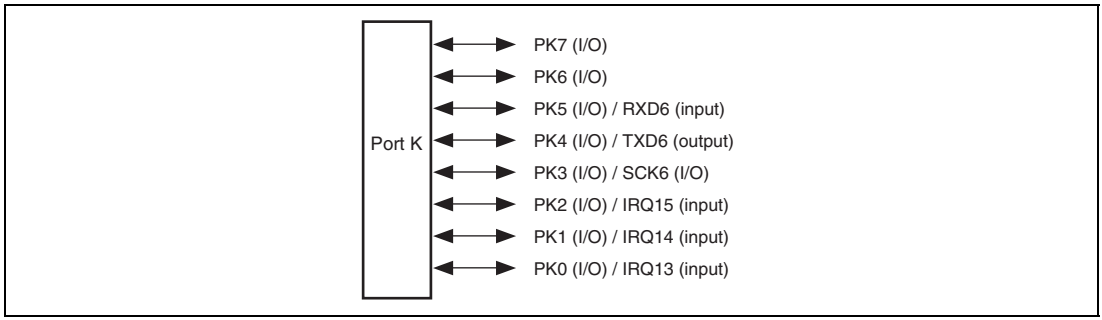


Figure 23.10 Port K

23.10.1 Register Descriptions

Port K has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.19 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port K data register L	PKDRL	R/W	H'0000	H'FFFE3D02	8, 16
Port K port register L	PKPRL	R	—	H'FFFE3D12	8, 16

23.10.2 Port K Data Register L (PKDRL)

PKDRL is a 16-bit readable/writable register that stores port K data. In this LSI, bits PK7DR to PK0DR correspond to pins PK7 to PK0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PKDRL, the value is output directly from the pin, and if PKDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PKDRL is read, the pin state, not the register value, is returned directly. If a value is written to PKDRL, although that value is written into PKDRL, it does not affect the pin state.

Table 23.20 summarizes read/write operations of port K data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK7 DR	PK6 DR	PK5 DR	PK4 DR	PK3 DR	PK2 DR	PK1 DR	PK0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PK7DR	0	R/W	See table 23.20.
6	PK6DR	0	R/W	
5	PK5DR	0	R/W	
4	PK4DR	0	R/W	
3	PK3DR	0	R/W	
2	PK2DR	0	R/W	
1	PK1DR	0	R/W	
0	PK0DR	0	R/W	

Table 23.20 Port K Data Register L (PKDRL) Read/Write Operations

- PKDRL bits 7 to 0

PKIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PKDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PKDRL, but it has no effect on pin state.
1	General output	PKDRL value	The value written is output from the pin.
	Other than general output	PKDRL value	Can write to PKDRL, but it has no effect on pin state.

23.10.3 Port K Port Register L (PKPRL)

PKPRL is a 16-bit read-only register, which returns the states of the pins regardless of the PFC setting. In this LSI, bits PK7PR to PK0PR correspond to pins PK7 to PK0, respectively (description of multiplexed functions are abbreviated here).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK7PR	PK6PR	PK5PR	PK4PR	PK3PR	PK2PR	PK1PR	PK0PR
Initial value:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PK7PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
6	PK6PR	Pin state	R	
5	PK5PR	Pin state	R	
4	PK4PR	Pin state	R	
3	PK3PR	Pin state	R	
2	PK2PR	Pin state	R	
1	PK1PR	Pin state	R	
0	PK0PR	Pin state	R	

23.11 Port L

Port L is an I/O port with 6 pins shown in figure 23.11.

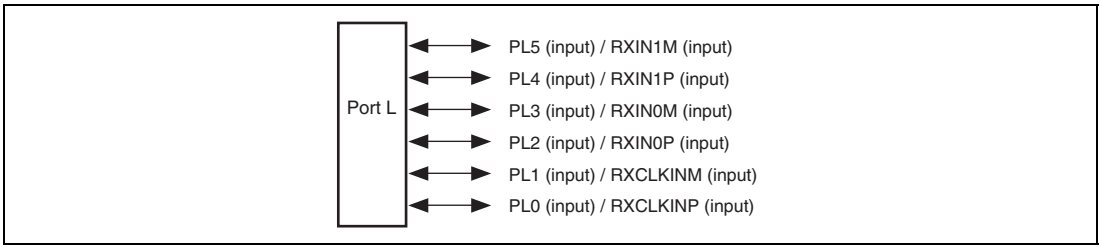


Figure 23.11 Port L

23.11.1 Register Descriptions

Port L has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 23.21 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port L data register L	PLDRL	R	—	H'FFFE3D82	8, 16

23.11.2 Port L Data Register L (PLDRL)

PLDRL is a 16-bit read-only register that stores port L data. In this LSI, bits PL5DR to PL0DR correspond to pins PL5 to PL0, respectively (description of multiplexed functions are abbreviated here). Even if a value is written to a bit, the pin state is not affected. If a PLDRL bit is read, the pin state, not the register value, is returned directly. In the SH72315A, when the LVDS module standby is cancelled (when the MSTP67 bit in the STBCR6 register is cleared to 0), 0 is read.

Table 23.22 summarizes read/write operations of port L data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PL5DR	PL4DR	PL3DR	PL2DR	PL1DR	PL0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PL5DR	Pin state	R	See table 23.22.
4	PL4DR	Pin state	R	
3	PL3DR	Pin state	R	
2	PL2DR	Pin state	R	
1	PL1DR	Pin state	R	
0	PL0DR	Pin state	R	

Table 23.22 Port L Data Register L (PLDRL) Read/Write Operations

- PLDRL bits 5 to 0

Product Type and State	Read	Write
SH72315A with the LVDS in module standby SH72315L/SH72314L	Pin state	Ignored (no effect on pin state)
SH72315A with the LVDS not in module standby	0	Ignored (no effect on pin state)

Section 24 LVDS Receive Interface (LVDS) (SH72315A only)

This LSI includes two channels of LVDS receive interface (LVDS).

24.1 Features

- Two input channels (reception with only one channel can be specified)
- Number of data bits: 16 bits/channel
- Input clock: 20 MHz (min.) to 160 MHz (max.)
- Data transfer rate: 40 Mbps (min.) to 320 Mbps (max.)
- The receiving unit has a 16-stage FIFO buffer.
- Differential input registor included
- Four types of interrupts

  Receive FIFO data full, end code detection, underrun error, and overrun error interrupts, and each interrupt can be requested independently. The data transfer controller (DTC) or direct memory access controller (DMAC) can be activated by the receive FIFO data full interrupt to transfer data.
- Module standby mode can be set.

Figure 24.1 shows a block diagram of the LVDS.

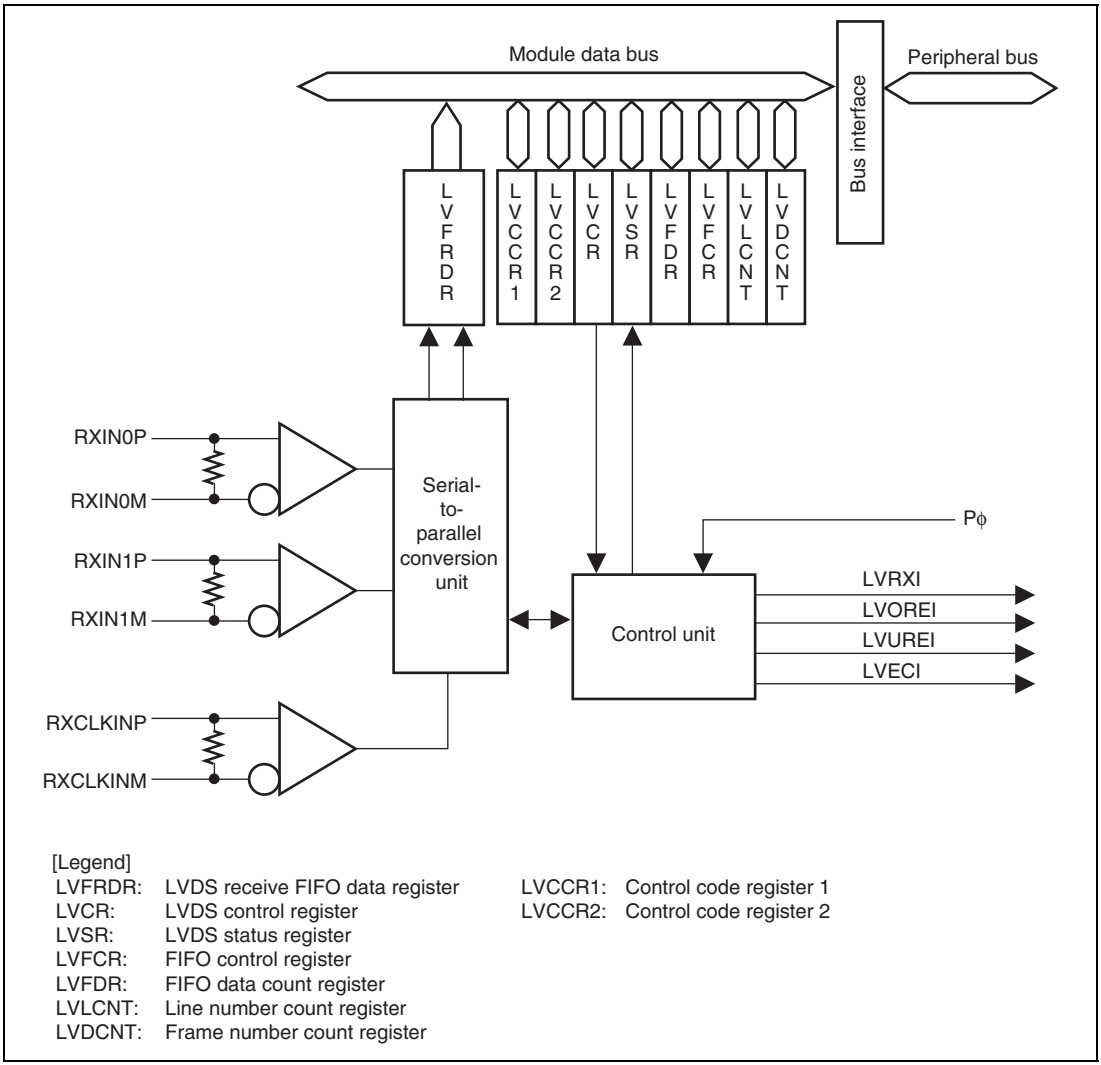


Figure 24.1 Block Diagram of LVDS

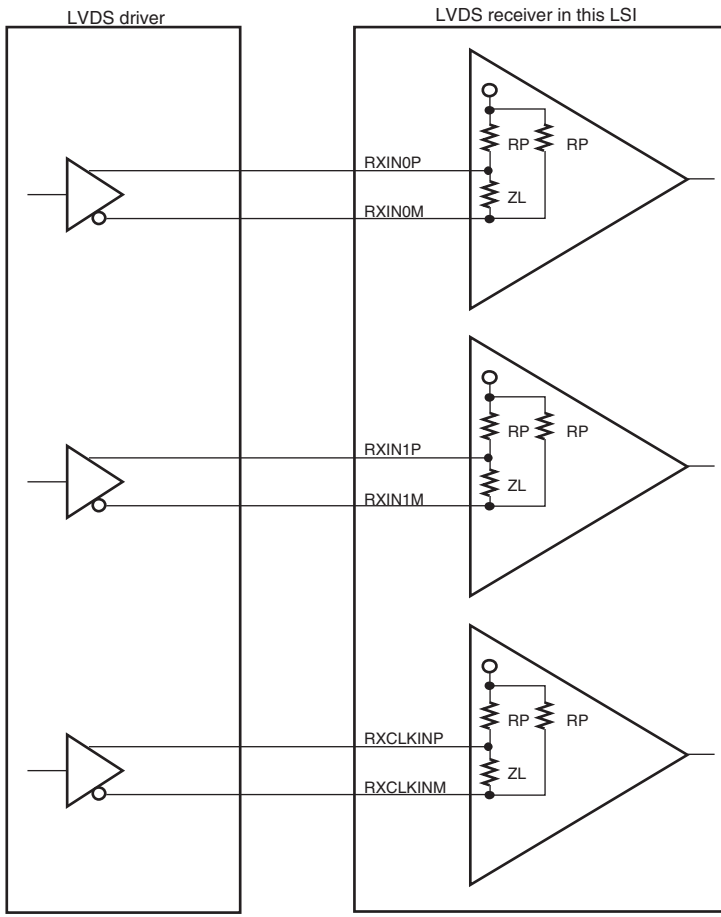
24.2 Input/Output Pin

Table 24.1 shows the LVDS pin configuration.

Table 24.1 Pin Configuration

Channel	Name	Pin Name	I/O	Function
Common	LVDS power supply pin	LVDSVCC	Input	LVDS power supply pin and reference potential
	LVDS power supply pin	LVDSVSS	Input	LVDS ground and reference potential
	LVDS clock pin	RXCLKINP	Input	LVDS clock input (+)
	LVDS clock pin	RXCLKINM	Input	LVDS clock input (–)
Channel 0	LVDS data pin	RXIN0P	Input	LVDS data (+)
	LVDS data pin	RXIN0M	Input	LVDS data (–)
Channel 1	LVDS data pin	RXIN1P	Input	LVDS data (+)
	LVDS data pin	RXIN1M	Input	LVDS data (–)

Figure 24.2 shows an example of connecting differential input circuits.



[Legend]
RP: Internal pull-up resistors
ZL: Internal differential input impedance resistors

Note: Supply power to this LSI before applying the voltage to the LVDS input pins.

Figure 24.2 Example of Connecting LVDS Differential Input Circuits

24.3 Register Descriptions

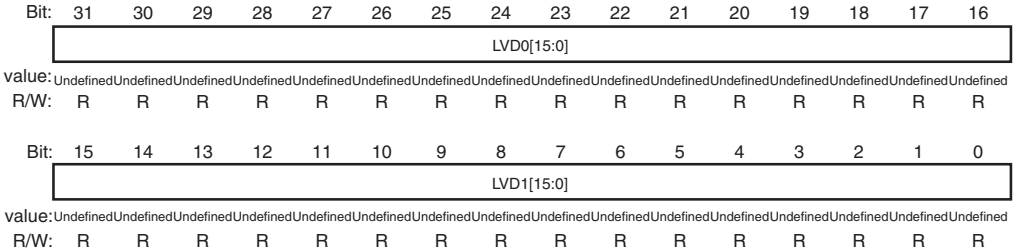
The LVDS has the following registers. See section 34, List of Registers for details on the register address and states in each operating mode.

Table 24.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
LVDS receive FIFO data register	LVFRDR	R	Undefined	H'FFFEB000	32
LVDS control register	LVCR	R/W	H'0000	H'FFFEB004	16
LVDS status register	LVSR	R/W	H'0000	H'FFFEB006	16
FIFO control register	LVFCR	R/W	H'0070	H'FFFEB008	16
FIFO data count register	LVFDR	R/W	H'0000	H'FFFEB00A	16
Control code register 1	LVCCR1	R/W	H'0000	H'FFFEB00C	16
Control code register 2	LVCCR2	R/W	H'0000	H'FFFEB00E	16
Line number count register	LVLCNT	R	H'0000	H'FFFEB010	16
Frame number count register	LVDCNT	R	H'0000	H'FFFEB012	16

24.3.1 LVDS Receive FIFO Data Register (LVFRDR)

LVFRDR is a 16-stage FIFO register for storing received data. Receiving data, the LVDS stores the data in LVFRDR. When LVFRDR is filled with received data, the subsequently received data is discarded. If LVFRDR is read when it contains no received data, the undefined value is read out.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	LVD0[15:0]	Undefined	R	Channel 0 Received Data Channel 0 received data is stored.
15 to 0	LVD1[15:0]	Undefined	R	Channel 1 Received Data Channel 1 received data is stored.

24.3.2 LVDS Control Register (LVCR)

LVCR enables or disables reception; specifies the number of channels; determines the end code; and enables or disables interrupts.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	LVECIE	LVRXIE	LVU REIE	LVO REIE	—	—	—	—	ENDC	—	CHSEL	RE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	LVECIE	0	R/W	End Code Detection Interrupt Enable Enables or disables an end code detection interrupt to be generated. 0: Disables an end code detection interrupt (LVECI). 1: Enables an end code detection interrupt (LVECI).
10	LVRXIE	0	R/W	Receive FIFO Data Full Interrupt Enable Enables or disables a receive FIFO data full interrupt to be generated. 0: Disables a receive FIFO data full interrupt (LVRXI). 1: Enables a receive FIFO data full interrupt (LVRXI).
9	LVUREIE	0	R/W	Underrun Error Interrupt Enable Enables or disables an underrun error interrupt to be generated. 0: Disables an underrun error interrupt (LVUREI). 1: Enables an underrun error interrupt (LVUREI).
8	LVOREIE	0	R/W	Overrun Error Interrupt Enable Enables or disables an overrun error interrupt to be generated. 0: Disables an overrun error interrupt (LVOREI). 1: Enables an overrun error interrupt (LVOREI).
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	ENDC	0	R/W	Data End Code Check Specifies whether to stop taking in the received data to FIFO upon receiving the end code (EOL or EOF) or to continue receiving data without checking an end code. 0: Does not check data whether it is an end code. 1: Stops taking in data to FIFO upon receiving an end code. When ENDC is 0, the LVLCNT and LVDCNT registers are invalid. When data is to be handled as two-dimensional image frames, be sure to set ENDC to 1. Note: This bit can be modified only when RE is 0.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CHSEL	0	R/W	Channel Number Select 0: Uses two channels. 1: Uses channel 0 (RXIN0) only. RXIN1 data is discarded. Note: This bit can be modified only when RE is 0.
0	RE	0	R/W	Reception Enable Enables or disables reception. 0: Disables reception. 1: Enables reception.

24.3.3 LVDS Status Register (LVSR)

LVSR is a 16-bit interrupt status flag register of the LVDS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	EOF	SOF	EOL	SOL	—	—	—	—	ECDET	RDRF	URER	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	EOF	0	R/(W)*	EOF Code Detection Indicates that the EOF code has been detected. 0: Indicates that the EOF code has not been detected. [Clearing conditions] • Power-on reset • A 0 is written to the EOF bit after 1 is read from the bit. 1: Indicates that the EOF code has been detected. [Setting condition] • The EOF code following the synchronization code is detected.
10	SOF	0	R/(W)*	SOF Code Detection Indicates that the SOF code has been detected. 0: Indicates that the SOF code has not been detected. [Clearing conditions] • Power-on reset • A 0 is written to the SOF bit after 1 is read from the bit. 1: Indicates that the SOF code has been detected. [Setting condition] • The SOF code following the synchronization code is detected.

Bit	Bit Name	Initial Value	R/W	Description
9	EOL	0	R/(W)*	EOL Code Detection Indicates that the EOL code has been detected. 0: Indicates that the EOL code has not been detected. [Clearing conditions] • Power-on reset • A 0 is written to the EOL bit after 1 is read from the bit. 1: Indicates that the EOL code has been detected. [Setting condition] • The EOL code following the synchronization code is detected.
8	SOL	0	R/(W)*	SOL Code Detection Indicates that the SOL code has been detected. 0: Indicates that the SOL code has not been detected. [Clearing conditions] • Power-on reset • A 0 is written to the SOL bit after 1 is read from the bit. 1: Indicates that the SOL code has been detected. [Setting condition] • The SOL code following the synchronization code is detected.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	ECDET	0	R/(W)*	End Code Detection Indicates that the EOF or EOL code has been detected. 0: Indicates that the end code has not been detected. [Clearing conditions] • Power-on reset • A 0 is written to the ECDET bit after 1 is read from the bit. 1: Indicates that the end code has been detected. [Setting condition] • The EOF or EOL code is detected while the ENDC bit in LVCR is 1.

Bit	Bit Name	Initial Value	R/W	Description
2	RDRF	0	R/(W)*	Receive FIFO Data Full Indicates that the received data has been stored in the LVDS receive FIFO data register (LVFRDR) and that the number of the received data units in LVFRDR has reached the receive trigger number specified by the LRTRG bits in the FIFO control register (LVFCR) as a result. 0: Indicates that the number of the received data units stored in LVFRDR is smaller than the specified receive trigger number. [Clearing conditions] • Power-on reset • Data is read from LVFRDR until the number of the received data units in LVFRDR becomes smaller than the specified receive trigger number and 0 is written to the RDRF bit after 1 is read from the bit. • Data is read from LVFRDR by the DMAC activated by the LVRXI interrupt for the times specified by the DMA transfer count register (DMATCR) leaving the smaller number of received data units in LVFRDR than the specified receive trigger number. • Data is read from LVFRDR by the DTC activated by the LVRXI interrupt while the DISEL bit in MRB of the DTC is 0 for the times specified by the DTC transfer count register (CRA) leaving the smaller number of received data units in LVFRDR than the specified receive trigger number. 1: Indicates that the number of the received data units stored in LVFRDR is equal to or larger than the specified receive trigger number. [Setting condition] • The larger number of received data units are stored in LVFRDR than the specified receive trigger number. Note: Attempting to write the larger number of received data units than the number of FIFO stages causes an overrun error thus disabling received data to be written.

Bit	Bit Name	Initial Value	R/W	Description
1	URER	0	R/(W)*	Underrun Error Indicates that an underrun error has occurred. 0: Indicates that no underrun error has occurred. [Clearing conditions] • Power-on reset • A 0 is written to the URER bit after 1 is read from the bit. 1: Indicates that an underrun error has occurred. [Setting condition] • LVFRDR is read when LVFRDR contains no received data.
0	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error has occurred. 0: Indicates that no overrun error has occurred. [Clearing conditions] • Power-on reset • A 0 is written to the ORER bit after 1 is read from the bit. 1: Indicates that an overrun error has occurred. [Setting condition] • The received data for the 17th stage is written to LVFRDR. Note: The data received before an overrun error occurs is retained in LVFRDR and the subsequently received data is discarded. In addition, received data in the LVDS cannot be stored in LVFRDR while ORER is 1.

Note: * Only 0 can be written to after reading 1, to clear the flag.

24.3.4 FIFO Control Register (LVFCR)

LVFCR specifies the trigger number of receive FIFO data units in the LVDS receive FIFO data register (LVFRDR) and resets LVFRDR. LVFCR should be modified after the RE bit in LVCR is cleared to 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	LRTRG[2:0]			—	—	—	FRST
Initial value:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/(W)*

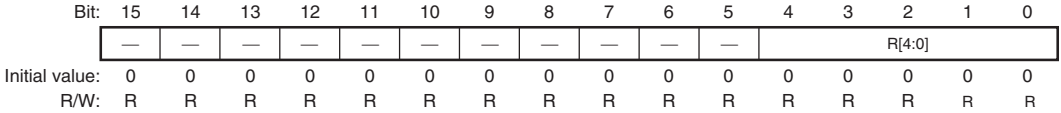
Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 4	LRTRG [2:0]	All 1	R/W	Receive FIFO Data Trigger Number Specifies the number of received data units (specified receive trigger number) based on which the RDF flag in the LVDS status register (LVSR) is set. When the number of the received data units in the LVDS receive FIFO data register (LVFRDR) reaches the specified trigger number, the RDF flag is set to 1. 000: Setting prohibited 001: 2 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	FRST	0	R/(W)*	LVDS Receive FIFO Data Register Reset Setting this bit to 1 initializes the R[4:0] bits in the FIFO data count register and resets (empties) the LVDS receive FIFO data register (LVFRDR). This bit is automatically cleared to 0 after the FIFO is reset. Note: Reset operation is provided at a power-on reset.

Note: * Only 0 can be written to after reading 1, to clear the flag.

24.3.5 FIFO Data Count Register (LVFDR)

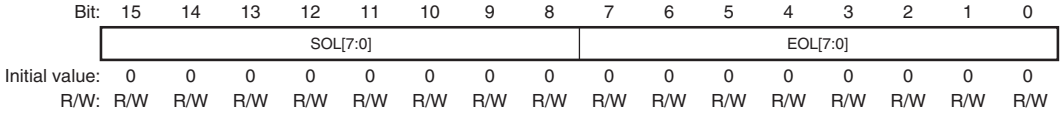
LVFDR indicates the number of data units stored in the LVDS receive FIFO data register (LVFRDR).



Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	R[4:0]	All 0	R	Indicates the number of received data units in LVFRDR. H'00 indicates that the receive FIFO contains no received data and H'10 indicates that all the sixteen stages of the receive FIFO contain received data.

24.3.6 Control Code Register 1 (LVCCR1)

LVCCR1 specifies the value of the control code to be input following the synchronization code.

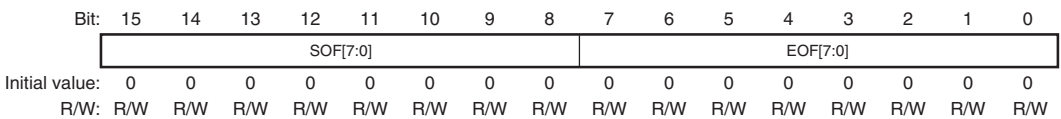


Bit	Bit Name	Initial Value	R/W	Description
15 to 8	SOL[7:0]	All 0	R/W	Line Start Code Specifies the value of the control code pointing out the start of line data.
7 to 0	EOL[7:0]	All 0	R/W	Line End Code Specifies the value of the control code pointing out the end of line data.

Note: Can be modified only when RE is 0.

24.3.7 Control Code Register 2 (LVCCR2)

LVCCR2 specifies the value of the control code input following to the synchronization code.



Bit	Bit Name	Initial Value	R/W	Description
15 to 8	SOF[7:0]	All 0	R/W	Image Frame Start Code Specifies the value of the control code pointing out the start of image frame.
7 to 0	EOF[7:0]	All 0	R/W	Image Frame End Code Specifies the value of the control code pointing out the end of image frame.

Note: Can be modified only when RE is 0.

24.3.8 Line Number Count Register (LVLCNT)

LVLCNT indicates the number of lines from the time of SOF code detection through the time of EOF code detection.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	LCNT [15:0]	All 0	R	Indicates the number of lines received from the time of SOF code detection through the time of EOF code detection. This register is updated when an EOF code is detected. Note: Reset operation is provided at a power-on reset. If RE is set to 1 and then an EOL code is detected without SOF code detection, this register indicates the number of lines received from the time of reception start through the time of EOF code detection.

Note: This register is invalid when the ENDC bit in LVCR is 0.

24.3.9 Frame Number Count Register (LVDCNT)

LVDCNT indicates the number of frames received from the time of SOF or SOL code detection through the time of EOL or EOF code detection.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DCNT [15:0]	All 0	R	Indicates the number of frames received from the time of SOF or SOL code detection through the time of EOL or EOF code detection. This register is updated when an EOL or an EOF code is detected. Note: Reset operation is provided at a power-on reset.

Note: This register is invalid when the ENDC bit in LVCR is 0.

24.4 Operation

The LVDS receives the data input via the RXIN0P/RXIN0M and RXIN1P/RXIN1M pins and takes the received data in the LVDS receive FIFO data register (LVFRDR) at the rising and falling edges of the clock signal pulses supplied via the RXCLKINP/RXCLKINM input pins. When the number of the data units in the receive FIFO reaches the number specified by the LRTRG bits in LVFCR, the LVDS sets the RDRF bit in the LVDS status register (LVSR) to 1. If the LVRXIE bit is 1 here, the LVDS issues the receive FIFO data full interrupt (LVRXI). The LVRXI can be used to activate the DMAC or DTC.

24.4.1 LVDS Reception Format

Figures 24.3 and 24.4 show the reception formats. An image frame is a set of line data and starts with SOF and ends with EOF. Lines are separated by EOL and SOL. Line data consists of the synchronization code, start code (SOF or SOL), received data for one or more frames, synchronization code, and end code (EOF or EOL), in this order.

When data is not to be handled as two-dimensional image frames, SOL and EOL should be used instead of SOF and EOF to separate data strings.

Note that the synchronization code and end code (EOF or EOL) following the received data can be omitted when the ENDC bit in the LVCR register is 0; however, such data cannot be used as two-dimensional image frames.

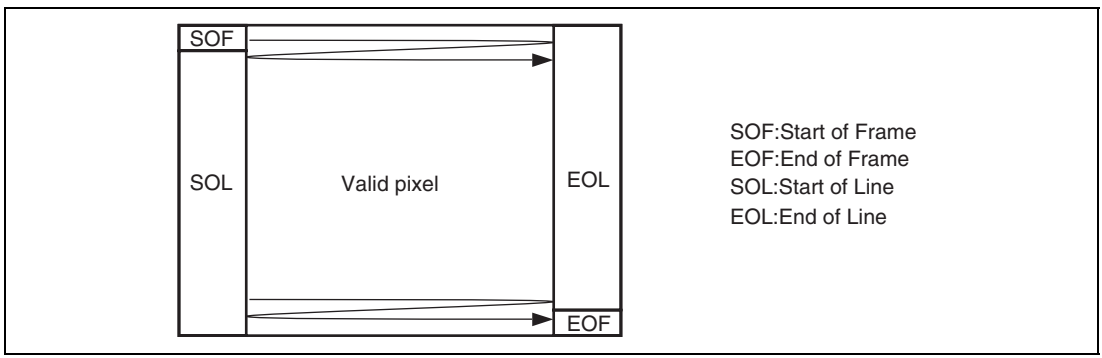


Figure 24.3 Reception Format (Image Frame)

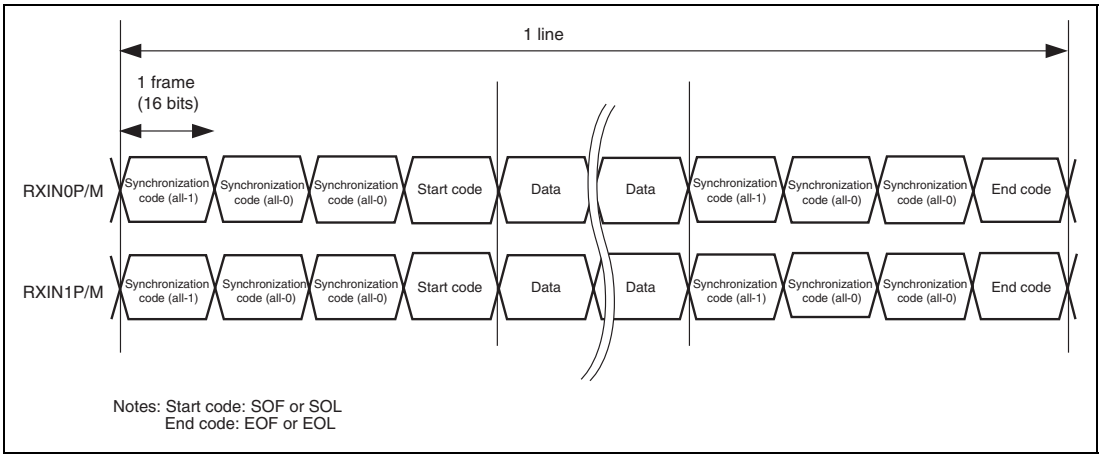


Figure 24.4 Reception Format (Line)

Figure 24.5 shows received data format.

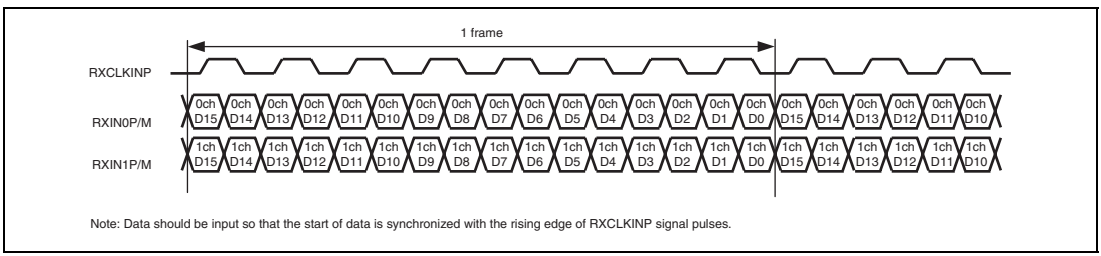


Figure 24.5 Received Data Bit Length

24.4.2 Synchronization Code

The LVDS recognizes the start of received data when it detects the synchronization code in the received data. Figure 24.6 shows the synchronization code. Data should be input so that the start of the synchronization code is synchronized with the rising edge of the RXCLKINP signal pulses.

Neither all-1 data or all-0 data must not be used as the standard data to prevent such data from being confused with the synchronization code.

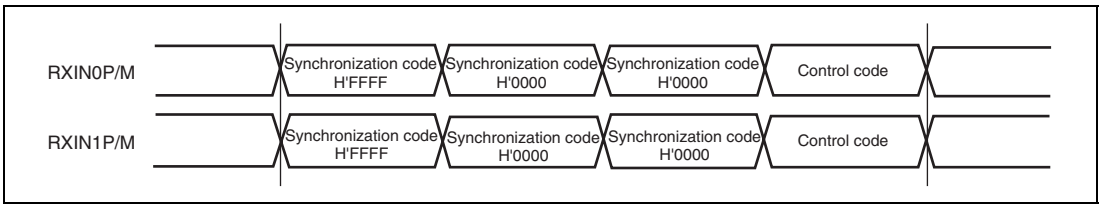


Figure 24.6 Synchronization Code (1)

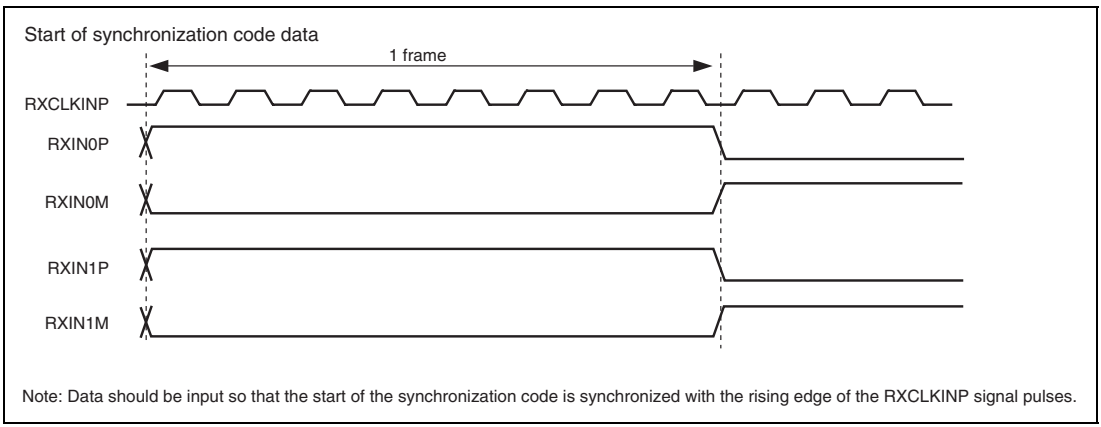
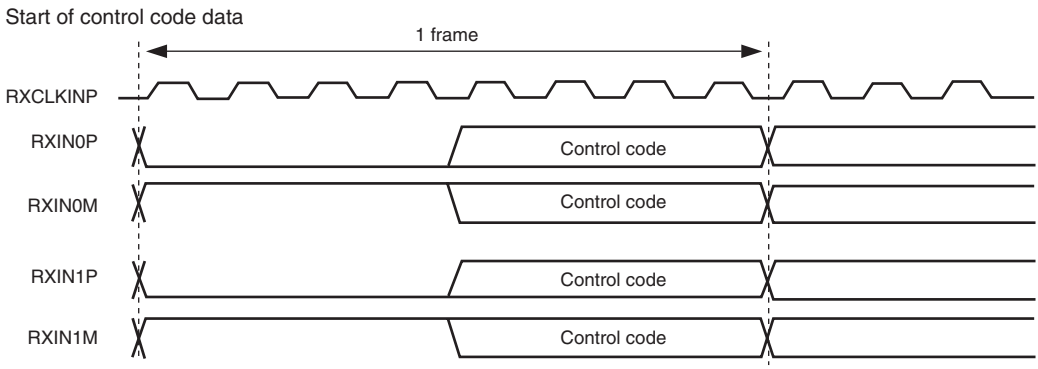


Figure 24.6 Synchronization Code (2)

24.4.3 Control Code

The LVDS recognizes the start of received data when it detects the control code following the synchronization code in the received data. Figure 24.7 shows the control code format.



Note: The control code is specified with lower 8 bits. 0 should be input to the upper bits.

Figure 24.7 Control Code

The detected control code is compared with the values of the SOF, EOF, SOL, and EOL bits in LVCCR1 and LVCCR2. When the control code agrees with any of these bit values, the code is identified as the corresponding control code.

24.4.4 Receiving LVDS Data

Figure 24.8 shows a sample flowchart for receiving LVDS data . Use the following procedure for receiving LVDS data.

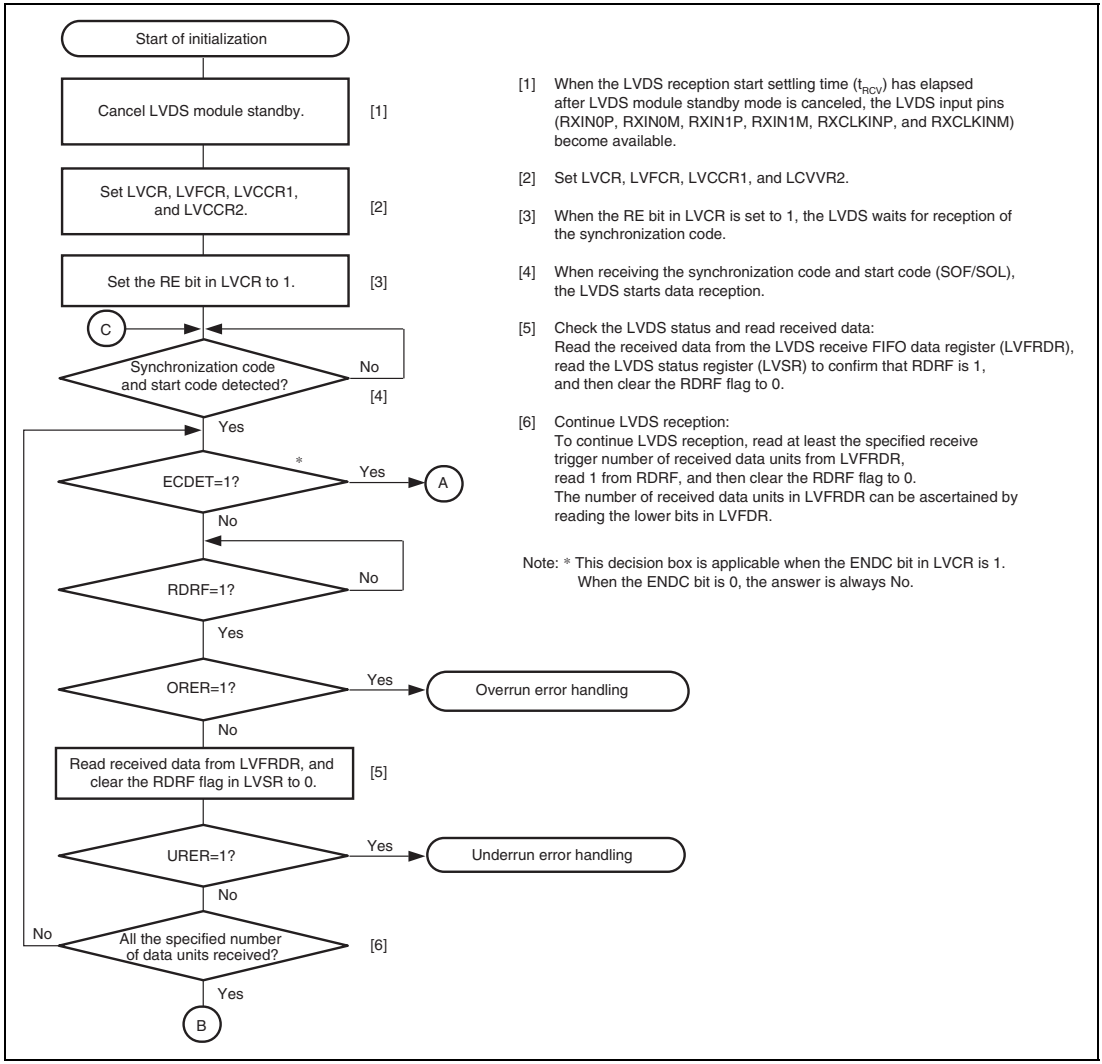


Figure 24.8 Sample Flowchart for Receiving LVDS Data (1)

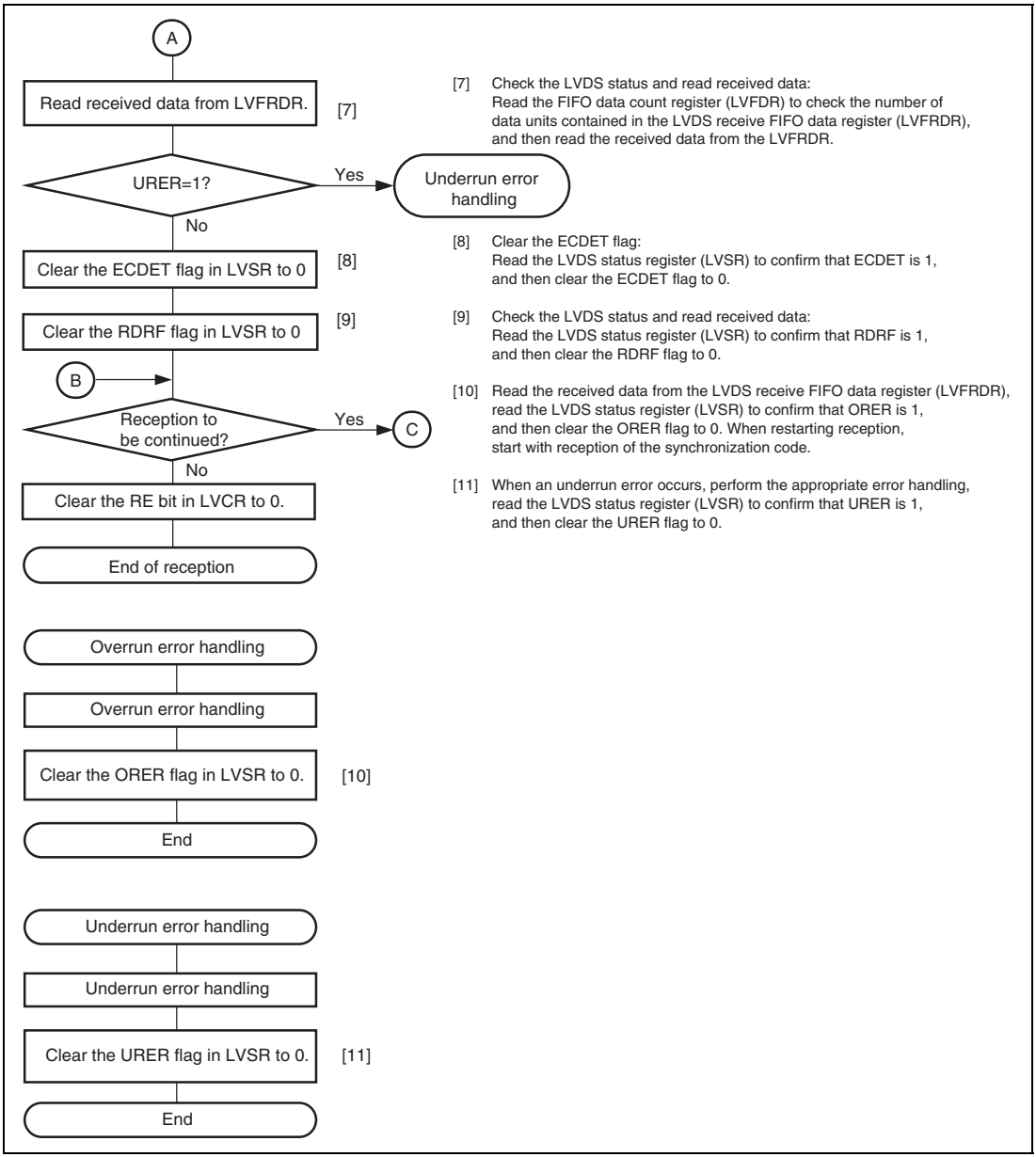


Figure 24.8 Sample Flowchart for Receiving LVDS Data (2)

The LVDS operates as described below during reception.

1. The LVDS automatically detects the synchronization code and starts reception.
2. Taking in data at the clock pulse rising edges and falling edges, the LVDS stores the data in the serial-to-parallel conversion unit, with the MSB first and LSB last. The LVDS checks the LVFDR value to see if LVFRDR is ready to accept the input data. If LVFRDR is ready, the LVDS stores the received data in LVFRDR. If an error occurs during error checking, the LVDS operates as shown in table 24.3.

Note: If a receive error is left unsolved, the following reception sequence is not performed.

3. When the number of the data units stored in the receive FIFO reaches the number specified with the LRTRG bits, the LVDS sets the RDRF flag to 1.
4. If the LVRXIE bit in LVCR is 1 when the RDRF flag is set to 1, the LVDS issues the receive FIFO data full interrupt (LVRXI) request. Similarly, if the LVOREIE bit in LVCR is 1 when the ORER flag is set to 1, the LVDS issues the overrun error interrupt (LVOREI) request.
5. When reading the received data from LVFRDR, the LVDS checks the LVFDR value to see if LVFRDR contains any data. If an error occurs during error checking, the LVDS operates as shown in table 24.3.

Table 24.3 Receive Errors and Error Generation Condition

Receive Error	Abbreviation	Error Generation Condition	Data Storage/Read Result
Overrun error	ORER	An attempt is made to write to LVFRDR when all the stages of it contain valid received data.	Received data is not stored in LVFRDR.
Underrun error	URER	LVFRDR is read when it contains no valid data.	Undefined data is read from LVFRDR.

24.4.5 Forcibly Terminating and Restarting Reception

Figure 24.9 shows a sample flowchart for forcibly terminating and restarting LVDS data reception.

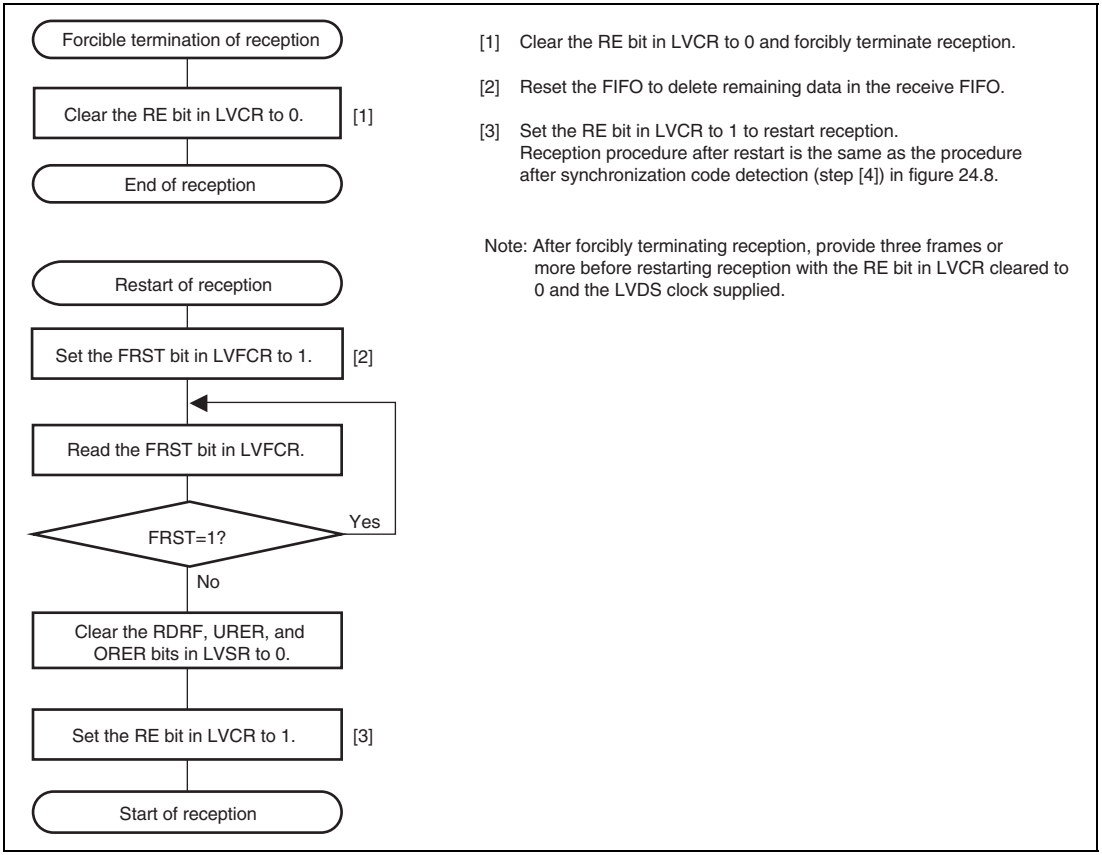


Figure 24.9 Sample Flowchart for Forcibly Terminating and Restarting LVDS Data Reception

24.5 Interrupt Sources and DMAC/DTC

The LVDS has four interrupt sources: overrun error interrupt (LVOREI), underrun error interrupt (LVUREI), end code detection interrupt (LVECI), and receive FIFO data full interrupt (LVRXI). Table 24.4 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the LVOREIE, LVUREIE, LVECIE, and LVRXIE bits in the LVDS control register (LVCR) and are separately issued to the interrupt controller.

If LVOREI is enabled by the LVOREIE bit in LVCR, LVOREI is issued when the ORER flag in the LVDS status register (LVSR) is set to 1.

If LVUREI is enabled by the LVUREIE bit in LVCR, LVUREI is issued when the URER flag in LVSR is set to 1.

If LVRXI is enabled by the LVRXIE bit in LVCR, LVRXI is issued when the RDRF flag in LVSR is set to 1. LVRXI can activate the direct memory access controller (DMAC) or data transfer controller (DTC) to transfer data.

If LVECI is enabled by the LVECIE bit in LVCR, LVECI is issued when the ECDET flag in LVSR is set to 1.

Table 24.4 LVDS Interrupt Sources

Interrupt Source	Interrupt	Interrupt Flag	Interrupt Enable Bit	DMAC/DTC Activation	Priority
LVOREI	Overrun error interrupt	ORER	LVOREIE	Impossible	High
LVUREI	Underrun error interrupt	URER	LVUREIE	Impossible	
LVRXI	Receive FIFO data full interrupt	RDRF	LVRXIE	Possible	
LVECI	End code detection interrupt	ECDET	LVECIE	Impossible	Low

24.6 Accessing LVDS Receive FIFO Data Register by DMAC/DTC

When activating the DMAC by the receive FIFO data full interrupt request to read the LVDS receive FIFO data register (LVFRDR), the TC bit in CHCR should be set to 1 to transfer data for the times specified with DMATCR by a single transfer request, which enables reading the entire received data.

While the R[4:0] bit value in LVFDR is H'0F or smaller and the RE bit in LVCR is 1, the subsequent data can be received.

24.6.1 Data Transfer by DMAC

Figure 24.10 shows an example of data transfer from the LVDS receive FIFO data register by the DMAC with the reload function, and figure 24.11 shows an example of the timing chart.

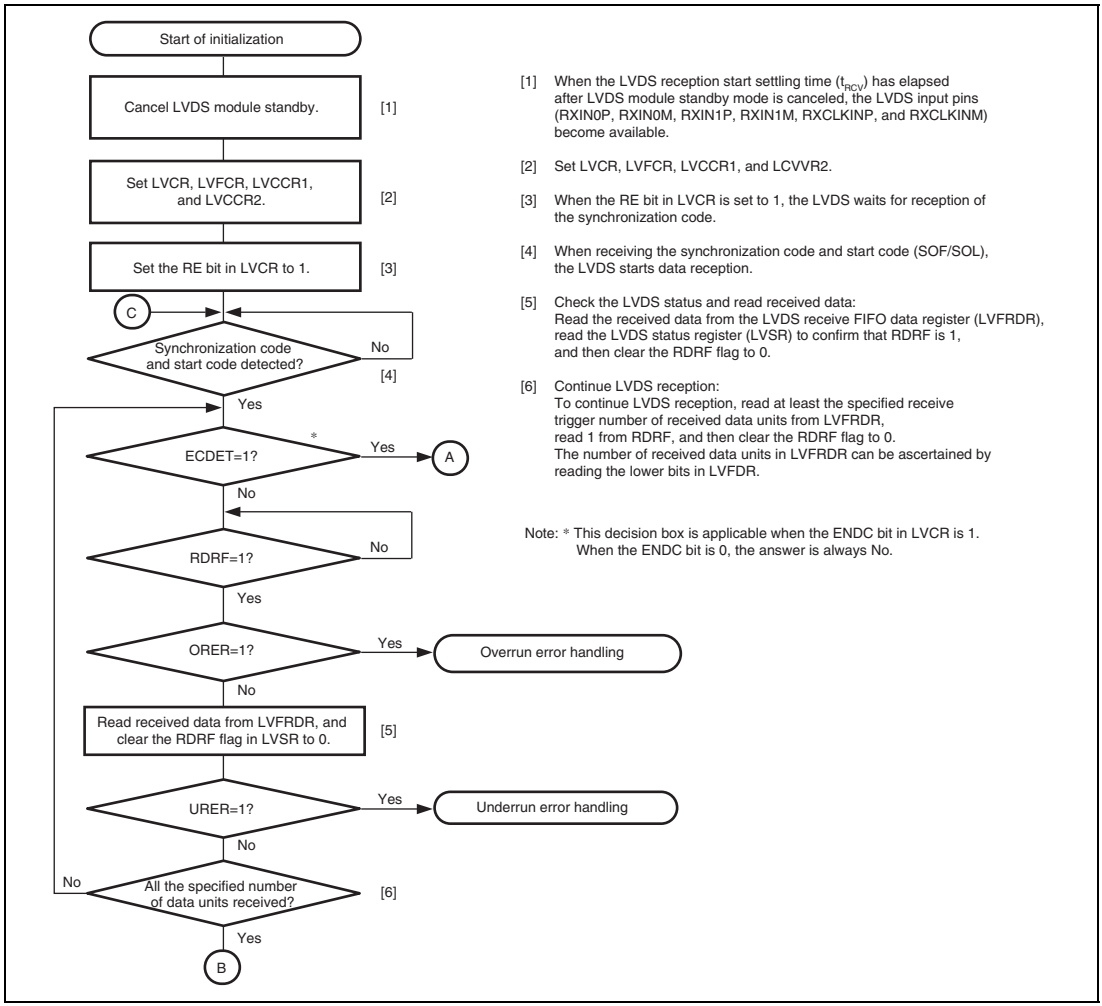


Figure 24.10 Example of Data Transfer from LVDS Receive FIFO Data Register by DMAC (1)

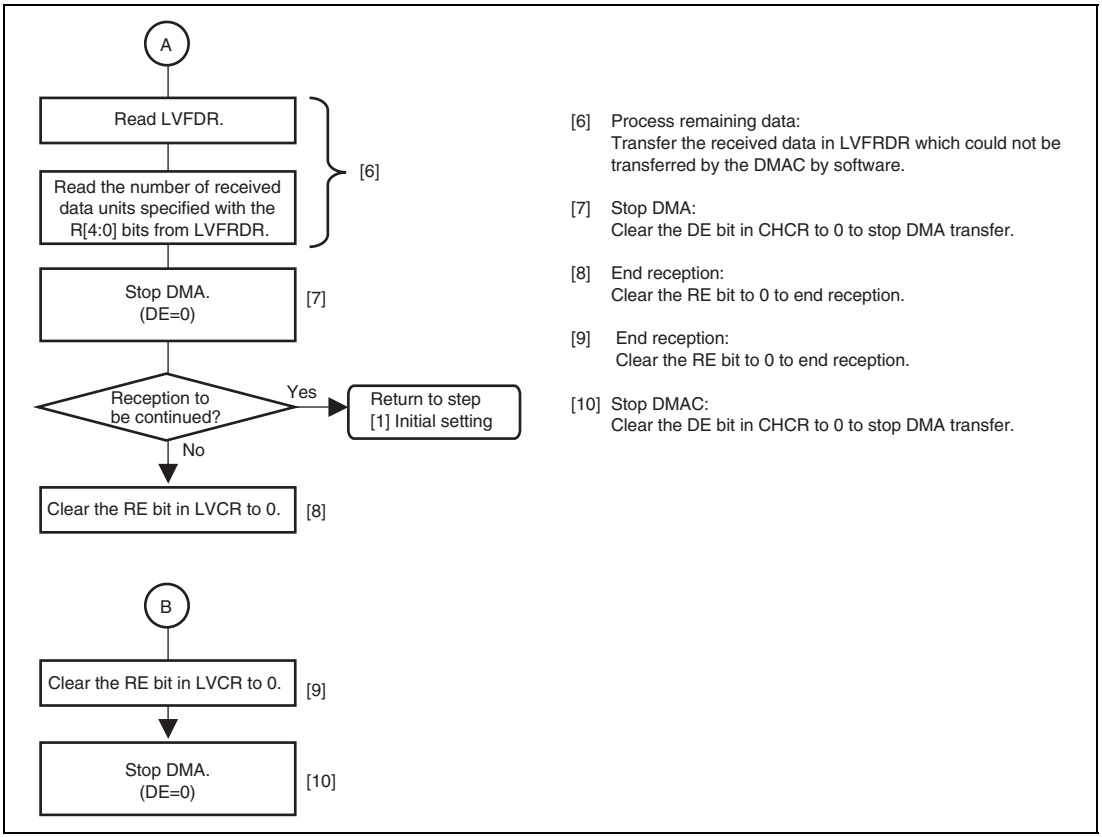


Figure 24.10 Example of Data Transfer in LVDS Receive FIFO Data Register by DMAC (2)

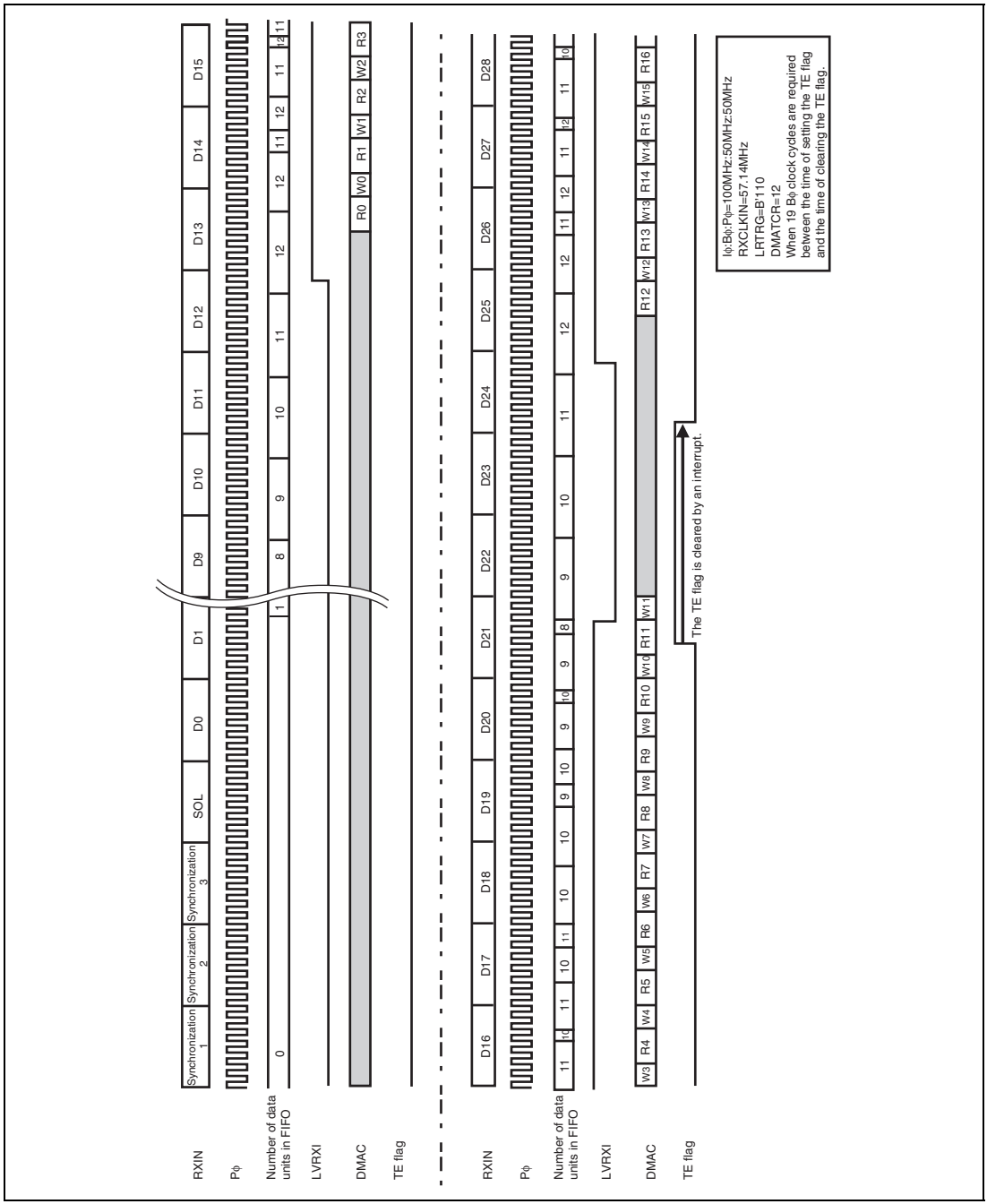


Figure 24.11 LVDS Reception Timing Chart by DMAC

24.6.2 Data Transfer by DTC

Figure 24.12 shows an example of data transfer from the LVDS receive FIFO data register by the DTC with block transfer, and figure 24.13 shows an example of the timing chart.

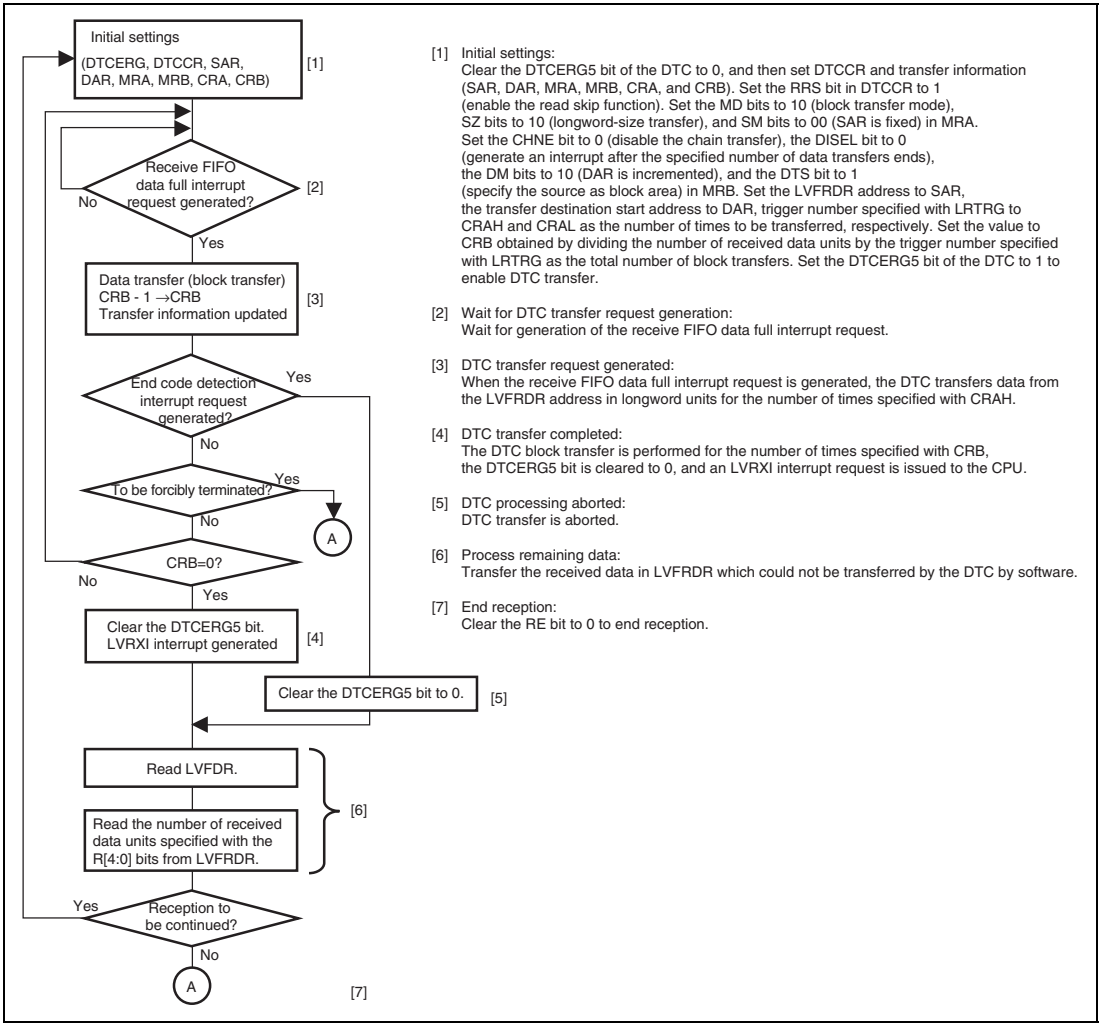


Figure 24.12 Example of Data Transfer from LVDS Receive FIFO Data Register by DTC (1)

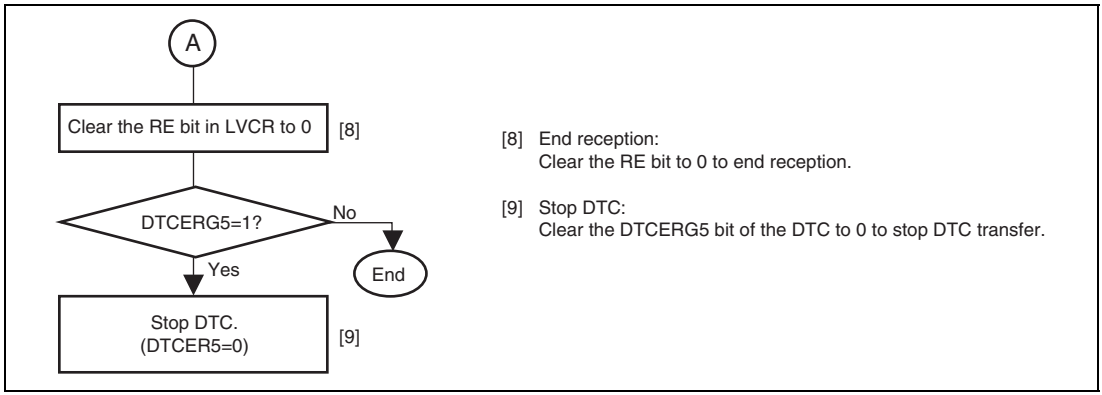


Figure 24.12 Example of Data Transfer from LVDS Receive FIFO Data Register by DTC (2)

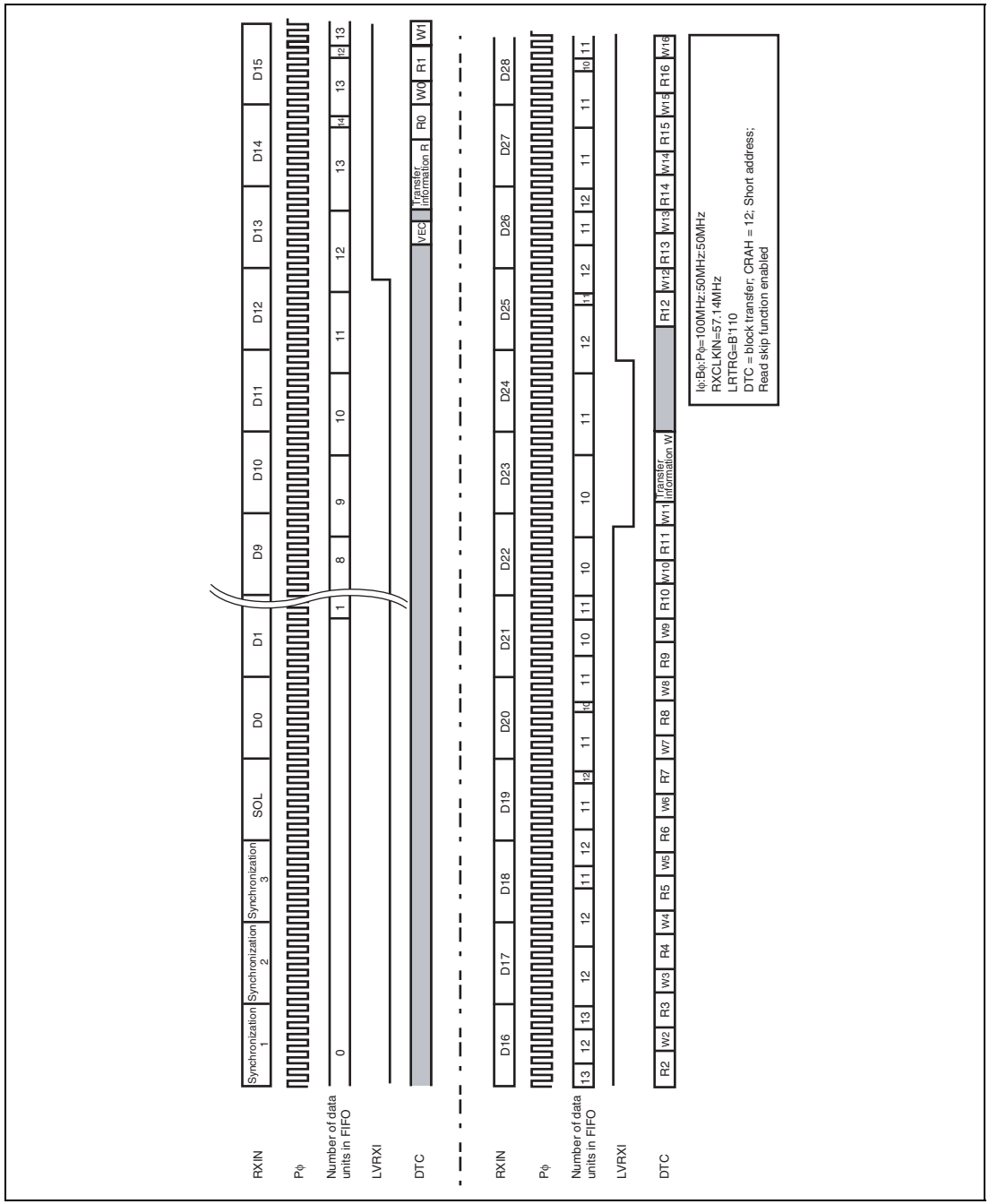


Figure 24.13 LVDS Receive Timing Chart by DTC

24.6.3 Data Transfer Rates

Table 24.5 shows the maximum transfer rates when the DMAC or DTC is used.

Table 24.5 Maximum Transfer Rates

Conditions: Iϕ:Bϕ:Pϕ = 100 MHz:50 MHz: 50 MHz

DMATCR = CRAH = trigger number specified with LRTRG

Clearing the TE flag in DMA requires 19 Bϕ clock cycles

		When DTC is Used [Mbps/ch]	
LRTRG[2:0]	When DMAC is Used [Mbps/ch]	When read Skip Function is Disabled	When Read Skip Function is Enabled
001	55.1	59.3	88.9
010	82.0	86.5	114.3
011	97.9	102.1	126.3
100	108.4	112.3	133.3
101	115.9	119.4	137.9
110	121.5	124.7	141.2
111	88.9	47.1	99.9
000	Continuous reception cannot be provided due to an overrun error.		

Notes: 1. The table shows the maximum transfer rates when the DMAC or DTC is used.

2. If there is an internal bus conflict, the actually available maximum transfer rates are lower than those shown in the table.

3. The transfer rates when the DMAC is used depend on the interrupt (software) process for clearing the TE flag. The table shows the transfer rates when 19 Bϕ cycles are required to clear the TE flag.

   Here, 19 Bϕ cycle is defined assuming the following conditions.

   Clock ratio of Iϕ:Bϕ:Pϕ = 2:1:1

   The DMAC transfer end interrupt routine includes the sequence below.

   MOVI20 #H'E100E,R0

   MOV.W @R0,R1 □read 1 from the TE flag bit.

   MOVI20 #H'FFFD,R4

   AND R4,R1

   MOV.W R1,@R0; clear the TE flag bit to 0.

   The DMAC transfer end interrupt immediately clears the TE flag.

   In the following cases, more cycles are required to clear the TE flag and thus the maximum transfer rates shown in the table are not available.

The interrupt wait time is not zero.

Saving registers is required because the register bank is not used.

A register bank overflows.

A different interrupt occurs.

The DMAC or DTC is activated by a different source.

4. The DTC read skip function is enabled only when the LVDS is the preceding DTC activation source. When the DTC is activated for the first time after power-on, the read skip function is always disabled.
5. When the DMAC or DTC is not used, the maximum transfer rates depend on the software processing performance.
6. The available LVDS transfer rates range from 40 Mbps (minimum) to the values shown in the table (maximum).
7. If data is input to the LVDS at the rate exceeding the maximum transfer rate, an overrun error interrupt (LVOREI) may be caused.
8. When LRTRG[2:0] bits are set to B'111, the transfer rate is lowered to prevent an overrun error.

24.6.4 Overrun Error Generation Timing

Figure 24.14 shows the overrun error generation timing. An overrun error is caused if an attempt is made to write the next received data when the LVDS receiving unit recognizes the FIFO to contain the specified number of data units. LVFRDR retains the received data having been written before the overrun error occurs thus discarding the subsequent data. Reception is disabled while the ORER bit is 1.

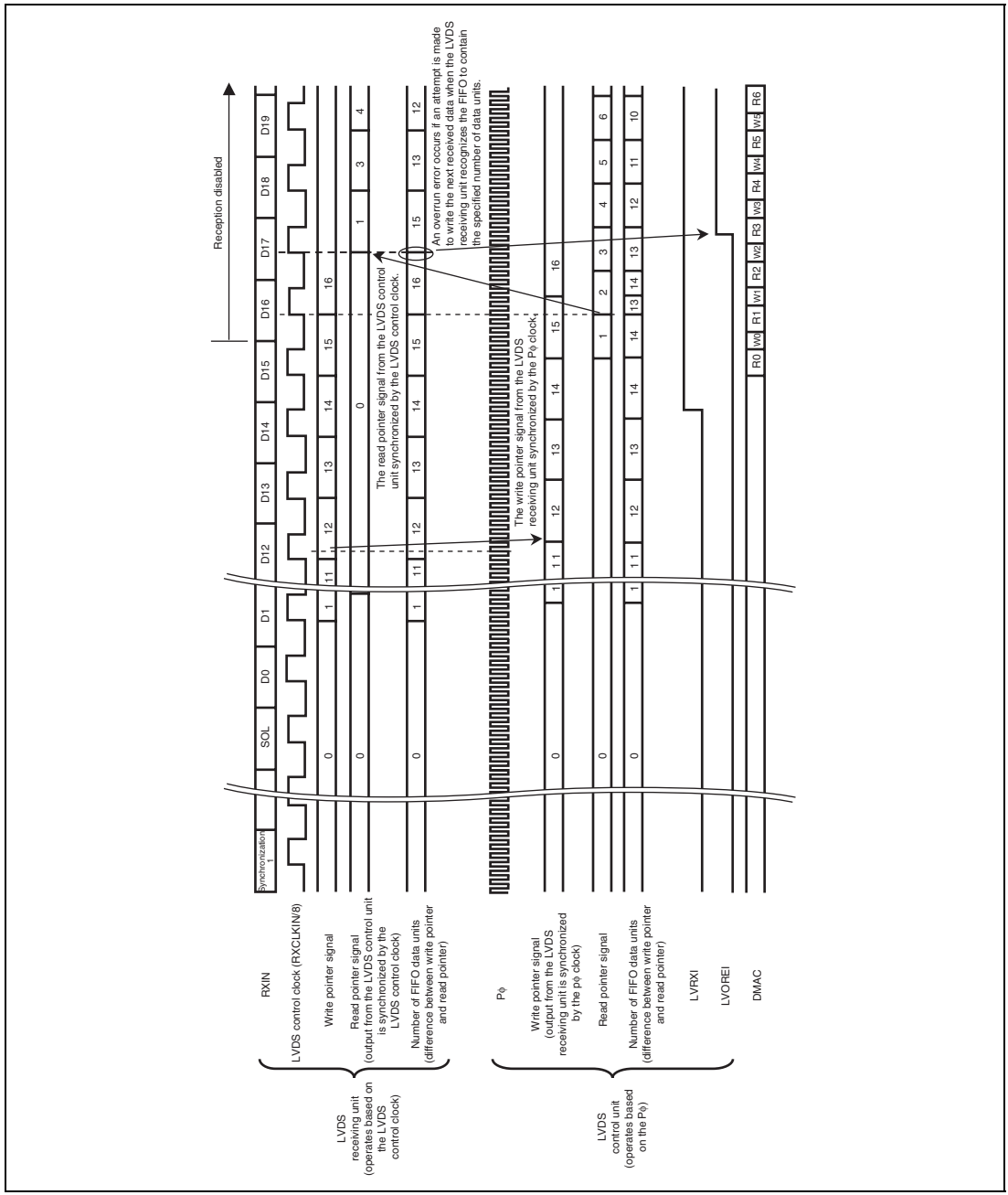


Figure 24.14 Overrun Error Generation Timing (when DMAC is used, Iϕ = 100 MHz, Bϕ = Pϕ = 50 MHz, RXCLKIN = 57.14 MHz, LRTRG = B'111, and DMATCR = 14)

24.7 Usage Notes

24.7.1 Setting Module Standby Mode

The LVDS module operation can be enabled or disabled using the standby control register. With the initial value, the LVDS is halted. Register access is enabled by canceling module standby mode. For details, refer to section 32, Power-Down Modes.

After canceling module standby mode, secure the LVDS reception start settling time (t_{RCV}) before starting synchronization code reception. Otherwise, correct reception is not available. For details, refer to figure 35.56 in section 35, Electrical Characteristics.

24.7.2 Relationship between LVDS Input Clock Signals and Peripheral Clock Signal (Pϕ)

The clock input cycle time (t_{CLKIN}) of the clock signals input to the LVDS (RXCLKINP and RXCLKINM) and the period of the peripheral clock signal (t_{PCYC}) should be set so that the following condition be satisfied:

$$t_{CLKIN} > (5/8) \times t_{PCYC}.$$

24.7.3 One-Channel Operation

1. Set the CHSEL bit to select the number of channels to be used while reception is disabled (RE bit is 0).
2. Even when only one channel is used, neither the PL4/RXIN1P or PL5/RXIN1M pins cannot be used for general input.

24.7.4 Stopping LVDS Clock Input

When the clock input to the LVDS (RXCLKINP and RXCLKINM) is to be stopped, input the clock signals for the time equivalent to the specified number of frames shown in table 24.6 after reception ends before actually stopping the clock.

Table 24.6 Number of Frames Necessary before Clock Input to the LVDS is Actually Stopped

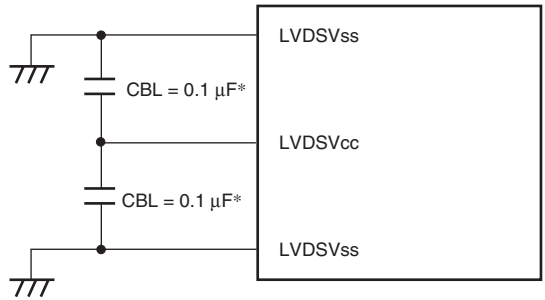
Frequency Ratio	Number of Frames Necessary before Clock Input to the LVDS is Actually Stopped
$P\phi \geq (t_{CLKIN}/8)$	10
$P\phi < (t_{CLKIN}/8)$	$10 + (t_{CLKIN}/P\phi/8)$

24.7.5 Internal Differential Input Impedance Resistors and Internal Pull-Up Resistors

Canceling module standby mode allows the internal pull-up resistors (RP) to be connected to the internal differential input impedance resistors (ZL) and LVDS input pins (RXCLKINP, RXCLKINM, RXIN0P, RXIN0M, RXIN1P, and RXIN1M). For resistor values, refer to table 35.3, DC Characteristics (3) [LVDS related pins (SH72315A only)] in section 35, Electrical Characteristics.

24.7.6 Notes on Board Design

Figure 24.15 shows the external circuitry recommended for stabilizing the LVDS operation. Separate LVDSVcc and LVDSVss from the board power supply source, and be sure to insert laminated ceramic capacitors close to the pins.



Note: * CBL is a laminated ceramic capacitor. (Recommended values are shown.)

Figure 24.15 Recommended External Circuitry for LVDS Circuit

Section 25 Renesas Serial Peripheral Interface (RSPI)

This LSI includes a channel of Renesas Serial Peripheral Interface (RSPI).

The RSPI is capable of full-duplex synchronous, high-speed serial communications with multiple processors and peripheral devices.

25.1 Features

The RSPI of this LSI has the following features:

1. RSPI Transfer Function
- Uses MOSI (Master Out Slave In), MISO (Maser In Slave Out), SSL (Slave Select), and RSPCK (RSPI Clock) signals to provide SPI mode (four-wire) and clock synchronous mode (three-wire) serial communications.
- Capable of master-slave mode serial communication.
- Capable of mode fault error detection.
- Capable of overrun error detection.
- Modifiable serial transfer clock polarity.
- Modifiable serial transfer clock phase.

2. Data Format
- Switchable MSB first/LSB first.
- Transfer bit length changeable to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, and 32 bits.
- Transmission/receive buffers of 128 bits
- Up to 4 frames (up to 32 bits per frame) can be transferred at a time in transmission or reception.

3. Bit Rate
- In master mode:
  An internal baud rate generator generates RSPCK by dividing Pϕ by up to 4096.
- In slave mode:
  The serial clock signal is generated with division by up to 8.
  An external input clock is used as the serial clock.

4. Buffer Configuration

- Transmission/receive buffers are provided in a double-buffer configuration.

5. SSL Control Function

- Provided with four SSL signals (SSL0 to SSL3).
- In single-master mode, SSL0 to SSL3 signals are for output.
- In multi-master mode, SSL0 signal is for input, and SSL1 to SSL3 signals are for either output or Hi-Z.
- In slave mode, SSL0 signal is for input, and SSL1 to SSL3 signals are for Hi-Z.
- A delay from SSL output assertion to RSPCK operation (RSPCK delay) can be set.
  Settable range: 1 to 8 RSPCK cycles
  Unit: 1 RSPCK cycle
- A delay from RSPCK stop to SSL output negation (SSL negation delay) can be set.
  Settable range: 1 to 8 RSPCK cycles
  Unit: 1 RSPCK cycle
- Wait for next-access SSL output assertion (next-access delay) can be set.
  Settable range: 1 to 8 RSPCK cycles
  Unit: 1 RSPCK cycle
- Switchable SSL polarity.

6. Master Mode Transfer Control Method

- A transfer comprised of a maximum of four commands can be executed in sequential loops.
- Each command can include:
  SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, LSB/MSB first, burst, RSPCK delay, SSL negation delay, and next-access delay.
- A transfer can be started upon writing to the transmit buffer by the DMAC.
- A transfer can be started upon writing to the transmit buffer by the DTC.
- A transfer can be started upon clearing the SPTEF bit by the CPU.
- MOSI signal values can be set during SSL negation.

7. Interrupt Sources

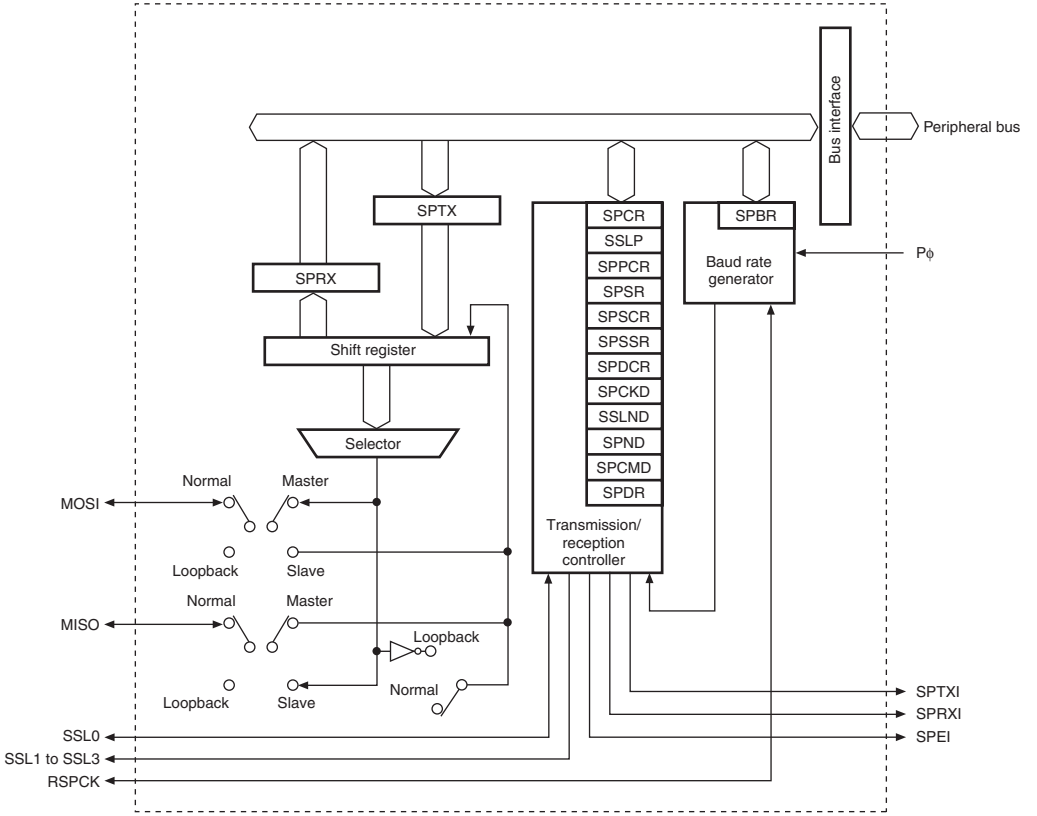
- Maskable interrupt sources are provided.
  - RSPI receive interrupt (receive buffer full)
  - RSPI transmit interrupt (transmit buffer empty)
  - RSPI error interrupt (mode fault and overrun)

8. Other Features

- Loopback mode is provided.
- The CMOS/open drain output switchover function is provided.
- The RSPI disable (initialization) function is provided.

25.1.1 Block diagram

Figure 25.1 shows an RSPI block diagram for one channel. When the CPU accesses the RSPI control registers, a peripheral bus (P-bus) is used.



[Legend]

SPCR:	RSPI control register	SPND:	RSPI next-access delay register
SSLP:	RSPI slave select polarity register	SPCMD:	RSPI command registers 0 to 3
SPPCR:	RSPI pin control register	SPBR:	RSPI bit rate register
SPSR:	RSPI status register	SPTX:	RSPI transmit buffer
SPSCR:	RSPI sequence control register	SPRX:	RSPI receive buffer
SPSSR:	RSPI sequence status register	SPTXI:	RSPI transmit interrupt
SPDCR:	RSPI data control register	SPRXI:	RSPI receive interrupt
SPCKD:	RSPI clock delay register	SPEI:	RSPI error interrupt
SSLND:	RSPI slave select negate delay register	SPDR:	RSPI data register

Figure 25.1 Block Diagram of RSPI (for One Channel)

25.2 Input/Output Pins

The RSPI has the serial pins shown in table 25.1. The RSPI automatically switches input/output directions of the pins. Pin SSL0 is set to output when the RSPI is in single master mode and set to input when the RSPI is in multi master or slave mode. Pins RSPCK0, MOSI0, and MISO0 are set to inputs or outputs according to the master/slave setting and input level of SSL0 (see section 25.4.2, Controlling RSPI Pins).

Table 25.1 Pin Configuration

Pin Name	Symbol	I/O	Function
RSPI clock pin	RSPCK0	I/O	RSPI clock input/output
Master transmit data pin	MOSI0	I/O	RSPI master transmit data
Slave transmit data pin	MISO0	I/O	RSPI slave transmit data
Slave select 0 pin	SSL0	I/O	RSPI slave select
Slave select 1 pin	SSL1	Output	RSPI slave select
Slave select 2 pin	SSL2	Output	RSPI slave select
Slave select 3 pin	SSL3	Output	RSPI slave select

Note: The 0 to indicate the individual interface is omitted from the pin names, which are henceforth given as RSPCK, MOSI, and MISO.

25.3 Register Descriptions

The RSPI has the registers shown in table 25.2. These registers enable the RSPI to perform the following controls: specifying master/slave modes, specifying a transfer format, and controlling the transmitter and receiver. See section 34, List of Registers, for the states of these registers in the various states of processing.

Table 25.2 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
RSPI control register	SPCR	R/W	H'00	H'FFFFB800	8, 16
RSPI slave select polarity register	SSLP	R/W	H'00	H'FFFFB801	8
RSPI pin control register	SPPCR	R/W	H'00	H'FFFFB802	8, 16
RSPI status register	SPSR	R/W	H'22	H'FFFFB803	8
RSPI data register	SPDR	R/W	H'00000000	H'FFFFB804	16, 32*
RSPI sequence control register	SPSCR	R/W	H'00	H'FFFFB808	8, 16
RSPI sequence status register	SPSSR	R	H'00	H'FFFFB809	8
RSPI bit rate register	SPBR	R/W	H'FF	H'FFFFB80A	8, 16
RSPI data control register	SPDCR	R/W	H'00	H'FFFFB80B	8
RSPI clock delay register	SPCKD	R/W	H'00	H'FFFFB80C	8, 16
RSPI slave select negation delay register	SSLND	R/W	H'00	H'FFFFB80D	8
RSPI next-access delay register	SPND	R/W	H'00	H'FFFFB80E	8
RSPI command register 0	SPCMD0	R/W	H'070D	H'FFFFB810	16
RSPI command register 1	SPCMD1	R/W	H'070D	H'FFFFB812	16
RSPI command register 2	SPCMD2	R/W	H'070D	H"FFFFB814	16
RSPI command register 3	SPCMD3	R/W	H'070D	H'FFFFB816	16

Notes: * Access is in the unit selected by the SPLW bit in register SPDCR.

25.3.1 RSPI Control Register (SPCR)

SPCR sets the operating mode of the RSPI. SPCR can be read from or written to by the CPU. If the MSTR and MODFEN bits are changed while the RSPI function is enabled by setting the SPE bit to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	–	SPMS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SPRIE	0	R/W	RSPI Receive Interrupt Enable If the RSPI has detected a receive buffer write after completion of a serial transfer and the SPRF bit in the RSPI status register (SPSR) is set to 1, this bit enables or disables the generation of an RSPI receive interrupt request. 0: Disables the generation of RSPI receive interrupt requests. 1: Enables the generation of RSPI receive interrupt requests.
6	SPE	0	R/W	RSPI Function Enable Setting this bit to 1 enables the RSPI function. When the MODF bit in the RSPI status register (SPSR) is 1, the SPE bit cannot be set to 1 (see section 25.4.7, Error Detection). Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function (see section 25.4.8, Initializing RSPI). 0: Disables the RSPI function 1: Enables the RSPI function

Bit	Bit Name	Initial Value	R/W	Description
5	SPTIE	0	R/W	RSPI Transmit Interrupt Enable Enables or disables the generation of RSPI transmit interrupt requests when the RSPI detects transmit buffer empty and sets the SPTEF bit in the RSPI status register (SPSR) to 1. In the RSPI disabled (with the SPE bit 0) status, the SPTEF bit is 1. Therefore, note that setting the SPTIE bit to 1 when the RSPI is in the disabled status generates an RSPI transmit interrupt request. 0: Disables the generation of RSPI transmit interrupt requests. 1: Enables the generation of RSPI transmit interrupt requests.
4	SPEIE	0	R/W	RSPI Error Interrupt Enable Enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the MODF bit in the RSPI status register (SPSR) to 1, or when the RSPI detects and sets the OVRF bit in SPSR to 1 (see section 25.4.7, Error Detection). 0: Disables the generation of RSPI error interrupt requests. 1: Enables the generation of RSPI error interrupt requests.
3	MSTR	0	R/W	RSPI Master/Slave Mode Select Selects master/slave mode of RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCK, MOSI, MISO, and SSL0 to SSL3. 0: Slave mode 1: Master mode
2	MODFEN	0	R/W	Mode Fault Error Detection Enable Enables or disables the detection of mode fault error (see section 25.4.7, Error Detection). In addition, the RSPI determines the input/output directions of the SSL0 pin based on combinations of the MODFEN and MSTR bits (see section 25.4.2, Controlling RSPI Pins). 0: Disables the detection of mode fault error 1: Enables the detection of mode fault error

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SPMS	0	R/W	RSPI Mode Select Selects SPI (4-wire) or clock synchronous (3-wire) mode. In clock synchronous mode, the SSL pin is not used and the RSPCK, MOSI, and MISO pins are used for communication. To enable clock synchronous mode, set the CPHA bit in the RSPI command register (SPCMD) to 1. If CPHA is set to 0, operation cannot be guaranteed. 0: SPI mode (4-wire) 1: Clock synchronous mode

25.3.2 RSPI Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL0 to SSL3 signals of the RSPI. SSLP can always be read from or written to by the CPU. If the contents of SSLP are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	SSL3P	SSL2P	SSL1P	SSL0P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SSL3P	0	R/W	SSL Signal Polarity Setting These bits set the polarity of the SSL signals. SSLiP (where i is 3 to 0) indicates the active polarity of the SSLi signal. 0: SSLi signal set to active-0 1: SSLi signal set to active-1
2	SSL2P	0	R/W	
1	SSL1P	0	R/W	
0	SSL0P	0	R/W	

25.3.3 RSPI Pin Control Register (SPPCR)

SPPCR sets the modes of the RSPI pins. SPPCR can be read from or written to by the CPU. If the contents of this register are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	MOIFE	MOIFV	–	SPOM	–	SPLP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5	MOIFE	0	R/W	MOSI Idle Value Fixing Enable Fixes the MOSI output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When MOIFE is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period. When MOIFE is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSI bit. 0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit
4	MOIFV	0	R/W	MOSI Idle Fixed Value If the MOIFE bit is 1 in master mode, the RSPI, according to MOIFV bit settings, determines the MOSI signal value during the SSL negation period (including the SSL retention period during a burst transfer). 0: MOSI Idle fixed value equals 0 1: MOSI Idle fixed value equals 1
3	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
2	SPOM	0	R/W	RSPI Output Pin Mode Sets the RSPI output pins to CMOS output/open drain output. 0: CMOS output 1: Open-drain output
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SPLP	0	R/W	RSPI Loopback When the SPLP bit is set to 1, the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects (reverses) the input path and the output path for the shift register (loopback mode). 0: Normal mode 1: Loopback mode

25.3.4 RSPI Status Register (SPSR)

SPSR indicates the operating status of the RSPI. SPSR can be read by the CPU. Writing 1 to the SPRF, SPTEF, MODF, and OVRF bits cannot be performed by the CPU. These bits can be cleared to 0 after they are read as 1.

Bit:	7	6	5	4	3	2	1	0
	SPRF	–	SPTEF	–	–	MODF	MIDLE	OVRF
Initial value:	0	0	1	0	0	0	1	0
R/W:	R/(W)*	R	R/(W)*	R	R	R/(W)*	R	R/(W)*

Note: * Only 0 can be written to this bit after reading it as 1 to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	SPRF	0	R/(W)*	RSPI Receive Buffer Full Flag Indicates the status of the receive buffer for the RSPI data register (SPDR). Upon completion of a serial transfer with the SPRF bit 0, the RSPI transfers the receive data from the shift register to SPDR, and sets this bit to 1. This also means that the last bit of transmit data has been sent because the RSPI performs full-duplex synchronous serial communication. If a serial transfer ends while the SPRF bit is 1, the RSPI does not transfer the received data from the shift register to SPDR. When the OVRF bit in SPSR is 1, the SPRF bit cannot be changed from 0 to 1 (see section 25.4.7, Error Detection). 0: No valid data in SPDR [Clearing conditions] • When 0 is written in SPRF after reading SPRF = 1. • When the DMAC is activated with an RXI interrupt and the DMAC reads data from SPDR as many as the number of states specified in SPFC. • When the DTC is activated with an RXI interrupt and the DTC reads data from SPDR as many as the number of states specified in SPFC (except when the transfer counter value of the DTC becomes H'0000 and the DISEL bit is 1). • Power-on reset 1: Valid data found in SPDR [Setting condition] • When serial reception of data as many as the number of states specified in SPFC is normally completed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SPTEF	1	R/(W)*	RSPI Transmit Buffer Empty Flag Indicates the status of the transmit buffer for the RSPI data register (SPDR). When the SPTEF bit is cleared and the shift register is empty, the data is copied from the transmit buffer to the shift register. The CPU, DMAC and DTC can write to SPDR only when the SPTEF bit is 1. If the CPU, the DMAC or the DTC writes to the transmit buffer of SPDR when the SPTEF bit is 0, the data in the transmit buffer is not updated. 0: Data found in the transmit buffer [Clearing conditions] • When 0 is written in SPTEF after reading SPTEF = 1. • When the DMAC is activated with a TXI interrupt and the DMAC writes data to SPDR as many as the number of states specified in SPFC. • When the DTC is activated with a TXI interrupt and the DTC writes data to SPDR as many as the number of states specified in SPFC (except when the transfer counter value of the DTC becomes H'0000 and the DISEL bit is 1). 1: No data in the transmit buffer [Setting conditions] • Power-on reset • When serial reception of data as many as the number of states specified in SPFC is normally completed.
4, 3	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MODF	0	R/(W)*	Mode Fault Error Flag Indicates the occurrence of a mode fault error. The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP). 0: No mode fault error occurs [Clearing conditions] • Power-on reset • When 0 is written in MODF after reading MODF = 1. 1: A mode fault error occurs [Setting conditions] • When the input of SSL0 is set to the active level in multi-master mode. • When the SSL0 pin is negated before the RSPCK cycle necessary for data transfer ends in slave mode
1	MIDLE	1	R	RSPI Idle Flag Indicates the status of RSPI transfer. 1: RSPI is in the idle state. [Setting conditions] In master mode: • The SPE bit in SPCR is 0 (RSPI initialization) • The SPTEF bit in SPSR is 1, the SPSSR bits in SPCP are 00, and the RSPI internal sequencer becomes idle. In slave mode: • The SPE bit in SPCR is 0. 0: RSPI transfers the data. [Clearing condition] When the setting condition is not satisfied.

Bit	Bit Name	Initial Value	R/W	Description
0	OVRF	0	R/(W)*	Overrun Error Flag Indicates the occurrence of an overrun error. 0: No overrun error occurs [Clearing conditions] • Power-on reset • When 0 is written in OVRF after reading OVRF = 1. 1: An overrun error occurs [Setting condition] • When serial transfer is ended while the SPRF bit is set to 1.

Note: * Only 0 can be written to this bit after reading it as 1 to clear the flag.

25.3.5 RSPI Data Register (SPDR)

SPDR is a buffer that stores RSPI transmit/receive data. The transmit buffer (SPTX) and receive buffer (SPRX) are allocated for SPDR and these buffers are independent of each other.

Data should be read from or written to SPDR in word or longword units according to the setting of the RSPI longword/word access setting bit (SPLW) in the RSPI data control register (SPDCR). When the SPLW bit is 0, SPDR is a 64-bit buffer consisting of 4 frames, each of which includes up to 16 bits. When the SPLW bit is 1, SPDR is a 128-bit buffer consisting of 4 frames, each of which includes up to 32 bits.

This register acts as the interface with the FIFO buffer. To read four frames of data, reading SPDR four times will lead to the data being read out in the order of reception. To transmit four frames of data, write to SPDR four times.

The frame length that SPDR uses is determined by the frame count setting bits (SPFC1 and SPFC0) in the RSPI data control register (SPDCR). The bit length to be used is determined by the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD).

If the CPU, DTC, or DMAC requests writing to SPDR when the SPTEF bit in the RSPI status register (SPSR) is 1, the RSPI writes data to the transmit buffer of SPDR. If the SPTEF bit is 0, the RSPI does not update the transmit buffer of SPDR.

When the CPU, DTC, or DMAC requests reading from SPDR, data is read from the receive buffer if the RSPI receive/transmit data select bit (SPRDTD) in the RSPI pin control register (SPPCR) is 0, or data is read from the transmit buffer if the SPRDTD bit is 1.

When reading data from the transmit buffer, the most recently written value is read. If the SPTEF bit in the RSPI status register (SPSR) is 0, no data is read from the transmit buffer.

In the normal operating method, the CPU, DTC, and DMAC read the receive buffer when the SPRF bit in SPSR is 1 (a condition in which unread data is stored in the receive buffer). When the SPRF or OVRF bit in SPSR is 1, the RSPI does not update the receive buffer of SPDR at the end of a serial transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

25.3.6 RSPI Sequence Control Register (SPSCR)

SPSCR sets the sequence control method when the RSPI operates in master mode. SPSCR can be read from or written to by the CPU. If the contents of SPSCR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function enabled, the subsequent operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	SPSLN[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SPSLN[2:0]	000	R/W	RSPI Sequence Length Setting These bits set a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes RSPI command registers 0 to 3 (SPCMD0 to SPCMD3) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN2 to SPSLN0 bits. When the RSPI is in slave mode, SPCMD0 is always referenced. The relationship among the setting in these bits, sequence length, and referenced SPCMD register number is shown below. SPSLN [1:0] / Sequence Length / Referenced SPCMD # 000: 1 0 → 0 → … 001: 2 0 → 1 → 0 → … 010: 3 0 → 1 → 2 → 0 → … 011: 4 0 → 1 → 2 → 3 → 0 → … 1xx: Setting prohibited

25.3.7 RSPI Sequence Status Register (SPSSR)

SPSSR indicates the sequence control status when the RSPI operates in master mode. SPSSR can be read by the CPU. Any writing to SPSSR by the CPU is ignored.

Bit:	7	6	5	4	3	2	1	0
	–	–	SPECM[1:0]		–	–	SPCP[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5, 4	SPECM[1:0]	00	R	RSPI Error Command These bits indicate RSPI command registers 0 to 3 (SPCMD0 to SPCMD3) that are pointed to by command pointers (SPCP1 and SPCP0 bits) when an error is detected during sequence control by the RSPI. The RSPI updates the bits SPECM1 and SPECM0 only when an error is detected. If both the OVRF and MODF bits in the RSPI status register (SPSR) are 0 and there is no error, the values of the bits SPECM1 and SPECM0 have no meaning. For the RSPI's error detection function, see section 25.4.7, Error Detection. For the RSPI's sequence control, see section 25.4.9 (2), Master Mode Operation. 00: SPCMD0 01: SPCMD1 10: SPCMD2 11: SPCMD3
3, 2	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	SPCP[1:0]	000	R	RSPI Command Pointer During RSPI sequence control, these bits indicate RSPI command registers 0 to 3 (SPCMD0 to SPCMD3), which are currently pointed to by the pointers. For the RSPI's sequence control, see 25.4.9 (2), Master Mode Operation. 00: SPCMD0 01: SPCMD1 10: SPCMD2 11: SPCMD3

25.3.8 RSPI Bit Rate Register (SPBR)

SPBR sets the bit rate in master mode. SPBR can be read from or written to by the CPU. If the contents of SPBR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed. When the RSPI is used in slave mode, the bit rate depends on the input clock regardless of the settings of SPBR and BRDV.

Bit:	7	6	5	4	3	2	1	0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bit rate is determined by combinations of SPBR settings and the bit settings in the BRDV1 and BRDV0 bits in the RSPI command registers (SPCMD0 to SPCMD3). The equation for calculating the bit rate is given below. In the equation, N denotes an SPBR setting (0, 1, 2, …, 255), and n denotes bit settings in the bits BRDV1 and BRDV0 (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(P\phi)}{2 \times (N + 1) \times 2^n}$$

Table 25.3 shows examples of the relationship between the SPBR register and BRDV1 and BRDV0 bit settings.

Table 25.3 Relationship between SPBR and BRDV[1:0] Settings

SPBR (N)	BRDV[1:0] (n)	Division Ratio	Bit Rate				
			Pϕ = 16 MHz	Pϕ = 20 MHz	Pϕ = 32 MHz	Pϕ = 40 MHz	Pϕ = 50 MHz
0	0	2	8.0 Mbps	10.0 Mbps	—	—	—
1	0	4	4.0 Mbps	5.0 Mbps	8.0 Mbps	10.0 Mbps	12.5 Mbps
2	0	6	2.67 Mbps	3.3 Mbps	5.33 Mbps	6.67 Mbps	8.33 Mbps
3	0	8	2.0 Mbps	2.5 Mbps	4.0 Mbps	5.0 Mbps	6.25 Mbps
4	0	10	1.6 Mbps	2.0 Mbps	3.2 Mbps	4.0 Mbps	5.00 Mbps
5	0	12	1.33 Mbps	1.67 Mbps	2.67 Mbps	3.33 Mbps	4.17 Mbps
5	1	24	667 kbps	833 kbps	1.33 Mbps	1.67 Mbps	2.08 Mbps
5	2	48	333 kbps	417 kbps	667 kbps	833 kbps	1.04 Mbps
5	3	96	167 kbps	208 kbps	333 kbps	417 kbps	520 kbps
255	3	4096	3.9 kbps	4.9 kbps	7.8 kbps	9.8 kbps	10 kbps

[Legend]

—: Prohibited setting (exceeds the limit on the frequency of toggling of the I/O buffers)

25.3.9 RSPI Data Control Register (SPDCR)

RSPI sets the number of frames that can be stored in the SPDR register, specifies from which buffer of the SPDR register data should be read, and sets the access size, word or longword, for the SPDR register.

Up to 4 frames can be transmitted or received at a time upon transmission or reception activation according to the setting combinations of the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD), RSPI sequence length setting bits (SPSLN2 to SPSLN0) in the RSPI sequence control register (SPSCR), and frame count setting bits (SPFC1 and SPFC0) in the RSPI data control register (SPDCR).

SPDCR can be read from or written to by the CPU. If the contents of SPDCR are changed by the CPU while the RSPI function is enabled with the SPE bit in the RSPI control register (SPCR) set to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	SPLW	SPRDTD	–	–	SPFC[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SPLW	0	R/W	RSPI Longword/Word Access Setting Sets the access size for the RSPI data register (SPDR). When SPLW is set to 0, SPDR is accessed in word units. When SPLW is set to 1, SPDR is accessed in longword units. When SPLW is 0, the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD) should be set to 8 to 16 bits. If these bits are set to 20, 24, or 32 bits, operation cannot be guaranteed. 0: Word access to SPDR register 1: Longword access to SPDR register

Bit	Bit Name	Initial Value	R/W	Description
4	SPRDTD	0	R/W	RSPI Receive/Transmit Data Select Selects whether data should be read from the receive buffer or transmit buffer of the RSPI data register (SPDR). When reading from the transmit buffer, most recently written value is read. Reading from the transmit buffer is allowed while the SPTEF bit in the RSPI status register (SPSR) is 1. 0: Read from receive buffer. 1: Read from transmit buffer (only when the SPTEF bit is 1).
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	SPFC[1:0]	00	R/W	Frame Count Setting These bits specify the number of frames that can be stored in the SPDR register. Up to 4 frames can be transmitted or received at a time upon transmission or reception activation according to the setting combinations of the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD), RSPI sequence length setting bits (SPSLN2 to SPSLN0) in the RSPI sequence control register (SPSCR), and frame count setting bits (SPFC1 and SPFC0) in the RSPI data control register (SPDCR). These bits also specify the number of received data to set the RSPI receive buffer full flag in the RSPI status register (SPSR) and the number of remaining data to be transmitted to clear the RSPI transmit buffer empty flag in SPSR. Table 25.4 shows combination examples of the frame formats that can be stored in the SPDR register and the transmission/reception settings. If any setting other than those listed in table 25.4 is made, subsequent operations cannot be guaranteed.

Table 25.4 Combinations of Frame Count Setting Bits

Setting No.	SPB3 to SPB0	SPSLN2 to SPSLN0	SPFC1 and SPFC0	Number of Frames to Transfer	Number of Frames to Set SPRF to 1 or to Clear SPTEF to 0
1-1	N	000	00	1	1 frame
1-2	N	000	01	2	2 frames
1-3	N	000	10	3	3 frames
1-4	N	000	11	4	4 frames
2-1	N, M	001	01	2	2 frames
2-2	N, M	001	11	4	4 frames
3	N, M, O	010	10	3	3 frames
4	N, M, O, P	011	11	4	4 frames

[Legend] N, M, O, P: Data lengths that can be set with SPB3 to SPB0.

Data can be transferred or received at a time upon transmission or reception activation according to the setting combinations, 1-1 to 4, as follows:

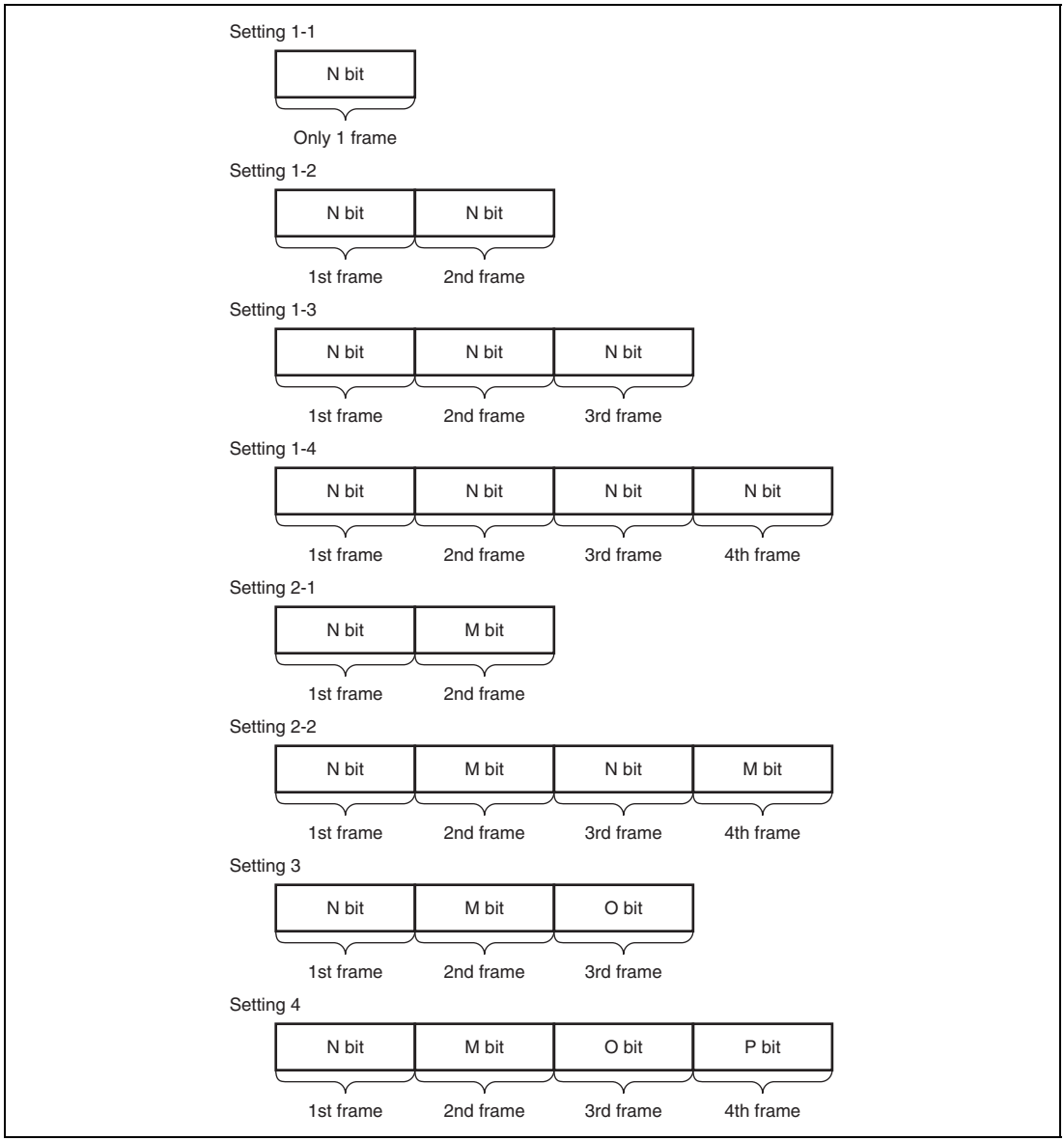
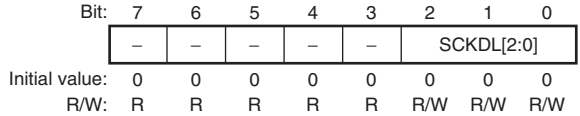


Figure 25.2 Data format for the RSPI

25.3.10 RSPI Clock Delay Register (SPCKD)

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SCKDEN bit in the RSPI command register (SPCMD) is 1. SPCKD can be read from or written to by the CPU. If the contents of SPCKD are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

When using the RSPI in slave mode, set 000 in SCKDL[2:0].



Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SCKDL[2:0]	000	R/W	RSPCK Delay Setting These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 1. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK

25.3.11 SPI Slave Select Negation Delay Register (SSLND)

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL signal during a serial transfer by the RSPI in master mode. SSLND can be read from or written to by the CPU. If the contents of SSLND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

When using the RSPI in slave mode, set 000 in SLNDL[2:0].

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	SLNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SLNDL[2:0]	000	R/W	SSL Negation Delay Setting These bits set an SSL negation delay value when the RSPI is in master mode. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK

25.3.12 RSPI Next-Access Delay Register (SPND)

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPNDEN bit in the RSPI command register (SPCMD) is 1. SPND can be read from or written to by the CPU. If the contents of SPND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

When using the RSPI in slave mode, set 000 in SPNDL[2:0].

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	SPNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SPNDL[2:0]	000	R/W	RSPI Next-Access Delay Setting These bits set a next-access delay when the SPNDEN bit in SPCMD is 1. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK

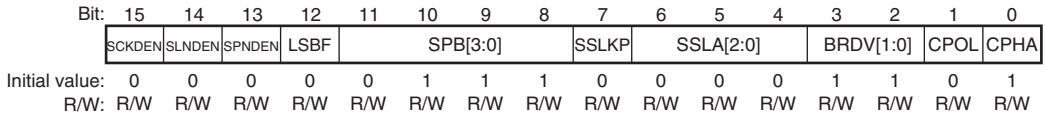
25.3.13 RSPI Command Register (SPCMD)

The RSPI has four RSPI command registers (SPCMD0 to SPCMD3). SPCMD0 to SPCMD3 are used to set a transfer format for the RSPI in master mode. Some of the bits in SPCMD0 are used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMD0 to SPCMD3 according to the settings in bits SPSLN1 and SPSLN0 in the RSPI sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD.

SPCMD can be read from or written to by the CPU.

Set the SPCMD register before setting data to be transferred referencing the SPCMD settings while the SPTEF bit in the RSPI status register (SPSR) is 1.

SPCMD that is referenced by the RSPI in master mode can be checked by means of bits SPCP1 and SPCP0 in the RSPI sequence status register (SPSSR). When the RSPI function in slave mode is enabled, operation cannot be guaranteed if the value set in SPCMD0 is changed by the CPU.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SCKDEN	0	R/W	RSPCK Delay Setting Enable Sets the period from the time the RSPI in master mode sets the SSL signal active until the RSPI oscillates RSPCK (RSPCK 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 RSPCK delay register (SPCKD) settings. To use the RSPI in slave mode, the SCKDEN bit should be set to 0. 0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay equal to SPCKD settings.

Bit	Bit Name	Initial Value	R/W	Description
14	SLNDEN	0	R/W	SSL Negation Delay Setting Enable Sets the period (SSL negation delay) from the time the master mode RSPI stops RSPCK oscillation until the RSPI sets the SSL signal inactive. 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 slave select negation delay register (SSLND) settings. To use the RSPI in slave mode, the SLNDEN bit should be set to 0. 0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay equal to SSLND settings.
13	SPNDEN	0	R/W	RSPI Next-Access Delay Enable Sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSL signal inactive until the RSPI enables the SSL 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. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with RSPI next-access delay register (SPND) settings. To use the RSPI in slave mode, the SPNDEN bit should be set to 0. 0: A next-access delay of 1 RSPCK 1: A next-access delay equal to SPND settings.
12	LSBF	0	R/W	RSPI LSB First Sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first. 0: MSB first 1: LSB first

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	SPB[3:0]	0111	R/W	SRPI Data Length Setting These bits set a transfer data length for the RSPI in master mode or slave mode. 0100 to 0111: 8 bits 1000: 9 bits 1001: 10 bits 1010: 11 bits 1011: 12 bits 1100: 13 bits 1101: 14 bits 1110: 15 bits 1111: 16 bits 0000: 20 bits 0001: 24 bits 0010 and 0011: 32 bits
7	SSLKP	0	R/W	SSL Signal Level Keeping When the RSPI in master mode performs a serial transfer, this bit specifies whether the SSL 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. To use the RSPI in slave mode, the SSLKP bit should be set to 0. 0: Negates all SSL signals upon completion of transfer. 1: Keeps the SSL signal level from the end of the transfer until the beginning of the next access.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	SSLA[2:0]	000	R/W	SSL Signal Assertion Setting These bits control the SSL signal assertion when the RSPI performs serial transfers in master mode. Setting these bits controls the assertion for the signals SSL3 to SSL0. When an SSL signal is asserted, its polarity is determined by the set value in the corresponding SSLP (RSPI slave select polarity register). When the SSLA2 to SSLA0 bits are set to 000 or 1** in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as SSL0 acts as input). When the SSLA2 to SSLA0 bits are set to 1** in single-master mode, serial transfers are performed with all the SSL signals in the negated state as well. When using the RSPI in slave mode, set 000 in SSLA2 to SSLA0. 000: SSL0 001: SSL1 010: SSL2 011: SSL3 1xx: —

Bit	Bit Name	Initial Value	R/W	Description
3, 2	BRDV[1:0]	11	R/W	Bit Rate Division Setting These bits are used to determine the bit rate. A bit rate is determined by combinations of bits BRDV1 and BRDV 0 and the settings in the RSPI bit rate register (SPBR). The settings in SPBR determine the base bit rate. The settings in bits BRDV1 and BRDV0 are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. For SPCMD0 to SPCMD3, different BRDV1 and BRDV0 settings can be specified. This permits the execution of serial transfers at a different bit rate for each command. 00: Select the base bit rate 01: Select the base bit rate divided by 2 10: Select the base bit rate divided by 4 11: Select the base bit rate divided by 8
1	CPOL	0	R/W	RSPCK Polarity Setting Sets the RSPCK polarity of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules. 0: RSPCK = 0 when idle 1: RSPCK = 1 when idle
0	CPHA	1	R/W	RSPCK Phase Setting Sets the RSPCK phase of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules. 0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge

25.4 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.

25.4.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave (SPI), single-master (SPI), and multi-master (SPI), slave (clock synchronous), and master (clock synchronous) modes. A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in the RSPI control register (SPCR). Table 25.5 gives the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 25.5 Relationship between RSPI Modes and SPCR and Description of Each Mode

Item	Slave (SPI)	Single-Master (SPI)	Multi-Master (SPI)	Slave (Clock Synchronous)	Master (Clock Synchronous)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0, 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCK signal	Input	Output	Output/Hi-Z	Input	Output/Hi-Z
MOSI signal	Input	Output	Output/Hi-Z	Input	Output/Hi-Z
MISO signal	Output/Hi-Z	Input	Input	Output/Hi-Z	Input
SSL0 signal	Input	Output	Input	Hi-Z	Hi-Z
SSL1 to SSL3 signals	Hi-Z	Output	Output/Hi-Z	Hi-Z	Hi-Z
Output pin mode	CMOS/ open-drain	CMOS/ open-drain	CMOS/ open-drain	CMOS/ open-drain	CMOS/ open-drain
SSL polarity modification function	Supported	Supported	Supported	—	—
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator

Item	Slave (SPI)	Single-Master (SPI)	Multi-Master (SPI)	Slave (Clock Synchronous)	Master (Clock Synchronous)
Clock polarity	Two	Two	Two	Two	Two
Clock phase	Two	Two	Two	One (CPHA = 1)	One (CPHA = 1)
First transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transfer data length	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer starting method	SSL input active or RSPCK oscillation	Writing to transmit buffer when SPTEF = 1	Writing to transmit buffer when SPTEF = 1	RSPCK oscillation	Writing to transmit buffer when SPTEF = 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported	Supported	Supported	Supported	Supported
Receive buffer full detection	Supported	Supported	Supported	Supported	Supported
Overrun error detection	Supported	Supported	Supported	Supported	Supported
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

25.4.2 Controlling RSPI Pins

According to the MSTR, MODFEN and SPMS bits in the RSPI control register (SPCR) and the SPOM bit in the RSPI pin control register (SPPCR), the RSPI can automatically switch pin directions and output modes. Table 25.6 shows the relationship between pin states and bit settings.

Table 25.6 Relationship between Pin States and Bit Settings

		Pin State*1	
Mode	**Pin**	**SPOM = 0**	**SPOM = 1**
Single-master mode (SPI) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3	CMOS output	Open-drain output
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Multi-master mode (SPI) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCK*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL3*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO	Input	Input
Slave mode (SPI) (MSTR = 0, SPMS = 0)	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL3	Hi-Z	Hi-Z
	MOSI	Input	Input
	MISO*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master (clock synchronous) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3*4	Hi-Z	Hi-Z
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input

Mode	Pin	Pin State*1	
		SPOM = 0	SPOM = 1
Slave (clock synchronous) (MSTR = 0, SPMS = 1)	RSPCK	Input	Input
	SSL0 to SSL3*4	Hi-Z	Hi-Z
	MOSI	Input	Input
	MISO	CMOS output	Open-drain output

Notes: 1. RSPI settings are not reflected to the multi-function pins for which the RSPI function is not applied.
2. When SSL0 is at the active level, the pin state is Hi-Z.
3. When SSL0 is at the active level or the SPE bit in SPCR is 0, the pin state is Hi-Z.
4. SSL0 to SSL3 can be used as the IO ports in clock synchronous mode.

The RSPI in single-master (SPI) and multi-master (SPI) modes determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to the settings of the MOIFE and MOIFV bits in SPPCR as shown in table 25.7.

Table 25.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE	MOIFV	MOSI Signal Value during SSL Negation Period*
0	0, 1	Final data from previous transfer
1	0	Always 0
1	1	Always 1

Note: * The SSL negation period includes the SSL retention period during a burst transfer.

25.4.3 RSPI System Configuration Example

(1) Single Master/Single Slave (with This LSI Acting as Master)

Figure 25.3 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSL0 to SSL3 outputs of this LSI (master) are not used. The SSL input of the RSPI slave is fixed to 0, and the RSPI slave is always maintained in a select state. In the transfer format corresponding to the case where the CPHA bit in the RSPI control register (SPCR) 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 SSL output of this LSI should be connected to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI signals. The RSPI slave always drives the MISO signal.

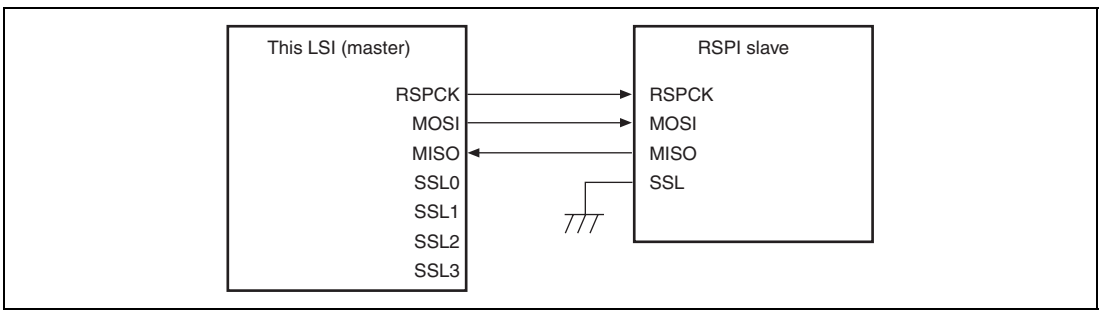


Figure 25.3 Single-Master/Single-Slave Configuration Example (This LSI = Master)

(2) Single Master/Single Slave (with This LSI Acting as Slave)

Figure 25.4 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSL0 pin is used as SSL input. The RSPI master always drives the RSPCK and MOSI signals. This LSI (slave) always drives the MISO signal*.

In the single-slave configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, the SSL0 input of this LSI (slave) is fixed to 0, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (figure 25.5).

Note: * When SSL0 is at the active level, the pin state becomes Hi-Z.

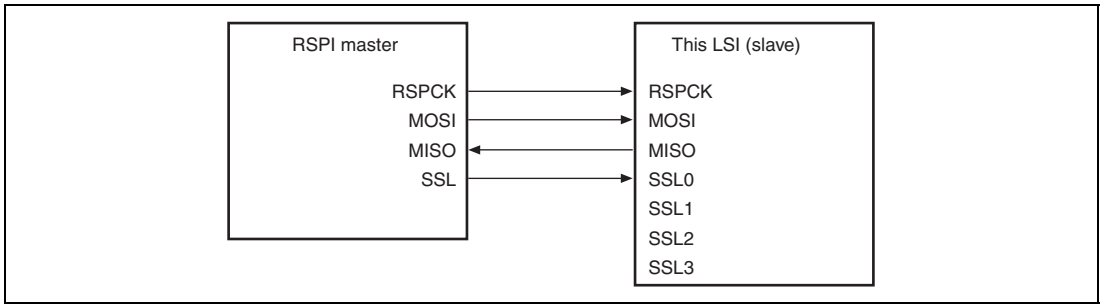


Figure 25.4 Single-Master/Single-Slave Configuration Example (This LSI = Slave)

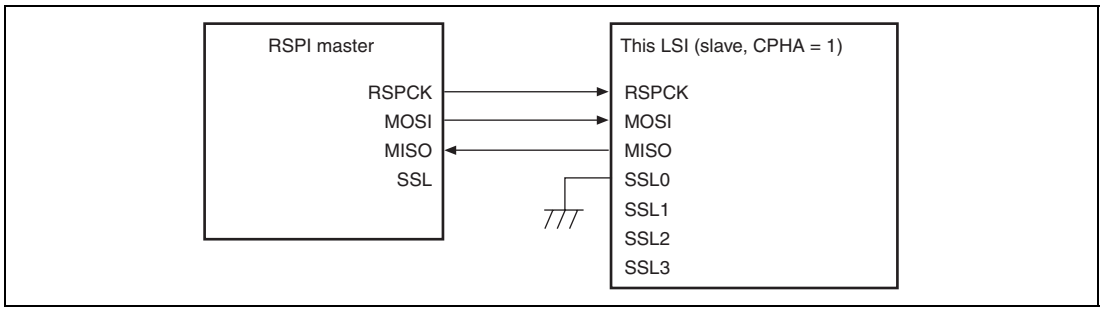


Figure 25.5 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

(3) Single Master/Multi-Slave (with This LSI Acting as Master)

Figure 25.6 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 25.6, the RSPI system is comprised of this LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCK and MOSI outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISO input of this LSI (master). SSL0 to SSL3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively.

This LSI (master) always drives the RSPCK, MOSI, and SSL0 to SSL3 signals. Of the RSPI slave 0 to RSPI slave 3, the slave that receives 0 into the SSL input drives the MISO signal.

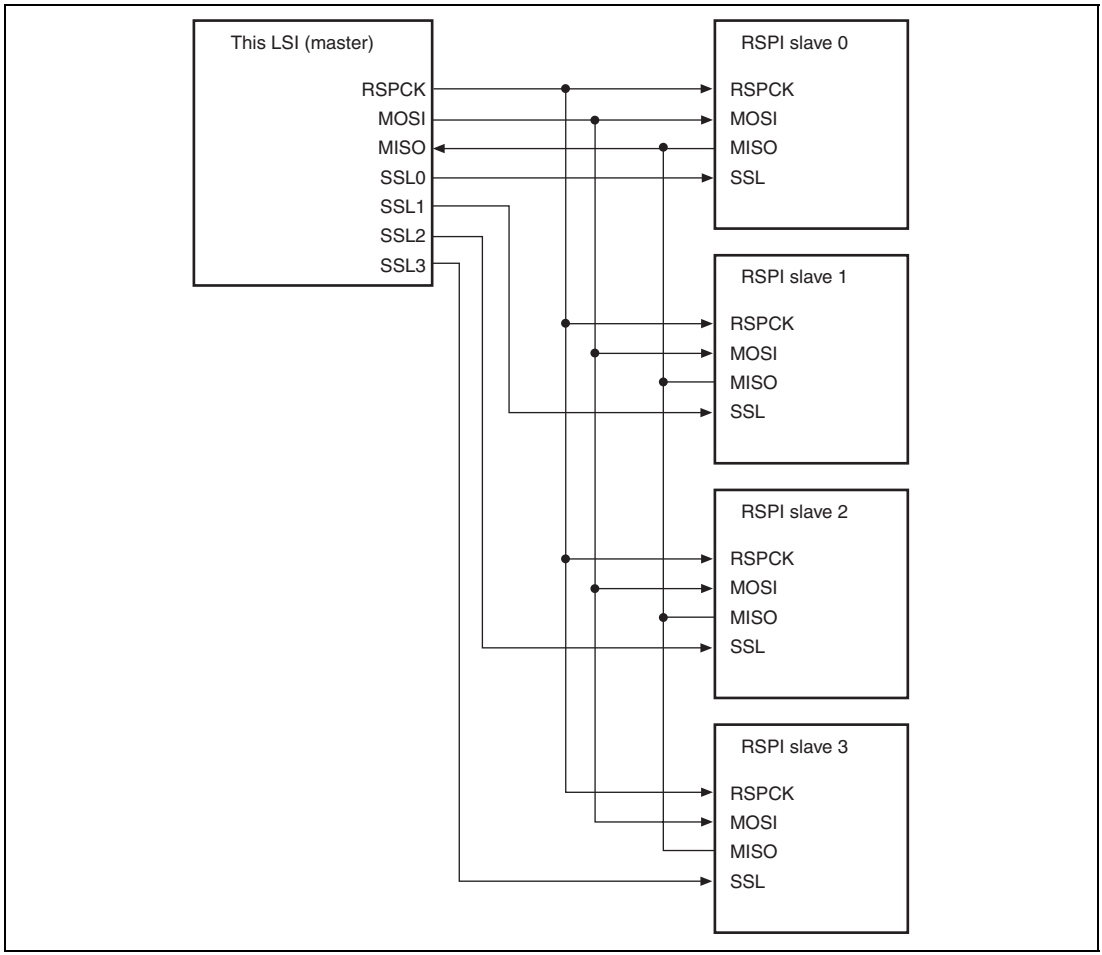


Figure 25.6 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

(4) Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 25.7 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of figure 25.7, the RSPI system is comprised of an RSPI master and these two LSIs (slave X and slave Y).

The RSPCK and MOSI outputs of the RSPI master are connected to the RSPCK and MOSI inputs of these LSIs (slave X and slave Y). The MISO outputs of these LSIs (slave X and slave Y) are all connected to the MISO input of the RSPI master. SSLX and SSLY outputs of the RSPI master are connected to the SSL0 inputs of the LSIs (slave X and slave Y), respectively.

The RSPI master always drives the RSPCK, MOSI, SSLX, and SSLY signals. Of these LSIs (slave X and slave Y), the slave that receives low level input into the SSL0 input drives the MISO signal.

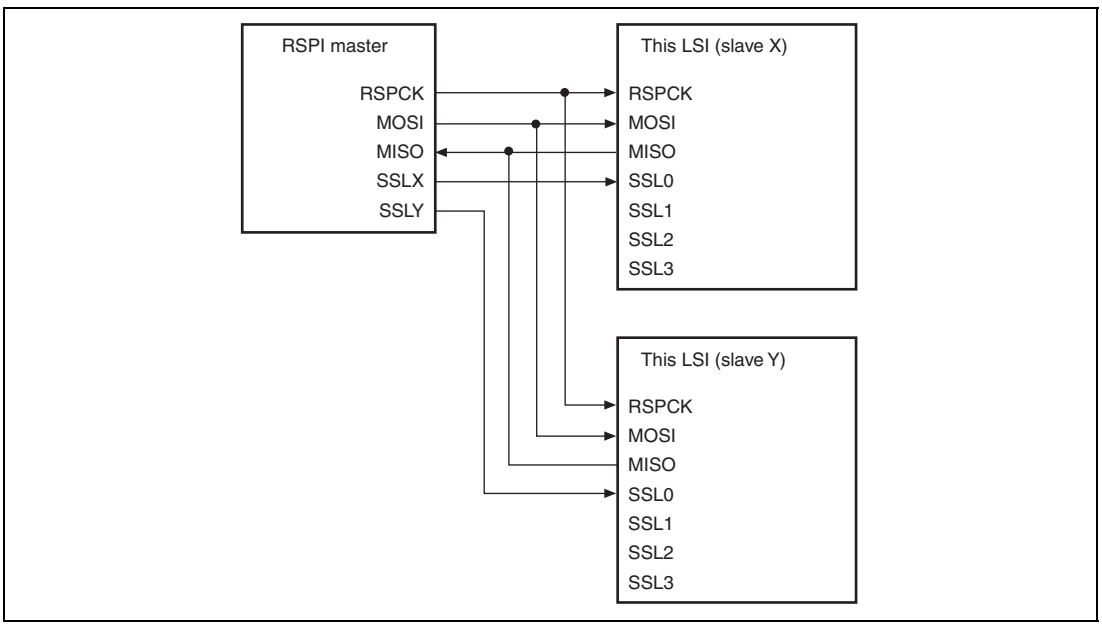


Figure 25.7 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

(5) Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 25.8 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 25.8, the RSPI system is comprised of these two LSIs (master X, master Y) and two RSPI slaves (RSPI slave 1, RSPI slave 2).

The RSPCK and MOSI outputs of this LSI (master X, master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISO inputs of this LSI (master X, master Y). Any generic port Y output from this LSI (master X) is connected to the SSL0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSL0 input of this LSI (master X). The SSL1 and SSL2 outputs of this LSI (master X, master Y) are connected to the SSL inputs of the RSPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSL0 input, and SSL1 and SSL2 outputs for slave connections, the output SSL3 of this LSI is not required.

This LSI drives the RSPCK, MOSI, SSL1, and SSL2 signals when the SSL0 input level is 1. When the SSL0 input level is 0, this LSI detects a mode fault error, sets RSPCK, MOSI, SSL1, and SSL2 to Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives 0 into the SSL input drives the MISO signal.

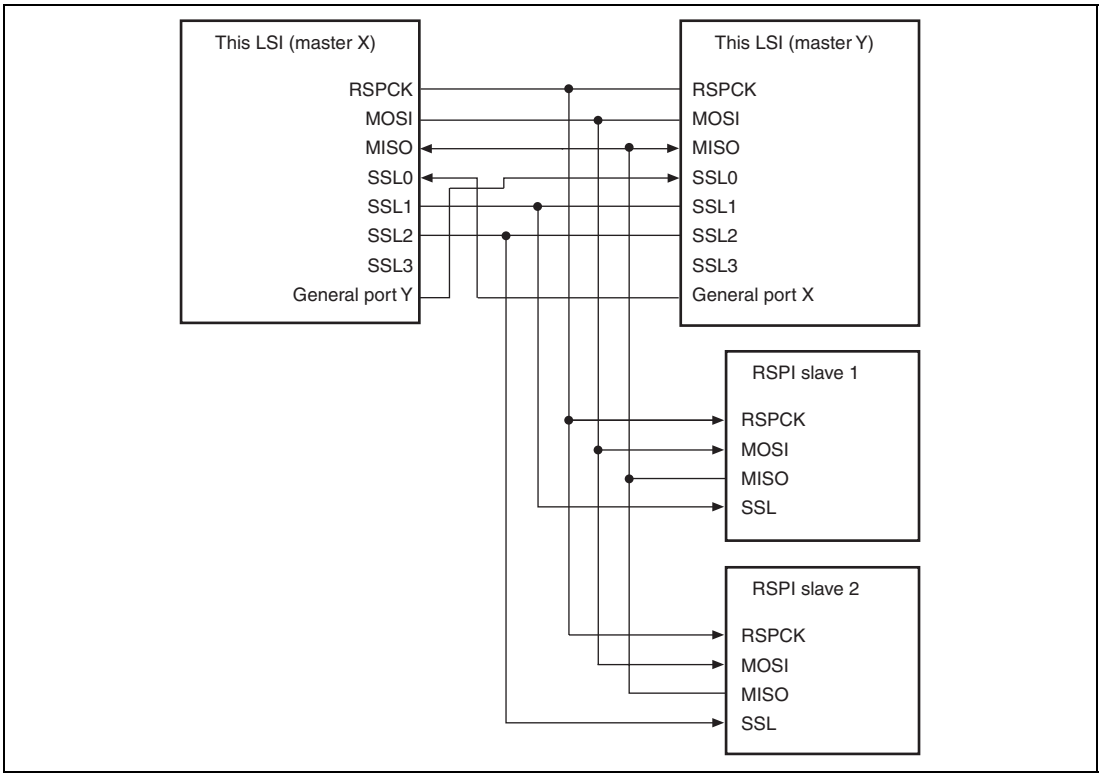


Figure 25.8 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

(6) Master (Clock Synchronous)/Slave (Clock Synchronous) (with This LSI Acting as Master)

Figure 25.9 shows a master (clock synchronous)/slave (clock synchronous) RSPI system configuration example when this LSI is used as a master. In the master (clock synchronous)/slave (clock synchronous) configuration, the SSL0 to SSL3 outputs of this LSI (master) are not used.

This LSI (master) always drives the RSPCK and MOSI signals. The RSPI slave always drives the MISO signal.

Only in the single-master configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, this LSI (master) can execute serial transfer.

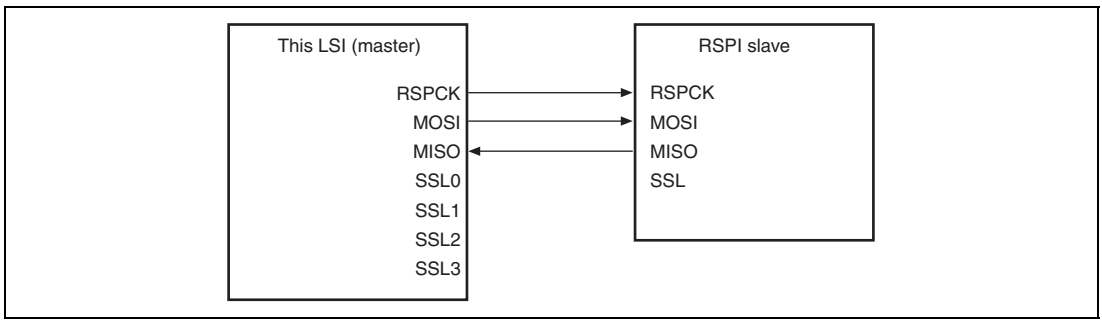


Figure 25.9 Master (Clock Synchronous)/Slave (Clock Synchronous) Configuration Example (This LSI = Master)

(7) Master (Clock Synchronous)/Slave (Clock Synchronous) (with This LSI = Slave)

Figure 25.10 shows a master (clock synchronous)/slave (clock synchronous) RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, this LSI always drives the MISO signal, and the RSPI master always drives the RSPCK and MOSI signals.

Only in the single-slave configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, this LSI (slave) can execute serial transfer.

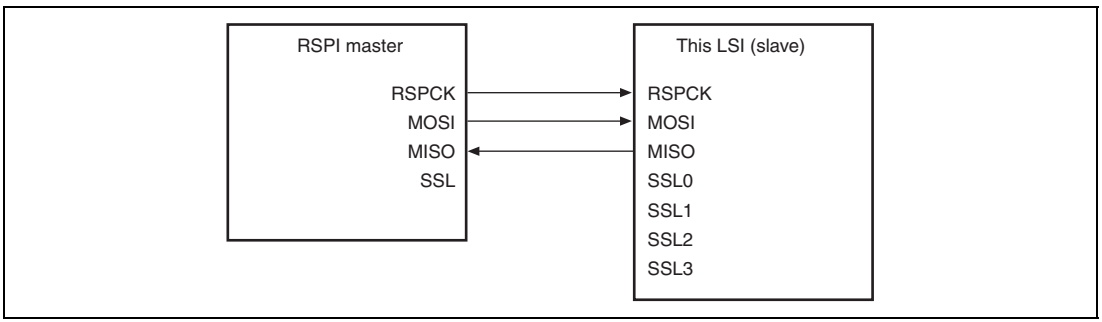


Figure 25.10 Master (Clock Synchronous)/Slave (Clock Synchronous) Configuration Example (This LSI = Slave, CPHA = 1)

25.4.4 Transfer Format

(1) CPHA = 0

Figure 25.11 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 0. Note that clock synchronous operation (with the SPMS bit in the RSPI control register (SPCR) set to 1) is not guaranteed when the CPHA bit is set to 0. In figure 25.11, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI settings. For details, see section 25.4.2, Controlling RSPI Pins.

When the CPHA bit is 0, the output of valid data to the MOSI signal and the driving of valid data to the MISO signal commence at an SSL signal assertion timing. The first RSPCK signal change timing that occurs after the SSL signal assertion becomes the first transfer data fetching timing. After this timing, data is sampled at every RSPCK cycle. The change timing for MOSI and MISO signals is always 1/2 RSPCK cycle after the transfer data fetch timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the cessation of RSPCK oscillation to an SSL signal negation (SSL negation delay). t3 denotes a period in which SSL signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 25.4.9, SPI Operation.

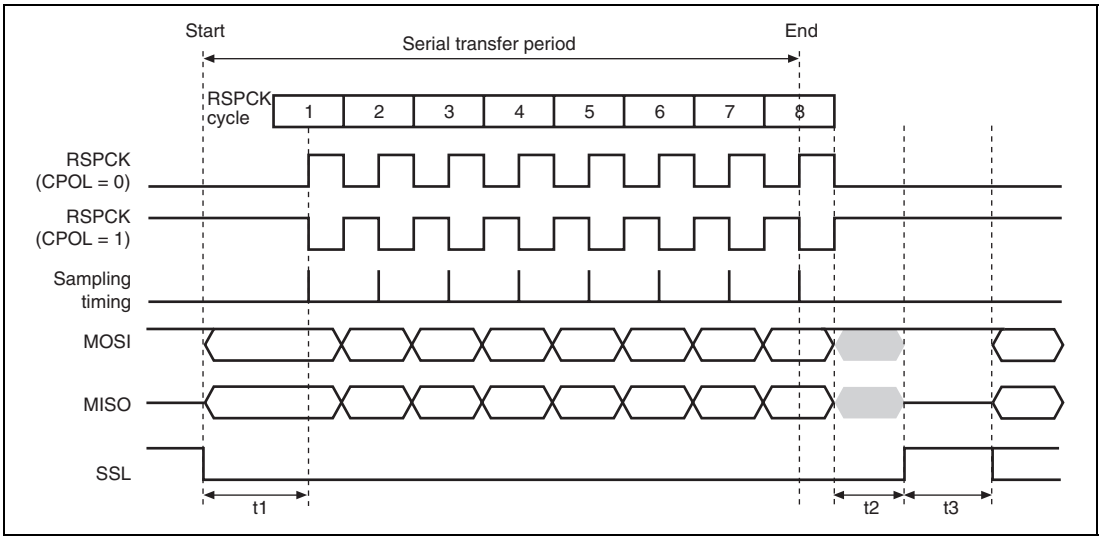


Figure 25.11 RSPI Transfer Format (CPHA = 0)

(2) CPHA = 1

Figure 25.12 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 1. Note that when the SPMS bit in the RSPI control register (SPCR) is 1, the SSL signal is not used and only the RSPCK, MOSI, and MISO signals are used for communication. In figure 25.12, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on RSPI mode (master or slave). For details, see section 25.4.2, Controlling RSPI Pins.

When the CPHA bit is 1, the driving of invalid data to the MISO signals commences at an SSL signal assertion timing. The driving of valid data to the MOSI and MISO signals commences at the first RSPCK signal change timing that occurs after the SSL signal assertion. After this timing, data is updated at every RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycle after the data update timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 25.4.9, SPI Operation.

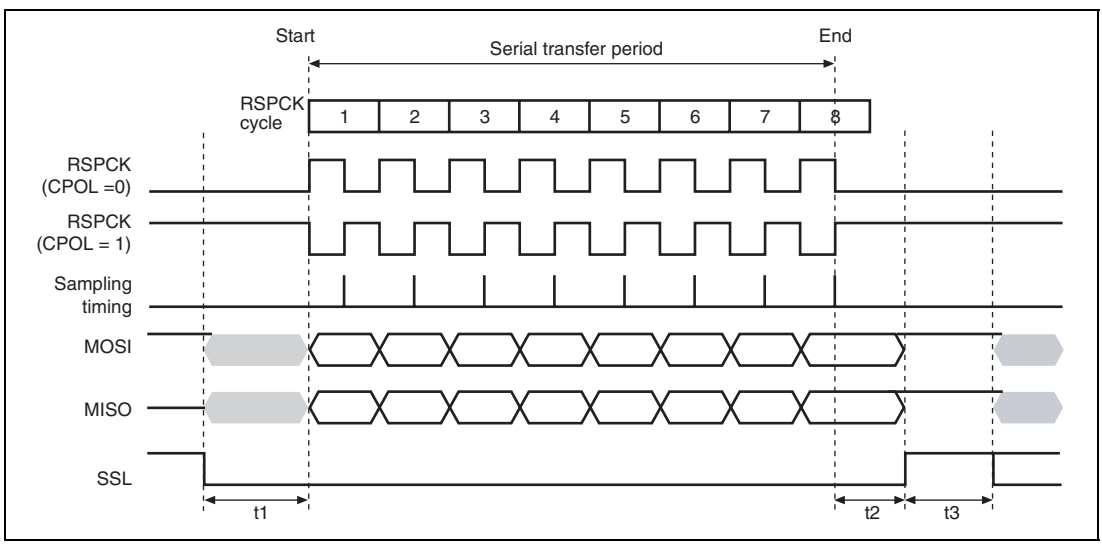


Figure 25.12 RSPI Transfer Format (CPHA = 1)

25.4.5 Data Format

The RSPI's data format depends on the settings in the RSPI command register (SPCMD). Irrespective of MSB/LSB first, the RSPI treats the assigned data length of data from the LSB of the RSPI data register (SPDR) as transfer data.

(1) MSB First Transfer (32-Bit Data)

Figure 25.13 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 32-bit MSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI copies the data in the transmit buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 32 bits has passed, data R31 to R00 is stored in the shift register. In this state, the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R31 to R00 is shifted out from the shift register.

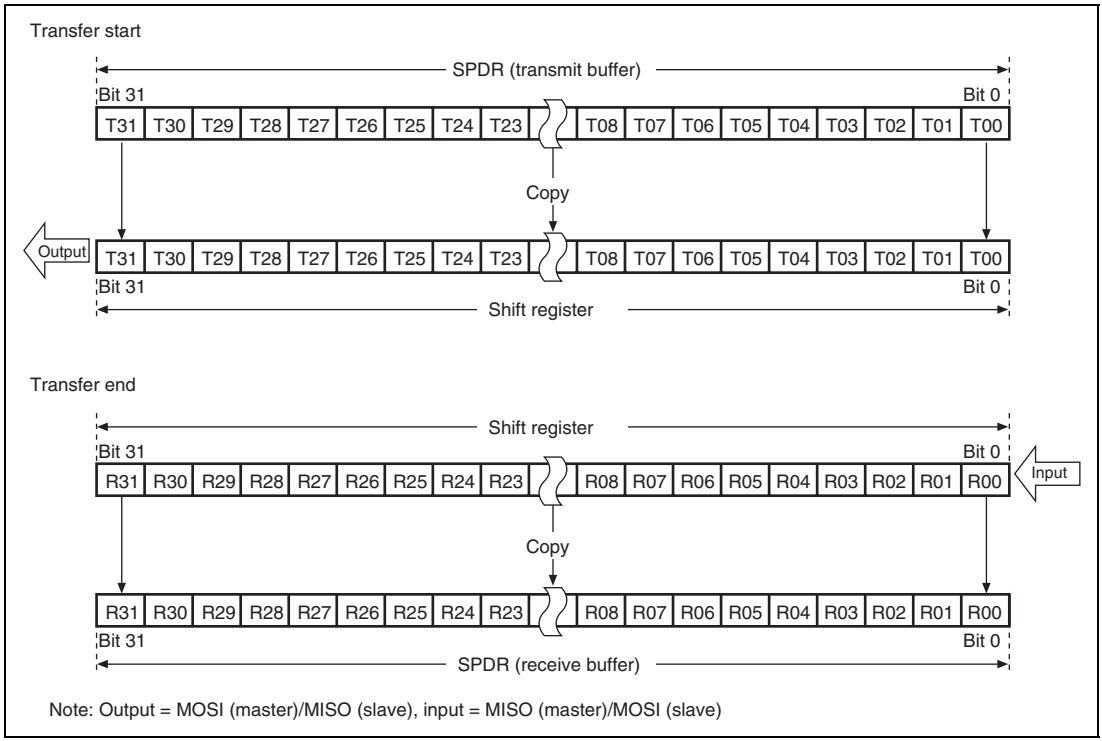


Figure 25.13 MSB First Transfer (32-Bit Data)

(2) MSB First Transfer (24-Bit Data)

Figure 25.14 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 24-bit data length MSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI copies the data in the transmit buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from bit 23 of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 24 bits has passed, received data R23 to R00 is stored in bits 23 to 0 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 24 in the shift register. In this state, the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R23 to R00 is shifted out from the shift register.

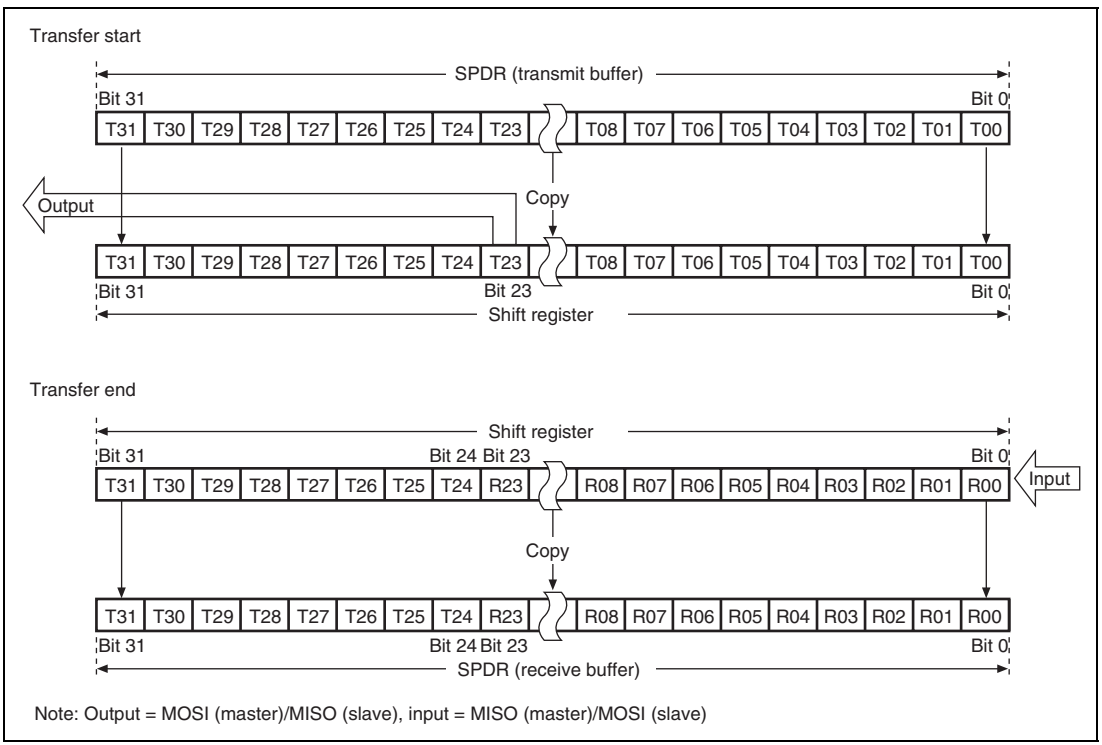


Figure 25.14 MSB First Transfer (24-Bit Data)

(3) LSB First Transfer (32-Bit Data)

Figure 25.15 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 32-bit data length LSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmit buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 32 bits has passed, data R00 to R31 is stored in the shift register. In this state, the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R00 to R31 is shifted out from the shift register.

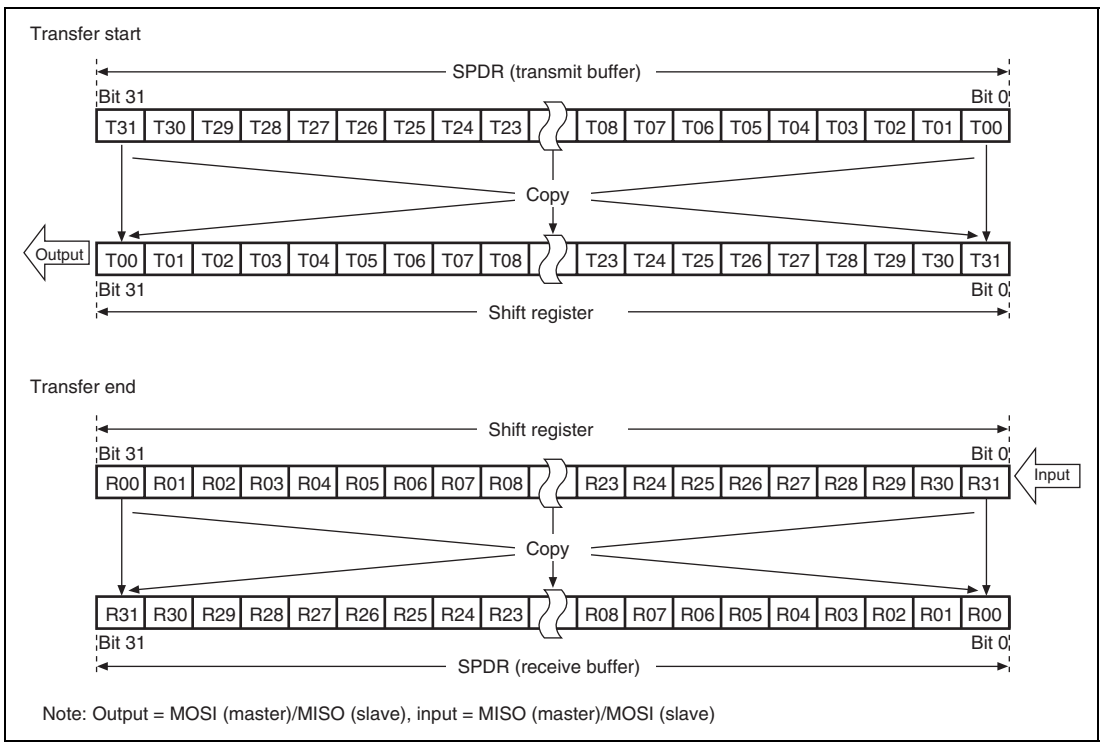


Figure 25.15 LSB First Transfer (32-Bit Data)

(4) LSB First Transfer (24-Bit Data)

Figure 25.16 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 24-bit data length LSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmit buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from bit 8 of the shift register. When the RSPCK cycle required for the serial transfer of 24 bits has passed, received data R00 to R23 is stored in bits 31 to 8 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 7 to 0 of the shift register. In this state, the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R00 to R23 is shifted out from the shift register.

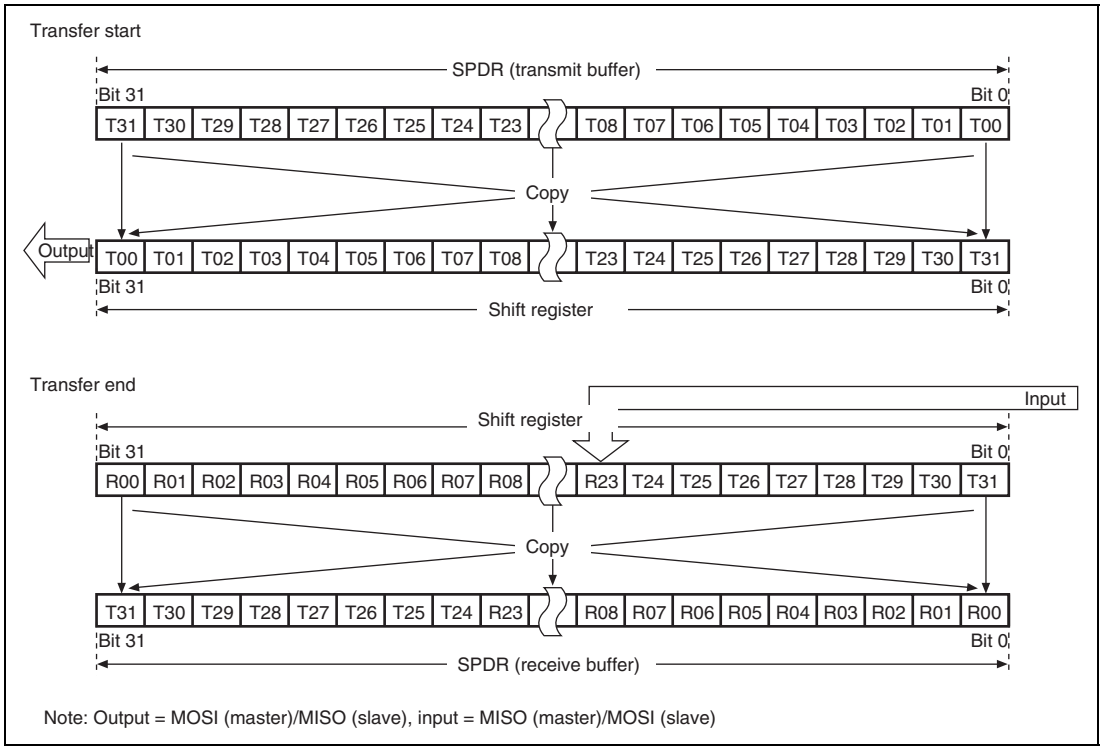


Figure 25.16 LSB First Transfer (24-Bit Data)

25.4.6 Transmit Buffer Empty/Receive Buffer Full Flags

Figure 25.17 shows an example of operation of the RSPI transmit buffer empty flag (SPTEF) and the RSPI receive buffer full flag in the RSPI status register (SPSR). The SPDR access depicted in figure 25.17 indicates the condition of access from the DTC/DMAC to the RSPI data register (SPDR), where I denotes an idle cycle, W a write cycle, and R a read cycle. In this example in figure 25.17, the RSPI executes an 8-bit serial transfer with the SPFC[1:0] bits in the RSPI data control register (SPDCR) set to 00, the CPHA bit in the RSPI command register (SPDR) set to 1, and the CPOL bit in SPDR set to 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

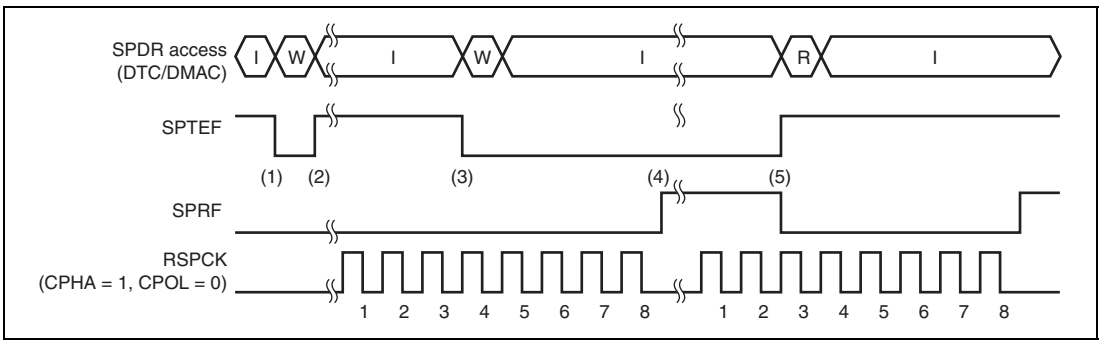


Figure 25.17 SPTEF and SPRF Bit Operation Example

The operation of the flags at timings shown in steps (1) to (5) in the figure is described below.

1. When the DTC/DMAC writes transmit data to SPDR when the transmit buffer of SPDR is empty, the RSPI sets the SPTEF bit to 0, and writes data to the transmit buffer, with no change in the SPRF flag.
2. If the shift register is empty, the RSPI sets the SPTEF bit to 1, and copies the data in the transmit buffer to the shift register, with no change in the SPRF flag. How a serial transfer is started depends on the mode of the RSPI. For details, see section 25.4.9, SPI Operation, and section 25.4.10, Clock Synchronous Operation.
3. When the DTC/DMAC writes transmit data to SPDR with the transmit buffer of SPDR being empty, the RSPI sets the SPTEF bit to 1, and writes data to the transmit buffer, while the SPRF flag remains unchanged. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in the transmit buffer to the shift register.

4. When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI sets the SPRF bit to 1, and copies the receive data in the shift register to the receive buffer. Because the shift register becomes empty upon completion of serial transfer, if the transmit buffer was full before the serial transfer ended, the RSPI sets the SPTEF bit to 1, and copies the data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer the RSPI determines that the shift register is empty, and as a result data transfer from the transmit buffer to the shift register is enabled.
5. When the DTC/DMAC reads SPDR with the receive buffer being full, the RSPI sets the SPRF bit to 0, and sends the data in the receive buffer to the bus inside the chip.

If the CPU or the DTC/DMAC writes to SPDR when the SPTEF bit is 0, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure that the SPTEF bit is 1. That the SPTEF bit is 1 can be checked by reading SPSR or by using an RSPI transmit interrupt. To use an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI is disabled (the SPE bit in SPCR being 0), the SPTEF bit is initialized to 1. For this reason, setting the SPTIE bit to 1 when the RSPI is disabled generates an RSPI transmit interrupt.

When serial transfer ends with the SPRF bit being 1, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 25.4.7, Error Detection). To prevent a receive data overrun error, set the SPRF bit to 0 before the serial transfer ends. That the SPRF bit is 1 can be checked by either reading SPSR or by using an RSPI receive interrupt. To use an RSPI receive interrupt, set the SPRIE bit in SPCR to 1.

25.4.7 Error Detection

In the normal RSPI serial transfer, the data written from the RSPI data register (SPDR) to the transmit buffer by either the CPU or the DTC is serially transmitted, and either the CPU or the DTC/DMAC can read the serially received data from the receive buffer of SPDR. If access is made to SPDR by either the CPU or the DTC, depending on the status of the transmit buffer/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error or a mode fault error. Table 25.8 shows the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 25.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
A	Either the CPU or the DTC/DMAC writes to SPDR when the transmit buffer is full.	Retains the contents of the transmit buffer. Missing write data.	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	Either the CPU or the DTC/DMAC reads from SPDR when the receive buffer is empty.	Previously received serial data is output to the CPU or the DMAC.	None
D	Serial transfer terminates when the receive buffer is full.	Retains the contents of the receive buffer. Missing serial receive data.	Overrun error
E	The SSL0 input signal is asserted when the serial transfer is idle in multi-master mode.	RSPI disabled. Driving of the RSPCK, MOSI, and SSL1 to SSL3 output signals stopped.	Mode fault error

	Occurrence Condition	RSPI Operation	Error Detection
F	The SSL0 input signal is asserted during serial transfer in multi-master mode.	Serial transfer suspended. Missing send/receive data. Driving of the RSPCK, MOSI, and SSL1 to SSL3 output signals stopped. RSPI disabled.	Mode fault error
G	The SSL0 input signal is negated during serial transfer in slave mode.	Serial transfer suspended. Missing send/receive data. Driving of the MISO output signal stopped. RSPI disabled.	Mode fault error

On operation A shown in table 25.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR by the CPU or the DTC/DMAC, write operations to SPDR should be executed when the SPTEF bit in the RSPI status register (SPSR) is 1.

Likewise, the RSPI does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Notice that the received data from the previous serial transfer is retained in the receive buffer of SPDR, and thus it can be correctly read by the CPU or the DTC/DMAC (if SPDR is not read before the end of the serial transfer, an overrun error may result).

Similarly, the RSPI does not detect an error on operation C. To prevent the CPU or the DTC/DMAC from reading extraneous data, SPDR read operation should be executed when the SPRF bit in SPSR is 1.

An overrun error shown in D is described in section 25.4.7 (1), Overrun Error. A mode fault error shown in E to G is described in section 25.4.7 (2), Mode Fault Error. On operations of the SPTEF and SPRF bits in SPSR, see section 25.4.6, Transmit Buffer Empty/Receive Buffer Full Flags.

(1) Overrun Error

If serial transfer ends when the receive buffer of the RSPI data register (SPDR) is full, the RSPI detects an overrun error, and sets the OVRF bit in SPSR to 1. When the OVRF bit 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 reset the OVRF bit in SPSR to 0, either execute a system reset, or write a 0 to the OVRF bit after the CPU has read SPSR with the OVRF bit set to 1.

Figure 25.18 shows an example of operation of the SPRF and OVRF bits in SPSR. The SPSR access depicted in figure 25.18 indicates the condition of access from the CPU to SPSR, and from the DTC/DMAC to SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of figure 25.18, the RSPI performs an 8-bit serial transfer in which the CPHA bit in the RSPI command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

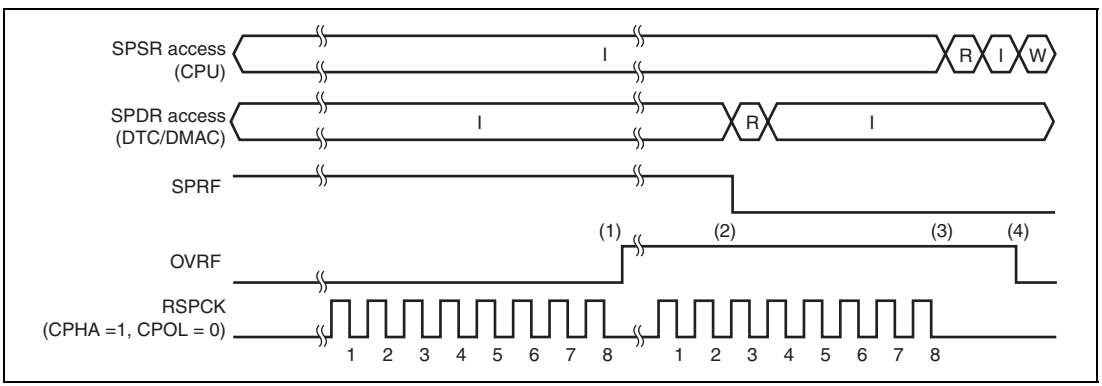


Figure 25.18 SPRF and OVRF Bit Operation Example

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 SPRF bit being 1 (receive buffer full), the RSPI detects an overrun error, and sets the OVRF bit to 1. The RSPI does not copy the data in the shift register to the receive buffer. In master mode, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM1 and SPECM0 in the RSPI sequence status register (SPSSR).
2. When the DTC/DMAC reads SPDR, the RSPI sets the SPRF bit to 0, and outputs the data in the receive buffer to an internal bus. The receive buffer becoming empty does not clear the OVRF bit.

3. If the serial transfer terminates with the OVRF bit being 1 (an overrun error), the RSPI keeps the SPRF bit at 0 and does not update it. Likewise, the RSPI does not copy the data in the shift register to the receive buffer. When in master mode, the RSPI does not update bits SPECM1 and SPECM0 of SPSSR. If, in an overrun error state, 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 is enabled from the transmit buffer to the shift register.
4. If the CPU writes a 0 to the OVRF bit after reading SPSR when the OVRF bit is 1, the RSPI clears the OVRF bit.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. When executing a serial transfer without using an RSPI error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is run in master mode, the pointer value to SPCMD can be checked by reading bits SPECM2 to SPECM0 of SPSSR.

If an overrun error occurs and the OVRF bit is set to 1, normal reception operations cannot be performed until such time as the OVRF bit is cleared. The OVRF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition in which the OVRF bit is set to 1, the CPU writes a 0 to the OVRF bit.
- System reset

(2) Mode Fault Error

The RSPI operates in multi-master mode when the MSTR bit is 1, the SPMS bit is 0 and the MODFEN bit is 1 in the RSPI control register (SPCR). If the active level is input with respect to the SSL0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR). The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit is 1 and the SPMS bit is 0 in the RSPI in slave mode and if the SSL0 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 the driving of output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the RSPI function is disabled (see section 25.4.8, Initializing RSPI). In multi-master configuration, it is possible to release the master right by using a mode fault error to stop the driving of output signals and the RSPI function.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. To detect a mode fault error without using an RSPI error interrupt, it is necessary to poll SPSR. When using the RSPI in master mode, one can read bits SPECM2 to SPECM0 of SPSSR to verify the value of the pointer to SPCMD when an error occurs.

When the MODF bit is 1, the RSPI ignores the writing of the value 1 to the SPE bit by the CPU. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0. The MODF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition where the MODF bit has turned 1, the CPU writes a 0 to the MODF bit.
- System reset

25.4.8 Initializing RSPI

If the CPU writes a 0 to the SPE bit in the RSPI control register (SPCR) or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes a part of the module function. If a system reset occurs, the RSPI initializes all of the module function. An explanation follows of initialization by the clearing of the SPE bit and initialization by a system reset.

(1) Initialization by Clearing SPE Bit

When the SPE bit in SPCR is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals only in slave mode (Hi-Z)
- Initializing the internal state of the RSPI
- Initializing the SPTEF bit in the RSPI status register (SPSR)

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 CPU resets the value 1 to the SPE bit.

The SPRF, OVRF, and MODF bits in SPSR are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.

The SPTEF bit in SPSR is initialized to 1. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized by the CPU, in order to disable any RSPI transmit interrupt, a 0 should be written to the SPTIE bit simultaneously with the writing of a 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write a 0 to the SPTIE bit.

(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 (1), Initialization by Clearing SPE Bit.

25.4.9 SPI Operation

(1) Slave Mode Operation

(1-1) Starting a Serial Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) is 0, when detecting an SSL0 input signal assertion, the RSPI needs to start driving valid data to the MISO output signal. For this reason, the asserting of the SSL0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCK edge in an SSL0 signal asserted condition, the RSPI needs to start driving valid data to the MSO signal. For this reason, when the CPHA bit is 1, the first RSPCK edge in an SSL0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register intact, in the full state.

Irrespective of CPHA bit settings, the timing at which the RSPI starts driving MISO output signals is the SSL0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on CPHA bit settings.

For details on the RSPI transfer format, see section 25.4.4, Transfer Format. The polarity of the SSL0 input signal depends on the setting of the SSL0P bit in the RSPI slave select polarity register (SSLP).

(1-2) Terminating a Serial Transfer

Irrespective of the CPHA bit in RSPI command register 0 (SPCMD0), the RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Irrespective of the value of the SPRF bit, upon termination of a serial transfer the RSPI changes the status of the shift register to "empty". A mode fault error occurs if the RSPI detects an SSL0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 25.4.7, Error Detection).

The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. The polarity of the SSL0 input signal depends on the setting in the SSL0P bit in the RSPI slave select polarity register (SSLP). For details on the RSPI transfer format, see section 25.4.4, Transfer Format.

(1-3) Notes on Single-Slave Operations

If the CPHA bit in RSPI command register 0(SPCMD0) is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSL0 input signal. In the type of configuration shown in figure 25.5 as an example, if the RSPI is used in single-slave mode, the SSL0 signal is always fixed at active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute send/receive operation by the RSPI in a configuration in which the SSL0 input signal is fixed at active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSL0 input signal should not be fixed.

(1-4) Burst Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSL0 input signal. If the CPHA bit is 1, the period from the first RSPCK edge to the sampling timing for the reception of the final bit in an SSL0 signal active state corresponds to a serial transfer period. Even when the SSL0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of access.

If the CPHA bit is 0, for the reason given in (1-3), Notes on Single-Slave Operations, second and subsequent serial transfers during the burst transfer cannot be executed correctly.

(1-5) Initialization Flowchart

Figure 25.19 shows an example of initialization flowchart for using the RSPI in slave mode during SPI operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

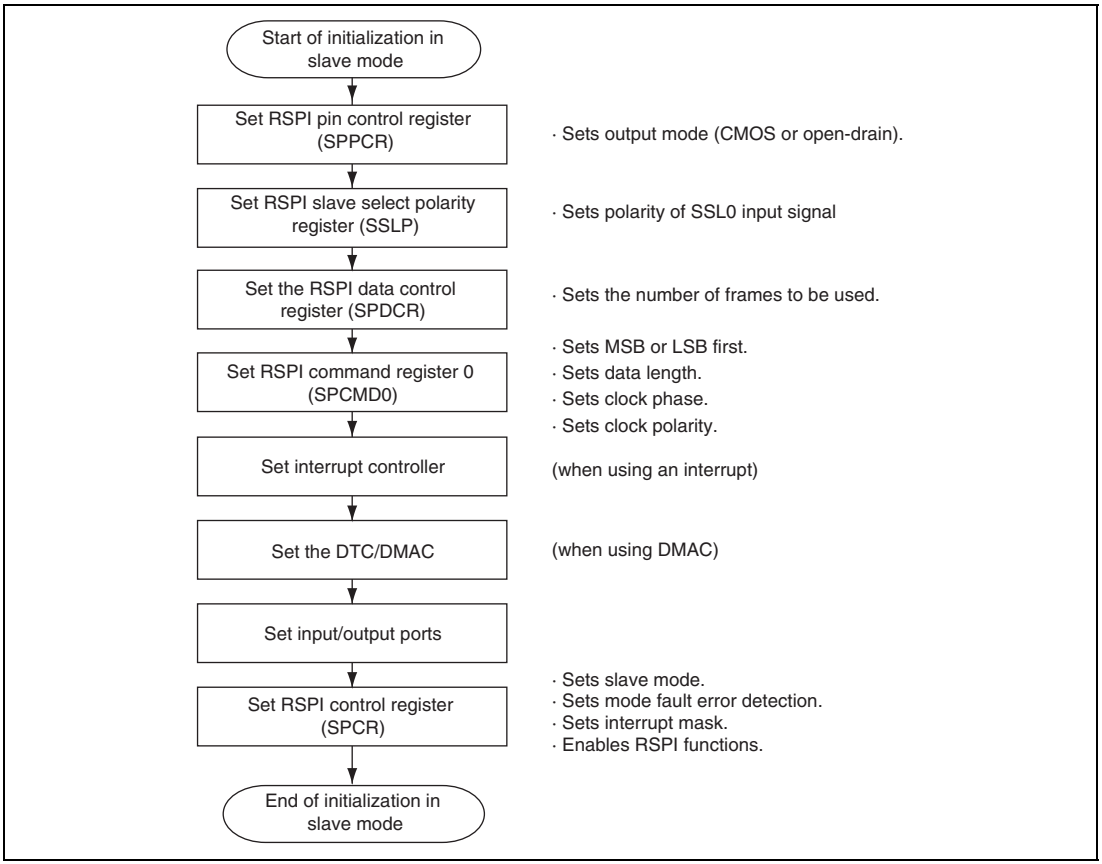


Figure 25.19 Example of Initialization Flowchart in Slave Mode

(1-6) Transfer Operation Flowchart (CPHA = 0)

Figure 25.20 shows an example of transfer operation flowchart for using the RSPI in slave mode during SPI operation, when the CPHA bit in RSPI command register 0 (SPCMD0) is 0.

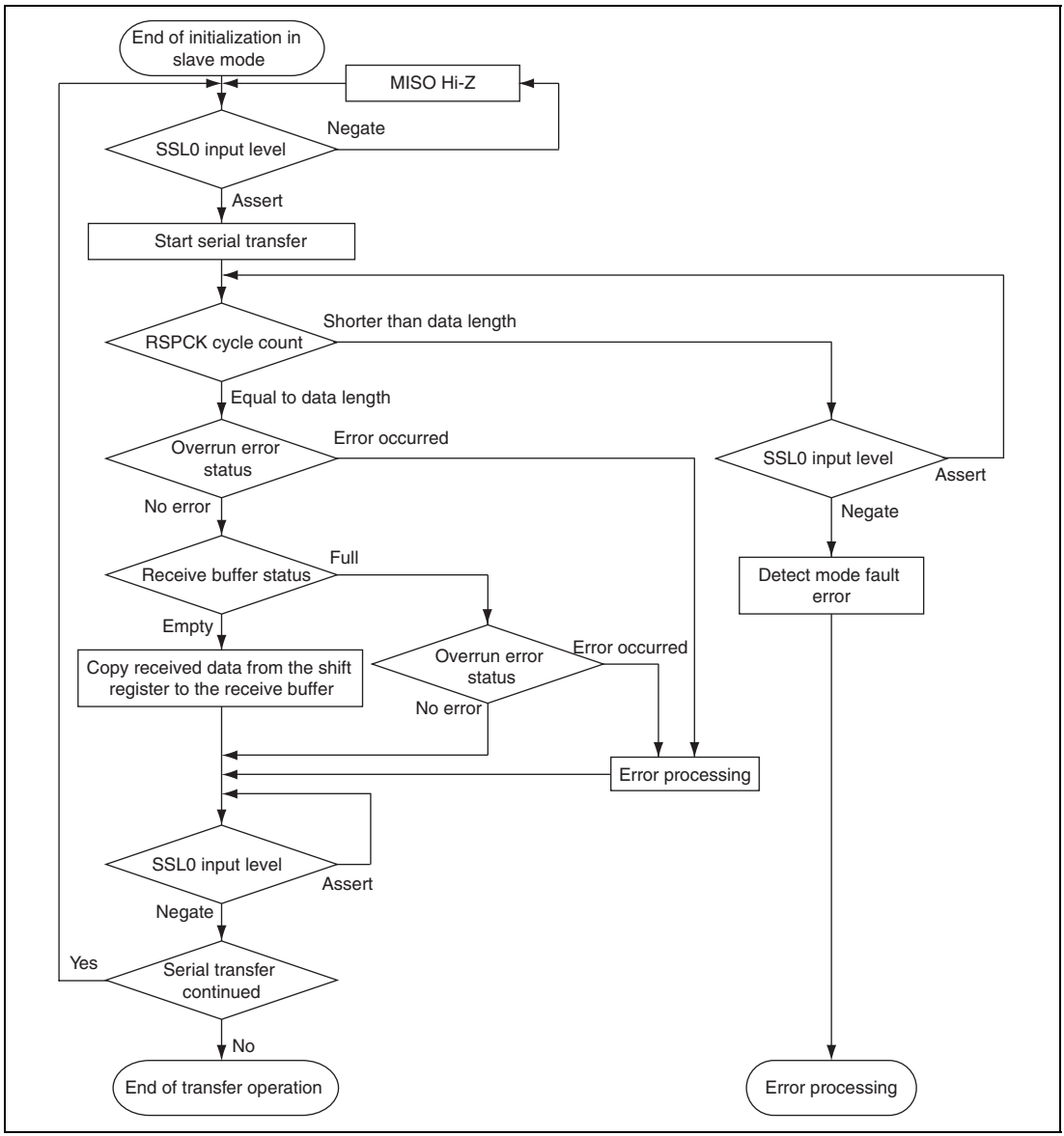


Figure 25.20 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 0)

(1-7) Transfer Operation Flowchart (CPHA = 1)

Figure 25.21 shows an example of transfer operation flowchart for using the RSPI in slave mode during SPI operation, when the CPHA bit in RSPI command register 0 (SPCMD0) is 1.

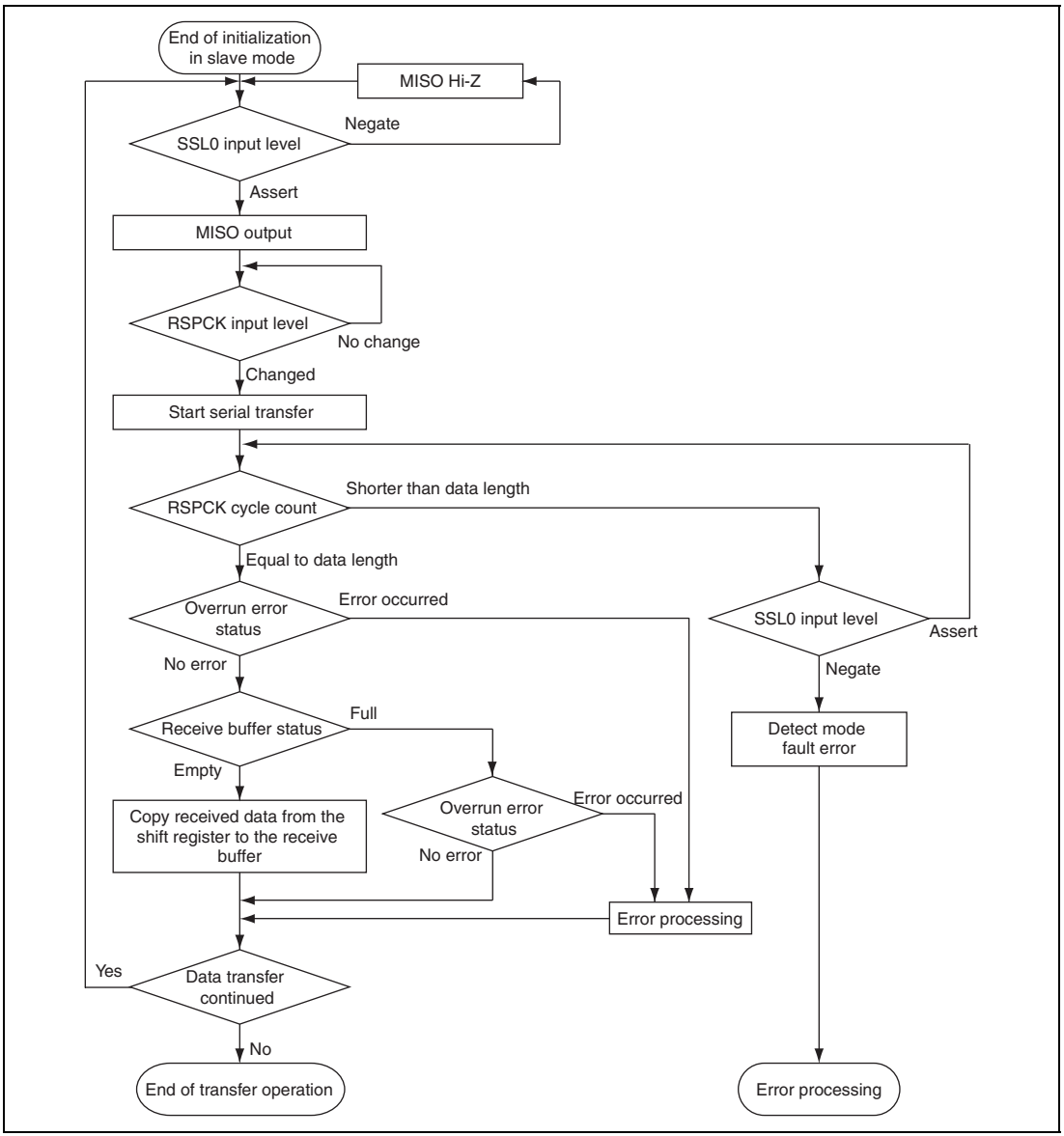


Figure 25.21 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

(2) Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 25.4.7, Error Detection). When operating in single-master mode (RSPI), 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-/multi-master modes.

(2-1) Starting Serial Transfer

The RSPI updates the data in the transmit buffer when the SPTEF bit in the RSPI status register (SPSR) is 1 and when either the CPU or the DTC/DMAC has written data to the RSPI data register (SPDR). If the shift register is empty in a condition where the SPTEF bit has been cleared to 0 due to the writing of 0 either after the writing to SPDR from the DTC/DMAC or by the writing of 0 after the value 1 is read from the SPTEF bit by the CPU, the RSPI copies the data in the transmit buffer to the shift register and starts a 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 from the CPU.

For details on the RSPI transfer format, see section 25.4.4, Transfer Format. The polarity of the SSL output signal depends on the setting in the RSPI slave select polarity register (SSLP).

(2-2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMD), the RSPI terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. If the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 in SPCMD. The polarity of the SSL output signal depends on the setting in the RSPI slave select polarity register (SSLP). For details on the RSPI transfer format, see section 25.4.4, Transfer Format.

(2-3) Sequence Control

The transfer format that is employed in master mode is determined by the RSPI sequence control register (SPSCR), RSPI command registers 0 to 3 (SPCMD0 to SPCMD3), the RSPI bit rate register (SPBR), the RSPI clock delay register (SPCKD), the RSPI slave select negation delay register (SSLND), and the RSPI next-access delay register (SPND).

The SPSCR register is used to determine the sequence configuration for serial transfers that are executed by a master mode RSPI. The following items are set in RSPI command registers SPCMD0 to SPCMD3: SSL output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMD0 to SPCMD3. The RSPI contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP[1:0] in the RSPI sequence status register (SPSSR). When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

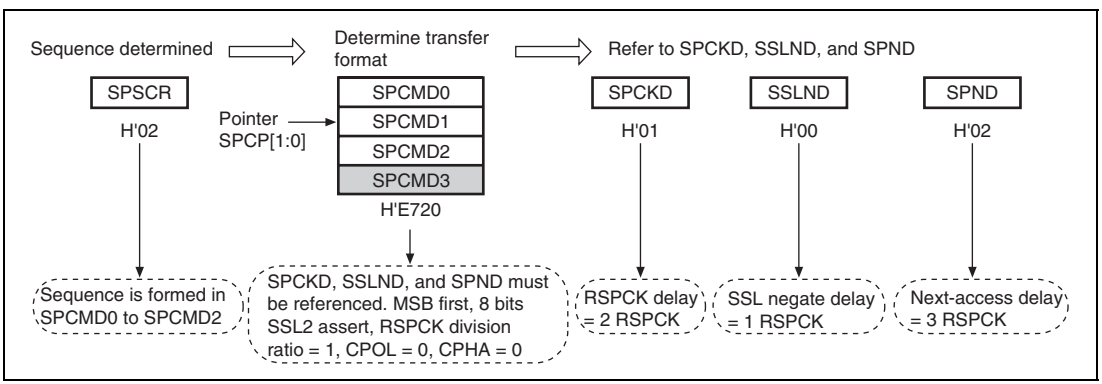


Figure 25.22 Determination Procedure of Serial Transfer Mode in Master Mode

(2-4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMD) which the RSPI refers to during the current serial transfer is 1, the RSPI keeps the SSL signal level during the serial transfer until the beginning of the SSL signal assertion for the next serial transfer. If the SSL signal level for the next serial transfer is the same as the SSL signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSL signal assertion status (burst transfer).

Figure 25.23 shows an example of an SSL signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 settings. The text below explains the RSPI operations (1) to (7) as depicted in figure 25.23. It should be noted that the polarity of the SSL output signal depends on the settings in the RSPI slave select polarity register (SSLP).

1. Based on SPCMD0, the RSPI asserts the SSL signal and inserts RSPCK delays.
2. The RSPI executes serial transfers according to SPCMD0.
3. The RSPI inserts SSL negation delays.
4. Because the SSLKP bit in SPCMD0 is 1, the RSPI keeps the SSL signal value on SPCMD0. This period is sustained for next-access delay of SPCMD0 + 2 Pϕ at a minimum. If the shift register is empty after the passage of a minimum period, this period is sustained until such time as the transmit data is stored in the shift register for another transfer.
5. Based on SPCMD1, the RSPI asserts the SSL signal and inserts RSPCK delays.
6. The RSPI executes serial transfers according to SPCMD1.
7. Because the SSLKP bit in SPCMD1 is 0, the RSPI negates the SSL signal. In addition, a next-access delay is inserted according to SPCMD1.

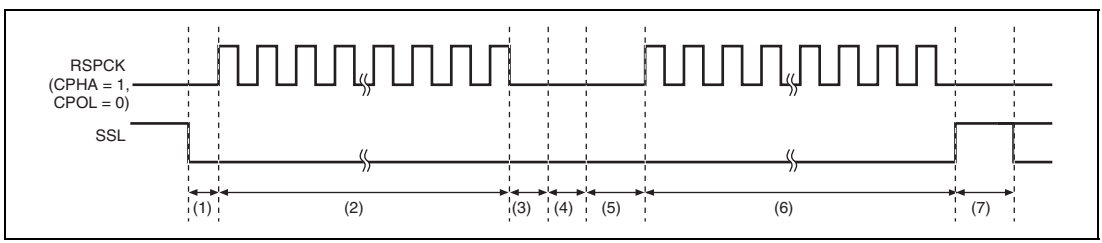


Figure 25.23 Example of Burst Transfer Operation using SSLKP Bit

If the SSL signal settings in the SPCMD in which 1 is assigned to the SSLKP bit are different from the SSL signal output settings in the SPCMD to be used in the next transfer, the RSPI switches the SSL signal status to SSL signal assertion ((5) in figure 25.23) corresponding to the command for the next transfer. Notice that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and the possibility arises of the collision of signal levels.

The RSPI in master mode refers to the SSL signal operation for the case where the SSLKP bit is not used within the module. Even when the CPHA bit in SPCMD is 0, the RSPI can accurately start serial transfers by asserting the SSL signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 25.4.9, SPI Operation).

(2-5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on SCKDEN bit settings in the RSPI command register (SPCMD) and on RSPCK delay register (SPCKD) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SCKDEN bit in the selected SPCMD and SPCKD, as shown in table 25.9. For a definition of RSPCK delay, see section 25.4.4, Transfer Format.

Table 25.9 Relationship among SCKDEN and SPCKD Settings and RSPCK Delay Values

SCKDEN	SPCKD	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(2-6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on SLNDEN bit settings in the RSPI command register (SPCMD) and on SSL negation delay register (SSLND) settings. The RSPI determines the SPCMD to be referred to during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SLNDEN bit in the selected SPCMD and SSLND, as shown in table 25.10. For a definition of SSL negation delay, see section 25.4.4, Transfer Format.

Table 25.10 Relationship among SLNDEN and SSLND Settings and SSL Negation Delay Values

SLNDEN	SSLND	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(2-7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on SPNDEN bit settings in the RSPI command register (SPCMD) and on next-access delay register (SPND) settings. The RSPI determines the SPCMD to be referred to during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPNDEN bit in the selected SPCMD and SPND, as shown in table 25.11. For a definition of next-access delay, see section 25.4.4, Transfer Format.

Table 25.11 Relationship among SPNDEN and SPND Settings and Next-Access Delay Values

SPNDEN	SPND	Next-Access Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(2-8) Initialization Flowchart

Figure 25.24 shows an example of initialization flowchart for using the RSPI in master mode during SPI operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

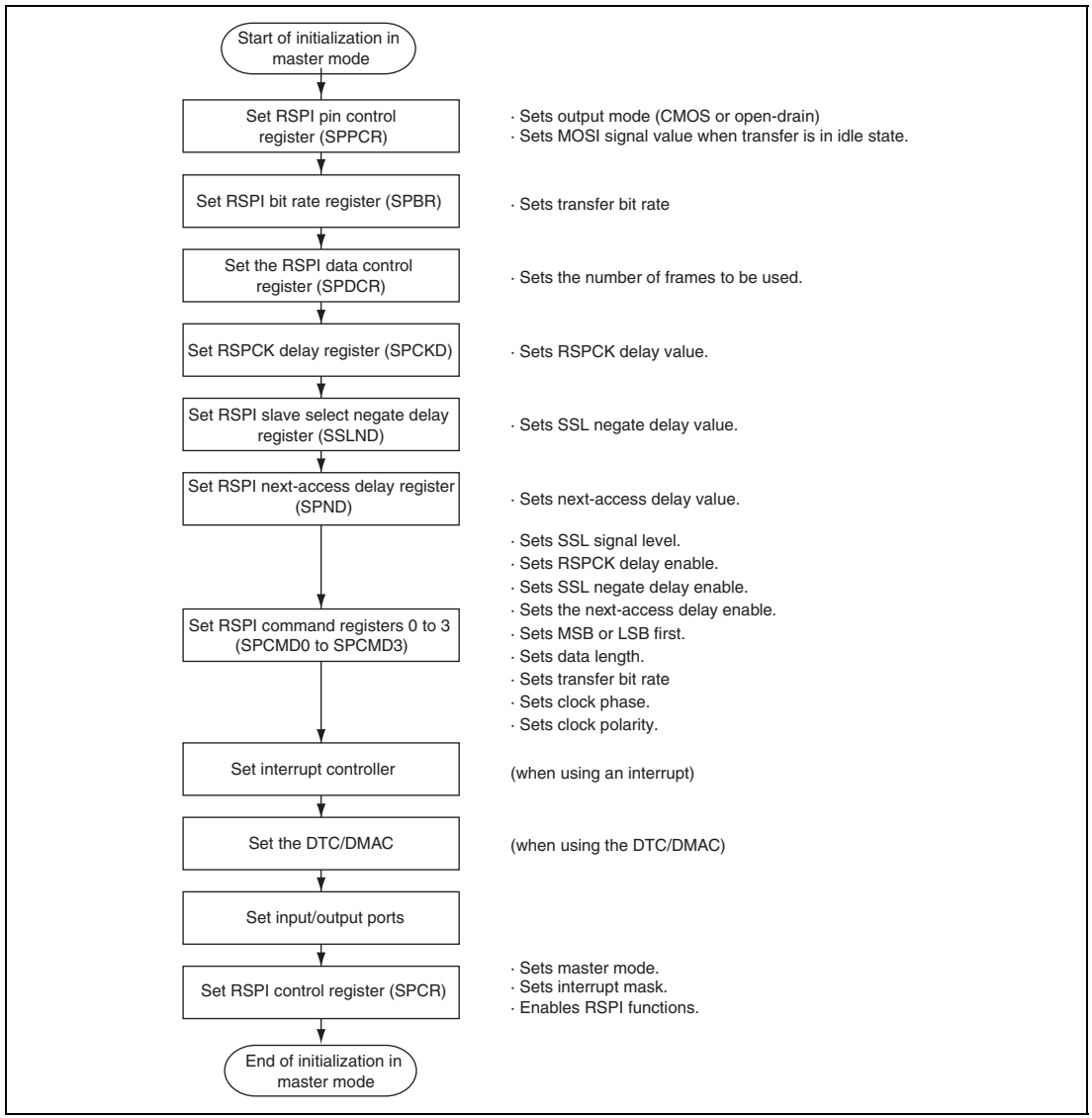
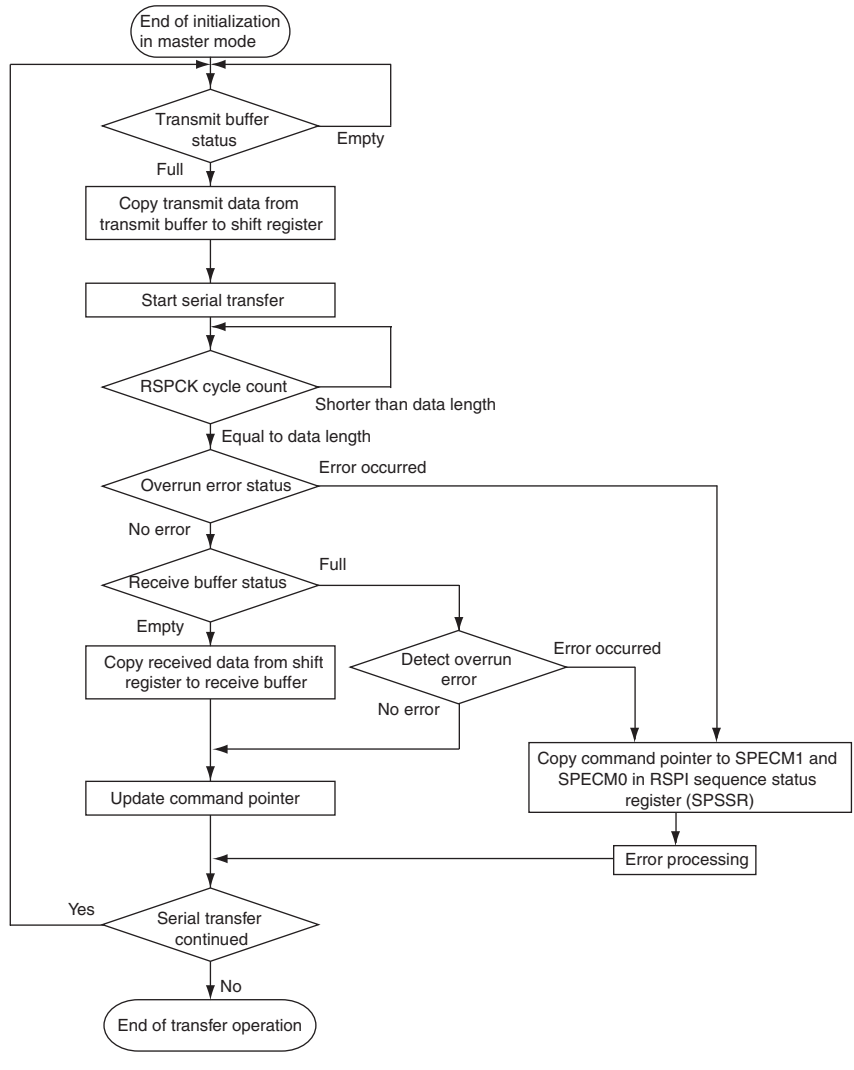


Figure 25.24 Example of Initialization Flowchart in Master Mode

(2-9) Transfer Operation Flowchart

Figure 25.25 shows an example of transfer operation flowchart for using the RSPI in master mode during SPI operation.



Note: When the SSL0 input signal is asserted in multi-master mode, the RSPI detects a mode fault error irrespective of the hardware status. When detecting the error, the RSPI copies the command pointer to the SPECM[1:0] bits in the RSPI sequence status register (SPSSR).

Figure 25.25 Example of Transfer Operation Flowchart in Master Mode

25.4.10 Clock Synchronous Operation

The RSPI selects clock synchronous operation when the SPMS bit in the RSPI control register (SPCR) is 1. During clock synchronous operation, the SSL pins are not used and the remaining three pins, RSPCK, MOSI, and MISO are used for communication. The SSL pins can be used as IO ports.

Although the SSL pins are not used for communication in clock synchronous operation, the internal operations within the modules are the same as those during SPI operation.

In both master and slave modes, communications can be performed with the same flows as the SPI operation except that mode fault error detection is not supported because the SSL pins are not used.

If the CPHA bit in the RSPI command register (SPCMD) is set in clock synchronous mode, operation cannot be guaranteed.

(1) Slave Mode Operation

(1-1) Starting a Serial Transfer

When the SPMS bit in the RSPI control register (SPCR) is 1, the first RSPCK edge triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register intact, in the full state.

When the SPMS bit is 1, the RSPI always drives the MISO output signal.

For details on the RSPI transfer format, see section 25.4.4, Transfer Format. Note that the SSL0 input signal is not used in clock synchronous operation.

(1-2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Irrespective of the value of the SPRF bit, upon termination of a serial transfer the RSPI changes the status of the shift register to "empty". The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. For details on the RSPI transfer format, see section 25.4.4, Transfer Format.

(1-3) Initialization Flowchart

Figure 25.26 shows an example of initialization flowchart for using the RSPI in slave mode during clock synchronous operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

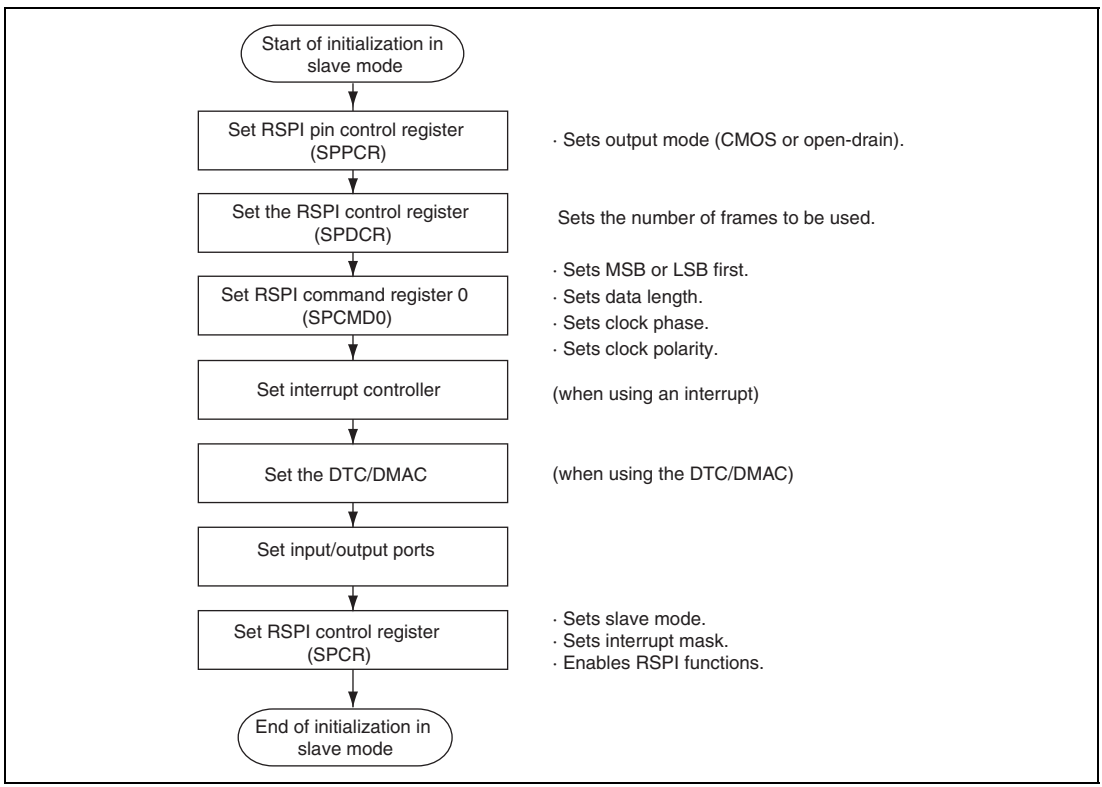


Figure 25.26 Example of Initialization Flowchart in Slave Mode

(1-4) Transfer Operation Flowchart (CPHA = 1)

Figure 25.27 shows an example of transfer operation flowchart for the RSPI during clock synchronous operation.

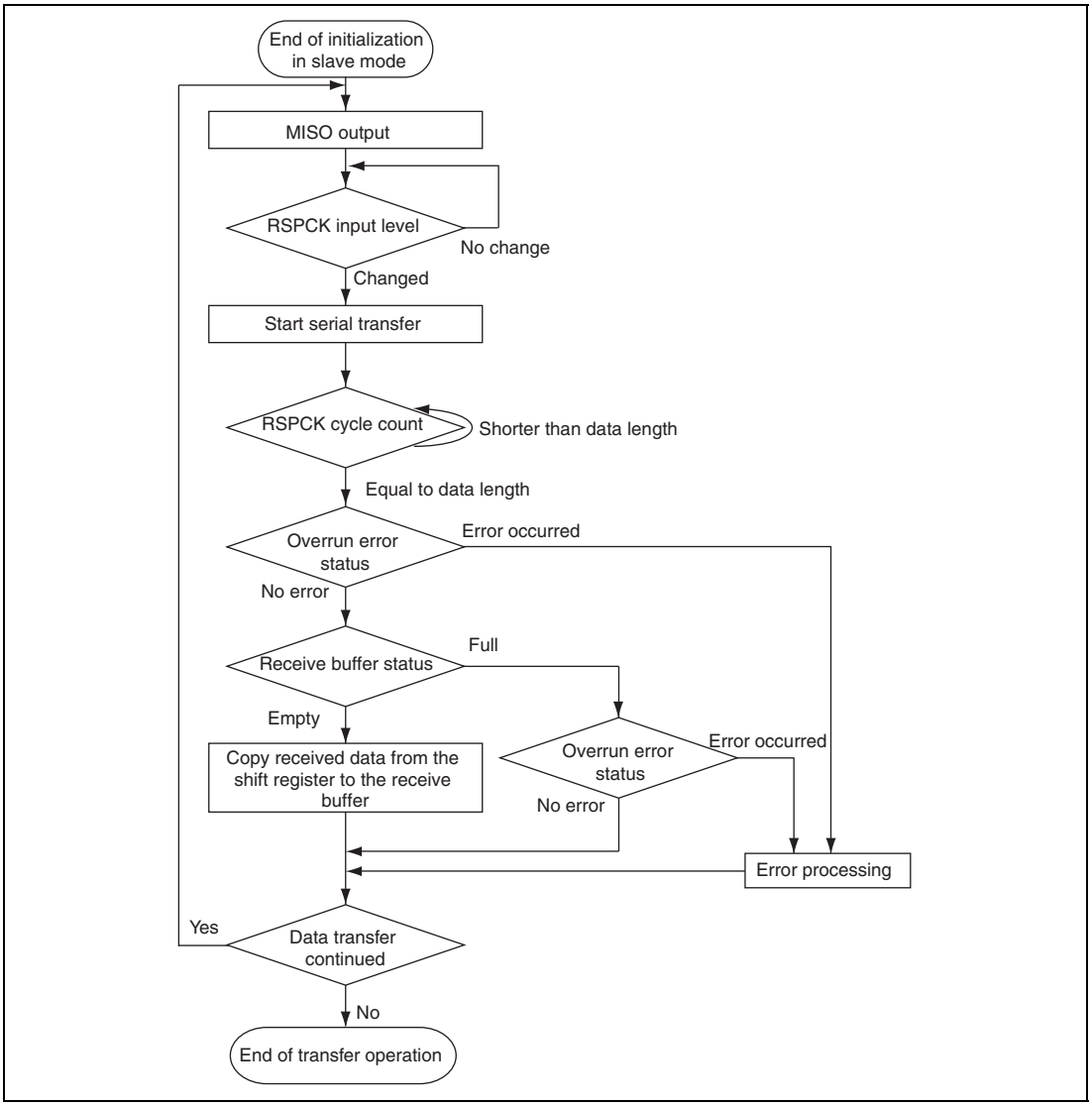


Figure 25.27 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

(2) Master Mode Operation

(2-1) Starting Serial Transfer

The RSPI updates the data in the transmit buffer when the SPTEF bit in the RSPI status register (SPSR) is 1 and when either the CPU or the DTC/DMAC has written data to the RSPI data register (SPDR). If the shift register is empty in a condition where the SPTEF bit has been cleared to 0 due to the writing of 0 either after the writing to SPDR from the DTC/DMAC or by the writing of 0 after the value 1 is read from the SPTEF bit by the CPU, the RSPI copies the data in the transmit buffer to the shift register and starts a 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 referred from the CPU.

For details on the RSPI transfer format, see section 25.4.4, Transfer Format. Note that the SSL0 output signal is not used for communication in clock synchronous operation.

(2-2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. If the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 in SPCMD. For details on the RSPI transfer format, see section 25.4.4, Transfer Format. Note that the SSL0 output signal is not used for communication in clock synchronous operation.

(2-3) Sequence Control

The transfer format that is employed in master mode is determined by the RSPI sequence control register (SPSCR), RSPI command registers 0 to 3 (SPCMD0 to SPCMD3), the RSPI bit rate register (SPBR), the RSPI clock delay register (SPCKD), the RSPI slave select negation delay register (SSLND), and the RSPI next-access delay register (SPND). Although no SSL signal is output in clock synchronous operation, these settings are valid.

The SPSCR register is used to determine the sequence configuration for serial transfers that are executed by a master mode RSPI. The following items are set in RSPI command registers SPCMD0 to SPCMD3: SSL output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be refererred to, whether SSLND is to be refererred to, and whether SPND is to be refererred to. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMD0 to SPCMD3. The RSPI contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP[1:0] in the RSPI sequence status register (SPSSR). When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

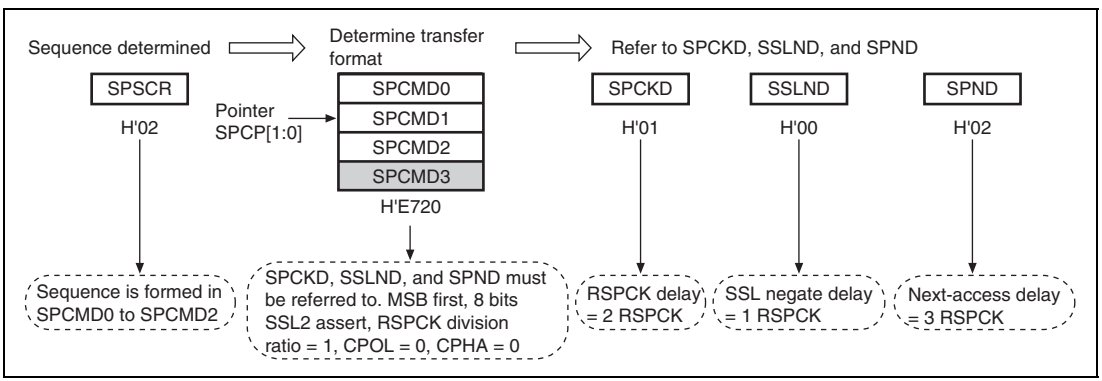


Figure 25.28 Determination Procedure of Serial Transfer Mode in Master Mode

(2-4) Initialization Flowchart

Figure 25.29 shows an example of initialization flowchart for using the RSPI in master mode during clock synchronous operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

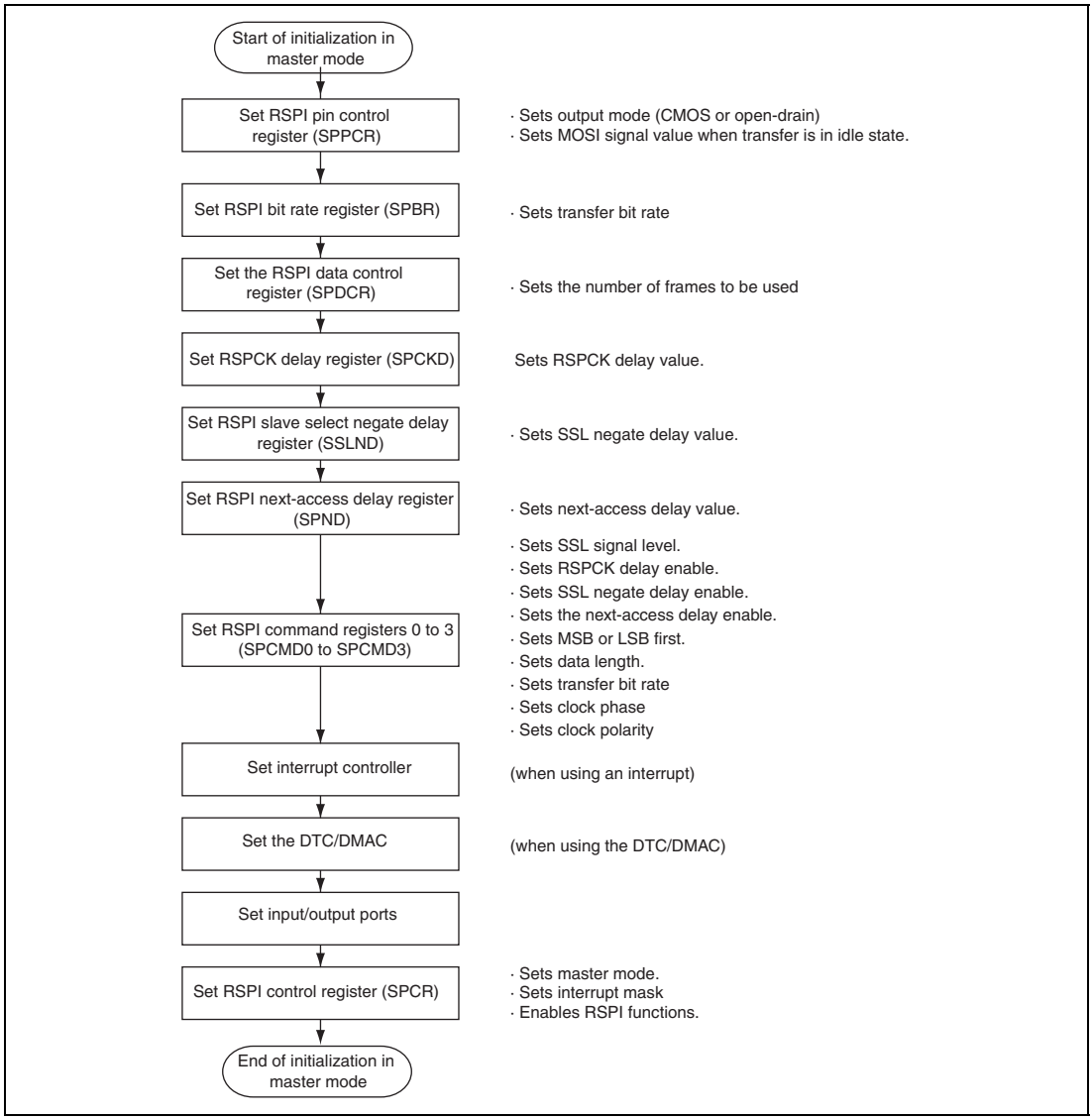


Figure 25.29 Example of Initialization Flowchart in Master Mode

(2-5) Transfer Operation Flowchart

Figure 25.30 shows an example of transfer operation flowchart in master mode during clock synchronous operation.

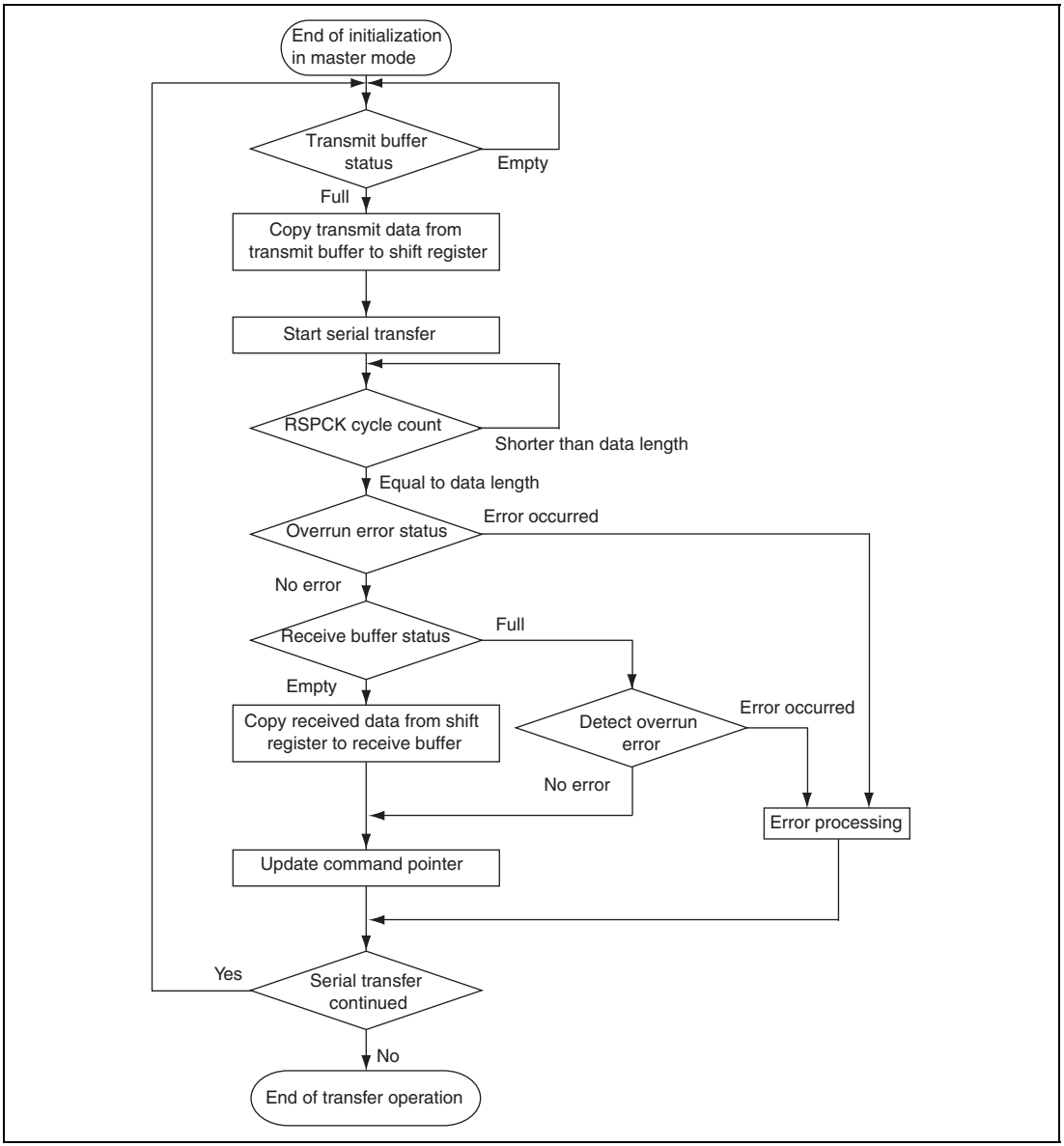


Figure 25.30 Example of Transfer Operation Flowchart in Master Mode

25.4.11 Error Processing

Figures 25.31 and 25.32 show error processing. The RSPI can recover from an error which may occur in master or slave mode, using the following error processing.

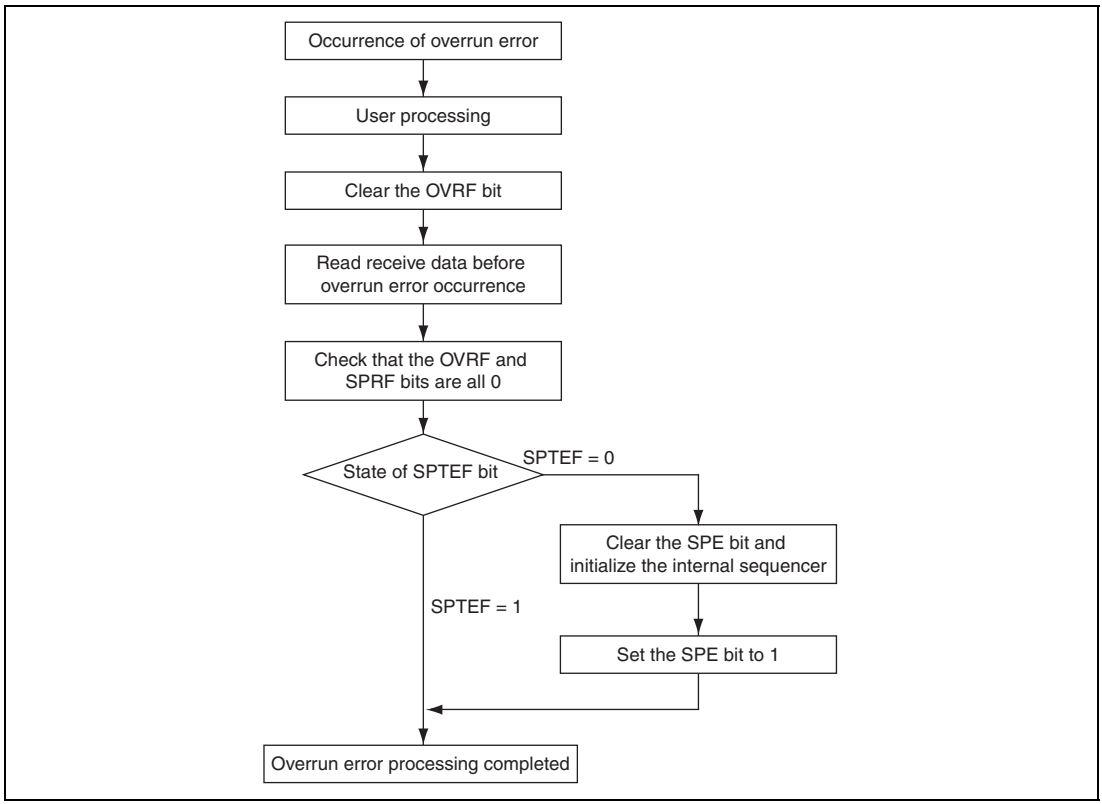


Figure 25.31 Error Processing (Overrun Error)

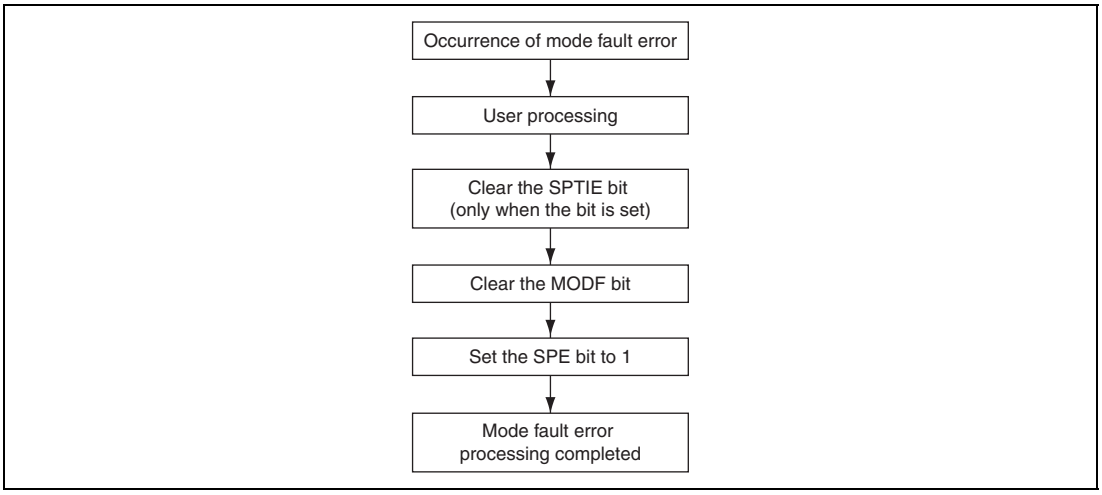


Figure 25.32 Error Processing (Mode Fault Error)

25.4.12 Loopback Mode

When the CPU writes 1 to the SPLP bit in the RSPI pin control register (SPPCR), the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects the input path and the output path (reversed) of the shift register. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI becomes the received data for the RSPI. Figure 25.33 shows the configuration of the shift register input/output paths for the case where the RSPI in master mode is set in loopback mode.

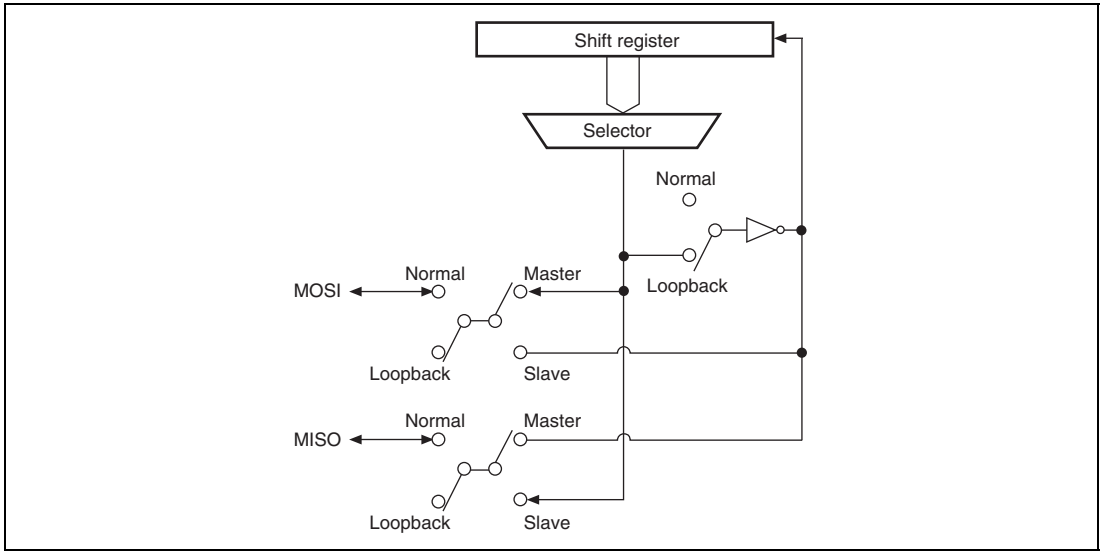


Figure 25.33 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

25.4.13 Interrupt Request

The interrupt sources for the RSPI include receive-buffer-full, transmission-buffer-empty, mode-fault, and overrun. With an interrupt request of receive-buffer-full or transmission-buffer-empty, the DTC or DMAC can start up and perform a data transfer.

The interrupt request of receive-buffer-full is allocated to the vector address of SPRXI, the interrupt request of transmission-buffer-empty is allocated to the vector address of SPTXI, and the interrupt requests of mode-fault and overrun are allocated to the vector address of SPEI. Therefore it is necessary to determine the interrupt source by the flag. Table 25.12 shows the interrupt sources for the RSPI.

When the interrupt condition is satisfied as shown in table 25.12, an interrupt occurs. Clear the interrupt source by executing a data transfer by the CPU or DTC/DMAC.

Table 25.12 RSPI Interrupt Sources

Name	Interrupt Source	Symbol	Interrupt Condition	DTC/DMAC Startup	Priority
SPEI	Mode-fault	MOI	(SPEIE=1) • (MODF=1)	—	High
	Overrun	OVI	(SPEIE=1) • (OVRF=1)	—	
SPRXI	Receive-buffer-full	RXI	(SPRIE=1) • (SPRF=1)	Startup	
SPTXI	Transmission-buffer-empty	TXI	(SPTIE=1) • (SPTEF=1)	Startup	Low

25.5 Usage Notes

25.5.1 Settings for Module Standby Mode

The RSPI can be made to operate or stop by the setting in the standby control register. The initial setting is for RSPI operation to be stopped. The registers of the module become accessible when it is released from module standby. Refer to section 32, Power-down Modes, for details.

25.5.2 DTC Block Transfer

To start a DTC block transfer due to SPRXI and SPTXI, set the block size in the DTC transfer count register (CRA) and the value in the block size counter to the same value as the number of frames set in the frame count setting bit. If these values are not the same, subsequent operations cannot be guaranteed.

25.5.3 DMAC Burst Transfer

To start a DMAC transfer due to SPRXI and SPTXI, set the value in the DMA transfer count register (DMATCR) to the same value as the number of frames set in the frame count setting bit. If these values are not the same, subsequent operations cannot be guaranteed.

25.5.4 Reading Receive Data

When reading the receive data by the CPU, clear the flag after the CPU reads the buffer for the specified number of times. If the flag is cleared before reaching the specified number of times, subsequent operations cannot be guaranteed.

25.5.5 DTC/DMAC and Mode Fault Error

If a mode fault error occurs when the SPTXI interrupt setting for DTC/DMAC is enabled while the SPTIE bit is valid, an unintended interrupt may occur. Clear the SPTIE bit while it is valid using the mode fault error processing (figure 25.31).

To use the DTC/DMAC after a mode fault error occurrence, reset the DTC/DMAC.

25.5.6 Usage of the RSPI Output Pins as Open Drain Outputs

When the RSPI output pins are to be used as open drain outputs, use a pull-up register to pull them up to the same electric potential as that on the PVcc1 pin.

Specify the pull-up resistance after enough evaluation to considerate whether the load satisfies the electrical characteristic requirements.

25.5.7 Usage with PVcc1 at 1.8 V

Ensure that the PJDRV bit in register DRVCR is 0 if the RSPI is to be used with PVcc1 at 1.8 V. Normal transfer may become impossible if the RSPI is used with PVcc1 at 1.8 V while the value of the PJDRV bit is 1.

Section 26 Controller Area Network (RCAN-ET)

26.1 Summary

26.1.1 Overview

This document primarily describes the programming interface for the RCAN-ET module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-ET implementation can ensure the design is successful.

26.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-ET module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

26.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the RCAN-ET user interface LSI engineers must use this document to understand the hardware requirements.

26.1.4 References

1. CAN Licence Specification, Robert Bosch GmbH, 1992
2. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
3. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
4. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
5. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)

26.1.5 Features

- supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 16 Mailbox version
- Clock 20 to 50 MHz
- 15 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- programmable CAN data rate up to 1MBit/s
- transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- data buffer access without SW handshake requirement in reception
- flexible micro-controller interface
- flexible interrupt structure

26.2 Architecture

The RCAN-ET device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control and CAN Interface. The figure below shows the block diagram of the RCAN-ET Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

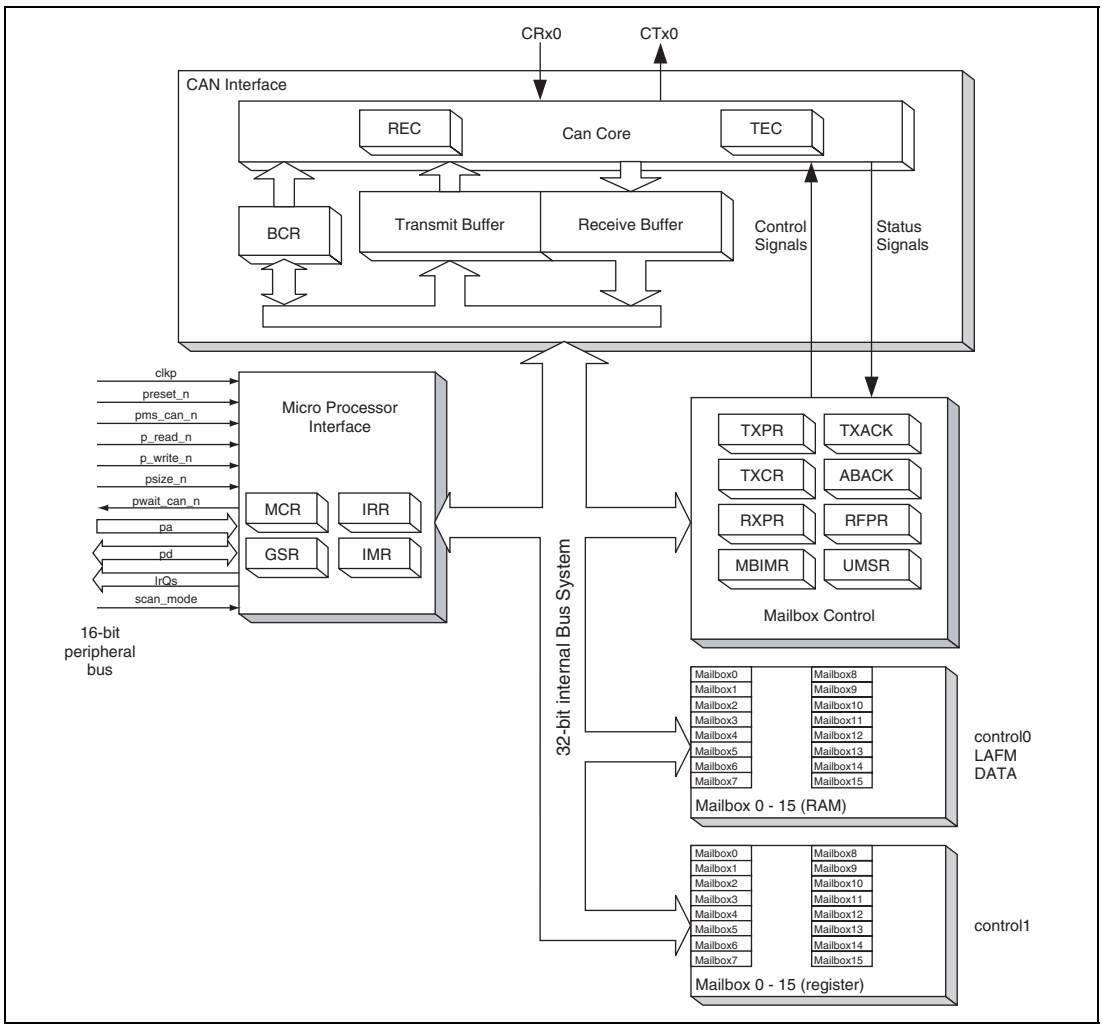


Figure 26.1 RCAN-ET Architecture

Important: Although core of RCAN-ET is designed based on a 32-bit bus system, the whole RCAN-ET including MPI for the CPU has 16-bit bus interface to CPU. In that case, LongWord (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this manual, LongWord access means the two consecutive accesses.

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and RCAN-ET's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-ET so that the RCAN-ET can automatically exit the Sleep mode.

  It contains registers such as MCR, IRR, GSR and IMR.

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There are 16 Mailboxes, and each mailbox has the following information.

  <RAM>

  — CAN message control (identifier, rtr, ide,etc)

  — CAN message data (for CAN Data frames)

  — Local Acceptance Filter Mask for reception

  <Registers>

  — CAN message control (dlc)

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

- Mailbox Control

  The Mailbox Control handles the following functions:

  — For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

  — To transmit messages, RCAN-ET will run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

- CAN Interface

  This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the

CAN Data Link Controller.

26.3 Programming Model - Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by the RCAN-ET IP. Different use of RCAN-ET is not allowed.

26.3.1 Memory Map

The diagram of the memory map is shown below.

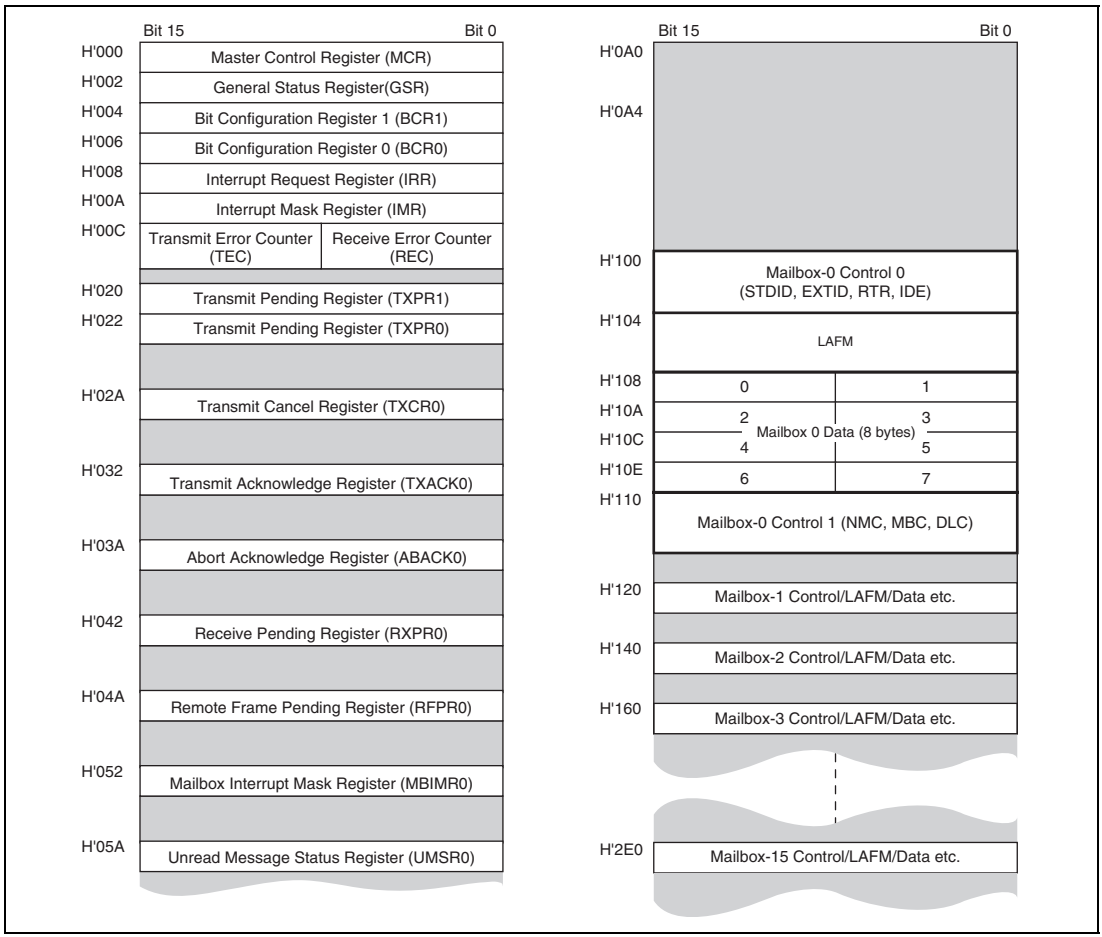


Figure 26.2 RCAN-ET Memory Map

The locations not used (between H'000 and H'2F2) are reserved and cannot be accessed.

26.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit / receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. The following table shows the address map for the control, LAFM, data and addresses for each mailbox.

	Address			
	Control0	**LAFM**	**Data**	**Control1**
Mailbox	**4 bytes**	**4 bytes**	**8 bytes**	**2 bytes**
0 (Receive Only)	100 – 103	104– 107	108 – 10F	110 – 111
1	120 – 123	124 – 127	128 – 12F	130 – 131
2	140 – 143	144 – 147	148 – 14F	150 – 151
3	160 – 163	164 - 167	168 – 16F	170 – 171
4	180 – 183	184 – 187	188 – 18F	190 – 191
5	1A0 – 1A3	1A4 – 1A7	1A8 – 1AF	1B0 – 1B1
6	1C0 – 1C3	1C4 – 1C7	1C8 – 1CF	1D0 – 1D1
7	1E0 – 1E3	1E4 – 1E7	1E8 – 1EF	1F0 – 1F1
8	200 – 203	204 – 207	208 – 20F	210 – 211
9	220 – 223	224 – 227	228 – 22F	230 – 231
10	240 – 243	244 – 247	248 – 24F	250 – 251
11	260 – 263	264 – 267	268 – 26F	270 – 271
12	280 – 283	284 – 287	288 – 28F	290 – 291
13	2A0 – 2A3	2A4 – 2A7	2A8 – 2AF	2B0 – 2B1
14	2C0 – 2C3	2C4 – 2C7	2C8 – 2CF	2D0 – 2D1
15	2E0 – 2E3	2E4 – 2E7	2E8 – 2EF	2F0 – 2F1

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

Table 26.1 Roles of Mailboxes

	Tx	Rx
MB15-1	OK	OK
MB0	—	OK

MB0 (reception MB)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord): 32-bit access

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

MBC[1] is fixed to "1"

MB15-1 (MB for transmission/reception)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

Figure 26.3 Mailbox-N Structure

Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be '0' and should not be relied upon.

2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.

3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

(1) Message Control Field

STDID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXTID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC=001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt), however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: In order to support automatic answer to remote frame when MBC=001(bin) is used and ATX=1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit) : Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

- Mailbox-0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: MBC[1] of MB0 is always "1".

- Mailbox-15 to 1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

NMC (New Message Control): When this bit is set to ‘0’, the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to ‘1’, the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

Important: Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

NMC	Description
0	Overrun mode (Initial value)
1	Overwrite mode

ATX (Automatic Transmission of Data Frame): When this bit is set to ‘1’ and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be ‘001’ (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

Important: When ATX is used and MBC=001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related RFPR flag is cleared by the CPU when the UMSR flag is set. In such case RFPR flag would get set again.

ATX	Description
0	Automatic Transmission of Data Frame disabled (Initial value)
1	Automatic Transmission of Data Frame enabled

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to ‘0’, RCAN-ET tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

DART	Description
0	Re-transmission enabled (Initial value)
1	Re-Transmission disabled

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC=111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC=’110’, ‘101’ and ‘100’ settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception. There is no hardware protection, and TXPR remains set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX* • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	Setting prohibited				
1	0	1	Setting prohibited				
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive (Initial value)				

Notes: * In order to support automatic retransmission, RTR shall be "0" when MBC=001(bin) and ATX=1.
When ATX=1 is used the filter for IDE must not be used

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 bytes (Initial value)
0	0	0	1	Data Length = 1 byte
0	0	1	0	Data Length = 2 bytes
0	0	1	1	Data Length = 3 bytes
0	1	0	0	Data Length = 4 bytes
0	1	0	1	Data Length = 5 bytes
0	1	1	0	Data Length = 6 bytes
0	1	1	1	Data Length = 7 bytes
1	x	x	x	Data Length = 8 bytes

(2) Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes.

LAFM: When MBC is set to 001, 010, 011 (Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

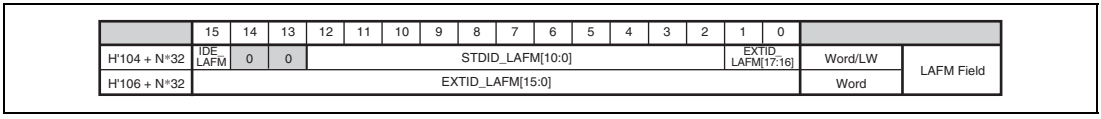


Figure 26.4 Acceptance Filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-ET searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

Important: RCAN-ET starts to find a matching identifier from Mailbox-15 down to Mailbox-0. As soon as RCAN-ET finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

STD_LAFM[10:0] — Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding STD_ID bit is cared
1	Corresponding STD_ID bit is "don't cared"

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE_ID bit is cared
1	Corresponding IDE_ID bit is "don't cared"

(3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

26.3.3 RCAN-ET Control Registers

The following sections describe RCAN-ET control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
Master Control Register	000	MCR	Word
General Status Register	002	GSR	Word
Bit Configuration Register 1	004	BCR1	Word
Bit Configuration Register 0	006	BCR0	Word
Interrupt Request Register	008	IRR	Word
Interrupt Mask Register	00A	IMR	Word
Error Counter Register	00C	TEC/REC	Word

Figure 26.5 RCAN-ET Control Registers

(1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-ET.

- MCR (Address = H'000)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	-	-	-	TST[2:0]			MCR7	MCR6	MCR5	-	-	MCR2	MCR1	MCR0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit 15 — ID Reorder (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

Bit15 : MCR15	Description
0	RCAN-ET is the same as HCAN2
1	RCAN-ET is not the same as HCAN2 (Initial value)

MCR15 (ID Reorder) = 0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	0	STDID[10:0]											RTR	IDE	EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	0	STDID_LAFM[10:0]											0	IDE_LAFM	EXTID_LAFM [17:16]		Word/LW	LAFM Field
H'106 + N*32	EXTID_LAFM[15:0]																Word	

MCR15 (ID Reorder) = 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM [17:16]		Word/LW	LAFM Field
H'106 + N*32	EXTID_LAFM[15:0]																Word	

Figure 26.6 ID Reorder

This bit can be modified only in reset mode.

Bit 14 — Auto Halt Bus Off (MCR14): If both this bit and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

Bit14 : MCR14	Description
0	RCAN-ET remains in BusOff for normal recovery sequence (128 x 11 Recessive Bits) (Initial value)
1	RCAN-ET moves directly into Halt Mode after it enters BusOff if MCR6 is set.

This bit can be modified only in reset mode.

Bit 13 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 12 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 11 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 10 - 8 — Test Mode (TST[2:0]): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move RCAN-ET into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 26.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-ET is used in normal operation.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	setting prohibited
1	1	1	setting prohibited

Bit 7 — Auto-wake Mode (MCR7): MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-ET automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-ET does not automatically cancel the sleep mode.

RCAN-ET cannot store the message that wakes it up.

Note: MCR7 cannot be modified while in sleep mode.

Bit7 : MCR7	Description
0	Auto-wake by CAN bus activity disabled (Initial value)
1	Auto-wake by CAN bus activity enabled

Bit 6 — Halt during Bus Off (MCR6): MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

Bit6 : MCR6	Description
0	If MCR[1] is set, RCAN-ET will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value)
1	Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.

Bit 5 — Sleep Mode (MCR5): Enables or disables Sleep mode transition. If this bit is set, while RCAN-ET is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position,
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-ET will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-ET will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-ET will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

Important: RCAN-ET is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-ET must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5]=1 and MCR[1]=0 at the same time).

Bit 5 : MCR5	Description
0	RCAN-ET sleep mode released (Initial value)
1	Transition to RCAN-ET sleep mode enabled

Bit 4 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 2 — Message Transmission Priority (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-15 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission).

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE=1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

Bit 2 : MCR2	Description
0	Transmission order determined by message identifier priority (Initial value)
1	Transmission order determined by mailbox number priority (Mailbox-15 → Mailbox-1)

Bit 1—Halt Request (MCR1): Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-ET remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-ET will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

In the Halt mode, the RCAN-ET configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a ‘0’ in order to re-join the CAN bus. After this bit has been cleared, RCAN-ET waits until it detects 11 recessive bits, and then joins the CAN bus.

Note: After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).

Note: Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

Bit 1 : MCR1	Description
0	Clear Halt request (Initial value)
1	Halt mode transition request

Bit 0 — Reset Request (MCR0): Controls resetting of the RCAN-ET module. When this bit is changed from ‘0’ to ‘1’ the RCAN-ET controller enters its reset routine, re-initialising the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all user registers are initialised.

RCAN-ET can be re-configured while this bit is set. This bit has to be cleared by writing a ‘0’ to join the CAN bus. After this bit is cleared, the RCAN-ET module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-ET needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

Bit 0 : MCR0	Description
0	Clear Reset Request
1	CAN Interface reset mode transition request (Initial value)

(2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-ET.

- GSR (Address = H'002)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-ET enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

Bit 5 : GSR5	Description
0	RCAN-ET is not in Error Passive or in Bus Off status (Initial value) [Reset condition] RCAN-ET is in Error Active state
1	RCAN-ET is in Error Passive (if GSR0=0) or Bus Off (if GSR0=1) [Setting condition] When TEC ≥ 128 or REC ≥ 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN-ET IP. RCAN-ET exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4 : GSR4	Description
0	RCAN-ET is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1=1) or Sleep mode (if MCR5=1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and RCAN-ET is in the halt mode or RCAN-ET is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-ET is in the reset state or not.

Bit 3 : GSR3	Description
0	RCAN-ET is not in the reset state
1	Reset state (Initial value) [Setting condition] After an RCAN-ET internal reset (due to SW or HW reset)

Bit 2 — Message Transmission in progress Flag (GSR2): Flag that indicates to the CPU if the RCAN-ET is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

Bit 2 : GSR2	Description
0	RCAN-ET is in Bus Off or a transmission is in progress
1	[Setting condition] Not in Bus Off and no transmission in progress (Initial value)

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

Bit 1 : GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC < 256 or 96 ≤ REC < 256

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that RCAN-ET is in the bus off state.

Bit 0 : GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC ≥ 256 (bus off state)

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is equivalent to GSR0.

(3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral bus frequency.

- BCR1 (Address = H'004)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				-	TSG2[2:0]			-	-	SJW[1:0]		-	-	-	BSP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bits 15 to 12 — Time Segment 1 (TSG1[3:0] = BCR1[15:12]): These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: TSG1[3]	Bit 14: TSG1[2]	Bit 13: TSG1[1]	Bit 12: TSG1[0]	Description
0	0	0	0	Setting prohibited (Initial value)
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	PRSEG + PHSEG1 = 4 time quanta
0	1	0	0	PRSEG + PHSEG1 = 5 time quanta
:	:	:	:	:
:	:	:	:	:
1	1	1	1	PRSEG + PHSEG1 = 16 time quanta

Bit 11 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 10 to 8 — Time Segment 2 (TSG2[2:0] = BCR1[10:8]): These bits are used to set the segment TSEG2 (=PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

Bit 10: TSG2[2]	Bit 9: TSG2[1]	Bit 8: TSG2[0]	Description
0	0	0	Setting prohibited (Initial value)
0	0	1	PHSEG2 = 2 time quanta (conditionally prohibited)
0	1	0	PHSEG2 = 3 time quanta
0	1	1	PHSEG2 = 4 time quanta
1	0	0	PHSEG2 = 5 time quanta
1	0	1	PHSEG2 = 6 time quanta
1	1	0	PHSEG2 = 7 time quanta
1	1	1	PHSEG2 = 8 time quanta

Bits 7 and 6 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

Bit 5: SJW[1]	Bit 4: SJW[0]	Description
0	0	Synchronisation Jump width = 1 time quantum (Initial value)
0	1	Synchronisation Jump width = 2 time quanta
1	0	Synchronisation Jump width = 3 time quanta
1	1	Synchronisation Jump width = 4 time quanta

Bits 3 to 1 : Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled.

Bit 0 : BSP	Description
0	Bit sampling at one point (end of time segment 1) (Initial value)
1	Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1)

- BCR0 (Address = H'006)

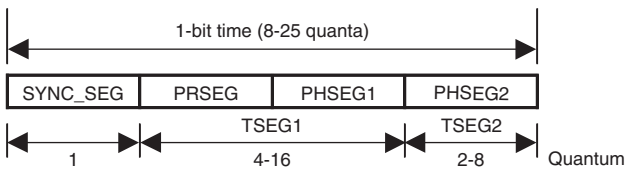
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 8 to 15 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 X peripheral bus clock (Initial value)
0	0	0	0	0	0	0	1	4 X peripheral bus clock
0	0	0	0	0	0	1	0	6 X peripheral bus clock
: :	: :	: :	: :	: :	: :	: :	: :	2*(register value+1) X peripheral bus clock
1	1	1	1	1	1	1	1	512 X peripheral bus clock

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation (resynchronisation) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronisation (resynchronisation) is established)

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

The RCAN-ET Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 * (BRP + 1) * (TSEG1 + TSEG2 + 1)}$$

where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+ 1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

f_{CLK} = Peripheral Clock

BCR Setting Constraints

TSEG1min > TSEG2 ≥ SJWmax (SJW = 1 to 4)

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	No	1-3	No	No	No	No	No	
0100	5	1-2	1-3	1-4	No	No	No	No	
0101	6	1-2	1-3	1-4	1-4	No	No	No	
0110	7	1-2	1-3	1-4	1-4	1-4	No	No	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	No	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

Example 1: To have a Bit rate of 500 Kbps with a frequency of fclk = 40 MHz it is possible to set: BRP = 3, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = 5200 and BCR0 = 0003.

Example 2: To have a Bit rate of 250 Kps with a frequency of 35 MHz it is possible to set: BPR = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

(4) Interrupt Request Register (IRR)

The interrupt register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	IRR13	IRR12	-	-	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bits 15 to 14: Reserved.

Bit 13 - Message Error Interrupt (IRR13): this interrupt indicates that:

- A message error has occurred when in test mode.
- Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set. When not in test mode this interrupt is inactive.

Bit 13: IRR13	Description
0	message error has not occurred in test mode (Initial value) [Clearing condition] Writing 1
1	[Setting condition] message error has occurred in test mode

Bit 12 – Bus activity while in sleep mode (IRR12): IRR12 indicates that a CAN bus activity is present. While the RCAN-ET is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

Bit 12: IRR12	Description
0	bus idle state (Initial value) [Clearing condition] Writing 1
1	[Setting condition] dominant bit level detection on the Rx line while in sleep mode

Bits 11 to 10: Reserved

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to ‘1’ and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing ‘1’) or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set . It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 9: IRR9	Description
0	No pending notification of message overrun/overwrite [Clearing condition] Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)
1	A receive message has been discarded due to overrun condition or a message has been overwritten [Setting condition] Message is received while the corresponding RXPR and/or RFPR =1 and MBIMR =0

Bit 8 - Mailbox Empty Interrupt Flag (IRR8): This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). The related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In effect, this bit is set by an OR’ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 8: IRR8	Description
0	Messages set for transmission or transmission cancellation request NOT progressed. (Initial value) [Clearing Condition] All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set
1	Message has been transmitted or aborted, and new message can be stored [Setting condition] When one of the TXPR bits is cleared by completion of transmission or completion of transmission abort, i.e., when a TXACK or ABACK bit is set (if MBIMR=0).

Bit 7 — Overload Frame (IRR7): Flag indicating that the RCAN-ET has detected a condition that should initiate the transmission of an overload frame. Note that on the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

Bit 7: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	[Setting conditions] Overload condition detected

Bit 6 — Bus Off Interrupt Flag (IRR6): This bit is set when RCAN-ET enters the Bus-off state or when RCAN-ET leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-ET node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-ET is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

Bit 6: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Enter Bus off state caused by transmit error or Error Active state returning from Bus-off [Setting condition] When TEC becomes ≥ 256 or End of Bus-off after 128X11 consecutive recessive bits or transition from Bus Off to Halt

Bit 5 — Error Passive Interrupt Flag (IRR5): Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-ET is in Error Passive or Bus Off status.

Bit 5: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128 or Error Passive test mode is used

Bit 4 — Receive Error Counter Warning Interrupt Flag (IRR4): This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when RCAN-ET is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 4: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96 and RCAN-ET is not in Bus Off

Bit 3 — Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 3: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 2 — Remote Frame Request Interrupt Flag (IRR2): flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 2: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (Initial value)
1	at least one remote request is pending [Setting condition] When remote frame is received and the corresponding MBIMR = 0

Bit 1 — Data Frame Received Interrupt Flag (IRR1): IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 1: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (Initial value)
1	Data frame received and stored in Mailbox [Setting condition] When data is received and the corresponding MBIMR = 0

Bit 0 — Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

1. Reset mode has been entered after a SW (MCR0) or HW reset
2. Halt mode has been entered after a Halt request (MCR1)
3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state RCAN-ET is in.

Important : When a Sleep mode request needs to be made, the Halt mode must be used beforehand. Please refer to the MCR5 description and figure 26.9.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-ET enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time * 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialisation.

Bit 0: IRR0	Description
0	[Clearing condition] Writing 1
1	Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value) [Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

(5) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0: Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

Bit[15:0]: IMRn	Description
0	Corresponding IRR is not masked (IRQ is generated for interrupt conditions)
1	Corresponding interrupt of IRR is masked (Initial value)

(6) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-ET needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100.
REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

26.3.4 RCAN-ET Mailbox Registers

The following sections describe RCAN-ET Mailbox registers that control / flag individual Mailboxes. The address is mapped as follows.

Important : LongWord access is carried out as two consecutive Word accesses.

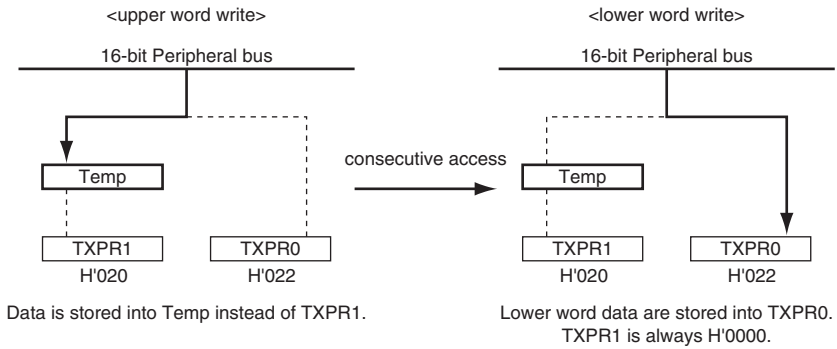
Description	Address	Name	Access Size (bits)
Transmit Pending 1	H'020	TXPR1	LW
Transmit Pending 0	H'022	TXPR0	—
	H'024		
	H'026		
	H'028		
Transmit Cancel 0	H'02A	TXCR0	
	H'02C		
	H'02E		
	H'030		
Transmit Acknowledge 0	H'032	TXACK0	Word
	H'034		
	H'036		
	H'038		
Abort Acknowledge 0	H'03A	ABACK0	Word
	H'03C		
	H'03E		
	H'040		
Data Frame Receive Pending 0	H'042	RXPR0	Word
	H'044		
	H'046		
	H'048		
Remote Frame Receive Pending 0	H'04A	RFPR0	Word
	H'04C		
	H'04E		
	H'050		
Mailbox Interrupt Mask Register 0	H'052	MBIMR0	Word
	H'054		
	H'056		
	H'058		
Unread message Status Register 0	H'05A	UMSR0	Word
	H'05C		
	H'05E		

Figure 26.7 RCAN-ET Mailbox Registers

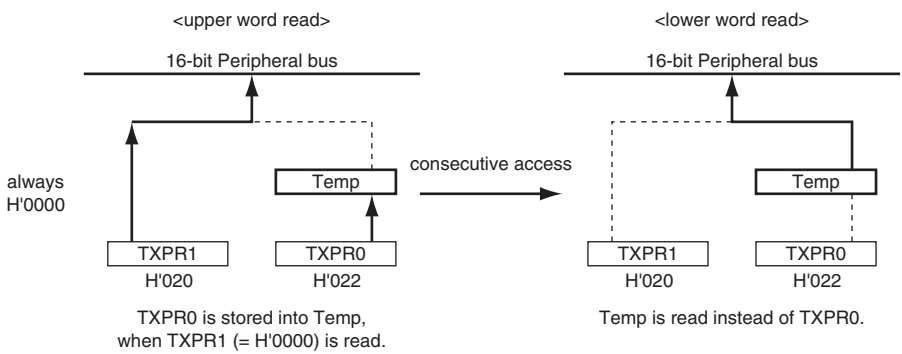
(1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

<Longword Write Operation>



<Longword Read Operation>



The TXPR1 register cannot be modified and it is always fixed to ‘0’. The TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a ‘0’ and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

The RCAN-ET will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-ET automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-ET shall ensure that in the identifier priority scheme (MCR2=0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to section 26.4, Application Note.

When the RCAN-ET changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Any write operation is ignored.

Read value is always H'0000. Long word access is mandatory when reading or writing TXPR1/TXPR0. Writing any value to TXPR1 is allowed, however, write operation to TXPR1 has no effect.

- TXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * it is possible only to write a ‘1’ for a Mailbox configured as transmitter.

Bit 15 to 1 — indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:1]:TXPR0	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

Bit 0— Reserved: This bit is always ‘0’ as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

(2) Transmit Cancel Register (TXCR0)

TXCR0 is a 16-bit read / conditionally-write registers. The TXCR0 controls Mailbox-15 to Mailbox-1. This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * Only writing a ‘1’ to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 1 — requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

Bit[15:1]:TXCR0	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(3) Transmit Acknowledge Register (TXACK0)

The TXACK0 is a 16-bit read / conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-ET sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 1 — notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:TXACK0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(4) Abort Acknowledge Register (ABACK0)

The ABACK0 is a 16-bit read / conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-ET sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-ET to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 1 — notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:ABACK0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(5) Data Frame Receive Pending Register (RXPR0)

The RXPR0 is a 16-bit read / conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

Bit[15:0]: RXPR0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

(6) Remote Frame Receive Pending Register (RFPR0)

The RFPR0 is a 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Request Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 15 to 0 respectively.

Bit[15:0]: RFPR0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

(7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Request Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

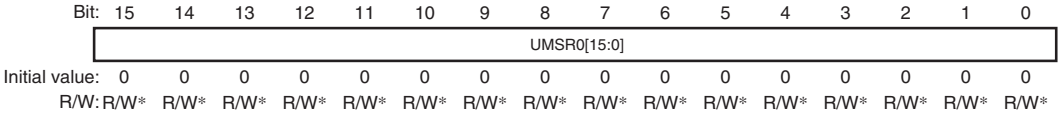
Bit[15:0]: MBIMR0	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

(8) Unread Message Status Register (UMSR)

This register is a 16-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR0



Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

Bit[15:0]: UMSR0	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

26.4 Application Note

26.4.1 Test Mode Settings

The RCAN-ET has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-ET test mode. The default (initialised) settings allow RCAN-ET to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Normal Mode: RCAN-ET operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the Tx Output is disabled so that RCAN-ET does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The Rx/Tx pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The Rx/Tx pins do not need to be connected to the CAN bus or any external devices, as the internal Tx is looped back to the internal Rx. Tx pin outputs only recessive bits and Rx pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-ET can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-ET can be forced to become an Error Warning by writing a value greater than 95 into them.

RCAN-ET needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.Error Passive Mode: RCAN-ET can be forced to enter Error Passive mode.

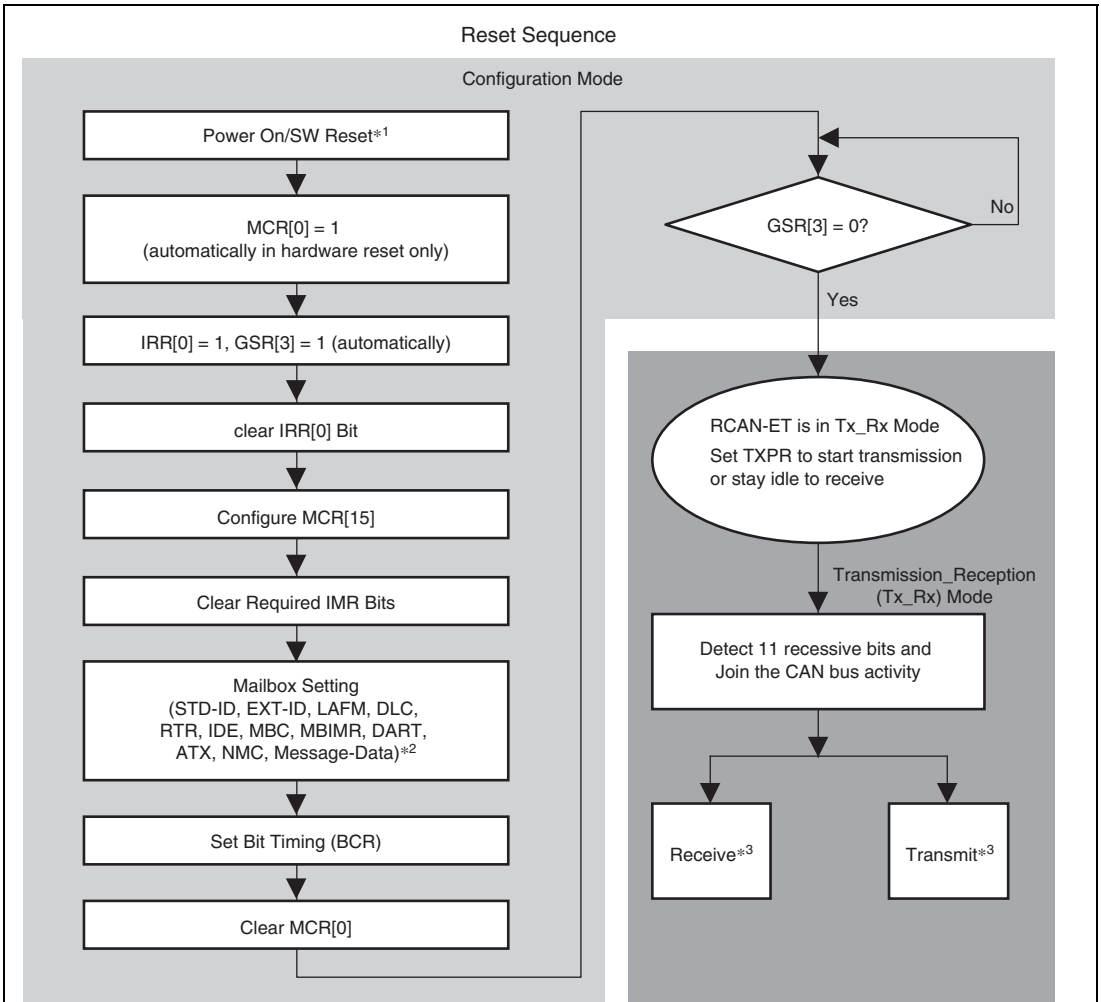
Note: the REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, RCAN-ET will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-ET will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, RCAN-ET will move to Error Passive and not to Error Active

When message error occurs, IRR13 is set in all test modes.

26.4.2 Configuration of RCAN-ET

RCAN-ET is considered in configuration mode or after a H/W (Power On Reset)/ S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-ET cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

- After a Reset request
  The following sequence must be implemented to configure the RCAN-ET after (S/W or H/W) reset. After reset, all the registers are initialised, therefore, RCAN-ET needs to be configured before joining the CAN bus activity. Please read the notes carefully.



Notes: 1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialise all the mailboxes enabled by MBC.
3. If there is no TXPR set, RCAN-ET will receive the next incoming message.
If there is a TXPR(s) set, RCAN-ET will start transmission of the message and will be arbitrated by the CAN bus.
If it loses the arbitration, it will become a receiver.

Figure 26.8 Reset Sequence

- Halt mode

  When RCAN-ET is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-ET to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-ET transit to Halt Mode, GSR4 is set.

  Once the configuration is completed the Halt request needs to be released. RCAN-ET will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

  When RCAN-ET is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

  The following diagram shows the flow to follow to move RCAN-ET into sleep mode.

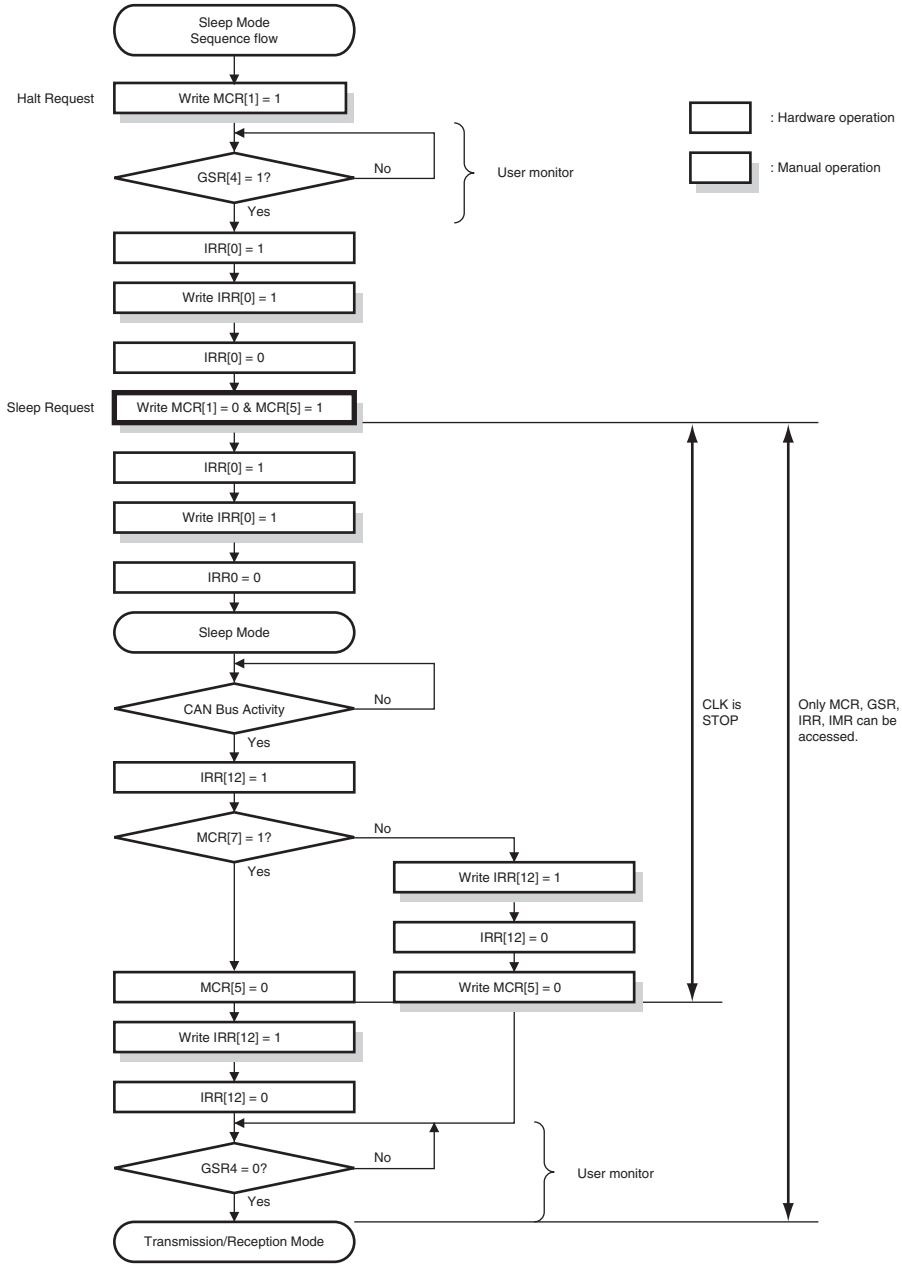

Sleep Mode Sequence flow
Halt Request
Write MCR[1] = 1
GSR[4] = 1?
No
Yes
User monitor
: Hardware operation
: Manual operation
IRR[0] = 1
Write IRR[0] = 1
IRR[0] = 0
Sleep Request
Write MCR[1] = 0 & MCR[5] = 1
IRR[0] = 1
Write IRR[0] = 1
IRR0 = 0
Sleep Mode
CAN Bus Activity
No
Yes
CLK is STOP
Only MCR, GSR, IRR, IMR can be accessed.
IRR[12] = 1
MCR[7] = 1?
No
Yes
Write IRR[12] = 1
IRR[12] = 0
MCR[5] = 0
Write MCR[5] = 0
Write IRR[12] = 1
IRR[12] = 0
GSR4 = 0?
No
Yes
User monitor
Transmission/Reception Mode

Figure 26.9 - Halt Mode / Sleep Mode shows allowed state transition.

— Please don't set MCR5 (Sleep Mode) without entering Halt Mode.
— After MCR1 is set, please don't clear it before GSR4 is set and RCAN-ET enters Halt Mode.

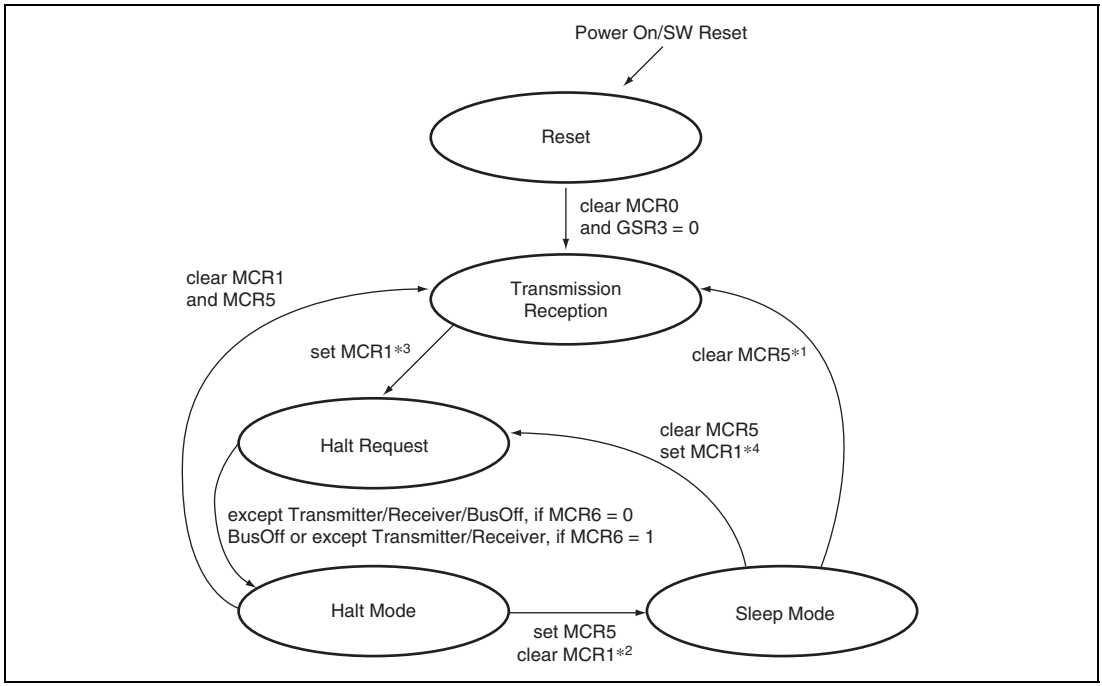


Figure 26.9 Halt Mode / Sleep Mode

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing "0"
2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.
3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-ET moves to Bus Off and MCR14 and MCR6 are both set.
4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-ET moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

	RCAN-ET Registers								
Status Mode	**MCR GSR**	**IRR IMR**	**BCR**	**MBIMR**	**Flag_register**	**mailbox (ctrl0, LAFM)**		**mailbox (data)**	**mailbox (ctrl1)**	
Reset	yes	yes	yes	yes	yes	yes		yes	yes	
Transmission Reception Halt Request	yes	yes	no*1	yes	yes	no*1	yes*2	yes*2	no*1	yes*2
Halt	yes	yes	no*1	yes	yes	yes		yes	yes	
Sleep	yes	yes	no	no	no	no		no	no	

Notes: 1. No hardware protection
2. When TXPR is not set.

26.4.3 Message Transmission Sequence

- Message Transmission Request

  The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

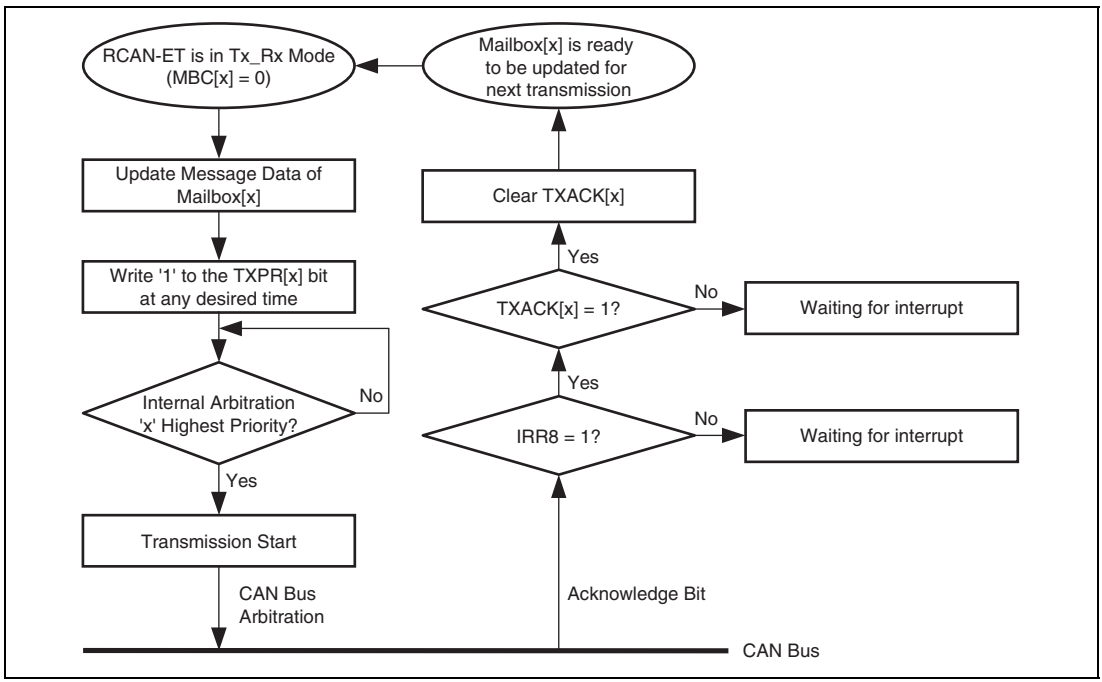


Figure 26.10 Transmission Request

- Internal Arbitration for transmission

  The following diagram explains how RCAN-ET manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

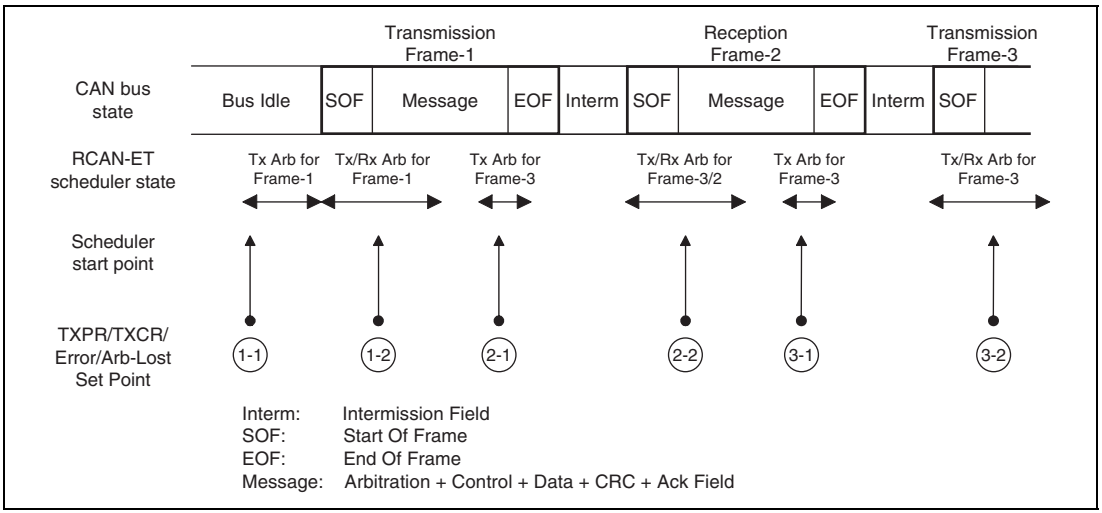


Figure 26.11 Internal Arbitration for Transmission

The RCAN-ET has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-ET becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-ET becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-ET becomes transmitter.

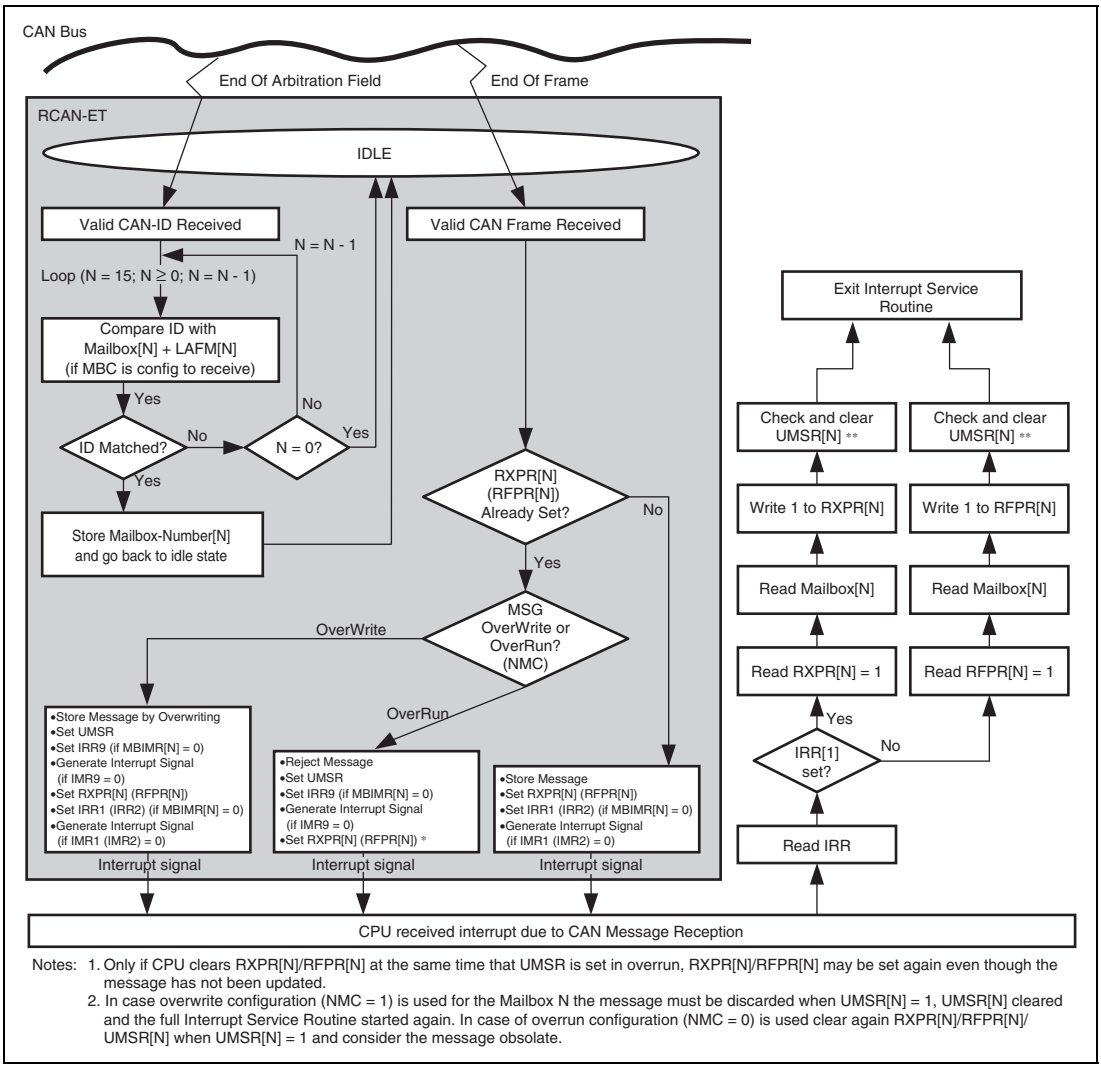
Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX=1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

26.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Notes: 1. Only if CPU clears RXPR[N]/RFPR[N] at the same time that UMSR is set in overrun, RXPR[N]/RFPR[N] may be set again even though the message has not been updated.
2. In case overwrite configuration (NMC = 1) is used for the Mailbox N the message must be discarded when UMSR[N] = 1, UMSR[N] cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/RFPR[N]/ UMSR[N] when UMSR[N] = 1 and consider the message obsolete.

Figure 26.12 Message Receive Sequence

When RCAN-ET recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-15 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-15 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-14 (if configured as receive). Once RCAN-ET finds a matching identifier, it stores the number of Mailbox-[N] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus. Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame request interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

In the Overrun Mode (NMC = ’0’), only the first Mailbox will cause the flags to be asserted. So, if a Data Frame is initially received, then RXPR and IRR1 are both asserted. If a Remote Frame is then received before the Data Frame has been read, then RFPR and IRR2 are NOT set. In this case UMSR of the corresponding Mailbox will still be set.

26.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box

  Two cases are possible.

  — Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART

  This change is possible only when MBC=3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.

  — Change from transmit to receive configuration (MBC)

  Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-ET to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt state.

  In case RCAN-ET is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box

  The configuration can be changed only in Halt Mode.

  RCAN-ET will not lose a message if the message is currently on the CAN bus and RCAN-ET is a receiver. RCAN-ET will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-ET is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt Mode.

  In case RCAN-ET is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

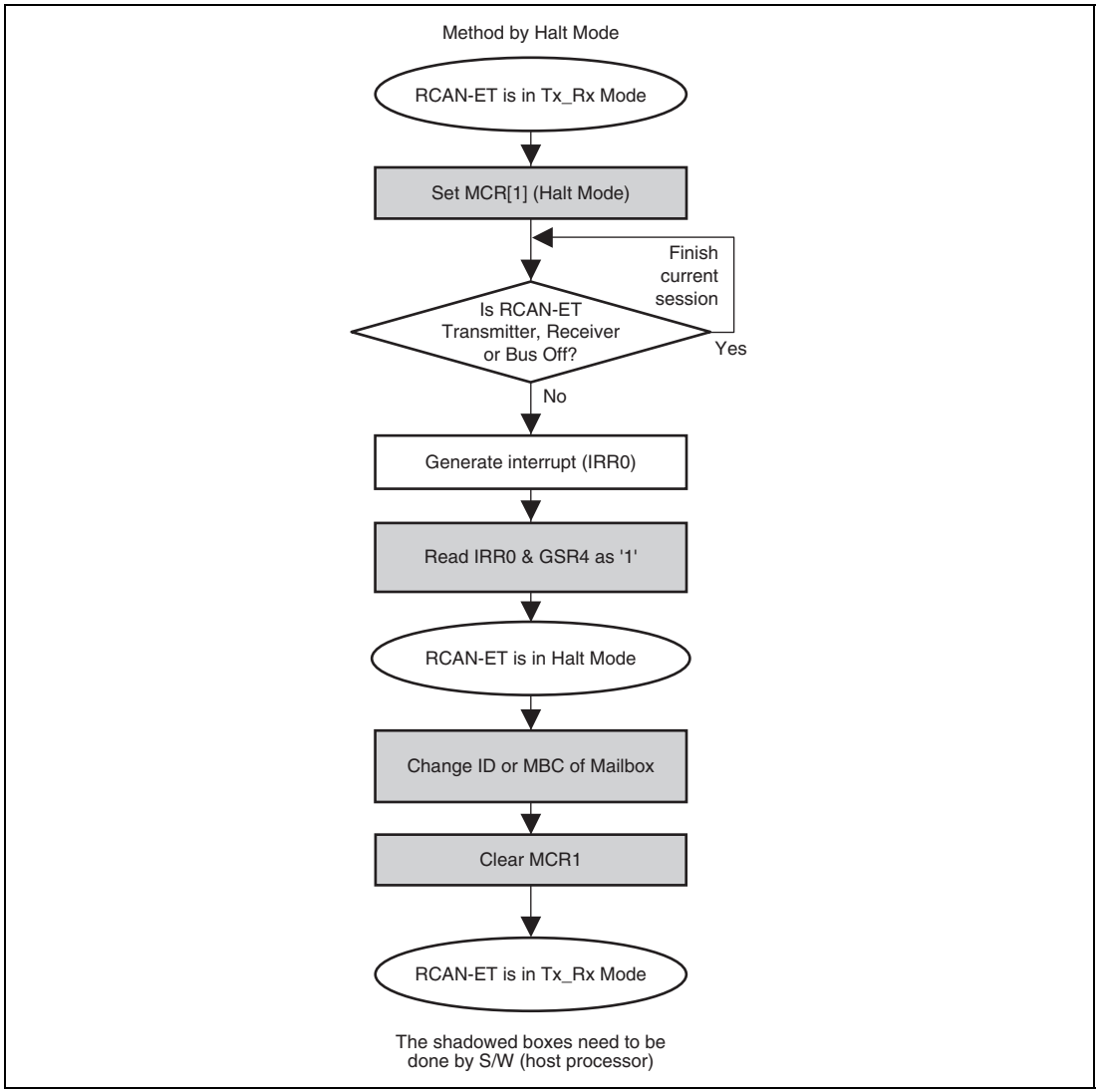


Figure 26.13 Change ID of Receive Box or Change Receive Box to Transmit Box

26.5 Interrupt Sources

Table 26.2 lists the RCAN-ET interrupt sources. With the exception of the reset processing interrupt (IRR0) by a power-on reset, these sources can be masked. Masking is implemented using the mailbox interrupt mask register 0 (MBIMR0) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 7, Interrupt Controller (INTC).

Table 26.2 RCAN-ET Interrupt Sources

Module	Interrupt	Description	Interrupt Flag	DTC Activation
RCAN-ET	ERS_0	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
		Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
		Error warning (TEC ≥ 96)	IRR3	
		Error warning (REC ≥ 96)	IRR4	
	OVR_0	Message error detection	IRR13*1	
		Reset/halt/CAN sleep transition	IRR0	
		Overload frame transmission	IRR7	
		Unread message overwrite (overrun)	IRR9	
		Detection of CAN bus operation in CAN sleep mode	IRR12	
	RM0_0*2	Data frame reception	IRR1*3	Possible*4
	RM1_0*2	Remote frame reception	IRR2*3	
	SLE_0	Message transmission/transmission disabled (slot empty)	IRR8	Not possible

Notes: 1. Available only in Test Mode.
2. RM0_0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1_0 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 15).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 15, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 15.
4. The DTC can be activated only by the RM0_0 interrupt.

26.6 DTC Interface

The DTC can be activated by the reception of a message in RCAN-ET mailbox 0. When DTC transfer ends after DTC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-ET cannot be sent to the CPU in this case. Figure 26.14 shows a DTC transfer flowchart.

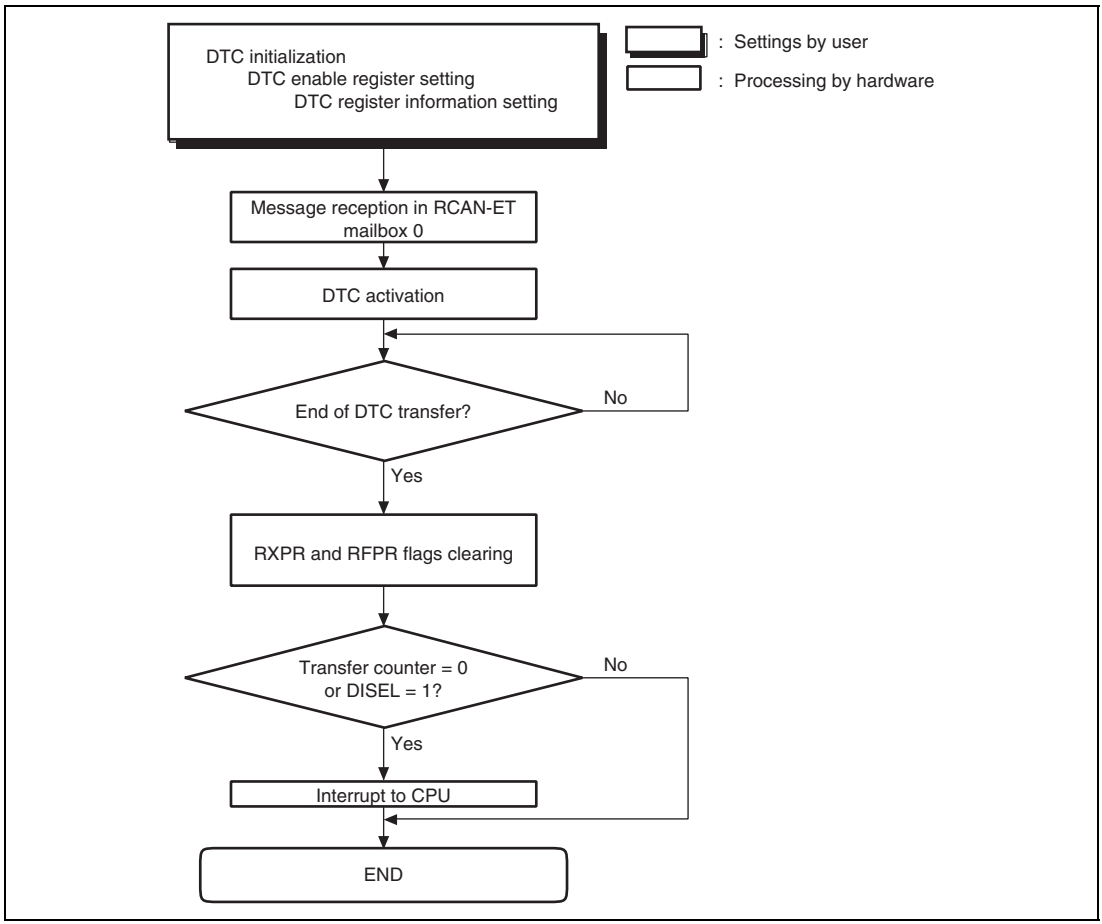
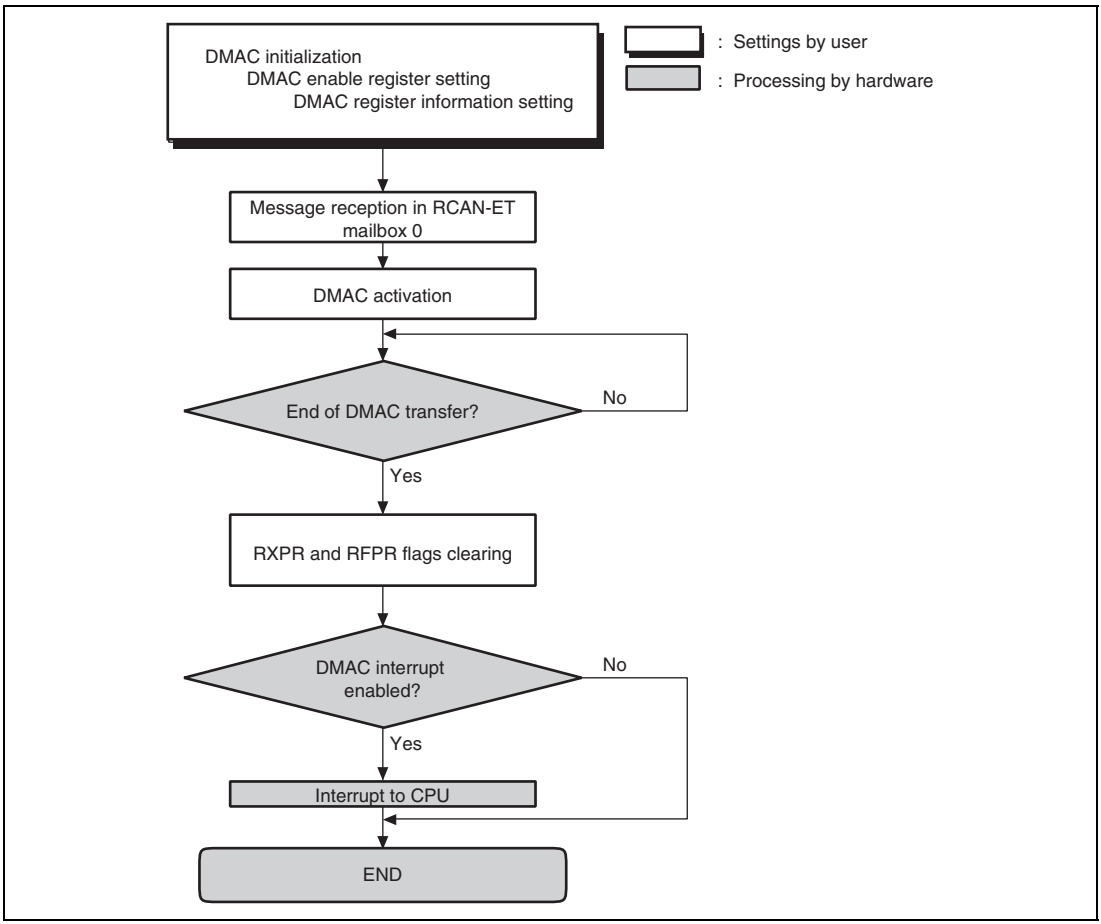


Figure 26.14 DTC Transfer Flowchart

26.7 DMAC Interface

The DMAC can be activated by the reception of a message in RCAN-ET mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-ET cannot be sent to the CPU in this case. Figure 26.15 shows a DMAC transfer flowchart.



26.15 DMAC Transfer Flowchart

26.8 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. The specification for this LSI circuit is a 3-V power-supply voltage, so use a level-shifter IC between its CRx0 pin and the Rxd pin of the HA13721. Figure 26.16 shows a sample connection diagram.

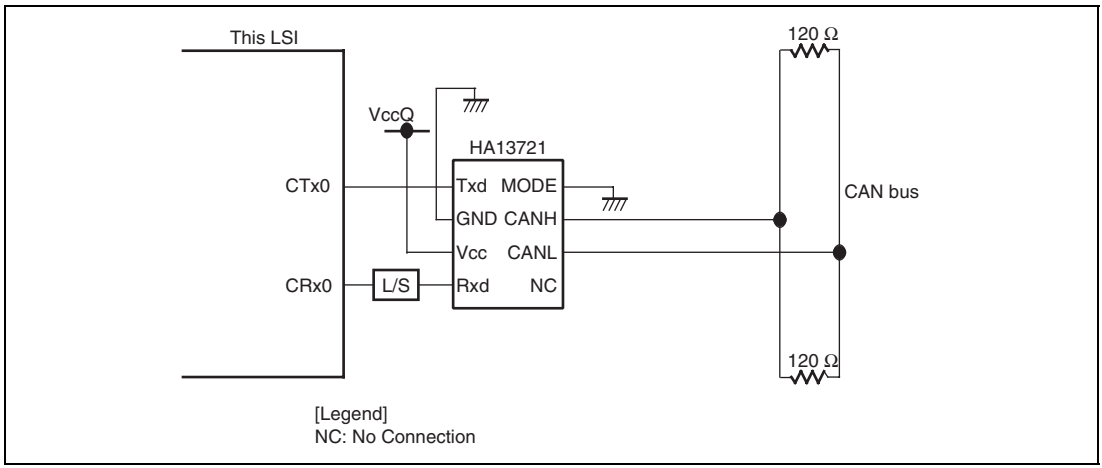


Figure 26.16 High-Speed CAN Interface Using HA13721

Section 27 32-kHz Timer (TIM32C)

This LSI incorporates a timer module (TIM32C) composed of two 8-bit timer channels and one 16-bit timer channel.

27.1 Features

- Two signals can be input to a two-phase counter of channels 0 and 1 each.
- One of four counter-clock signals can be separately selected for each channel.
- Channels 0 and 1 operate in two two-phase counter modes, allowing an interrupt to be requested on occurrence of incrementing/decrementing.
- Channel 2 operates in compare match mode, allowing an interrupt to be requested on occurrence of a compare match.
- Channel 2 can be incremented up to 32 seconds.
- Operation is possible even in software standby mode and deep software standby mode.

Table 27.1 lists the TIM32C functions.

Table 27.1 List of TIM32C Functions

Function	Channel 0	Channel 1	Channel 2
Counter clock frequency choice	• 1/2 the frequency of the clock signal input via the EXTAL32 pin • 1/4 the frequency of the clock signal input via the EXTAL32 pin • 1/8 the frequency of the clock signal input via the EXTAL32 pin • 1/16 the frequency of the clock signal input via the EXTAL32 pin	• 1/2 the frequency of the clock signal input via the EXTAL32 pin • 1/4 the frequency of the clock signal input via the EXTAL32 pin • 1/8 the frequency of the clock signal input via the EXTAL32 pin • 1/16 the frequency of the clock signal input via the EXTAL32 pin	• 1/2 the frequency of the clock signal input via the EXTAL32 pin • 1/4 the frequency of the clock signal input via the EXTAL32 pin • 1/8 the frequency of the clock signal input via the EXTAL32 pin • 1/16 the frequency of the clock signal input via the EXTAL32 pin
Counter	8-bit up/down counter (TI32CNT8_0)	8-bit up/down counter (TI32CNT8_1)	16-bit up counter (TI32CNT16_2)
Compare register	—	—	TI32COR_2
Counter clear sources	—	—	• Compare match between TI32CNT16 and TI32COR_2 • Clearance of the counter start bit (CST2)
Input pins	TI32I0A, TI32I0B	TI32I1A, TI32I1B	—
Two-phase increment/decrement mode	Normal or high speed	Normal or high speed	—
Interrupt sources	Two sources • TI32CNT8_0 increment • TI32CNT8_0 decrement	Two sources • TI32CNT8_1 increment • TI32CNT8_1 decrement	One source • Compare match between TI32CNT16 and TI32COR_2

Figure 27.1 shows a block diagram of the TIM32C.

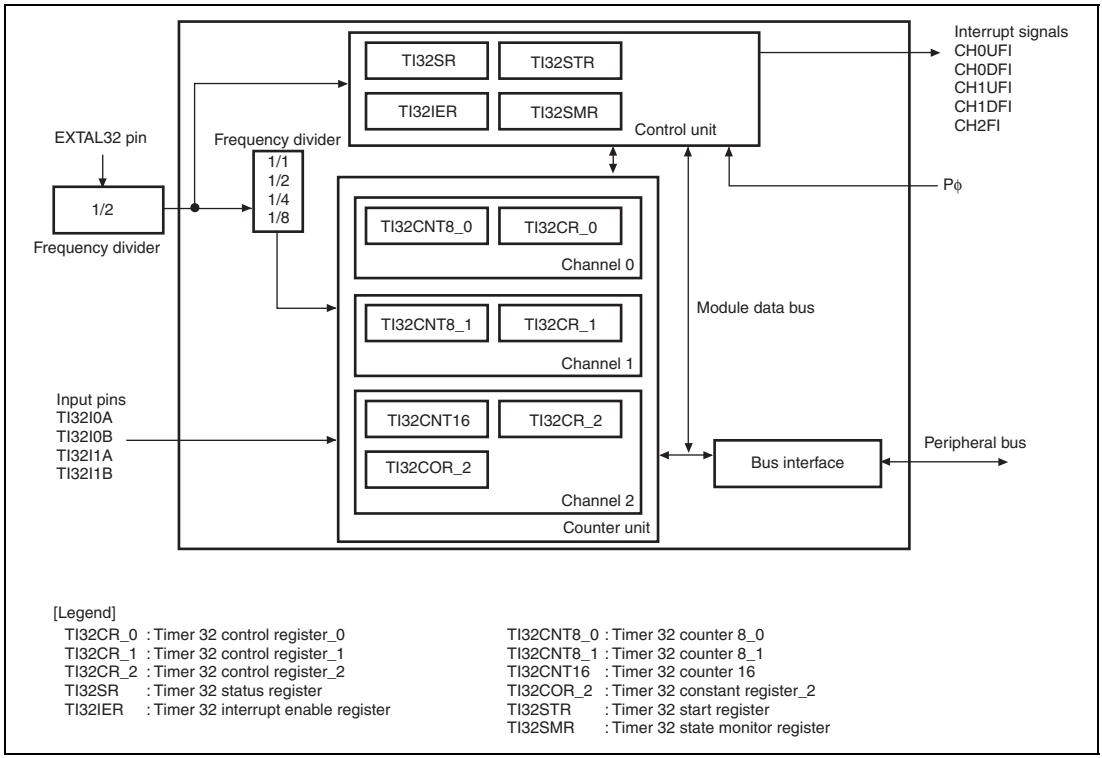


Figure 27.1 Block Diagram of TIM32C

27.2 Input/Output Pins

Table 27.2 lists the TIM32C I/O pins.

Table 27.2 Pin Configuration

Pin Name	Pin	I/O	Description
Timer input	TI32I0A	Input	Input A to the two-phase counter of channel 0
	TI32I0B	Input	Input B to the two-phase counter of channel 0
	TI32I1A	Input	Input A to the two-phase counter of channel 1
	TI32I1B	Input	Input B to the two-phase counter of channel 1

27.3 Register Descriptions

Table 27.3 lists the TIM32C registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 27.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Timer 32 control register_0	TI32CR_0	R/W	H'00	H'FFFEC200	8
Timer 32 control register_1	TI32CR_1	R/W	H'00	H'FFFEC204	8
Timer 32 control register_2	TI32CR_2	R/W	H'00	H'FFFEC208	8
Timer 32 status register	TI32SR	R/W	H'00	H'FFFEC20C	8
Timer 32 interrupt enable register	TI32IER	R/W	H'00	H'FFFEC210	8
Timer 32 counter 8_0	TI32CNT8_0	R/W	H'00	H'FFFEC214	8
Timer 32 counter 8_1	TI32CNT8_1	R/W	H'00	H'FFFEC218	8
Timer 32 counter 16	TI32CNT16	R/W	H'0000	H'FFFEC21C	16
Timer 32 constant register_2	TI32COR_2	R/W	H'0000	H'FFFEC220	16
Timer 32 start register	TI32STR	R/W	H'00	H'FFFEC224	8
Timer 32 state monitor register	TI32SMR	R	H'00	H'FFFEC280	8

27.3.1 Timer 32 Control Registers_0 to _2 (TI32CR_0 to TI32CR_2)

TI32CR_0 to TI32CR_2 are 8-bit readable/writable registers that set the operating mode, counter clock signal, and other parameters. TI32CR_0 to TI32CR_2 should be set while the timer counter of each channel (TI32CNT8_0, TI32CNT8_1, and TI32CNT16) is halted.

TI32CR_0 to TI32CR_2 are initialized only by the power-on reset signal via the $\overline{\text{RES}}$ pin. TI32CR_0 to TI32CR_2 are not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode.

(1) Timer 32 Control Registers_0 and _1 (TI32CR_0 and TI32CR_1)

Bit:	7	6	5	4	3	2	1	0
	MS	-	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MS	0	R/W	Two-Phase Counter Operating Mode Select This bit selects the operating mode of the two-phase counters. 0: Normal mode 1: High-speed mode
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R	Counter Clock Select These bits select the frequency of the counter clock signal for TI32CNT8. 00: 1/2 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 62.5 μs) 01: 1/4 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 125 μs) 10: 1/8 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 250μs) 11: 1/16 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 500 μs)

(2) Timer 32 Control Register_2 (TI32CR_2)

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	CCLR	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CCLR	0	R/W	Counter Clear This bit enables or disables TI32CNT16 to be cleared on a compare match between TI32CNT16 and TI32COR_2. 0: Disables TI32CNT16 to be cleared. 1: Enables TI32CNT16 to be cleared.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]·	00	R	Counter Clock Select These bits select the frequency of the counter clock signal for TI32CNT16. 00: 1/2 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 62.5 μs) 01: 1/4 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 125 μs) 10: 1/8 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 250μs) 11: 1/16 the frequency of the clock signal input via the EXTAL32 pin (one counter clock period: 500 μs)

27.3.2 Timer 32 Status Register (TI32SR)

TI32SR is an 8-bit readable/writable register that contains flags to indicate interrupt status. TI32SR is initialized only by the power-on reset signal via the $\overline{\text{RES}}$ pin. TI32SR is not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode.

Bit:	7	6	5	4	3	2	1	0
	CH2F	-	-	-	CH1UF	CH1DF	CH0UF	CH0DF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
7	CH2F	0	R/(W)*	Channel 2 Compare Match Flag This bit indicates that a compare match has occurred between TI32CNT16 and TI32COR_2. 0: Indicates that TI32CNT16 value and TI32COR_2 value do not match. [Clearing condition] • A 0 is written to the CH2F bit after 1 is read from the bit. 1: Indicates that TI32CNT16 value and TI32COR_2 value match. [Setting condition] • TI32CNT16 value and TI32COR_2 value match.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	CH1UF	0	R/(W)*	Channel 1 Increment Flag This bit indicates that an increment condition of TI32CNT8_1 has been satisfied. 0: Indicates that an increment condition of TI32CNT8_1 has not been satisfied. [Clearing condition] • A 0 is written to the CH1UF bit after 1 is read from the bit. 1: Indicates that an increment condition of TI32CNT8_1 has been satisfied. [Setting condition] • An increment condition of TI32CNT8_1 is satisfied.
2	CH1DF	0	R/(W)*	Channel 1 Decrement Flag This bit indicates that a decrement condition of TI32CNT8_1 has been satisfied. 0: Indicates that a decrement condition of TI32CNT8_1 has not been satisfied. [Clearing condition] • A 0 is written to the CH1DF bit after 1 is read from the bit. 1: Indicates that a decrement condition of TI32CNT8_1 has been satisfied. [Setting condition] • A decrement condition of TI32CNT8_1 is satisfied.

Bit	Bit Name	Initial Value	R/W	Description
1	CH0UF	0	R/(W)*	Channel 0 Increment Flag This bit indicates that an increment condition of TI32CNT8_0 has been satisfied. 0: Indicates that an increment condition of TI32CNT8_0 has not been satisfied. [Clearing condition] • A 0 is written to the CH0UF bit after 1 is read from the bit. 1: Indicates that an increment condition of TI32CNT8_0 has been satisfied. [Setting condition] • An increment condition of TI32CNT8_0 is satisfied.
0	CH0DF	0	R/(W)*	Channel 0 Decrement Flag This bit indicates that a decrement condition of TI32CNT8_0 has been satisfied. 0: Indicates that a decrement condition of TI32CNT8_0 has not been satisfied. [Clearing condition] • A 0 is written to the CH0DF bit after 1 is read from the bit. 1: Indicates that a decrement condition of TI32CNT8_0 has been satisfied. [Setting condition] • A decrement condition of TI32CNT8_0 is satisfied.

Note: * Only 0 can be written to the flag bit after 1 is read out to clear the flag bit.

27.3.3 Timer 32 Interrupt Enable Register (TI32IER)

TI32IER is an 8-bit readable/writable register that enables or disables interrupts. TI32IER is initialized only by the power-on reset signal via the $\overline{\text{RES}}$ pin. TI32IER is not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode.

Bit:	7	6	5	4	3	2	1	0
	CH2IE	-	-	-	CH1UIE	CH1DIE	CH0UIE	CH0DIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CH2IE	0	R/W	Channel 2 Compare Match Interrupt Enable This bit enables or disables a channel 2 compare match interrupt (CH2FI) to be requested when a compare match occurs between TI32CNT16 and TI32COR_2 (CH2F = 1). 0: Disables CH2FI to be requested. 1: Enables CH2FI to be requested.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CH1UIE	0	R/W	Channel 1 Increment Interrupt Enable This bit enables or disables a channel 1 increment interrupt (CH1UFI) to be requested when an increment condition of TI32CNT8_1 is satisfied (CH1UFI = 1). 0: Disables CH1UFI to be requested. 1: Enables CH1UFI to be requested.
2	CH1DIE	0	R/W	Channel 1 Decrement Interrupt Enable This bit enables or disables a channel 1 decrement interrupt (CH1DFI) to be requested when a decrement condition of TI32CNT8_1 is satisfied (CH1DFI = 1). 0: Disables CH1DFI to be requested. 1: Enables CH1DFI to be requested.

Bit	Bit Name	Initial Value	R/W	Description
1	CH0UIE	0	R/W	Channel 0 Increment Interrupt Enable This bit enables or disables a channel 0 increment interrupt (CH10UFI) to be requested when an increment condition of TI32CNT8_0 is satisfied (CH0UFI = 1). 0: Disables CH0UFI to be requested. 1: Enables CH0UFI to be requested.
0	CH0DIE	0	R/W	Channel 0 Decrement Interrupt Enable This bit enables or disables a channel 0 decrement interrupt (CH0DFI) to be requested when a decrement condition of TI32CNT8_0 is satisfied (CH0DFI = 1). 0: Disables CH0DFI to be requested. 1: Enables CH0DFI to be requested.

27.3.4 Timer 32 Counters 8_0 and 8_1 (TI32CNT8_0 and TI32CNT8_1)

TI32CNT8_0 and TI32CNT8_1 are 8-bit readable/writable up/down counters. With the TIM32C, TI32CNT8_0 and TI32CNT8_1 are provided in channels 0 and 1, respectively. TI32CNT8_0 and TI32CNT8_1 are initialized only by the power-on reset signal via the $\overline{\text{RES}}$ pin. TI32CNT8_0 and TI32CNT8_1 are not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode. TI32CNT8_0 and TI32CNT8_1 should be modified while they are halted (while the CST0 and CST1 bits in TI32STR are 0).

Bit:	7	6	5	4	3	2	1	0
	TI32CNT8[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TI32CNT8 [7:0]	All 0	R/W	8-Bit Up/Down Counter

27.3.5 Timer 32 Counter 16 (TI32CNT16)

TI32CNT16 is a 16-bit readable/writable up counter. With the TIM32C, TI32CNT16 is provided in channel 2. TI32CNT16 is initialized by the power-on reset signal via the $\overline{\text{RES}}$ pin or when its value and the TI32COR_2 value match or the counter start bit (CST2) in TI32STR is cleared to 0. TI32CNT16 is not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode. TI32CNT16 should be modified while it is halted (while the CST2 bit in TI32STR is 0).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI32CNT16[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TI32CNT16 [15:0]	All 0	R/W	16-Bit Up/Down Counter

27.3.6 Timer 32 Start Register (TI32STR)

TI32STR is an 8-bit readable/writable register that enables or disables the operation of the counters. TI32STR is initialized only by the power-on reset signal via the $\overline{RES}$ pin. TI32STR is not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 This bit enables or disables the operation of TI32CNT16. 0: Disables the counter operation. 1: Enables the counter operation.
1	CST1	0	R/W	Counter Start 1 This bit enables or disables the operation of TI32CNT8_1. 0: Disables the counter operation. 1: Enables the counter operation.
0	CST0	0	R/W	Counter Start 0 This bit enables or disables the operation of TI32CNT8_0. 0: Disables the counter operation. 1: Enables the counter operation.

27.3.7 Timer 32 Constant Register_2 (TI32COR_2)

TI32COR_2 is a 16-bit readable/writable register provided in channel 2. TI32COR_2 is initialized to H'FFFF by the power-on reset signal via the $\overline{\text{RES}}$ pin. TI32COR_2 is not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

27.3.8 Timer 32 State Monitor Register (TI32SMR)

TI32SMR is an 8-bit read-only register that indicates whether channels 0 and 1 are waiting for the increment/decrement conditions to be detected in normal two-phase counter mode. TI32SMR is initialized to H'00 by the power-on reset signal via the $\overline{\text{RES}}$ pin. TI32SMR is not initialized by the power-on reset signal from the WDT or the power-on reset signal used to return from deep software standby mode.

Bit:	7	6	5	4	3	2	1	0
	UP1	DWN1	UP0	DWN0	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	UP1	0	R	This bit indicates whether channel 1 is waiting for the increment condition to be detected in normal two-phase counter mode. In high-speed mode, this bit is always 0. 0: Indicates that channel 1 is not waiting for the increment condition to be detected. 1: Indicates that channel 1 is waiting for the increment condition to be detected.
6	DWN1	0	R	This bit indicates whether channel 1 is waiting for the decrement condition to be detected in normal two-phase counter mode. In high-speed mode, this bit is always 0. 0: Indicates that channel 1 is not waiting for the decrement condition to be detected. 1: Indicates that channel 1 is waiting for the decrement condition to be detected.
5	UP0	0	R	This bit indicates whether channel 0 is waiting for the increment condition to be detected in normal two-phase counter mode. In high-speed mode, this bit is always 0. 0: Indicates that channel 0 is not waiting for the increment condition to be detected. 1: Indicates that channel 0 is waiting for the increment condition to be detected.

Bit	Bit Name	Initial Value	R/W	Description
4	DWN0	0	R	This bit indicates whether channel 0 is waiting for the decrement condition to be detected in normal two-phase counter mode. In high-speed mode, this bit is always 0. 0: Indicates that channel 0 is not waiting for the decrement condition to be detected. 1: Indicates that channel 0 is waiting for the decrement condition to be detected.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.4 Operation

27.4.1 Two-Phase Counter Operation

By setting channels 0 and 1 appropriately, TI32CNT8_0 and TI32CNT8_1 are incremented or decremented according to the phase difference detected between two externally input signals. When the increment condition of TI32CNT8_0 and TI32CNT8_1 is satisfied, the CH0UF and CH1UF bits in TI32SR are set to 1, respectively; and when the decrement condition is satisfied, the CH0DF and CH1DF bits are set to 1, respectively. If the corresponding enable bits in TI32IER are 1 here, the TIM32C issues an interrupt request to the CPU.

(1) Procedure for Setting Two-Phase Counter Operation

Figure 27.2 shows an example of procedure for setting two-phase counter operation.

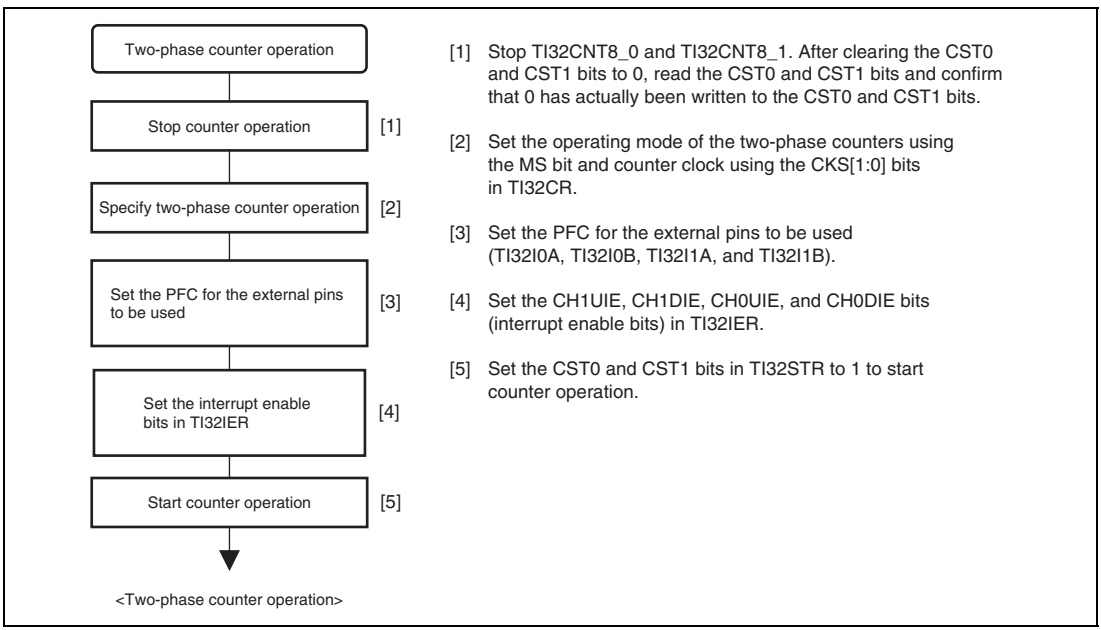


Figure 27.2 Example of Procedure for Setting Two-Phase Counter Operation

(2) Example of Two-Phase Counter Operation

Two-phase counters can be operated in normal mode or high-speed mode according to the MS bit setting in TI32CR.

(a) Normal Mode

Figures 27.3 and 27.4 show examples of TI32CNT8 increment and decrement operations in normal two-phase counter mode, respectively.

Upon detection of the transition from state 1 to 2, TI32CNT8 starts waiting for the increment condition to be detected. Upon detection of the increment condition (transition from state 4 to 1) while waiting, TI32CNT8 increments its value.

Upon detection of the transition from state 1 to 4 while waiting for the increment condition to be detected, TI32CNT8 stops waiting for the increment condition to be detected and starts waiting for the decrement condition to be detected instead.

Upon detection of the transition from state 1 to 4, TI32CNT8 starts waiting for the decrement condition to be detected. Upon detection of the decrement condition (transition from state 2 to 1) while waiting, TI32CNT8 decrements its value.

Upon detection of the transition from state 1 to 2 while waiting for the decrement condition to be detected, TI32CNT8 stops waiting for the decrement condition to be detected and starts waiting for the increment condition to be detected instead.

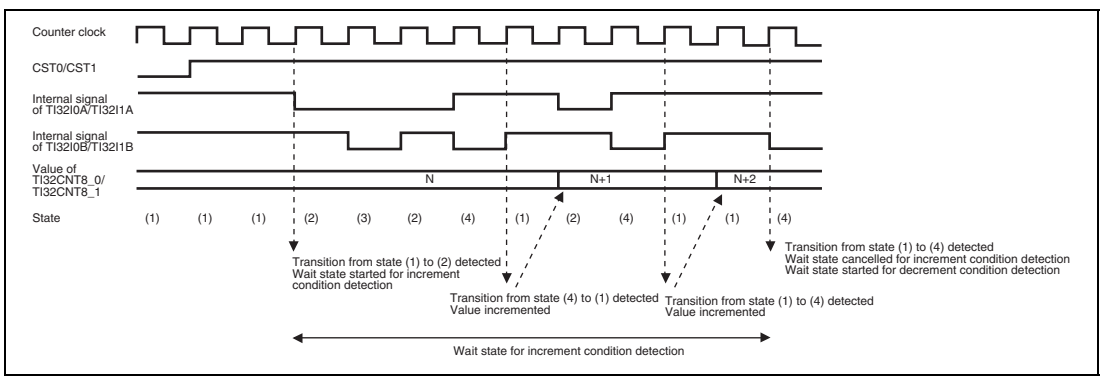


Figure 27.3 Example of TI32CNT8 Increment Operation in Normal Two-Phase Counter Mode

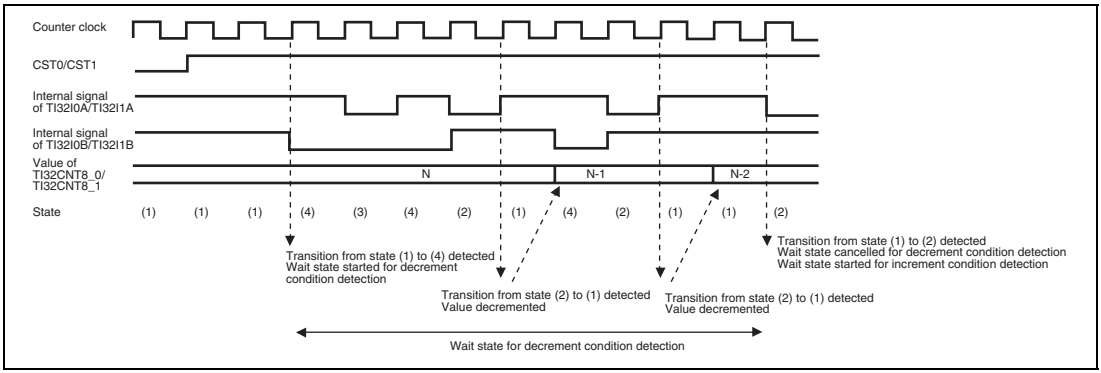


Figure 27.4 Example of TI32CNT8 Decrement Operation in Normal Two-Phase Counter Mode

(b) High-Speed Mode

Table 27.4 shows the TI32CNT8 increment/decrement conditions in high-speed two-phase counter mode, and figure 27.5 shows an example of TI32CNT8 operations in the mode.

Table 27.4 Increment/Decrement Conditions

	TI32I0A and TI32I1A Input Pins	TI32I0B and TI32I1B Input Pins		TI32I0A and TI32I1A Input Pins	TI32I0B and TI32I1B Input Pins
Increment	High level	Falling edge	Decrement	High level	Falling edge
	Low level	Rising edge		Low level	Rising edge
	Rising edge	Low level		Rising edge	Low level
	Falling edge	High level		Falling edge	High level

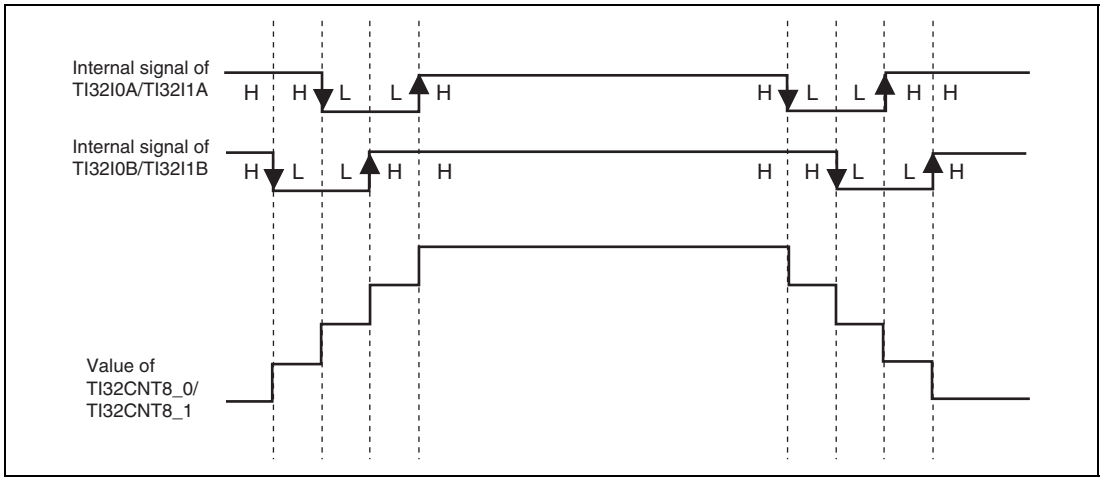


Figure 27.5 Example of TI32CNT8 Operation in High-Speed Two-Phase Counter Mode

27.4.2 Compare Match Operation

Channel 2 operates as a compare match timer. When the CST2 bit in TI32STR is set to 1 after the counter clock is selected using the CKS1 and CKS0 bits in TI32CR_2, TI32CNT16 starts incrementing based on the selected clock signal. When the TI32CNT16 value and TI32COR_2 value match, the CH2F bit in TI32SR is set to 1. If the CCLR bit in TI32CR_2 is 1 here, TI32CNT16 is cleared to H'0000; and if the CH2IE bit in TI32IER is 1 here, TI32CNT16 issues a CH2FI interrupt request. TI32CNT16 restarts incrementing from H'0000.

Figure 27.6 shows an example of procedure for setting compare match operation and figure 27.7 shows an example of compare match operation.

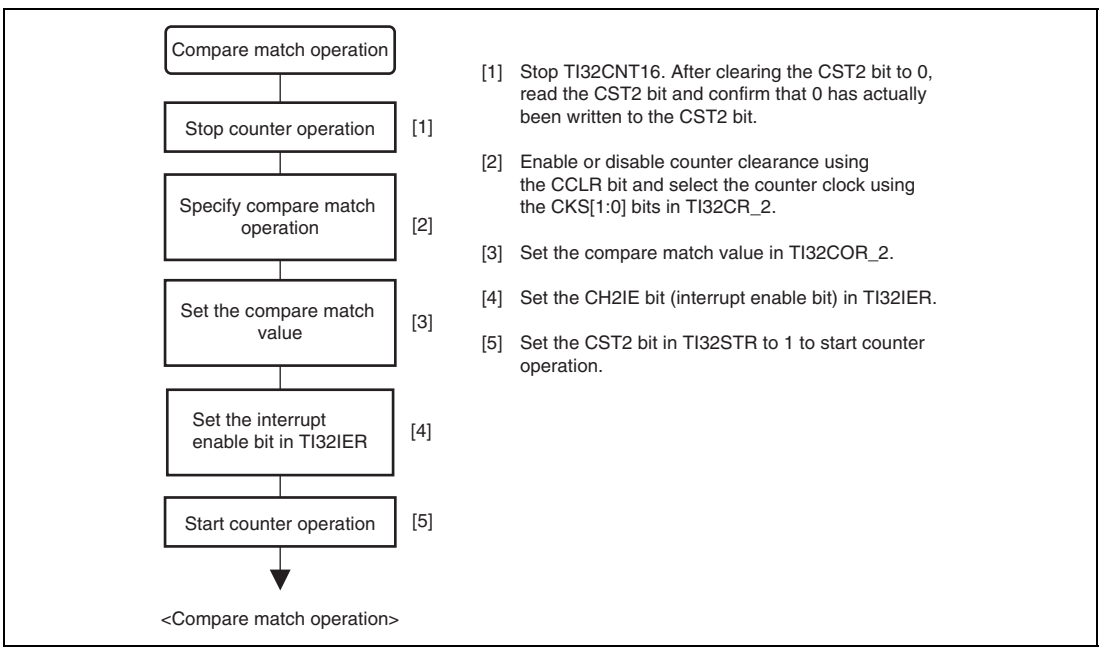


Figure 27.6 Example of Procedure for Setting Compare Match Operation

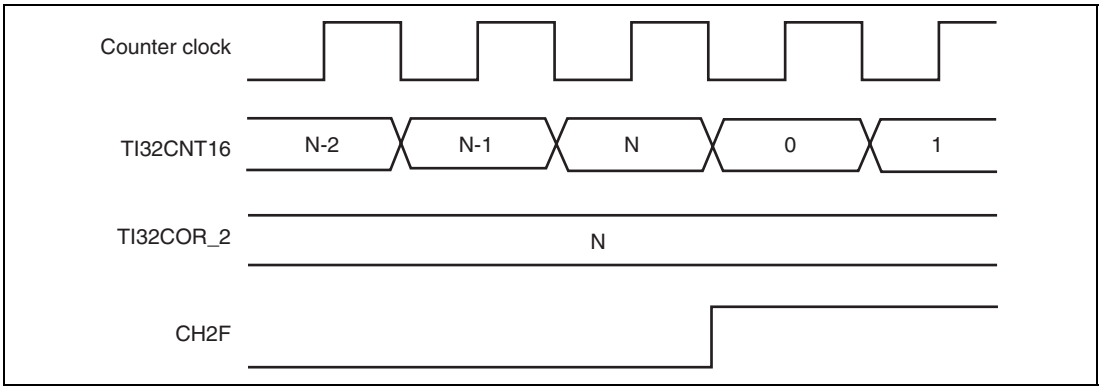


Figure 27.7 Example of Compare Match Operation (CCLR = 1)

27.5 Interrupt Sources

The TIM32C has five interrupt sources: channel 0 decrement interrupt (CH0DFI), channel 0 increment interrupt (CH0UFI), channel 1 decrement interrupt (CH1DFI), channel 1 increment interrupt (CH1UFI), and channel 2 compare match interrupt (CH2FI).

Table 27.5 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the CH0DIE, CH0UIE, CH1DIE, CH1UIE, and CH2IE bits in TI32IER and are separately issued to the interrupt controller.

When software standby mode is canceled using the TIM32C interrupt, the SSRF bit in the standby interrupt flag register (SIFR) is set to 1.

Table 27.5 TIM32C Interrupt Sources and Priority

Interrupt Source	Interrupt Name	Interrupt Enable Bit	Priority
CH0DFI	Channel 0 decrement interrupt (CH0DFI)	CH0DIE	High
CH0UFI	Channel 0 increment interrupt (CH0UFI)	CH0UIE	
CH1DFI	Channel 1 decrement interrupt (CH0DFI)	CH1DIE	
CH1UFI	Channel 1 increment interrupt (CH0DFI)	CH1UIE	
CH2FI	Channel 2 compare match interrupt (CH2FI)	CH2IE	Low

27.6 Usage Notes

27.6.1 Setting Module Standby Mode

The TIM32C operation can be disabled or enabled using the standby control register. With the initial value, the TIM32C and 32-kHz clock oscillator used by the TIM32C are halted. Register access and TIM32C operation are enabled by canceling module standby mode. For details, refer to section 32, Power-Down Modes.

When the 32-kHz crystal resonator is used, secure the EXTAL32 clock settling time (tOSC32) after module standby mode is canceled before accessing the TIM32C registers. Otherwise, the TIM32C registers cannot be accessed correctly. For details, refer to figure 35.10 in section 35, Electrical Characteristics.

27.6.2 Restarting Suspended Counters

Before restarting the suspended counters, clear the CST0 to CST2 bits in TI32STR and read the CST0 to CST2 bits to confirm that 0 has actually been written to the bits.

27.6.3 Consecutive Writing to the Same Register

The value written to a register is actually used for the operation after synchronization based on the TIM32C operating clock signal (the clock signal obtained by dividing EXTAL32 input signal by two). Therefore, when writing to the same register consecutively, wait one or more TIM32C operating clock cycle after a write access before the next write access, or read the register before the next write access to confirm that the desired value has actually been written to the register.

27.6.4 Counter Value Modification

The counters (TI32CNT8_0, TI32CNT8_1, and TI32CNT16) should be modified while they are halted (while the CST0 to CST2 bits in TI32STR are 0).

Section 28 Key Scan Controller (KEYC)

This LSI incorporates a key scan controller (KEYC).

28.1 Features

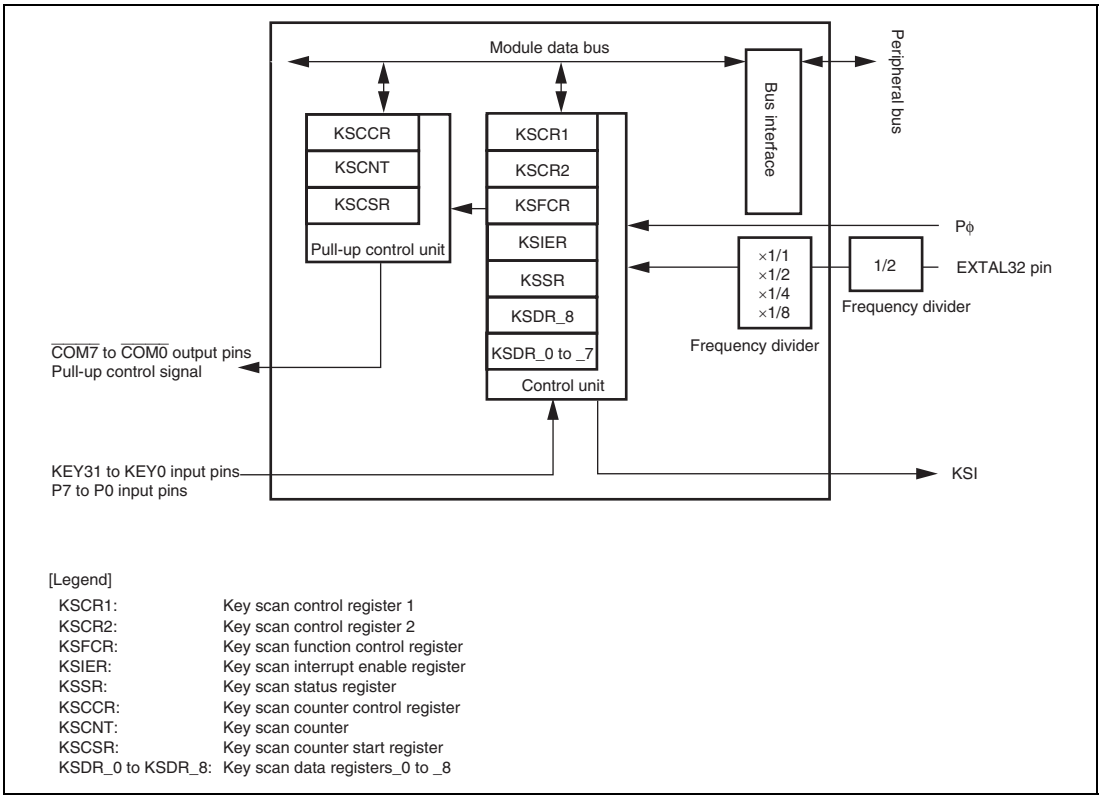
- A maximum of 32 inputs are available, and one of the following combinations in terms of I/O and keys and key matrix can be selected.
  - Thirty-two inputs via keys
  - Twenty-four inputs via keys and four inputs and outputs each via the key matrix
  - Sixteen inputs via keys and eight inputs and outputs each via the key matrix
- One interrupt source

  An interrupt can be generated when the values input via the keys and/or key matrix are different from the previously stored values. This interrupt can be used to cancel software standby mode or deep software standby mode.
- Intermittent pull up

  Key input pins and key matrix input pins are provided with pull-up MOSs, for which it is possible to select the desired intermittent pull-up period and duration. It is also possible to select the duration of the low-level signal output via the key matrix pins.

  At the end of pull-up duration, the values input via the keys and/or key matrix are stored in the data registers.
- Module standby mode can be set.

Figure 28.1 shows a block diagram of the KEYC.



[Legend]

KSCR1:	Key scan control register 1
KSCR2:	Key scan control register 2
KSFCR:	Key scan function control register
KSIER:	Key scan interrupt enable register
KSSR:	Key scan status register
KSCCR:	Key scan counter control register
KSCNT:	Key scan counter
KSCSR:	Key scan counter start register
KSDR_0 to KSDR_8:	Key scan data registers_0 to _8

Figure 28.1 Block Diagram of KEYC

28.2 Input/Output Pins

Table 28.1 lists the KEYC I/O pins and table 28.2 shows the pin functions.

Table 28.1 Pin Configuration

Pin	I/O	Description
KEY31 to KEY0	Input	Key input
P7 to P0	Input	Key matrix input
$\overline{\text{COM7}}$ to $\overline{\text{COM0}}$	Output	Key matrix output

Table 28.2 Pin Functions

Initial Function	Available Functions when KMS Bit in KSCR1 is 0	Available Functions when KMS Bit in KSCR1 is 1
KEY31	KEY31/P3	KEY31/P7
KEY30	KEY30/P2	KEY30/P6
KEY29	KEY29/P1	KEY29/P5
KEY28	KEY28/P0	KEY28/P4
KEY27	KEY27/$\overline{\text{COM3}}$	KEY27/P3
KEY26	KEY26/$\overline{\text{COM2}}$	KEY26/P2
KEY25	KEY25/$\overline{\text{COM1}}$	KEY25/P1
KEY24	KEY24/$\overline{\text{COM0}}$	KEY24/P0
KEY23	KEY23	KEY23/$\overline{\text{COM7}}$
KEY22	KEY22	KEY22/$\overline{\text{COM6}}$
KEY21	KEY21	KEY21/$\overline{\text{COM5}}$
KEY20	KEY20	KEY20/$\overline{\text{COM4}}$
KEY19	KEY19	KEY19/$\overline{\text{COM3}}$
KEY18	KEY18	KEY18/$\overline{\text{COM2}}$
KEY17	KEY17	KEY17/$\overline{\text{COM1}}$
KEY16	KEY16	KEY16/$\overline{\text{COM0}}$
KEY15	KEY15	KEY15
KEY14	KEY14	KEY14

Initial Function	Available Functions when KMS Bit in KSCR1 is 0	Available Functions when KMS Bit in KSCR1 is 1
KEY13	KEY13	KEY13
KEY12	KEY12	KEY12
KEY11	KEY11	KEY11
KEY10	KEY10	KEY10
KEY9	KEY9	KEY9
KEY8	KEY8	KEY8
KEY7	KEY7	KEY7
KEY6	KEY6	KEY6
KEY5	KEY5	KEY5
KEY4	KEY4	KEY4
KEY3	KEY3	KEY3
KEY2	KEY2	KEY2
KEY1	KEY1	KEY1
KEY0	KEY0	KEY0

28.3 Register Descriptions

Table 28.3 lists the KEYC registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 28.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Key scan control register 1	KSCR1	R/W	H'0000	H'FFFEC300	16
Key scan control register 2	KSCR2	R/W	H'0000	H'FFFEC302	16
Key scan function control register	KSFCR	R/W	H'00	H'FFFEC304	8
Key scan data register_0	KSDR_0	R	H'FF	H'FFFEC310	8
Key scan data register_1	KSDR_1	R	H'FF	H'FFFEC311	8
Key scan data register_2	KSDR_2	R	H'FF	H'FFFEC312	8
Key scan data register_3	KSDR_3	R	H'FF	H'FFFEC313	8
Key scan data register_4	KSDR_4	R	H'FF	H'FFFEC314	8
Key scan data register_5	KSDR_5	R	H'FF	H'FFFEC315	8
Key scan data register_6	KSDR_6	R	H'FF	H'FFFEC316	8
Key scan data register_7	KSDR_7	R	H'FF	H'FFFEC317	8
Key scan data register_8	KSDR_8	R	H'FFFF	H'FFFEC318	16
Key scan interrupt enable register	KSIER	R/W	H'00	H'FFFEC320	8
Key scan status register	KSSR	R/W	H'00	H'FFFEC322	8
Key scan counter control register	KSCCR	R/W	H'00	H'FFFEC324	8
Key scan counter	KSCNT	R	H'00	H'FFFEC326	8
Key scan counter start register	KSCSR	R/W	H'00	H'FFFEC328	8

28.3.1 Key Scan Control Registers 1 and 2 (KSCR1 and KSCR2)

KSCR1 and KSCR2 are 16-bit readable/writable registers. KSCR1 selects the number of key matrix pins to be used (= key matrix configuration); resets key scan data registers_0 to _8 (KSDR_0 to KSDR_8); and enables or disables the key input or key matrix input/output via the KEY31 to KEY16 pins. KSCR2 enables or disables the key input via the KEY15 to KEY0 pins.

KSCR1 and KSCR2 are initialized to H'0000 by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSCR1 and KSCR2 are not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

(1) Key Scan Control Register 1 (KSCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRRST	-	-	-	-	-	-	-	-	-	-	-	-	KMS	KSE17	KSE16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	DRRST	0	R/W	Key Scan Data Register Reset This bit initializes key scan data registers_0 to _8 (KSDR_0 to KSDR_8). This bit should be held to 0 during key scan operation. 0: Does not initialize key scan data registers_0 to _8 (KSDR_0 to KSDR_8). 1: Initializes key scan data registers_0 to _8 (KSDR_0 to KSDR_8). Note: When the power-on reset signal is input via the $\overline{\text{RES}}$ pin, KSDR_0 to KSDR_8 are initialized regardless of the setting of this bit.
14 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	KMS	0	R/W	Key Matrix Input/Output Pin Select This bit sets the key matrix input/output pins. 0: Sets the 4 × 4-key matrix configuration (output pins: $\overline{COM3}$ to $\overline{COM0}$; input pins: P3 to P0). 1: Sets the 8 × 8-key matrix configuration (output pins: $\overline{COM7}$ to $\overline{COM0}$; input pins: P7 to P0). Note: This bit should be modified when the CST bit in KSCSR is 0.
1	KSE17	0	R/W	Key Scan Input/Output Enable 17 This bit enables or disables the key input or key matrix input/output via the KEY31 to KEY24 pins. 0: Disables key input or key matrix input/output.* 1: Enables key input or key matrix input/output.
0	KSE16	0	R/W	Key Scan Input/Output Enable 16 This bit enables or disables the key input or key matrix input/output via the KEY23 to KEY16 pins. 0: Disables key input or key matrix input/output.* 1: Enables key input or key matrix input/output.

Note: * When the KSE bits are 0, a 1 is always read from the corresponding key or key matrix input pins. No scanned data is output from the corresponding key matrix output pins. The intermittent pull-up MOS of the corresponding key or key matrix input pins are invalid.

(2) Key Scan Control Register 2 (KSCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KSE15	KSE14	KSE13	KSE12	KSE11	KSE10	KSE9	KSE8	KSE7	KSE6	KSE5	KSE4	KSE3	KSE2	KSE1	KSE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	KSE15	0	R/W	Key Scan Input/Output Enable 15 This bit enables or disables the key input via the KEY15 pin. 0: Disables key input.* 1: Enables key input.
14	KSE14	0	R/W	Key Scan Input/Output Enable 14 This bit enables or disables the key input via the KEY14 pin. 0: Disables key input.* 1: Enables key input.
13	KSE13	0	R/W	Key Scan Input/Output Enable 13 This bit enables or disables the key input via the KEY13 pin. 0: Disables key input.* 1: Enables key input.
12	KSE12	0	R/W	Key Scan Input/Output Enable 12 This bit enables or disables the key input via the KEY12 pin. 0: Disables key input.* 1: Enables key input.
11	KSE11	0	R/W	Key Scan Input/Output Enable 11 This bit enables or disables the key input via the KEY11 pin. 0: Disables key input.* 1: Enables key input.

Bit	Bit Name	Initial Value	R/W	Description
10	KSE10	0	R/W	Key Scan Input/Output Enable 10 This bit enables or disables the key input via the KEY10 pin. 0: Disables key input.* 1: Enables key input.
9	KSE9	0	R/W	Key Scan Input/Output Enable 9 This bit enables or disables the key input via the KEY9 pin. 0: Disables key input.* 1: Enables key input.
8	KSE8	0	R/W	Key Scan Input/Output Enable 8 This bit enables or disables the key input via the KEY8 pin. 0: Disables key input.* 1: Enables key input.
7	KSE7	0	R/W	Key Scan Input/Output Enable 7 This bit enables or disables the key input via the KEY7 pin. 0: Disables key input.* 1: Enables key input.
6	KSE6	0	R/W	Key Scan Input/Output Enable 6 This bit enables or disables the key input via the KEY6 pin. 0: Disables key input.* 1: Enables key input.
5	KSE5	0	R/W	Key Scan Input/Output Enable 5 This bit enables or disables the key input via the KEY5 pin. 0: Disables key input.* 1: Enables key input.

Bit	Bit Name	Initial Value	R/W	Description
4	KSE4	0	R/W	Key Scan Input/Output Enable 4 This bit enables or disables the key input via the KEY4 pin. 0: Disables key input.* 1: Enables key input.
3	KSE3	0	R/W	Key Scan Input/Output Enable 3 This bit enables or disables the key input via the KEY3 pin. 0: Disables key input.* 1: Enables key input.
2	KSE2	0	R/W	Key Scan Input/Output Enable 2 This bit enables or disables the key input via the KEY2 pin. 0: Disables key input.* 1: Enables key input.
1	KSE1	0	R/W	Key Scan Input/Output Enable 1 This bit enables or disables the key input via the KEY1 pin. 0: Disables key input.* 1: Enables key input.
0	KSE0	0	R/W	Key Scan Input/Output Enable 0 This bit enables or disables the key input via the KEY0 pin. 0: Disables key input.* 1: Enables key input.

Note: * When the KSE bits are 0, a 1 is always read from the corresponding key input pins. The intermittent pull-up MOS of the corresponding key input pins are invalid.

28.3.2 Key Scan Function Control Register (KSFCR)

KSFCR is an 8-bit readable/writable register that selects either the key input or key matrix input/output function for the KEY pins. KSFCR is initialized to H'00 by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSFCR is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

KSFCR should be modified when the CST bit in KSCSR is 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	KSFCR2	KSFCR1	KSFCR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	KSFCR2	0	R/W	Key Scan Input/Output Control 2 This bit selects the function of the KEY31 to KEY28 pins. For details, refer to table 28.4. 0: Key input 1: Key matrix input
1	KSFCR1	0	R/W	Key Scan Input/Output Control 1 This bit selects the function of the KEY27 to KEY24 pins. For details, refer to table 28.4. 0: Key input 1: Key matrix input or key matrix output*
0	KSFCR0	0	R/W	Key Scan Input/Output Control 0 This bit selects the function of the KEY23 to KEY16 pins. For details, refer to table 28.4. 0: Key input 1: Key matrix input

Note: * Key matrix output is selected when the KMS bit in KSCR1 is 0 and key matrix input is selected when 1.

Table 28.4 Function Selection of KEY31 to KEY16 Pins by Register Set

Item			Key Input	4 × 4-Key Matrix Input/Output	8 × 8-Key Matrix Input/Output
Register settings	KSFCR	KSFCR2 bit	0	1	1
		KSFCR1 bit	0	1	1
		KSFCR0 bit	0	0	1
	KSCR1	KMS bit	— (invalid setting)	0	1
		KSE17 bit	1	1	1
		KSE16 bit	1	1	1
Multiplexed pins	KEY31/P3/P7 pin		KEY31 input	P3 input	P7 input
	KEY30/P2/P6 pin		KEY30 input	P2 input	P6 input
	KEY29/P1/P5 pin		KEY29 input	P1 input	P5 input
	KEY28/P0/P4 pin		KEY28 input	P0 input	P4 input
	KEY27/$\overline{\text{COM3}}$/P3 pin		KEY27 input	$\overline{\text{COM3}}$ output	P3 input
	KEY26/$\overline{\text{COM2}}$/P2 pin		KEY26 input	$\overline{\text{COM2}}$ output	P2 input
	KEY25/$\overline{\text{COM1}}$/P1 pin		KEY25 input	$\overline{\text{COM1}}$ output	P1 input
	KEY24/$\overline{\text{COM0}}$/P0 pin		KEY24 input	$\overline{\text{COM0}}$ output	P0 input
	KEY23/$\overline{\text{COM7}}$ pin		KEY23 input	KEY23 input	$\overline{\text{COM7}}$ output
	KEY22/$\overline{\text{COM6}}$ pin		KEY22 input	KEY22 input	$\overline{\text{COM6}}$ output
	KEY21/$\overline{\text{COM5}}$ pin		KEY21 input	KEY21 input	$\overline{\text{COM5}}$ output
	KEY20/$\overline{\text{COM4}}$ pin		KEY20 input	KEY20 input	$\overline{\text{COM4}}$ output
	KEY19/$\overline{\text{COM3}}$ pin		KEY19 input	KEY19 input	$\overline{\text{COM3}}$ output
	KEY18/$\overline{\text{COM2}}$ pin		KEY18 input	KEY18 input	$\overline{\text{COM2}}$ output
	KEY17/$\overline{\text{COM1}}$ pin		KEY17 input	KEY17 input	$\overline{\text{COM1}}$ output
	KEY16/$\overline{\text{COM0}}$ pin		KEY16 input	KEY16 input	$\overline{\text{COM0}}$ output

28.3.3 Key Scan Data Registers_0 to _8 (KSDR_0 to KSDR_8)

KSDR_8 is a 16-bit read-only register and KSDR_0 to KSDR_7 are 8-bit read-only registers.

When the key input function is selected using the key scan function control register (KSFCR), the values of the KEY15 to KEY0 pins are stored in KSDR_8; the values of the KEY23 to KEY16 pins are stored in KSDR_7; and the values of the KEY31 to KEY24 pins are stored in KSDR_6 at the end of pull-up duration.

When the key matrix input function is selected using KSFCR, the values of the key matrix input pins are stored in KSDR_0 to KSDR_7 at the end of pull-up duration. The values to be stored in KSDR_0 to KSDR_7 depend on the KMS bit setting in key scan control register 1 (KSCR1) as shown in table 28.5.

KSDR_0 to KSDR_8 are initialized by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSDR_0 to KSDR_8 are not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

(1) Key Scan Data Register_8 (KSDR_8)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KSDR15	KSDR14	KSDR13	KSDR12	KSDR11	KSDR10	KSDR9	KSDR8	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	KSDR15	1	R	The values of the KEY15 to KEY0 pins are stored in these bits at the end of pull-up duration. For details, refer to table 28.5.
14	KSDR14	1	R	
13	KSDR13	1	R	
12	KSDR12	1	R	
11	KSDR11	1	R	
10	KSDR10	1	R	
9	KSDR9	1	R	
8	KSDR8	1	R	
7	KSDR7	1	R	
6	KSDR6	1	R	
5	KSDR5	1	R	
4	KSDR4	1	R	
3	KSDR3	1	R	
2	KSDR2	1	R	
1	KSDR1	1	R	
0	KSDR0	1	R	

(2) Key Scan Data Registers_0 to _7 (KSDR_0 to KSDR_7)

Bit:	7	6	5	4	3	2	1	0
	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	KSDR7	1	R	For details, refer to table 28.5.
6	KSDR6	1	R	
5	KSDR5	1	R	
4	KSDR4	1	R	
3	KSDR3	1	R	
2	KSDR2	1	R	
1	KSDR1	1	R	
0	KSDR0	1	R	

Table 28.5 Function Assignment to KSDR_0 to KSDR_8

Particular KSDR	When KEY31 to KEY16 are Used for Key Input	When KMS Bit in KSCR1 is 0 (key matrix input: P3 to P0; key matrix output: $\overline{\text{COM3}}$ to $\overline{\text{COM0}}$)*	When KMS Bit in KSCR1 is 0 (key matrix input: P3 to P0; key matrix output: $\overline{\text{COM3}}$ to $\overline{\text{COM0}}$)
KSDR_0	Not used	The result of reading the values output via key matrix 0	The result of reading the values output via key matrix 0
KSDR_1	Not used	The result of reading the values output via key matrix 1	The result of reading the values output via key matrix 1
KSDR_2	Not used	The result of reading the values output via key matrix 2	The result of reading the values output via key matrix 2
KSDR_3	Not used	The result of reading the values output via key matrix 3	The result of reading the values output via key matrix 3
KSDR_4	Not used	Not used	The result of reading the values output via key matrix 4
KSDR_5	Not used	Not used	The result of reading the values output via key matrix 5
KSDR_6	The result of reading the values input via KEY[31:24]	Not used	The result of reading the values output via key matrix 6
KSDR_7	The result of reading the values input via KEY[23:16]	The result of reading the values input via KEY[23:16]	The result of reading the values output via key matrix 7
KSDR_8	The result of reading the values input via KEY[15:0]	The result of reading the values input via KEY[15:0]	The result of reading the values input via KEY[15:0]

Note: * The result of reading the values output via the key matrix is stored in the lower four bits. 1s are stored in the upper four bits.

28.3.4 Key Scan Interrupt Enable Register (KSIER)

KSIER is an 8-bit readable/writable register that enables or disables key scan interrupts. KSIER is initialized to H'00 by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSIER is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	KSIE2	KSIE1	KSIE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	KSIE2	0	R/W	Key Scan Interrupt Enable 2 This bit enables or disables a key scan interrupt (KSI) to be requested when the KSF2 bit in KSSR is set to 1. 0: Disables KSI to be requested. 1: Enables KSI to be requested.
1	KSIE1	0	R/W	Key Scan Interrupt Enable 1 This bit enables or disables a key scan interrupt (KSI) to be requested when the KSF1 bit in KSSR is set to 1. 0: Disables KSI to be requested. 1: Enables KSI to be requested.
0	KSIE0	0	R/W	Key Scan Interrupt Enable 0 This bit enables or disables a key scan interrupt (KSI) to be requested when the KSF0 bit in KSSR is set to 1. 0: Disables KSI to be requested. 1: Enables KSI to be requested.

28.3.5 Key Scan Status Register (KSSR)

KSSR is an 8-bit readable/writable register that indicates interrupt status. KSSR is initialized to H'00 by the power-on reset signal via the $\overline{RES}$ pin or when placed in module standby mode. KSSR is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	KSF2	KSF1	KSF0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to after reading 1, to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	KSF2	0	R/(W)*	Key Scan Flag 2 This bit indicates whether a key scan interrupt source has been input from any of the KEY31 to KEY24 pins. 0: Indicates that no key scan interrupt source has been input from any of the KEY31 to KEY24 pins. [Clearing condition] • A 0 is written to the KSF2 bit after 1 is read from the bit. 1: Indicates that a key scan interrupt source has been input from any of the KEY31 to KEY24 pins. [Setting conditions] • When the KEY31 to KEY24 pins are used for key input, the values having been stored in KSDR_6 are different from the previously stored values. • When the KEY31 to KEY24 pins are used for the 4 × 4-key matrix, the values having been stored in KSDR_0 to KSDR_3 are different from the previously stored values. • When the KEY31 to KEY24 pins are used for the 8 × 8-key matrix, the values having been stored in KSDR_0 to KSDR_7 are different from the previously stored values.

Bit	Bit Name	Initial Value	R/W	Description
1	KSF1	0	R/(W)*	Key Scan Flag 1 This bit indicates whether a key scan interrupt source has been input from any of the KEY23 to KEY16 pins. 0: Indicates that no key scan interrupt source has been input from any of the KEY23 to KEY16 pins. [Clearing condition] • A 0 is written to the KSF1 bit after 1 is read from the bit. 1: Indicates that a key scan interrupt source has been input from any of the KEY23 to KEY16 pins. [Setting conditions] • When the KEY23 to KEY16 pins are used for key input, the values having been stored in KSDR_7 are different from the previously stored values.
0	KSF0	0	R/(W)*	Key Scan Flag 0 This bit indicates whether a key scan interrupt source has been input from any of the KEY15 to KEY0 pins. 0: Indicates that no key scan interrupt source has been input from any of the KEY15 to KEY0 pins. [Clearing condition] • A 0 is written to the KSF1 bit after 1 is read from the bit. 1: Indicates that a key scan interrupt source has been input from any of the KEY15 to KEY0 pins. [Setting conditions] • When the KEY15 to KEY0 pins are used for key input, the values having been stored in KSDR_8 are different from the previously stored values.

Note: * Only 0 can be written to the flag bit after 1 is read out to clear the flag bit.

28.3.6 Key Scan Counter Control Register (KSCCR)

KSCCR is an 8-bit readable/writable register that selects the duration of the low-level signal output from the key matrix output pins; selects the intermittent pull-up period and duration for the internal pull-up MOSs of the key input pins and key matrix input pins; and selects the counter clock signal for the key scan counter (KSCNT).

KSCCR is initialized to H'00 by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSCCR is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow. KSCCR should be modified when the CST bit in KSCSR is 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	KSOC	KSPC	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	KSOC	0	R/W	Key Scan Output Control This bit selects the duration of the low-level signal output from the key matrix output pins. For details, refer to section 28.5, Setting Intermittent Pull-Up Times of Key Input and Key Matrix Input Pins, and Duration of Low-Level Signal Output from Key Matrix Pins. 0: The low-level signal is output for the duration as long as the intermittent pull-up duration. 1: The low-level signal is output for the duration as long as the intermittent pull-up period.

Bit	Bit Name	Initial Value	R/W	Description
4	KSPC	0	R/W	Key Scan Pull-up Control This bit is used in combination with the CKS[1:0] bits to select the intermittent pull-up period and duration for the internal pull-up MOSs of the key input pins and key matrix input pins. For details, refer to section 28.5, Setting Intermittent Pull-Up Times of Key Input and Key Matrix Input Pins, and Duration of Low-Level Signal Output from Key Matrix Pins. Note: When the KSE bits in KSCR1 and KSCR2 are 0, the intermittent pull-up MOSs of the corresponding key or key matrix input pins are invalid.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Counter Clock Select 1 and 0 These bits select the frequency of the counter clock signal for KSCNT. 00: 1/2 the frequency of the clock signal input via the EXTAL32 pin 01: 1/4 the frequency of the clock signal input via the EXTAL32 pin 10: 1/8 the frequency of the clock signal input via the EXTAL32 pin 11: 1/16 the frequency of the clock signal input via the EXTAL32 pin

28.3.7 Key Scan Counter (KSCNT)

KSCNT is an 8-bit read-only counter register that increments the value based on the clock signal selected by the CKS[1:0] bits in the key scan counter control register (KSCCR). KSCNT starts incrementing when the CST bit in the key scan counter start register (KSCSR) is set to 1. KSCNT is cleared to H'00 when the KSCNT value agrees with the period selected by KSCCR or 0 is written to the CST bit.

KSCNT is initialized to H'00 by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSCNT is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

28.3.8 Key Scan Counter Start Register (KSCSR)

KSCSR is an 8-bit readable/writable register that enables or disables the operation of the key scan counter (KSCNT). KSCSR is initialized to H'00 by the power-on reset signal via the $\overline{\text{RES}}$ pin or when placed in module standby mode. KSCSR is not initialized by the internal reset signal used to return from deep software standby mode or the internal reset signal caused by a WDT overflow.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CST
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	CST	0	R/W	Counter Start This bit enables or disables the operation of KSCNT. 0: Disables the counter operation. 1: Enables the counter operation.

28.4 Operation

28.4.1 Key Input

Key input is available when key input is enabled using key scan control registers 1 and 2 (KSCR1 and KSCR2) and the key input function is selected using the key scan function control register (KSFCR). It is possible to select the desired intermittent pull-up duration and period for the internal pull-up MOSs of the key input pins using the key scan counter control register (KSCCR).

At the end of pull-up duration, the values of the KEY15 to KEY0 pins are stored in key scan data register_8 (KSDR_8), the values of the KEY23 to KEY16 pins are stored in key scan data register_7 (KSDR_7); and the values of the KEY31 to KEY24 pins are stored in key scan data register_6 (KSDR_6).

If the values having been stored in each KSDR are different from the values stored in the register previously, the corresponding KSF bit in the key scan status register (KSSR) is set to 1. When the KSIE bit in the key scan interrupt enable register (KSIER) corresponding to the KSF flag bit is 1 here, an interrupt request is issued. This interrupt can be used to cancel software standby mode or deep software standby mode.

Figure 28.2 shows an example of key input operation.

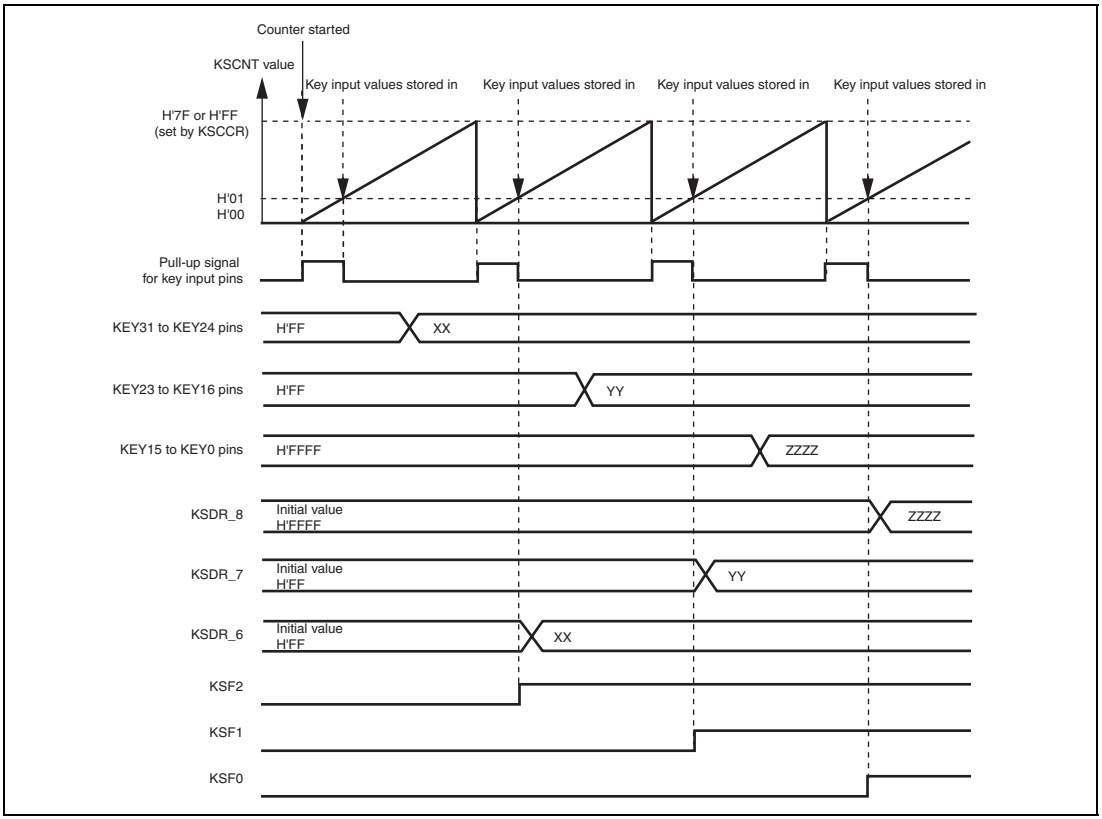


Figure 28.2 Example of Key Input Operation (32-Key Input)

(1) Key Input Setting Procedure

Figure 28.3 shows the key input setting procedure.

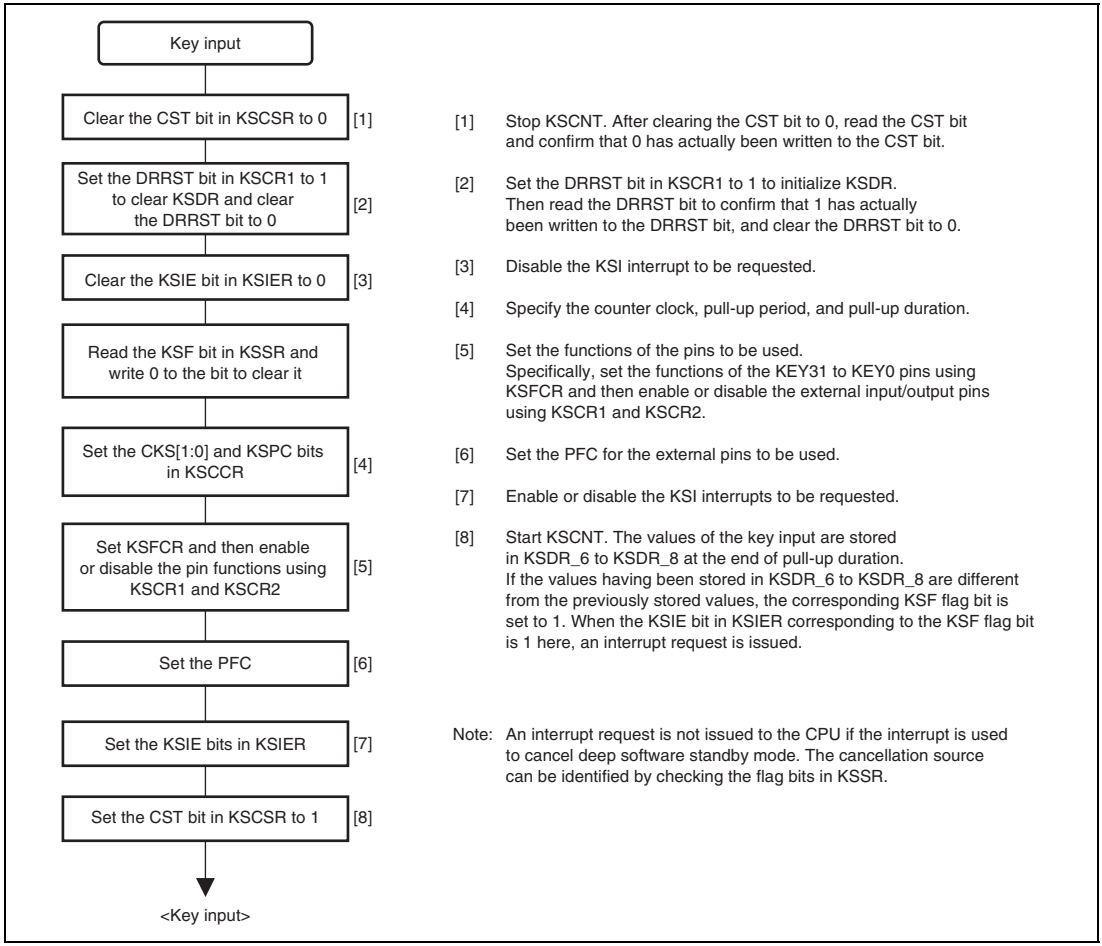


Figure 28.3 Key Input Setting Procedure

28.4.2 Key Scan

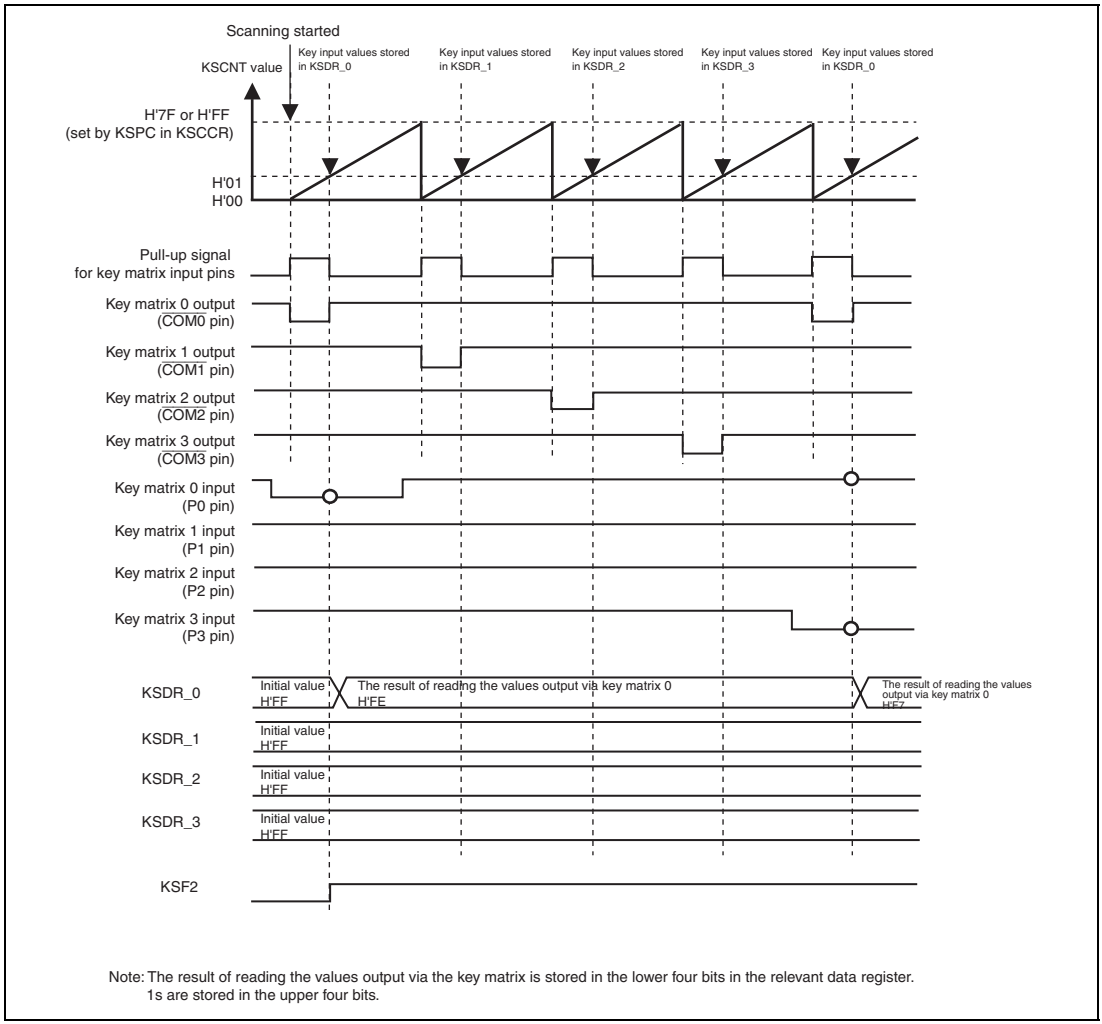
Either the 4 × 4-key matrix or 8 × 8-key matrix can be selected. Setting the KMS bit in key scan control register 1 (KSCR1) selects the 4 × 4-key matrix and setting the bit to 0 selects the 8 × 8-key matrix. The selected key matrix is available when key matrix input and output is enabled using KSCR1 and the key matrix input and output functions are selected using the key scan function control register (KSFCR). It is possible to select the desired intermittent pull-up duration and period for the internal pull-up MOSs of the key matrix input pins and the duration of the low-level signal output via the key matrix pins both using the key scan counter control register (KSCCR).

With a key matrix used, the values of the key matrix input pins are stored in key scan data registers_0 to _7 (KSDR_0 to KSDR_7) at the end of pull-up duration.

With the 4 × 4-key matrix selected, scanning is performed four times and the values of the key matrix input pins are stored in KSDR_0 to KSDR_3. Similarly, with the 8 × 8-key matrix selected, scanning is performed eight times and the values of the key matrix input pins are stored in KSDR_0 to KSDR_7.

If the values having been stored in any of KSDR_0 to KSDR_7 are different from the values stored in the same register previously, the KSF2 bit in the key scan status register (KSSR) is set to 1. When the KSIE2 bit in the key scan interrupt enable register (KSIER) is 1 here, an interrupt request is issued. This interrupt can be used to cancel software standby mode or deep software standby mode. Even when the KSF2 bit is set to 1, scanning continues until the counter is stopped.

Figure 28.4 shows an example of key scan operation.



Note: The result of reading the values output via the key matrix is stored in the lower four bits in the relevant data register. 1s are stored in the upper four bits.

Figure 28.4 Example of Key Scan Operation (4 Scan Outputs: KSOC = 1)

(1) Key Scan Setting Procedure

Figure 28.5 shows the key scan setting procedure.

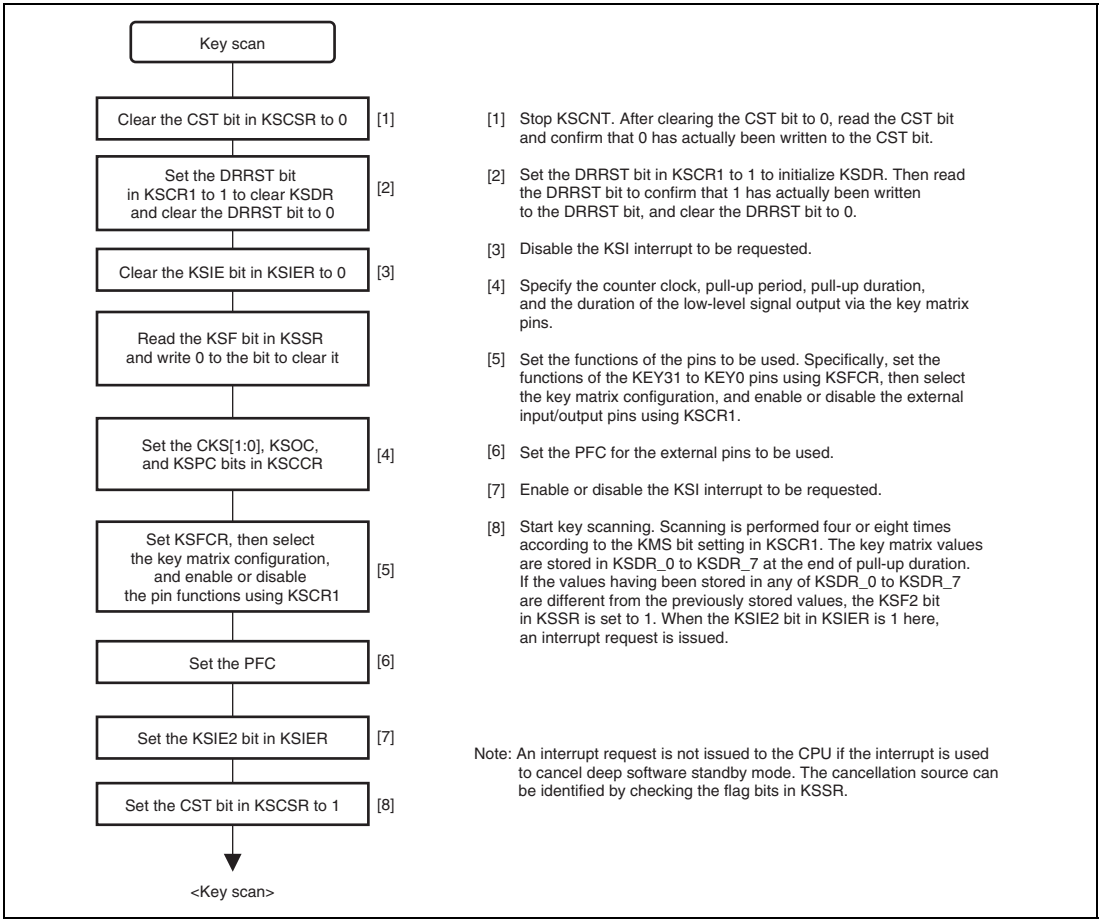


Figure 28.5 Key Scan Setting Procedure

28.5 Setting Intermittent Pull-Up Times of Key Input and Key Matrix Input Pins, and Duration of Low-Level Signal Output from Key Matrix Pins

Table 28.6 shows the bit settings and the selected intermittent pull-up times of key input and key matrix input, and duration of low-level signal output from the key matrix pins. Figures 28.6 and 28.7 show operation examples when KSPC is 0 and 1, respectively.

Table 28.6 Register Bit Settings and Selected Intermittent Pull-Up Times of Key Input and Key Matrix Input, and Duration of Low-Level Signal Output from Key Matrix Pins (32-kHz signal input via EXTAL32 pin)

		Intermittent Pull-Up Times		Duration of Low-Level Signal Output from Key Matrix Pins	
CKS[1:0] Bits	KSPC Bit	Period	Duration	KSOC = 0 (same as pull-up duration)	KSOC = 1 (same as pull-up period)
00	0	8 ms	62.5 µs	62.5 µs	8 ms
	1	16 ms			16 ms
01	0	16 ms	125 µs	125 µs	16 ms
	1	32 ms			32 ms
10	0	32 ms	250 µs	250 µs	32 ms
	1	64 ms			64 ms
11	0	64 ms	500 µs	500 µs	64 ms
	1	128 ms			128 ms

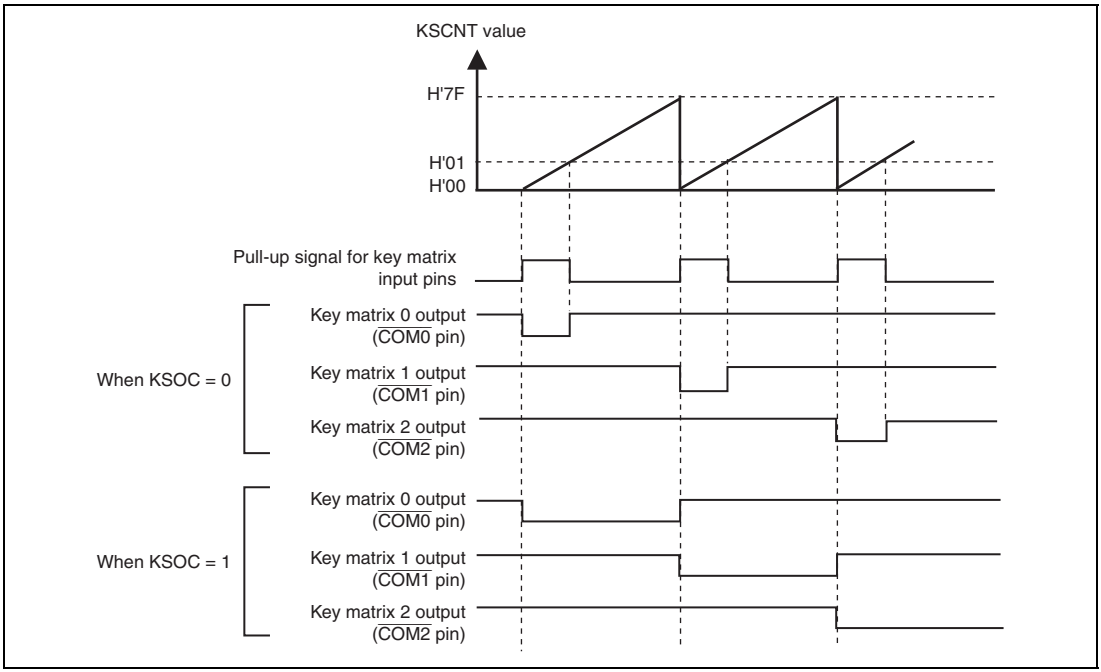


Figure 28.6 Operation Example with KSPC = 0

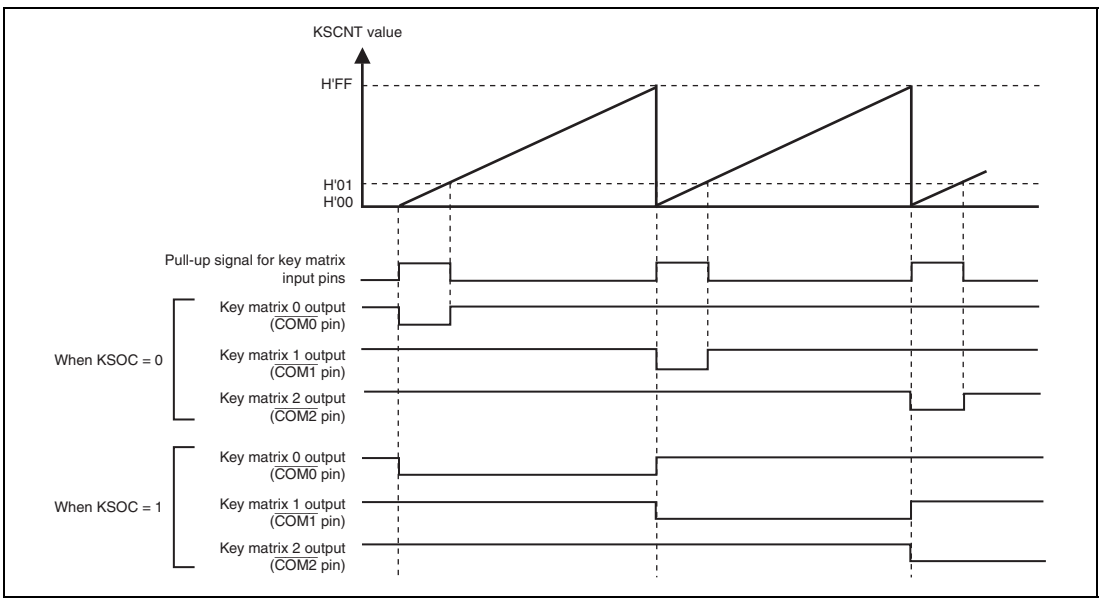


Figure 28.7 Operation Example with KSPC = 1

28.6 Interrupt Sources

Table 28.7 shows an interrupt source.

The interrupt source can be enabled or disabled according to the settings of the key scan interrupt enable bits 0 to 2 (KSIE0 to KSIE2) in the key scan interrupt enable register (KSIER).

When the key scan flag bit (KSF) in the key scan status register (KSSR) is 1 and the corresponding interrupt enable bit (KSIE) in the key scan interrupt enable register (KSIER) is 1 here, the key scan interrupt (KSI) is requested. The interrupt request is canceled when the KSF flag bit is cleared to 0.

When software standby mode is canceled using the key scan interrupt (KSI), the SSRF bit in the standby interrupt flag register (SIFR) is set to 1. For details, refer to section 32, Power-Down Modes.

Table 28.7 Interrupt Source Specifications

Interrupt Source	Interrupt Enable Bits	Interrupt Flags	Interrupt-Generation Conditions
Key scan interrupt	KSIE0, KSIE1, KSIE2	KSF0, KSF1, KSF2	KSIE0 • KSF0 + KSIE1 • KSF1 + KSIE2 • KSF2

28.7 Usage Notes

28.7.1 Setting Module Standby Mode

The KEYC module operation can be enabled or disabled using the standby control register. With the initial value, the KEYC and 32-kHz clock oscillator used by the KEYC are halted. Register access and KEYC operation is enabled by canceling module standby mode. For details, refer to section 32, Power-Down Modes.

When the 32-kHz crystal resonator is used, secure the EXTAL32 clock settling time (tOSC32) after module standby mode is canceled before accessing the KEYC registers. Otherwise, the KEYC registers cannot be accessed correctly. For details, refer to figure 35.10 in section 35, Electrical Characteristics.

28.7.2 Operation after a Key Scan Interrupt is Generated

Even after key input or a key scan interrupt is generated, operation continues. Therefore, if multiple sources of key input or key scan input are generated, the data register used for identifying the interrupt source is overwritten each time.

28.7.3 When a Pin Function other than Key Input or Key Matrix Input is Selected by PFC

When a pin function other than key input or key matrix input is selected for any of the key input pins or key matrix input pins using the pin function controller (PFC), a 1 is always stored in the corresponding bit in key scan data registers_0 to _8 (KSDR_0 to KSDR_8). The intermittent pull-up MOS of the corresponding pin is invalid. When a pin function other than key matrix output is selected for any of the key matrix output pins using the PFC, no scanned data is output from the corresponding pin.

28.7.4 Modifying KSE Bits in KSCR1 and KSCR2 during Key Input and Key Scan Operation

When the KSE bits in key scan control registers 1 and 2 (KSCR1 and KSCR2) are modified from 1 (function enabled) to 0 (function disabled) during key input and key scan operation, a 1 is always read from the corresponding pin. Therefore, if the value previously stored in KSDR is 0, the KSF flag in the key scan status register (KSSR) is set. When the interrupt request is enabled using the key scan interrupt enable register (KSIER) here, an interrupt request is issued to the CPU. To prevent an interrupt request to be issued in such a case, disable the interrupt request using KSIER before modifying KSCR1 and KSCR2.

28.7.5 Restarting Suspended Counter

Before restarting the suspended counter, clear the CST bit in the key scan counter start register (KSCSR) and read the CST bit to confirm that 0 has actually been written to the bit.

28.7.6 Consecutive Writing to the Same KEYC Register

The value written to a register is actually used for the operation after synchronization based on the KEYC operating clock signal (the clock signal obtained by dividing EXTAL32 input signal by two). Therefore, when writing to the same KEYC register consecutively, wait one or more KEYC operating clock cycle after a write access before the next write access, or read the register before the next write access to confirm that the desired value has actually been written to the register.

28.7.7 Operation while DRRST Bit is 1

With the DRRST bit in key scan control register 1 (KSCR1) set to 1, key scan data registers_0 to _8 (KSDR_0 to KSDR_8) are initialized: therefore, clear the DRRST bit to 0 and start the counter (set the CST bit in the key scan counter start register (KSCSR) to 1). When clearing KSDR_0 to KSDR_8, first set the DRRST bit to 1, read the DRRST bit to confirm that 1 has actually been written to the bit, and clear the DRRST bit to 0 again.

28.7.8 Output Levels on the $\overline{\text{COM}}$ Pins When the Kay Matrix is in Use

The $\overline{\text{COM}}$ pins that act as outputs for the key matrix only output the low level to the pins that handle scan operation and output the high level for other pins. Since selecting the switch for a different $\overline{\text{COM}}$ pin at the same time may lead to a conflict between the output levels, include diodes as a measure to prevent conflict between output levels on $\overline{\text{COM}}$ pins.

Section 29 Flash Memory (ROM)

The SH72315A and SH72315L incorporate up to 1 Mbyte and the SH72314L incorporates 768 Kbytes of flash memory (ROM) for the storage of program code. The ROM has the following features.

29.1 Features

- Two types of flash-memory MATs

  The ROM has two types of memory areas (hereafter referred to as memory MATs) in the same address space. These two MATs can be switched by the start-up mode or bank switching through the control register. For addresses H'00008000 to H'000FFFFF, undefined data is read and programming and erasing are ignored when the user boot MAT is selected.

  User MAT: 1 Mbyte (SH72315A, SH72315L), 768 Kbytes (SH72314L)

  User boot MAT: 32 Kbytes

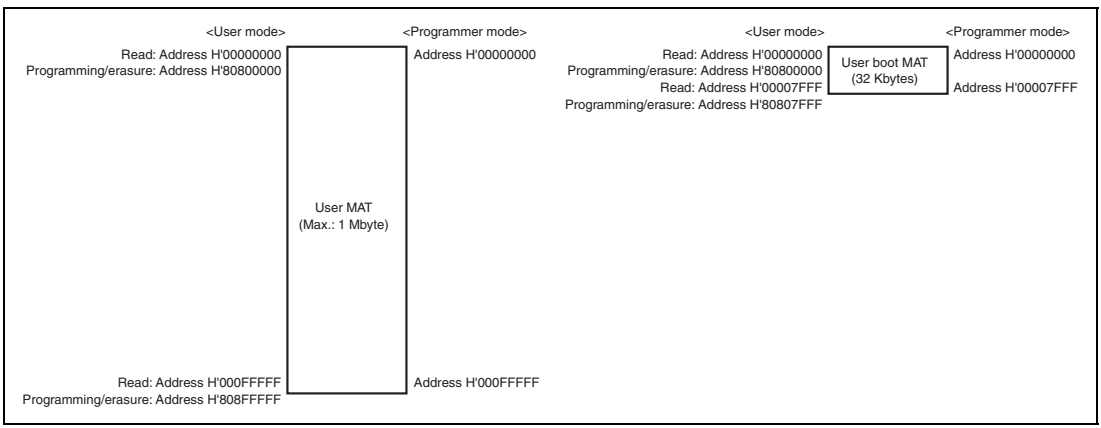
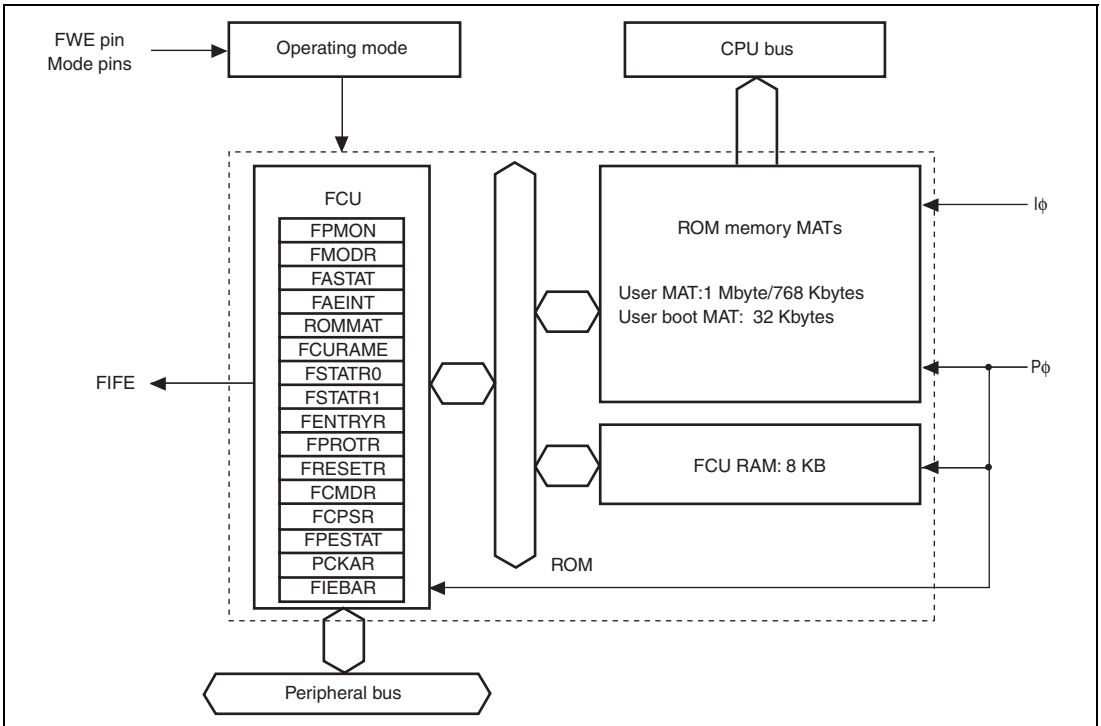


Figure 29.1 Memory MAT Configuration in ROM

- Programming and erasing methods

  The ROM can be programmed and erased by commands issued through the peripheral bus to the ROM/data flash (FLD) dedicated sequencer (FCU).

  While the flash control unit (FCU) is programming or erasing the ROM, the CPU can execute a program located outside the ROM. While the FCU is programming or erasing the FLD, the CPU can execute a program in the ROM. When the FCU suspends programming or erasure, the CPU can execute a program in the ROM, and then the FCU can resume programming or erasure. While the FCU suspends erasure, areas other than the erasure-suspended area can be programmed.



[Legend]

FPMON:	Flash pin monitor register
FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
ROMMAT:	ROM MAT select register
FCURAME:	FCU RAM enable register
FSTATR0, FSTATR1:	Flash status registers 0 and 1
FENTRYR:	Flash P/E mode entry register
FPROTR:	Flash protect register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FCPSR:	FCU processing switch register
FPESTAT:	Flash P/E status register
PCKAR:	Peripheral clock notification register
FIEBAR:	Erasure block notification register

Figure 29.2 Block Diagram of ROM

- Programming/erasing unit

  The user MAT and user boot MAT are programmed in 256-byte units. The entire area of the user boot MAT is always erased at one time. The user MAT can be erased in block units if the mode is not programmer mode. The entire area of the user MAT is erased in programmer mode.

  Figure 29.3 shows the block configuration of the user MAT. The user MAT is divided into eight 8-Kbyte blocks, nine 64-Kbyte blocks, two 128-Kbyte blocks, and two 256-Kbyte blocks in the SH72315A and SH72315K or one 256-Kbyte block in the SH72314L.

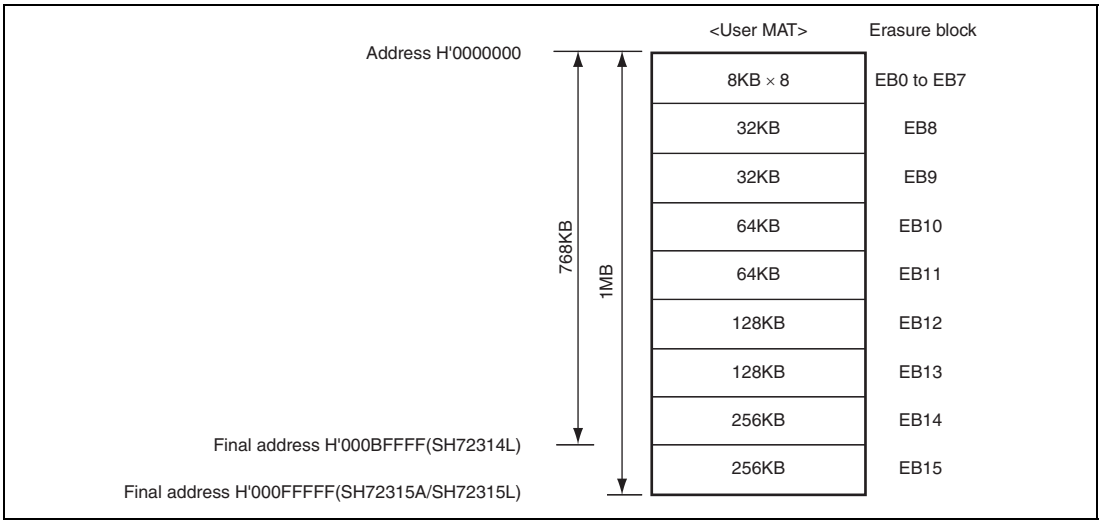


Figure 29.3 Block Configuration of User MAT

- Three types of on-board programming modes
  - Boot mode

    The user MAT and user boot MAT can be programmed using the SCI. The bit rate for SCI communications between the host and this LSI can be automatically adjusted.
  - User program mode

    The user MAT can be programmed with a desired interface. A transition from MCU mode 2 (MCU extended mode) or mode 3 (MCU single-chip mode) to this mode is enabled simply by changing the level on the FWE pin.
  - User boot mode

    The user MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.
- One type of off-board programming mode
  - Programmer mode

    The user MAT and user boot MAT can be programmed in programmer mode using the PROM programmer.
- Protection modes

  This LSI supports two modes to protect memory against programming or erasure: hardware protection by the levels on the FWE and mode pins and software protection by the FENTRY0 bit in FENTRYR or lock bit settings. The FENTRY0 bit enables or disables ROM programming or erasure by the FCU. A lock bit is included in each erasure block of the user MAT to protect memory against programming or erasure.

  The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure.
- Programming and erasing time and count

  Refer to section 35, Electrical Characteristics.

29.2 Input/Output Pins

Table 29.1 shows the input/output pins used for the ROM. The combination of MD1 and MD0 pin levels and the FWE pin level determines the ROM programming mode (see section 29.4, Overview of ROM-Related Modes). In boot mode, the ROM can be programmed or erased by the host connected via the PA0/RxD1 and PA1/TxD0 pins (see section 29.5, Boot Mode).

Table 29.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	This LSI enters the power-on reset state when this signal goes low.
Test reset	$\overline{\text{TRST}}$	Input	Input pin for the initialization signal.
Flash programming enable	FWE	Input	Hardware protection against reprogramming of flash memory.
Mode 1	MD1	Input	Set the operating mode for the LSI.
Mode 0	MD0	Input	Set the operating mode for the LSI.
SCI transmit data	TXD0(PA0)	Input	Output for transmission of serial data (for use in boot mode)
SCI receive data	RXD0(PA0)	Output	Input for receiving serial data (for use in boot mode)

29.3 Register Descriptions

Table 29.2 shows the ROM-related registers. Some of these registers have data flash (FLD) related bits, but this section only describes the ROM-related bits. For the FLD-related bits, refer to section 30.3, Register Descriptions. The ROM-related registers are initialized by a power-on reset.

Refer to 34, List of Registers, for the states of these registers in the various states of processing.

Table 29.2 Register Configuration

Register Name	Symbol	R/W*1	Initial Value	Address	Access Size
Flash pin monitor register	FPMON	R	H'00 H'80	H'FFFFA800	8
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*2	H'00	H'FFFFA810	8
Flash access error interrupt enable register	FAEINT	R/W	H'9F	H'FFFFA811	8
ROM MAT select register	ROMMAT	R/(W)*3	H'0000 H'0001	H'FFFFA820	8, 16
FCU RAM enable register	FCURAME	R/(W)*3	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*5	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*5	H'FFFFA901	8
Flash P/E mode entry register	FENTRYR	R/(W)*4	H'0000*5	H'FFFFA902	8, 16
Flash protect register	FPROTR	R/(W)*4	H'0000*5	H'FFFFA904	8, 16
Flash reset register	FRESETR	R/(W)*3	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*5	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W	H'0000*5	H'FFFFA918	8, 16
Flash P/E status register	FPESTAT	R	H'0000*5	H'FFFFA91C	8, 16
Peripheral clock notification register	PCKAR	R/W	H'0000*5	H'FFFFA938	8, 16
Erasure block notification register	FIEBAR	R/W	H'0000*5	H'FFFFA93A	8, 16

Notes: 1. In on-chip ROM disabled mode, the ROM-related registers are always read as 0 and writing to them is ignored.
2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.

3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.
4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.
5. These registers can be initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

29.3.1 Flash Pin Monitor Register (FPMON)

FPMON monitors the FWE pin state. FPMON is read as H'00 in on-chip ROM disabled mode. FPMON is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	FWE	—	—	—	—	—	—	—
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Write Enable Monitors the FWE pin level. The initial value depends on the FWE pin level. 0: Disables ROM programming and erasure 1: Enables ROM programming and erasure
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

29.3.2 Flash Mode Register (FMODR)

FMODR specifies the FCU operation mode. In on-chip ROM disabled mode, FMODR is read as H'00 and writing to it is ignored. FMODR is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FRDMD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	FRDMD	0	R/W	FCU Read Mode Select The bit selects permission for or prohibition of reading of the ROM/FLD by the FCU. In the case of the ROM, this bit specifies permission or prohibition of lock bit reading (see section 29.6.1, FCU Command List, and section 29.6.3 (13), Reading Lock Bit), whereas this bit must be set to make the blank check command available for use in the FLD (see section 30, Data Flash (FLD)). 0: Prohibit lock bit reading. 1: Permit look bit reading.
3 to 0	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.

29.3.3 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and FLD. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 29.9.3, Error Protection). To cancel a command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	Access Error Indicates whether or not a ROM access error has been generated. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters a command-locked state. 0: No ROM access error has occurred. 1: A ROM access error has occurred. [Setting conditions] • An access command is issued to ROM program/erase addresses H'80800000 to H'808FFFFF while the FENTRY0 bit in FENTRYR is 1 in ROM P/E normal mode. • An access command is issued to ROM program/erase addresses H'80800000 to H'808FFFFF while the FENTRY0 bit in FENTRYR is 0. • A read access command is issued to ROM read addresses H'00000000 to H'000FFFFF while the FENTRYR register value is not H'0000. • A block erase, program, or lock bit program command is issued while the user boot MAT is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	• An access command is issued to an address other than ROM program/erase addresses H'80800000 to H'80807FFF while the user boot MAT is selected. [Clearing condition] • A 0 is written to this bit after reading a 1 from the ROMAE bit.
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLK	0	R	FCU Command Lock Indicates whether the FCU is in command-locked state (see section 29.9.3, Error Protection). 0: The FCU is not in a command-locked state 1: The FCU is in a command-locked state [Setting condition] • The FCU detects an error and enters command-locked state. [Clearing condition] • The FCU completes the status-clear command processing while FASTAT is H'10.
3	EEPAE	0	R/(W)*	FLD Access Error Refer to section 30, Data Flash (FLD).
2	EEPIFE	0	R/(W)*	FLD Instruction Fetch Error Refer to section 30, Data Flash (FLD).
1	EEPRPE	0	R/(W)*	FLD Read Protect Error Refer to section 30, Data Flash (FLD).
0	EEPWPE	0	R/(W)*	FLD Program/Erase Protect Error Refer to section 30, Data Flash (FLD).

Note: * Only 0 can be written to clear the flag after 1 is read.

29.3.4 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupts. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
Initial value:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAEIE	1	R/W	ROM Access Error Interrupt Enable Enables or disables an FIFE interrupt request when a ROM access error occurs and the ROMAE bit in FASTAT becomes 1. 0: Prohibit FIFE interrupt requests when ROMAE = 1. 1: Permit FIFE interrupt requests when ROMAE = 1.
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLKIE	1	R/W	FCU Command Lock Interrupt Enable Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. 0: Prohibit FIFE interrupt requests when CMDLK = 1 1: Permit FIFE interrupt requests when CMDLK = 1
3	EEPAEIE	1	R/W	FLD Access Error Interrupt Enable Refer to section 30, Data Flash (FLD).
2	EEPIFEIE	1	R/W	FLD Instruction Fetch Error Interrupt Enable Refer to section 30, Data Flash (FLD).
1	EEPRPEIE	1	R/W	FLD Read Protect Error Interrupt Enable Refer to section 30, Data Flash (FLD).
0	EEPWPEIE	1	R/W	FLD Program/Erase Protect Error Interrupt Enable Refer to section 30, Data Flash (FLD).

29.3.5 ROM MAT Select Register (ROMMAT)

ROMMAT switches memory MATs in the ROM. In on-chip ROM disabled mode, ROMMAT is read as H'0000 and writing to it is ignored. ROMMAT is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	ROM SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable ROMSEL bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ROMSEL	0/1	R/W	ROM MAT Select Selects a memory MAT in the ROM. The initial value is 1 when the LSI is started in user boot mode; otherwise, the initial value is 0. Writing to this bit is enabled only when this register is accessed in word size and H'3B is written to the KEY bits. 0: Selects the user MAT 1: Selects the user boot MAT

Note: * Write data is not retained.

29.3.6 FCU RAM Enable Register (FCURAME)

FCURAME enables or disables access to the FCU RAM area. In on-chip ROM disabled mode, FCURAME is read as H'00 and writing to it is ignored. FCURAME is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	FCRME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable FCRME bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FCRME	0	R/W	FCU RAM Enable Enables or disables access to the FCU RAM. Writing to this bit is enabled only when this register is accessed in word size and H'C4 is written to the KEY bits. Before writing to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. 0: Disables access to FCU RAM 1: Enables access to FCU RAM

Note: * Write data is not retained.

29.3.7 Flash Status Register 0 (FSTATR0)

FSTATR0 indicates the FCU status. In on-chip ROM disabled mode, FSTATR0 is read as H'00. FRTATR0 is initialized by a power-on reset, or setting the FRESET bit of the FRESETR register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	FRDY	ILG LERR	ERS ERR	PRG ERR	SUS RDY	—	ERS SPD	PRG SPD
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FRDY	1	R	Flash Ready Indicates the processing state in the FCU. 0: Programming or erasure processing, programming or erasure suspension processing, lock bit read 2 command processing, or FLD blank check is in progress (see section 30, Data Flash (FLD)). 1: None of the above is in progress.
6	ILGLERR	0	R	Illegal Command Error Indicates that the FCU has detected an illegal command or illegal ROM or FLD access. When this bit is 1, the FCU is in command-locked state (see section 29.9.3, Error Protection). 0: The FCU has not detected any illegal command or illegal ROM/FLD access 1: The FCU has detected an illegal command or illegal ROM/FLD access [Setting conditions] • The FCU has detected an illegal command. • The FCU has detected an illegal ROM/FLD access (the ROMAE, EEPAE, EEPIFE, EEPRPE, or EEPWPE bit in FASTAT is 1). • The FENTRYR setting is illegal. • In erasure, reading of the lock bits, or programming of the lock bits, command was issued at an address outside the erasure block set in FIEBAR. [Clearing condition] • The FCU completes the status-clear command processing while FASTAT is H'10.

Bit	Bit Name	Initial Value	R/W	Description
5	ERSERR	0	R	Erasure Error Indicates the result of ROM or FLD erasure by the FCU. When this bit is 1, the FCU is in command-locked state (see section 29.9.3, Error Protection). 0: Erasure processing has been completed successfully 1: An error has occurred during erasure [Setting conditions] • An error has occurred during erasure. • A block erase command has been issued for the area protected by a lock bit. [Clearing condition] • The FCU completes the status-clear command processing.
4	PRGERR	0	R	Programming Error Indicates the result of ROM or FLD programming by the FCU. When this bit is 1, the FCU is in command-locked state (see section 29.9.3, Error Protection). 0: Programming has been completed successfully 1: An error has occurred during programming [Setting conditions] • An error has occurred during programming. • A programming command has been issued for the area protected by a lock bit. [Clearing condition] • The FCU completes the status-clear command processing.

Bit	Bit Name	Initial Value	R/W	Description
3	SUSRDY	0	R	Suspend Ready Indicates whether the FCU is ready to accept a P/E suspend command. 0: The FCU cannot accept a P/E suspend command 1: The FCU can accept a P/E suspend command [Setting condition] • After initiating programming/erasure, the FCU has entered a state where it is ready to accept a P/E suspend command. [Clearing conditions] • The FCU has accepted a P/E suspend command. • The FCU has entered a command-locked state during programming or erasure.
2	—	0	R	Reserved This bit is always read as 0. Correct operation is not guaranteed if 1 is written to this bit.
1	ERSSPD	0	R	Erasure-Suspended Status Indicates that the FCU has entered an erasure suspension process or an erasure-suspended status (see section 29.6.4, Suspending Operation). 0: The FCU is in a status other than the below-mentioned. 1: The FCU is in an erasure suspension process or an erasure-suspended status. [Setting condition] • The FCU has initiated an erasure suspend command. [Clearing condition] • The FCU has accepted a resume command.

Bit	Bit Name	Initial Value	R/W	Description
0	PRGSPD	0	R	Programming-Suspended Status Indicates that the FCU has entered a write suspension process or a write suspend status (see section 29.6.4, Suspending Operation). 0: The FCU is in a status other than the below-mentioned. 1: The FCU is in a write suspension process or a write-suspended status. [Setting condition] • The FCU has initiated a write suspend command. [Clearing condition] • The FCU has accepted a resume command.

29.3.8 Flash Status Register 1 (FSTATR1)

FSTATR1 indicates the FCU status. In on-chip ROM disabled mode, FSTATR1 is read as H'00. FSTATR1 is initialized by a power-on reset, or setting the FRESET bit of the FRESETR register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	FCUERR	—	—	FLOCKST	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FCUERR	0	R	FCU Error Indicates an error has occurred during the CPU processing in the FCU. 0: No error has occurred during the CPU processing in the FCU 1: An error has occurred during the CPU processing in the FCU [Clearing condition] • The FRESET bit in FRESETR is set to 1. When FCUERR is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FLOCKST	0	R	Lock Bit Status Reflects the lock bit data read through lock bit read 2 command execution. When the FRDY bit becomes 1 after the lock bit read 2 command is issued, valid data is stored in this bit. This bit value is retained until the next lock bit read 2 command is completed. 0: Protected state 1: Non-protected state

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

29.3.9 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or FLD. To specify the P/E mode for the ROM or FLD so that the FCU can accept commands, set either of FENTRYD and FENTRY0 bits to 1. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR can be initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

In access to the FENTRYR for a mode transition of the FCU, write to the register and then read it, and only proceed with programming, erasure or reading of the ROM after confirming the register setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FEKEY								FENTRYD	—	—	—	—	—	—	FENTRY0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FEKEY	All 0	R/(W)*	Key Code These bits enable or disable rewriting of the FENTRYD and FENTRY0 bits. Data written to these bits are not retained.
7	FENTRYD	0	R/W	FLD P/E Mode Entry Bit Refer to section 30, Data Flash (FLD).
6 to 1	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
0	FENTRY0	0	R/W	ROM P/E Mode Entry Bit 0 These bits specify the P/E mode for ROM. 0: ROM is in read mode 1: ROM is in P/E mode Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Note: * Write data is not retained.

29.3.10 Flash Protect Register (FPROTR)

FPROTR enables or disables the protection function through the lock bits against programming and erasure. In on-chip ROM disabled mode, FPROTR is read as H'0000 and writing to it is ignored. FPROTR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FPKEY								—	—	—	—	—	—	—	FPROTCN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FPKEY	All 0	R/(W)*	Key Code These bits enable or disable FPROTCN bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FPROTCN	0	R/W	Lock Bit Protect Cancel Enables or disables protection through the lock bits against programming and erasure. 0: Enables protection through the lock bits 1: Disables protection through the lock bits [Setting condition] • H'55 is written to FPKEY and 1 is written to FPROTCN in word access while the FENTRYR register value is not H'0000. [Clearing conditions] • This register is written to in byte access. • A value other than H'55 is written to FPKEY in word access. • H'55 is written to FPKEY and 0 is written to FPROTCN in word access. • The FENTRYR register value is H'0000.

Note: * Write data is not retained.

29.3.11 Flash Reset Register (FRESETR)

FRESETR is used for the initialization of FCU. In on-chip ROM disabled mode, FRESETR is read as H'0000 and writing to it is ignored. FRESETR is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRKEY								—	—	—	—	—	—	—	FRESET
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

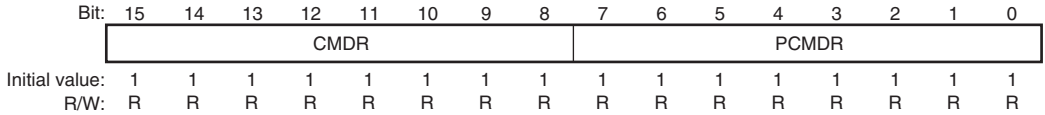
Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FRKEY	All 0	R/(W)*	Key Code These bits enable or disable FRESET bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FRESET	0	R/W	Flash Reset Setting this bit to 1 forcibly terminates programming/erasure of ROM or FLD and initializes the FCU. A high voltage is applied to the ROM/FLD memory units during programming and erasure. To ensure sufficient time for the voltage applied to the memory unit to drop, keep the value of the FRESET bit at 1 for a period of t_{RESW2} (see section 35, Electrical Characteristics) when the FCU is initialized. Do not read from the ROM/FLD units while the value of the FRESET bit is kept at 1. The FCU commands are unavailable for use while the FRESET bit is set to 1, since this initializes the FENTRYR register. This bit can be written only when H'CC is written to FRKEY in word access. 0: Issue no reset to the FCU. 1: Issues a reset to the FCU.

Note: * Write data is not retained.

29.3.12 FCU Command Register (FCMDR)

FCMDR stores the commands that the FCU has accepted. In on-chip ROM disabled mode, FCMDR is read as H'0000 and writing to it is ignored. FCMDR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.



Bit	Bit Name	Initial Value	R/W	Description
15 to 8	CMDR	H'FF	R	Command Register These bits store the latest command accepted by the FCU.
7 to 0	PCMDR	H'FF	R	Precommand Register These bits store the previous command accepted by the FCU.

Table 29.3 shows the states of FCMDR after acceptance of the various commands.

Table 29.3 FCMDR Status after a Command is Accepted

Command	CMDR	PCMDR
Normal mode transition	H'FF	Previous command
Status read mode transition	H'70	Previous command
Lock bit read mode transition	H'71	Previous command
Program	H'E8	Previous command
Block erase	H'D0	H'20
P/E suspend	H'B0	Previous command
P/E resume	H'D0	Previous command
Status register clear	H'50	Previous command
Lock bit read 2 blank check	H'D0	H'71
Lock bit program	H'D0	H'77
Peripheral clock notification	H'E9	Previous command

29.3.13 FCU Processing Switch Register (FCPSR)

FCPSR selects a function to make the FCU suspend erasure. In on-chip ROM enabled mode, FCPSR is read as H'0000 and writing to it is ignored. FCPSR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSPMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ESUSPMD	0	R/W	Erasure-Suspended Mode Selects the erasure-suspended mode to be entered when a P/E suspend command is issued while the FCU is erasing the ROM or FLD (see section 29.6.4, Suspending Operation). 0: Suspension-priority mode 1: Erasure-priority mode

29.3.14 Flash P/E Status Register (FPESTAT)

FPESTAT indicates the result of programming/erasure of the ROM/FLD. In on-chip ROM enabled mode, FPESTAT is read as H'0000 and writing to it is ignored. FPESTAT is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PEERRST							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	PEERRST	H'00	R	P/E Error Status Indicates the source of an error that occurs during programming/erasure. This bit value is only valid if the PRGERR or ERSERR bit value in FSTATR0 is 1; otherwise the bit retains the value to indicate the source of an error that previously occurred. H'01: A write attempt made to an area protected by the lock bits H'02: A write error caused by other source than the above H'11: An erase attempt made to an area protected by the lock bits H'12: An erase error caused by other source than the above Other than above: Reserved

29.3.15 Peripheral Clock Notification Register (PCKAR)

PCKAR is used to notify the sequencer of information regarding the frequency setting of the peripheral clock (Pϕ) for programming or erasure of the ROM or data flash memory. The setting governs the time programming or erasure takes. In modes where the internal ROM is disabled, the value read from the PCKAR will be H'0000 and writing to the PCKAR will be ineffective. PCKAR is initialized by a power-on reset or by writing 1 to the FRESET bit in FRESETR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCKA							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. When writing to the register, always write 0 to these bits. Operation is not guaranteed if 1 is written to any or all of these bits.
7 to 0	PCKA	H'00	R/W	Peripheral Clock Notification These bits are used to notify the peripheral clock (Pϕ) for programming or erasure of the ROM/FLD. Set the frequency of Pϕ by setting these bits before programming or erasure, and then issue a peripheral clock notification command. Do not change the frequency while the ROM/FLD is being programmed or erased. Follow the procedure below to calculate the setting. • Convert the frequency expressed in MHz units to binary notation, and write the value to the PCKA bits. For example, if the frequency of the peripheral clock is 35.9 MHz, the setting is derived as follows. • Round 35.9 up to obtain 36. • Convert 36 into binary form and set the PCKA bits to H'24 (B'00100100). Notes: 1. Do not issue the command for overwriting the ROM or data flash memory if the setting of the PCKA bits is for a frequency outside the range from 20 to 50 MHz. 2. If the frequency set by the PCKA bits differs from the actual frequency, there is a possibility of destroying the ROM/FLD.

29.3.16 Erasure block notification register (FIEBAR)

FIEBAR notifies the sequencer of erasure block information. Setting is required when erasing the user MAT and when programming or erasing lock bits. In operating modes where the internal ROM is disabled, the value read from FIEBAR will be H'0000 and writing to FIEBAR will be ineffective. FIEBAR is initialized by a power-on reset or by writing 1 to the FRESET bit in FRESETR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FIEBAR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. When writing to the register, always write 0 to these bits. Operation is not guaranteed if 1 is written to any or all of these bits.
7 to 0	FIEBAR [7:0]	H'00	R/W	Erasure block notification bit The bit notifies the sequencer of an erasure block. Issue commands after setting the erasure block information corresponding to the target erasure block. The correspondence between values of the erasure block information and erasure blocks is as follows; Set H'00 for EB0 Set H'01 for EB1 Set H'02 for EB2 Set H'03 for EB3 Set H'04 for EB4 Set H'05 for EB5 Set H'06 for EB6 Set H'07 for EB7 Set H'08 for EB8 Set H'09 for EB9 Set H'0A for EB10 Set H'0B for EB11 Set H'0C for EB12 Set H'0D for EB13 Set H'0E for EB14 Set H'0F for EB15

29.4 Overview of ROM-Related Modes

Figure 29.4 shows the ROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD1, MD0 and FWE pin settings, refer to section 4, MCU Operating Modes.

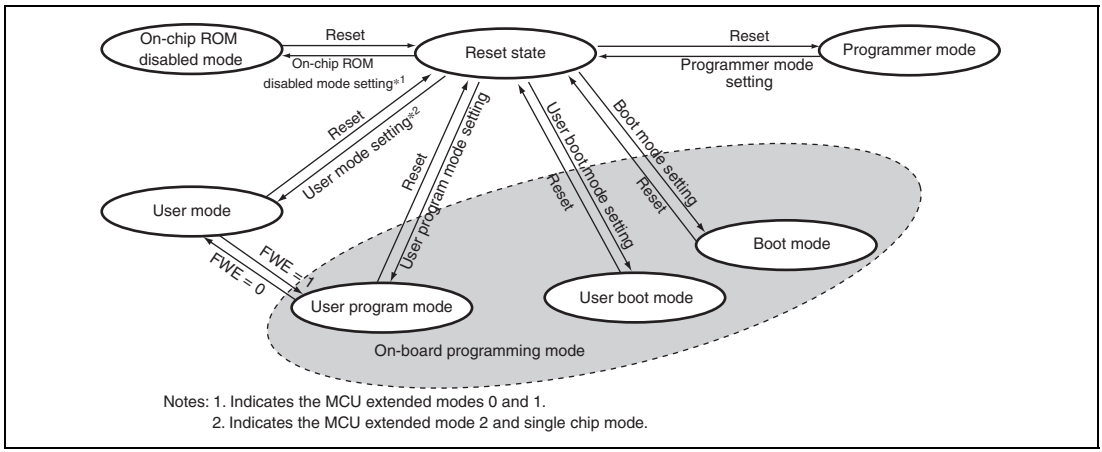


Figure 29.4 ROM-Related Mode Transition

- The ROM cannot be read, programmed, or erased in on-chip ROM disabled mode (MCU extended modes 0 and 1).
- The ROM can be read but cannot be programmed or erased in user mode (MCU extended mode 2 and single chip mode).
- The ROM can be read, programmed, and erased on the board in user program mode, user boot mode, and boot mode.

Table 29.4 compares programming- and erasure-related items for the boot mode, user program mode, user boot mode, and programmer mode.

Table 29.4 Comparison of Programming Modes

Item	Boot Mode	User Program Mode	User Boot Mode	Programmer Mode
Programming/ erasure environment	On-board programming			Off-board programming
Programming/ erasure enabled MAT	User MAT and user boot MAT	User MAT	User MAT	User MAT and user boot MAT
Programming/ erasure control	Host	FCU	FCU	Programmer
Entire area erasure	Available (automatic)	Available	Available	Available (automatic)
Block erasure	Available*1	Available	Available	Not available
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM	Via programmer
Reset-start MAT	Embedded program stored MAT	User MAT	User boot MAT*2	Embedded program stored MAT
Transition to MCU operating mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	—

Notes: 1. The entire area is erased when the LSI is started. After that, a specified block can be erased.
2. After the LSI is started in the embedded program stored MAT and the boot program provided by Renesas Corp. is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- In boot mode, the user MAT, user boot MAT, and FLD data MAT are all erased immediately after the LSI is started. The user MAT, user boot MAT, and data MAT can then be programmed from the host via the SCI. The ROM can also be read after this entire area erasure.
- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user program mode.

29.5 Boot Mode

29.5.1 System Configuration

To program or erase the user MAT and user boot MAT in boot mode, send control commands and programming data from the host. The on-chip SCI of this LSI is used in asynchronous mode for communications between the host and this LSI. The tool for sending control commands and programming data must be prepared in the host. When this LSI is started in boot mode, the program in the embedded program stored MAT is executed. This program automatically adjusts the SCI bit rate and performs communications between the host and this LSI by means of the control command method.

Figure 29.5 shows the system configuration in boot mode. The NMI and any other interrupts are ignored in this mode, but these pins must be fixed to non-active state. Note that the AUD cannot be used in this mode.

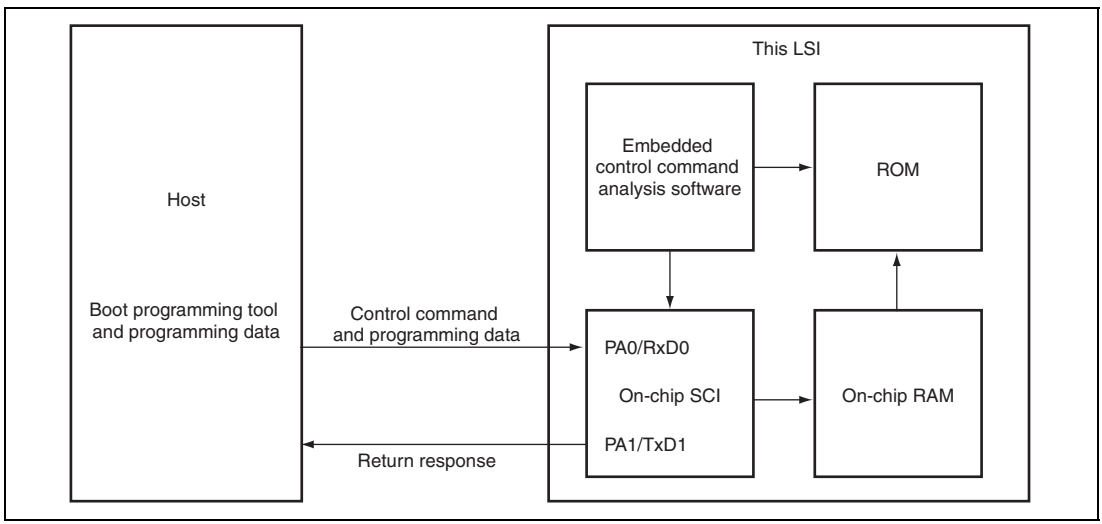


Figure 29.5 System Configuration in Boot Mode

29.5.2 State Transition in Boot Mode

Figure 29.6 shows the state transition in boot mode.

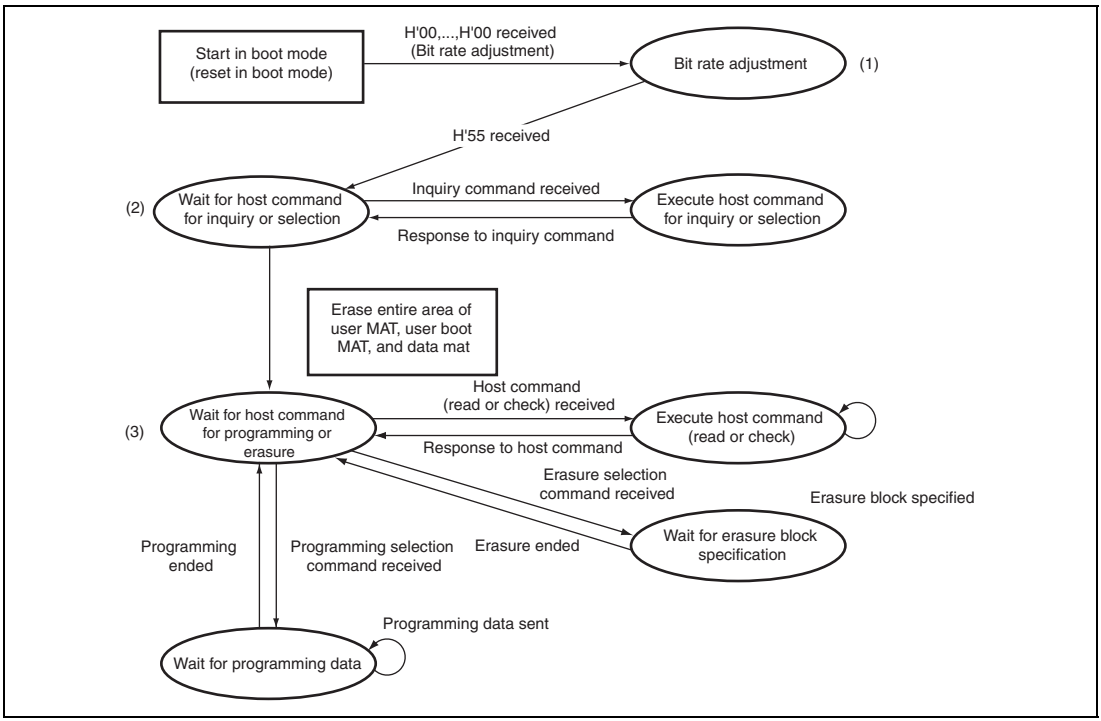


Figure 29.6 State Transition in Boot Mode

(1) Bit Rate Adjustment

After this LSI is started in boot mode, it automatically adjusts the bit rate for communications between the host and SCI. After automatic adjustment of the bit rate, the LSI sends H'00 to the host. After the LSI has successfully received H'55 sent from the host, the LSI waits for a host command for inquiry or selection. For details on bit rate adjustment, see section 29.5.3, Automatic Adjustment of Bit Rate.

(2) Waiting for Host Command for Inquiry or Selection

In this state, the host inquires regarding MAT information (such as the size, configuration, and start address) and the supported functions, and selects the device, clock mode, and bit rate. Upon reception of a programming/erasure state transition command sent from the host, this LSI erases the entire area of each of the user MAT, user boot MAT, and FLD data MAT and waits for a host command for programming or erasure. For details of inquiry/selection host commands, see section 29.5.4, Inquiry/Selection Host Command Wait State.

(3) Waiting for Host Command for Programming or Erasure

In this state, this LSI performs programming or erasure according to the command sent from the host. The LSI enters programming data wait state, erasure block specification wait state, or command (read or check) processing state depending on the received command.

Upon reception of a programming selection command, the LSI waits for programming data. After the programming selection command, send the programming start address and programming data from the host. Specifying H'FFFFFFFF as the programming start address terminates programming processing and the LSI makes a transition from the programming data wait state to programming/erasure command wait state.

Upon reception of an erasure selection command, the LSI waits for erasure block specification. After the erasure selection command, send the erasure block number from the host. Specifying H'FF as the erasure block number terminates erasure processing and the LSI makes a transition from the erasure block specification wait state to programming/erasure command wait state. As the entire area of each of the user MAT, user boot MAT, and FLD data MAT is erased before the LSI enters programming/erasure command wait state after it is started in boot mode, erasure processing is not needed except for the case when the data programmed in boot mode should be erased without resetting the LSI.

In addition to programming and erasing commands, many other host commands are provided for use in programming/erasure command wait state; these include commands for checksum, blank check (erasure check), memory read, and status inquiry. For details on these host commands, see section 29.5.5, Programming/Erasing Host Command Wait State.

(4) Initial values for register FRQCR in boot mode

The initial value of the FRQCR register in boot mode is adjusted so that the settings for frequency-division rate for Iϕ, Bϕ, Pϕ, Mϕ, and Aϕ are all 1/8.

29.5.3 Automatic Adjustment of Bit Rate

When this LSI is started in boot mode, it measures the low-level (H'00) period of the data that is continuously sent from the host in asynchronous SCI communications. During this measurement, set the SCI transmit/receive format to 8-bit data, 1 stop bit, and no parity, and set the bit rate to 9,600 bps or 19,200 bps. This LSI calculates the bit rate of the host SCI by means of the measured low-level period, and then sends H'00 to the host after completing the bit rate adjustment. When the host has received H'00 successfully, it must send H'55 to this LSI. If the host has failed to receive H'00, restart this LSI in boot mode to calculate and adjust the bit rate again. When this LSI has received H'55, it returns H'E6 to the host, or when it has failed to receive H'55, it returns H'FF.

Figure 29.7 shows the format transmitted and received for automatic adjustment of bit rate by the SCI , and figure 29.8 shows the sequence of communications between the host and LSI.

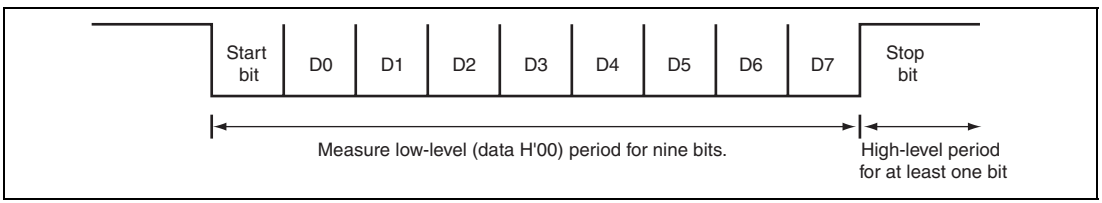


Figure 29.7 SCI Transmit/Receive Format for Automatic Adjustment of Bit Rate

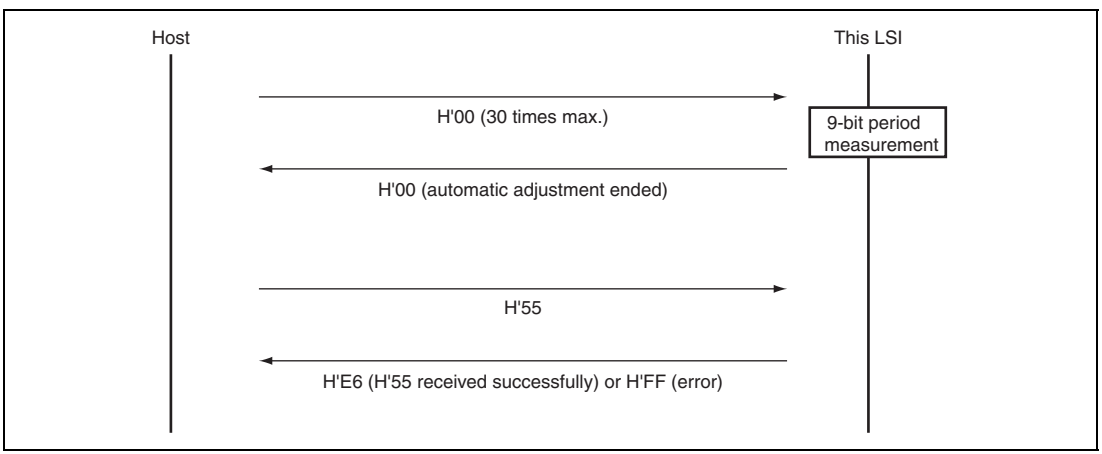


Figure 29.8 Communication Sequence between Host and This LSI

The bit rate may not be adjusted correctly depending on the bit rate of the host SCI or the peripheral clock frequency of this LSI. Satisfy the SCI communications condition as shown in table 29.5.

Table 29.5 Condition for Automatic Adjustment of Bit Rate

Host SCI Bit Rate	Peripheral Clock Frequency of this LSI
9,600 bps	5 to 50 MHz
19,200 bps	5 to 50 MHz

29.5.4 Inquiry/Selection Host Command Wait State

Table 29.6 shows the host commands available in inquiry/selection host command wait state. The boot program status inquiry command can also be used in programming/erasure host command wait state. The other commands can only be used in inquiry/selection host command wait state.

Table 29.6 Inquiry/Selection Host Commands

Host Command Name	Function
Supported device inquiry	Inquires regarding the device codes and the product codes for the embedded programs
Device selection	Selects a device code
Clock mode inquiry	Inquires regarding the clock mode
Clock mode selection	Selects a clock mode
Multiplication ratio inquiry	Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication /division ratios
Operating frequency inquiry	Inquires regarding the number of clock types and the maximum and minimum operating frequencies
User boot MAT information inquiry	Inquires regarding the number of user boot MATs and the start and end addresses
User MAT information inquiry	Inquires regarding the number of user MATs and the start and end addresses
Erasure block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
New bit rate selection	Modifies the bit rate of SCI communications between the host and this LSI
Programming/erasure state transition	Erases the entire area of each of the user MAT, user boot MAT, and FLD data MAT and makes this LSI enter programming/erasure host command wait state
Boot program status inquiry	Inquires regarding the state of this LSI

If the host has sent an undefined command, this LSI returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

Error response	H'80	Command

In inquiry/selection host command wait state, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up this LSI according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command. If commands are issued in an incorrect order, this LSI returns a response indicating a command error. Figure 29.9 shows an example of the procedure to use inquiry/selection host commands.

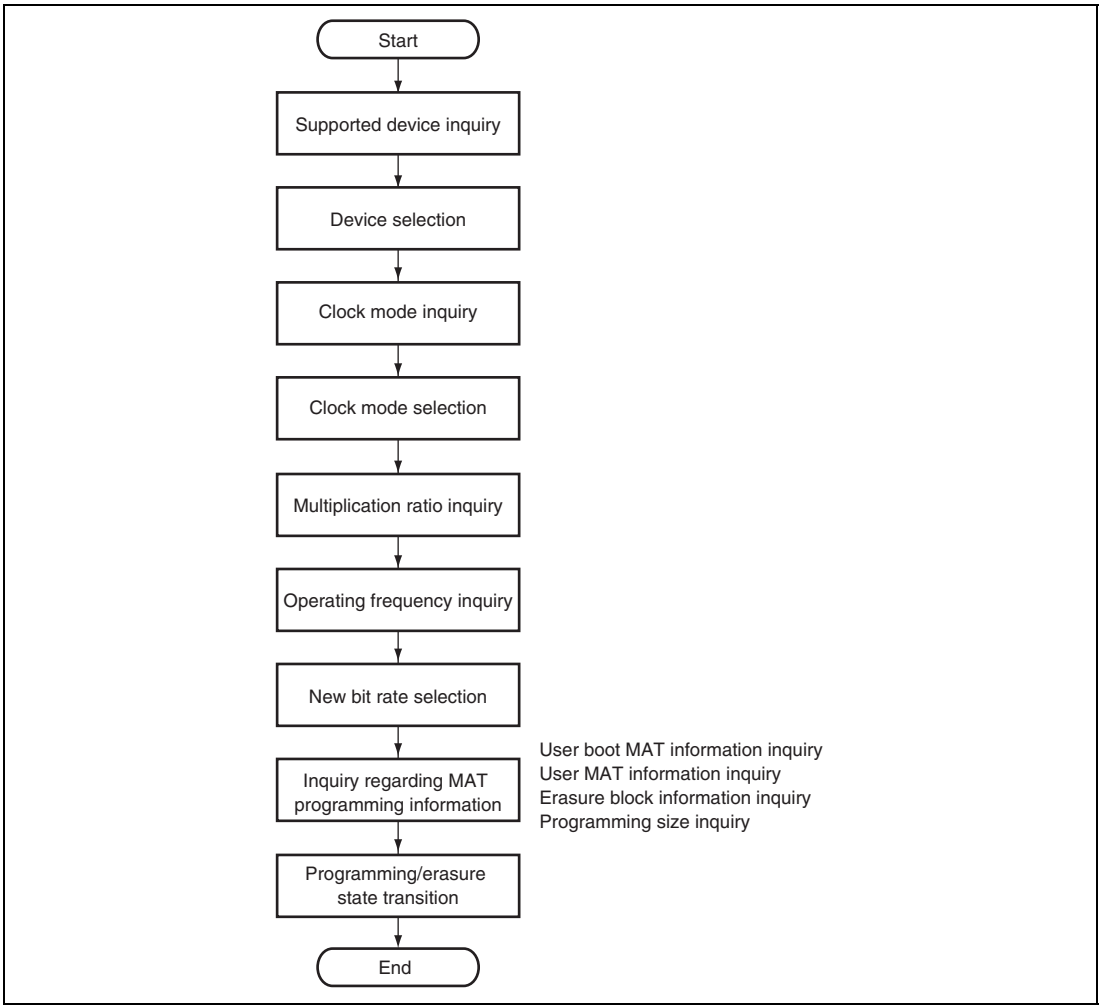


Figure 29.9 Example of Procedure to Use Inquiry/Selection Host Commands

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, this LSI returns the information concerning the devices supported by the embedded program for boot mode. If the supported device inquiry command comes after the host has selected a device, this LSI only returns the information concerning the selected device.

Command

H'20

Response

H'30	Size	Device count	
Character count	Device code		Product code
Character count	Device code		Product code
:	:		:
Character count	Device code		Product code
SUM			

[Legend]

Size (1 byte): Total number of bytes in the device count, character count, device code, and product code fields

Device count (1 byte): Number of device types supported by the embedded program for boot mode

Character count (1 byte): Number of characters included in the device code and product code fields

Device code (4 bytes): ASCII code for the product name of the chip

Product code (n bytes): ASCII code for the supported device

SUM (1 byte): Checksum

(2) Device Selection

In response to a device selection command sent from the command, this LSI checks if the selected device is supported. When the selected device is supported, this LSI specifies this device as the device for use and returns a response (H'06). If the selected device is not supported or the sent command is illegal, this LSI returns an error response (H'90).

Even when H'01 has been returned as the number of supported devices in response to a supported device inquiry command, issue a device selection command to specify the device code that has been returned as the result of the inquiry.

Command	H'10	Size	Device code	SUM
Response	H'06			
Error response	H'90	Error		

[Legend]

Size (1 byte): Number of characters in the device code field (fixed at four)

Device code (4 bytes): ASCII code for the product name of the chip (one of the device codes returned in response to the supported device inquiry command)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'21: Incorrect device code error

(3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, this LSI returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, this LSI only returns the information concerning the selected clock mode.

Command	H'21

Response				
	H'31	Size		
	Mode	Mode	. . .	Mode
	SUM			

[Legend]

Size (1 byte): Total number of bytes in the mode count and mode fields

Mode (1 byte): Supported clock mode (for example, H'01 indicates clock mode 1)

SUM (1 byte): Checksum

(4) Clock Mode Selection

In response to a clock mode selection command sent from the host, this LSI checks if the selected clock mode is supported. When the selected mode is supported, this LSI specifies this clock mode for use and returns a response (H'06). If the selected mode is not supported or the sent command is illegal, this LSI returns an error response (H'91).

Be sure to issue a clock mode selection command only after issuing a device selection command. Even when H'00 or H'01 has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

Command	H'11	Size	Mode	SUM
Response	H'06			
Error response	H'91	Error		

[Legend]

Size (1 byte): Number of characters in the mode field (fixed at 1)

Mode (1 byte): Clock mode (one of the clock modes returned in response to the clock mode inquiry command)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'22: Incorrect clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, this LSI returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command

H'22

Response

H'32	Size	Clock type count		
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
:	:	:	. . .	:
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
SUM				

[Legend]

Size (1 byte): Total number of bytes in the clock type count, multiplication ratio count, and multiplication ratio fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for example, H'03 indicates that three multiplication ratios are supported for the internal clock (x4, x6, and x8))

Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example, H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example, H'FE = -2 = division by 2)

SUM (1 byte): Checksum

(6) Operating Clock Frequency Inquiry

In response to an operating clock frequency inquiry command sent from the host, this LSI returns the minimum and maximum frequencies for each clock.

Command

H'23

Response

H'33	Size	Clock type count
Minimum frequency		Maximum frequency
Minimum frequency		Maximum frequency
:		:
Minimum frequency		Maximum frequency
SUM		

[Legend]

Size (1 byte): Total number of bytes in the clock type count, minimum frequency, and maximum frequency fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Minimum frequency (2 bytes): Minimum value of the operating frequency (for example, H'07D0 indicates 20.00 MHz).

This value should be calculated by multiplying the frequency value (MHz) to two decimal places by 100.

Maximum frequency (2 bytes): Maximum value of the operating frequency represented in the same format as the minimum frequency

SUM (1 byte): Checksum

(7) User Boot MAT Information Inquiry

In response to a user boot MAT information inquiry command sent from the host, this LSI returns the number of user boot MATs and their addresses.

Command

H'24

Response

H'34	Size	MAT count
MAT start address		
MAT end address		
MAT start address		
MAT end address		
:		
MAT start address		
MAT end address		
SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of user boot MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of the user boot MAT

MAT end address (4 bytes): End address of the user boot MAT

SUM (1 byte): Checksum

(8) User MAT Information Inquiry

In response to a user MAT information inquiry command sent from the host, this LSI returns the number of user MATs and their addresses.

Command	H'25		
Response	H'35	Size	MAT count
	MAT start address		
	MAT end address		
	MAT start address		
	MAT end address		
	:		
	MAT start address		
	MAT end address		
	SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of user MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of the user MAT

MAT end address (4 bytes): End address of the user MAT

SUM (1 byte): Checksum

(9) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, this LSI returns the number of erasure blocks in the user MAT and their addresses.

Command	H'26		
Response	H'36	Size	Block count
	Block start address		
	Block end address		
	Block start address		
	Block end address		
	:		
	Block start address		
	Block end address		
	SUM		

[Legend]

Size (2 bytes): Total number of bytes in the block count, block start address, and block end address fields

Block count (1 byte): Number of erasure blocks in the user MAT

Block start address (4 bytes): Start address of the erasure block

Block end address (4 bytes): End address of the erasure block

SUM (1 byte): Checksum

(10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, this LSI returns the programming size.

Command	H'27			
Response	H'37	Size	Programming size	SUM

[Legend]

Size (1 byte): Number of characters included in the programming size field (fixed at two)

Programming size (2 bytes): Programming unit (bytes)

SUM (1 byte): Checksum

(11) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, this LSI checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate, this LSI returns a response (H'06) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, this LSI returns an error response (H'BF). Upon reception of response H'06, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host bit rate to the new one. After that, the host sends confirmation data (H'06) in the new bit rate, and this LSI returns a response (H'06) to the confirmation data.

Be sure to issue a new bit rate selection command only after a clock mode selection command.

Figure 29.10 shows the sequence for selection of a new bit rate.

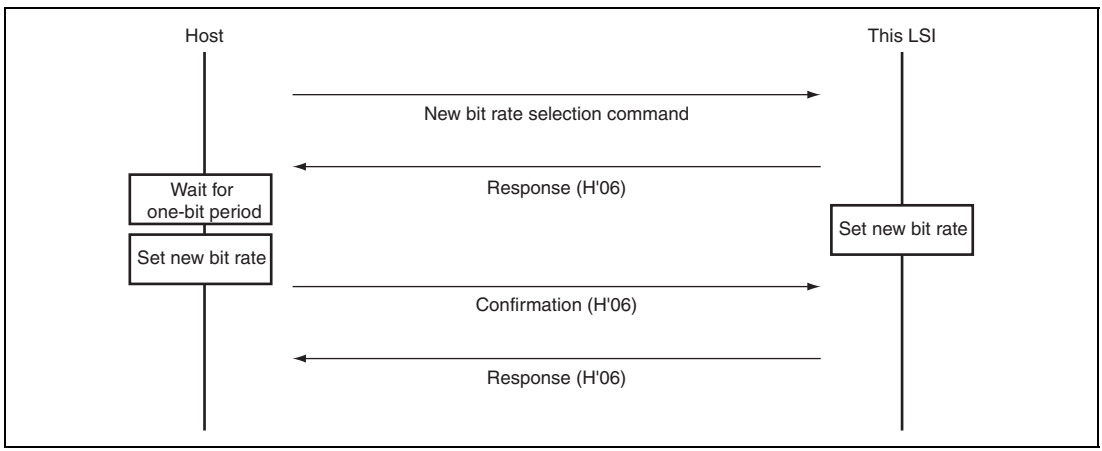


Figure 29.10 New Bit Rate Selection Sequence

Command	H'3F	Size	Bit rate	Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2	
	SUM			
Response	H'06			
Error response	H'BF	Error		
Confirmation	H'06			
Response	H'06			

[Legend]

Size (1 byte): Total number of bytes in the bit rate, input frequency, clock type count, and multiplication ratio fields

Bit rate (2 bytes): New bit rate (for example, H'00C0 indicates 19200 bps)

1/100 of the new bit rate value should be specified.

Input frequency (2 bytes): Clock frequency input to this LSI (for example, H'07D0 indicates 20.00 MHz)

This value should be calculated by multiplying the input frequency value to two decimal places by 100.

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Multiplication ratio 1 (1 byte): Multiplication/division ratio of the input frequency to obtain the internal clock

A positive value indicates a multiplication ratio (for example, H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example, HFE = -2 = division by 2)

Multiplication 2 (1 byte): Multiplication/division ratio of the input frequency to obtain the peripheral clock

This value is represented in the same format as multiplication ratio 1

SUM (1 byte): Checksum

Error: Error code

H'11: Checksum error

H'24: Bit rate selection error

H'25: Input frequency error

H'26: Multiplication ratio error

H'27: Operating frequency error

- Bit rate selection error

  A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of this LSI within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (fEX), multiplication ratio 2 (Pϕ), the SCBRR setting (N) in SCI, and the CKS[1:0] bit value (N) in SCSMR.

$$\text{Error (\%)} = \frac{f_{EX} \times P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1$$

- Input frequency error

  An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock mode selection command.

- Multiplication ratio error

  A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

- Operating frequency error

  An operating frequency error occurs when this LSI cannot operate at the operating frequencies selected through a new bit rate selection command. This LSI calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating clock frequency inquiry command.

(12) Programming/Erasure State Transition

In response to a programming/erasure state transition command sent from the host, this LSI erases the entire area of each of the user MAT, user boot MAT, and FLD data MAT. After completing erasure, this LSI returns a response (H'06) and waits for a programming/erasure host command. If this LSI has failed to complete erasure due to an error, it returns an error response (sends H'C0 and H'51 in that order).

Do not issue a programming/erasure state transition command before device selection, clock mode selection, and new bit rate selection commands.

Command	H'40	
Response	H'06	
Error response	H'C0	H'51

(13) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, this LSI returns its current status. The boot program status inquiry command can be issued in both inquiry/selection host command wait state and programming/erasure host command wait state.

Command	H'4F			
Response	H'5F	Size	Status	Error

[Legend]

Size (1 byte): Total number of bytes in the status and error fields (fixed at two)

Status (1 byte): Current status in this LSI (see table 29.7)

Error (1 byte): Error status in this LSI (see table 29.8)

Table 29.7 Status Code

Code	Description
H'11	Waiting for device selection
H'12	Waiting for clock mode selection
H'13	Waiting for bit rate selection
H'1F	Waiting for transition to programming/erasure host command wait state (bit rate has been selected)
H'31	Erasing the user MAT and user boot MAT
H'3F	Waiting for a programming/erasure host command
H'4F	Waiting for reception of programming data
H'5F	Waiting for erasure block selection

Table 29.8 Error Code

Code	Description
H'00	No error
H'11	Checksum error
H'21	Incorrect device code error
H'22	Incorrect clock mode error
H'24	Bit rate selection error
H'25	Input frequency error
H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data size error
H'51	Erasure error
H'52	Incomplete erasure error
H'53	Programming error
H'54	Selection error
H'80	Command error
H'FF	Bit rate adjustment verification error

29.5.5 Programming/Erasing Host Command Wait State

Table 29.9 shows the host commands available in programming/erasure host command wait state.

Table 29.9 Programming/Erasure Host Commands

Host Command Name	Function
User boot MAT programming selection	Selects the program for user boot MAT programming
User MAT programming selection	Selects the program for user MAT programming
Simultaneous two-user MAT programming selection	Selects the program for simultaneous two-user MAT programming
256-byte programming	Programs 256 bytes of data
Erasure selection	Selects the erasure program
Block erasure	Erases block data
Memory read	Reads data from memory
User boot MAT checksum	Performs checksum verification for the user boot MAT
User MAT checksum	Performs checksum verification for the user MAT
User boot MAT blank check	Checks whether the user boot MAT is blank
User MAT blank check	Checks whether the user MAT is blank
Read lock bit status	Reads from the lock bit
Lock bit program	Writes to the lock bit
Lock bit enabled	Enables the lock bit protect
Lock bit disable	Disables the lock bit protect
Boot program status inquiry	Inquires regarding the state of this LSI

If the host has sent an undefined command, this LSI returns a response indicating a command error. For the format of this response, see section 29.5.4, Inquiry/Selection Host Command Wait State.

Figure 29.11 shows the procedure for ROM programming in boot mode.

To program the ROM, issue a programming selection command (user boot MAT programming selection or user MAT programming selection command) and then a 256-byte programming command from the host. Upon reception of a programming selection command, this LSI enters programming data wait state (see section 29.5.2, State Transition in Boot Mode). In response to a 256-byte programming command sent from the host in this state, this LSI starts programming the ROM. When the host sends a 256-byte programming command specifying H'FFFFFFFF as the programming start address, this LSI detects it as the end of programming and enters programming/erasure host command wait state.

Figure 29.12 shows the procedure for ROM erasure in boot mode.

To erase the ROM, issue an erasure selection command and then a block erasure command from the host. Upon reception of an erasure selection command, this LSI enters erasure block selection wait state (see section 29.5.2, State Transition in Boot Mode). In response to a block erasure command sent from the host in this state, this LSI erases the specified block in the ROM. When the host sends a block erasure command specifying H'FF as the block number, this LSI detects it as the end of erasure and enters programming/erasure host command wait state.

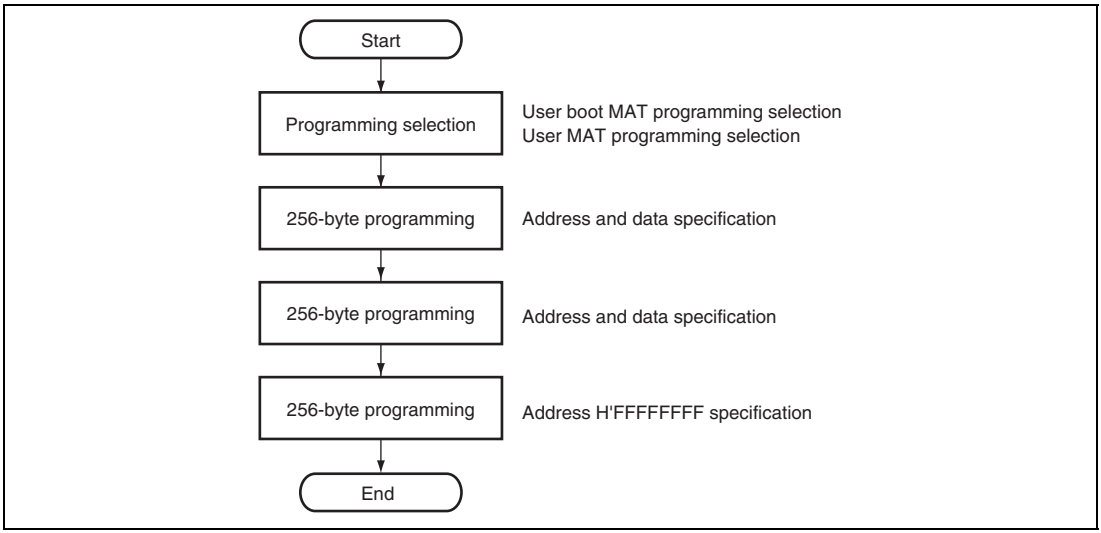


Figure 29.11 Procedure for ROM Programming in Boot Mode

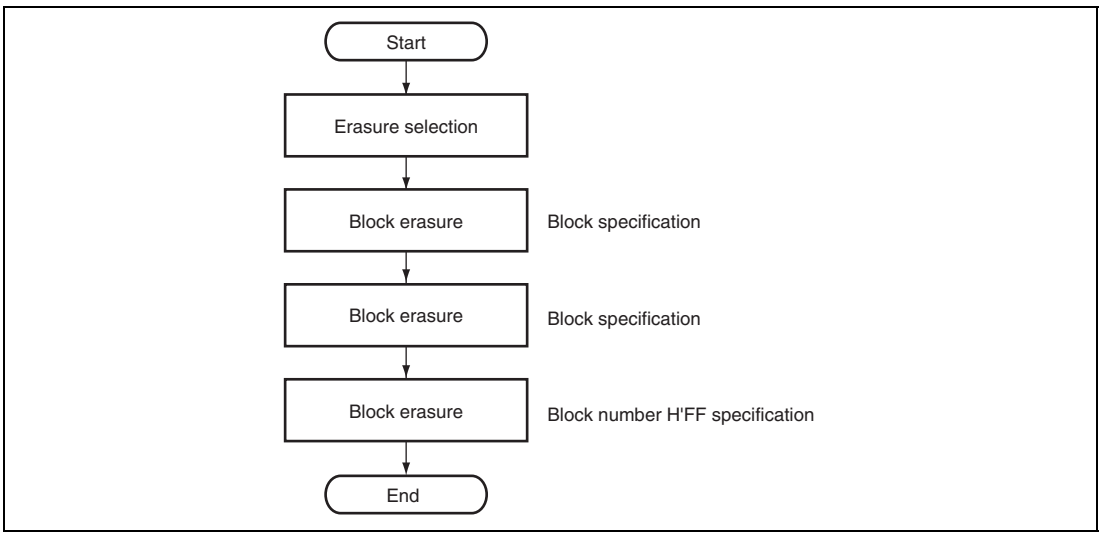


Figure 29.12 Procedure for ROM Erasure in Boot Mode

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) User Boot MAT Programming Selection

In response to a user boot MAT programming selection command sent from the host, this LSI selects the program for user boot MAT programming and waits for programming data.

Command	H'42
Response	H'06

(2) User MAT Programming Selection

In response to a user MAT programming selection command sent from the host, this LSI selects the program for user MAT programming and waits for programming data.

Command	H'43
Response	H'06

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, this LSI programs the ROM. After completing ROM programming successfully, this LSI returns a response (H'06). If an error has occurred during ROM programming, this LSI returns an error response (H'D0).

Command	H'50	Programming Address		
	Data	Data	. . .	Data
	SUM			
Response	H'06			
Error response	H'D0	Error		

[Legend]

Programming address (4 bytes): Target address of programming
To program the ROM, a 256-byte boundary address should be specified.
To terminate programming, H'FFFFFFFF should be specified.

Data (256 bytes): Programming data
H'FF should be specified for the bytes that do not need to be programmed.
When terminating programming, no data needs to be specified (only the programming address and SUM should be sent in that order).

SUM (1 byte): Checksum

Error (1 byte): Error code
H'11: Checksum error
H'2A: Address error (the specified address is not in the target MAT)
H'53: Programming cannot be done due to a programming error

(4) Erasure Selection

In response to an erasure selection command sent from the host, this LSI selects the erasure program and waits for erasure block specification.

Command	H'48
Response	H'06

(5) Block Erasure

In response to a block erasure command sent from the host, this LSI erases the ROM. After completing ROM erasure successfully, this LSI returns a response (H'06). If an error has occurred during ROM erasure, this LSI returns an error response (H'D8).

Command	H'58	Size	Block	SUM
Response	H'06			
Error response	H'D8	Error		

[Legend]

Size (1 byte): Number of bytes in the block specification field (fixed at 1)

Block (1 byte): Block number whose data is to be erased
To terminate erasure, H'FF should be specified.

SUM (1 byte): Checksum

Error (1 byte): Error code
H'11: Checksum error
H'29: Block number error (an incorrect block number is specified)
H'51: Erasure cannot be done due to an erasure error

(6) Memory Read

In response to a memory read command sent from the host, this LSI reads data from the ROM. After completing ROM reading, this LSI returns the data stored in the address specified by the memory read command. If this LSI has failed to read the ROM, this LSI returns an error response (H'D2).

Command	H'52	Size	Area	Read start address	
	Reading size				SUM

Response	H'52	Reading size		
	Data	Data	. . .	Data
	SUM			

Error response	H'D2	Error

[Legend]

Size (1 byte): Total number of bytes in the area, read start address, and reading size fields

Area (1 byte): Target MAT to be read
H'00: User boot MAT
H'01: User MAT

Read start address (4 bytes): Start address of the area to be read

Reading size (4 bytes): Size of data to be read (bytes)

SUM (1 byte): Checksum

Data (1 byte): Data read from the ROM

Error (1 byte): Error code
H'11: Checksum error
H'2A: Address error
- The value specified for area selection is neither H'00 nor H'01.
- The specified read start address is outside the selected MAT.

H'2B: Data size error
- H'00 is specified for the reading size.
- The reading size is larger than the MAT.
- The end address calculated from the read start address and the reading size is outside the selected MAT.

(7) User Boot MAT Checksum

In response to a user boot MAT checksum command sent from the host, this LSI sums the user boot MAT data in byte units and returns the result (checksum).

Command	H'4A			
Response	H'5A	Size	MAT checksum	SUM

[Legend]
Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)
MAT checksum (4 bytes): Checksum of the user boot MAT data
SUM (1 byte): Checksum (for the response data)

(8) User MAT Checksum

In response to a user MAT checksum command sent from the host, this LSI sums the user MAT data in byte units and returns the result (checksum).

Command	H'4B			
Response	H'5B	Size	MAT checksum	SUM

[Legend]
Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)
MAT checksum (4 bytes): Checksum of the user MAT data
The user MAT also stores the key code for debugging function authentication. Note that the checksum includes this key code value.
SUM (1 byte): Checksum (for the response data)

(9) User Boot MAT Blank Check

In response to a user boot MAT blank check command sent from the host, this LSI checks whether the user boot MAT is completely erased. When the user boot MAT is completely erased, this LSI returns a response (H'06). If the user boot MAT has an unerased area, this LSI returns an error response (sends H'CC and H'52 in that order).

Command	H'4C	
Response	H'06	
Error response	H'CC	H'52

(10) User MAT Blank Check

In response to a user MAT blank check command sent from the host, this LSI checks whether the user MAT is completely erased. When the user MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'CD and H'52 in that order).

Command	H'4D	
Response	H'06	
Error response	H'CD	H'52

(11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, this LSI reads data from the lock bit. After completing the lock bit reading, this LSI returns the data stored in the address specified by the read lock bit status command. If this LSI has failed to read the lock bit, this LSI returns an error response (H'F1).

Command	H'71	Size	Area	Medium address	Upper address	SUM
Response	Status					
Error response	H'F1	Error				

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3 in this LSI)

Area (1 byte): Target MAT to be read

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Status (1 byte): Bit 6 locked at "0"

Bit 6 unlocked at "1"

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

(12) Lock Bit Program

In response to a lock bit program command sent from the host, this LSI writes to a lock bit and locks the specified block. After completing the lock bit blocking, this LSI returns a response (H'06). If this LSI has failed to lock, this LSI returns an error response (H'F7).

Command	H'77	Size	Area	Medium address	Upper address	SUM
Response	H'06					
Error response	H'F7	Error				

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3 in this LSI)

Area (1 byte): Target MAT to be locked

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

H'53: Locking cannot be done due to a programming error

(13) Lock Bit Enable

In response to a lock bit enable command sent from the host, this LSI enables a lock bit.

Command	H'7A
Response	H'06

(14) Lock Bit Disable

In response to a lock bit enable command sent from the host, this LSI disables a lock bit.

Command	H'75
Response	H'06

(15) Boot Program Status Inquiry

For details, refer to section 29.5.4, Inquiry/Selection Host Command Wait State.

29.6 User Program Mode

29.6.1 FCU Command List

To program or erase the user MAT in user program mode, issue FCU commands to the FCU. Table 29.10 is a list of FCU commands for ROM programming and erasure.

Table 29.10 FCU Command List (ROM-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 29.6.2, Conditions for FCU Command Acceptance)
Status read mode transition	Moves to the status read mode (see section 29.6.2, Conditions for FCU Command Acceptance)
Lock bit read mode transition	Moves to the lock bit read mode (see section 29.6.2, Conditions for FCU Command Acceptance)
Program	Programs ROM (in 256-byte units)
Block erase	Erases ROM (in block units; erasing the lock bit)
P/E suspend	Suspends programming or erasure
P/E resume	Resumes programming or erasure
Status register clear	Clears the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state
Lock bit read 2	Reads the lock bit of a specified erasure block (updates the FLOCKST bit in FSTATR1 to reflect the lock bit state)
Lock bit program	Writes to the lock bit of a specified erasure block
Peripheral clock notification	Specifies the peripheral clock frequency

FCU commands other than the lock bit read 2 program and lock bit program are also used for FLD programming and erasure. When a lock bit read 2 command is issued to the FLD, an FLD blank check is executed. When a lock bit program command is issued to the FLD, it is detected as an illegal command and generates an error (see section 30, Data Flash (FLD)).

To issue a command to the FCU, write to a ROM program/erase address via the peripheral bus. Table 29.11 shows the FCU command format. Write access via the peripheral bus as shown in table 29.11 under the respective conditions starts processing of the corresponding command by the FCU. For the conditions for FCU command acceptance, refer to section 29.6.2, Conditions for FCU Command Acceptance. For details of each FCU command, refer to section 29.6.3, FCU Command Usage.

If a lock bit is to be read, the FRDMD bit has to be set to 1, When H'71 is sent in the first cycle of the FCU command while the FRDMD bit is 1 , the FCU waits for the second-cycle data (H'D0) of the lock bit read 2 command. When a ROM program/erase address is written to through the peripheral bus in this state, the FCU copies the lock bit of the erasure block corresponding to the accessed address into the FLOCKST bit in FSTATR1.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 29.6.4, Suspending Operation.

Table 29.11 FCU Command Format

Command	Number of Command Cycles*	First Cycle		Second Cycle		Third Cycle		Fourth and Fifth Cycles		Sixth Cycle		Seventh to 130th Cycles		131st Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Normal mode transition	1	RA	H'FF	—	—	—	—	—	—	—	—	—	—	—	—
Status read mode transition	1	RA	H'70	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read mode transition	1	RA	H'71	—	—	—	—	—	—	—	—	—	—	—	—
Program	131	RA	H'E8	RA	H'80	WA	WD1	RA	WDn	RA	WDn	RA	WDn	RA	H'D0
Block erase	2	RA	H'20	BA	H'D0	—	—	—	—	—	—	—	—	—	—
P/E suspend	1	RA	H'B0	—	—	—	—	—	—	—	—	—	—	—	—
P/E resume	1	RA	H'D0	—	—	—	—	—	—	—	—	—	—	—	—
Status register clear	1	RA	H'50	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read 2	2	RA	H'71	BA	H'D0	—	—	—	—	—	—	—	—	—	—
Lock bit program	2	RA	H'77	BA	H'D0	—	—	—	—	—	—	—	—	—	—
Peripheral clock notification	6	RA	H'E9	RA	H'03	WA	H'0F0F	WA	H'0F0F	RA	H'D0	—	—	—	—

[Legend]

RA: ROM program/erase address
An address in the range from H'80800000 to H'808FFFFF

WA: ROM program address
Start address of 256-byte programming data

BA: ROM erasure block address
An address in the target erasure block (specified by the ROM program/erase address)

WDn: n-th word of programming data (n = 1 to 128)

Note: * The numbers of cycles for commands are numbers of cycles of write access to programming/erasure addresses over the peripheral bus (P bus) by the CPU.

29.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 29.13 is an FCU mode transition diagram.

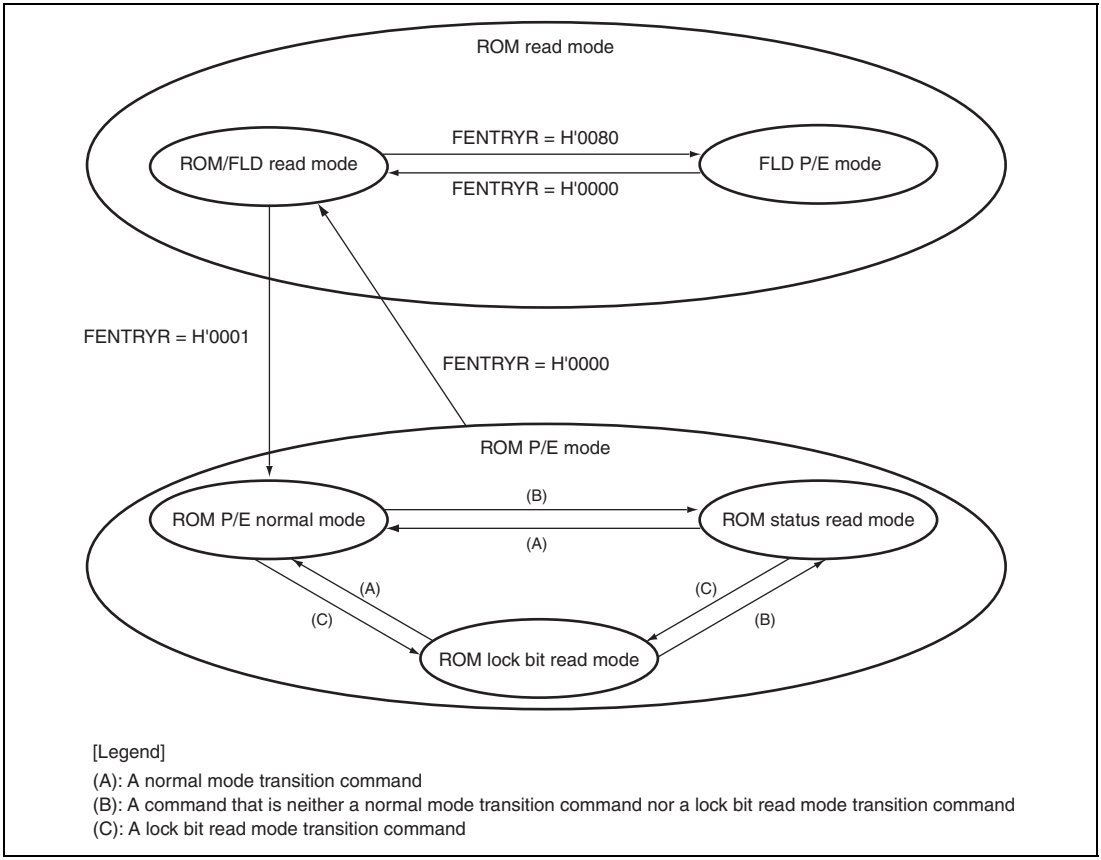


Figure 29.13 FCU Mode Transition Diagram (ROM-Related Modes)

(1) ROM Read Mode

- ROM/FLD read mode

  The ROM and FLD respectively can be read at high speed via the CPU and peripheral buses. The FCU does not accept commands. The FCU enters this mode when the FENTRY0 bit in FENTRYR is set to 0 and the FENTRYD bit to 0 in FENTRYR.
- FLD P/E mode

  The ROM can be read at a high speed. The FCU accepts commands for FLD, but does not accept commands for ROM. The FCU enters this mode when the FENTRY0 bit is set to 0 and the FENTRYD bit to 1. For details of the FLD P/E mode, refer to section 29.6.2, Conditions for FCU Command Acceptance.

(2) ROM P/E Mode

- ROM P/E normal mode

  The FCU enters this mode when the FENTRYD bit is set to 0 and the FENTRY0 bit is set to 1 in ROM read mode, or when a normal mode transition command is accepted in ROM P/E mode. Table 29.12 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80800000 to H'808FFFFF is read through the peripheral bus while the FENTRY0 bit is set to 1, a ROM access error occurs and the FCU enters the command-locked state (see section 29.9.3, Error Protection).
- ROM status read mode

  The FCU enters this mode when the FCU accepts a command that is neither a normal mode transition command nor a lock bit read mode transition command in ROM P/E mode. The ROM status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 29.12 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80800000 to H'808FFFFF is read through the peripheral bus while the FENTRY0 bit is set to 1, the FSTATR0 value is read.
- ROM lock bit read mode

  The FCU enters this mode when the FCU accepts a lock bit read mode transition command in ROM P/E mode. Table 29.12 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. The FENTRYR value is the same as that in ROM P/E normal mode.

Table 29.12 shows the acceptable commands in each FCU mode/state. When a command that cannot be accepted is issued, the FCU enters the command-locked state (see section 29.9.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR bit value in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR bit value in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 29.12, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and lock bit read 2 processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 29.12 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 29.12 FCU Modes/States and Acceptable Commands

	P/E Normal Mode			Status Read Mode									Lock Bit Read Mode		
Item	Programming-Suspended	Erasure-Suspended	Other State	Programming/Erasure Processing	Programming processing while erasure is suspended	Programming/Erasure Suspension Processing	Lock Bit Read 2 Processing	Programming-Suspended	Erasure-Suspended	Command-Locked (FRDY = 0)	Command-Locked (FRDY = 1)	Other State	Programming-Suspended	Erasure-Suspended	Other State
FRDY bit in FSTATR0	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1
SUSRDY bit in FSTATR0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0
PRGSPD bit in FSTATR0	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0
CMDLK bit in FASTAT	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Normal mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock bit read mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Program	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	×	A	A	×	×	×	A	A	×
Status register clear	A	A	A	×	×	×	×	A	A	×	A	A	A	A	A
Lock bit read 2	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock bit program	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A

[Legend]

A: Acceptable

*: Only programming is acceptable for the areas other than the erasure-suspended block

×: Not acceptable

29.6.3 FCU Command Usage

This section shows examples of user processing procedures for firmware transfer to the FCU RAM and the issuing of FCU commands. In some procedures given in this section, the FCU state is not checked before an FCU command is issued but the command result is checked before the processing is completed. To make sure that the FCU accepts a command, check the FCU state before starting processing (see section 29.6.2, Conditions for FCU Command Acceptance).

In a flow used in this section, the current state of FCU command handling and error occurrence is checked via the FRDY, ILGLERR, ERSERR, PRGERR, SUSRDY, ERSSPD, and PRGSPD bits in FSTATR0 and the FCUERR bit in FSTATR1. Since both FSTATR0 and FSTATR1 can be read in word access at a time, the FCU state can be checked by making register access only once. If the FCU state is checked via the FRDY bit of FSTATR0 and the CMDLK bit of FASTAT, register access must be made twice. However, the state of error occurrence can checked via the CMDLK bit only.

The FRDY bit retains 0, if the FRDTCT and FRCRCT bits are set to 1 to put the FCU into a command-locked state in the middle of its command handling while the FCUERR bit is 1. Since the FCU in a command-locked state halts its processes, the FRDY bit is never set to 1 from 0. If the FRDY retains 0 for a longer period than programming/erasing time or suspend delay time (see section 35, Electrical Characteristics), abnormal operation such as the FCU process halt may have occurred. In such case, initialize the FCU by a FCU reset. If the FRDY is set to 1 upon completion of the FCU command handling, the FCUERR bit is also 0. Therefore, the state of error occurrence can be checked via the ILGLERR, ERSERR, and PRGERR bits.

Figure 29.14 gives an overview of the flow of processing for programming and erasure.

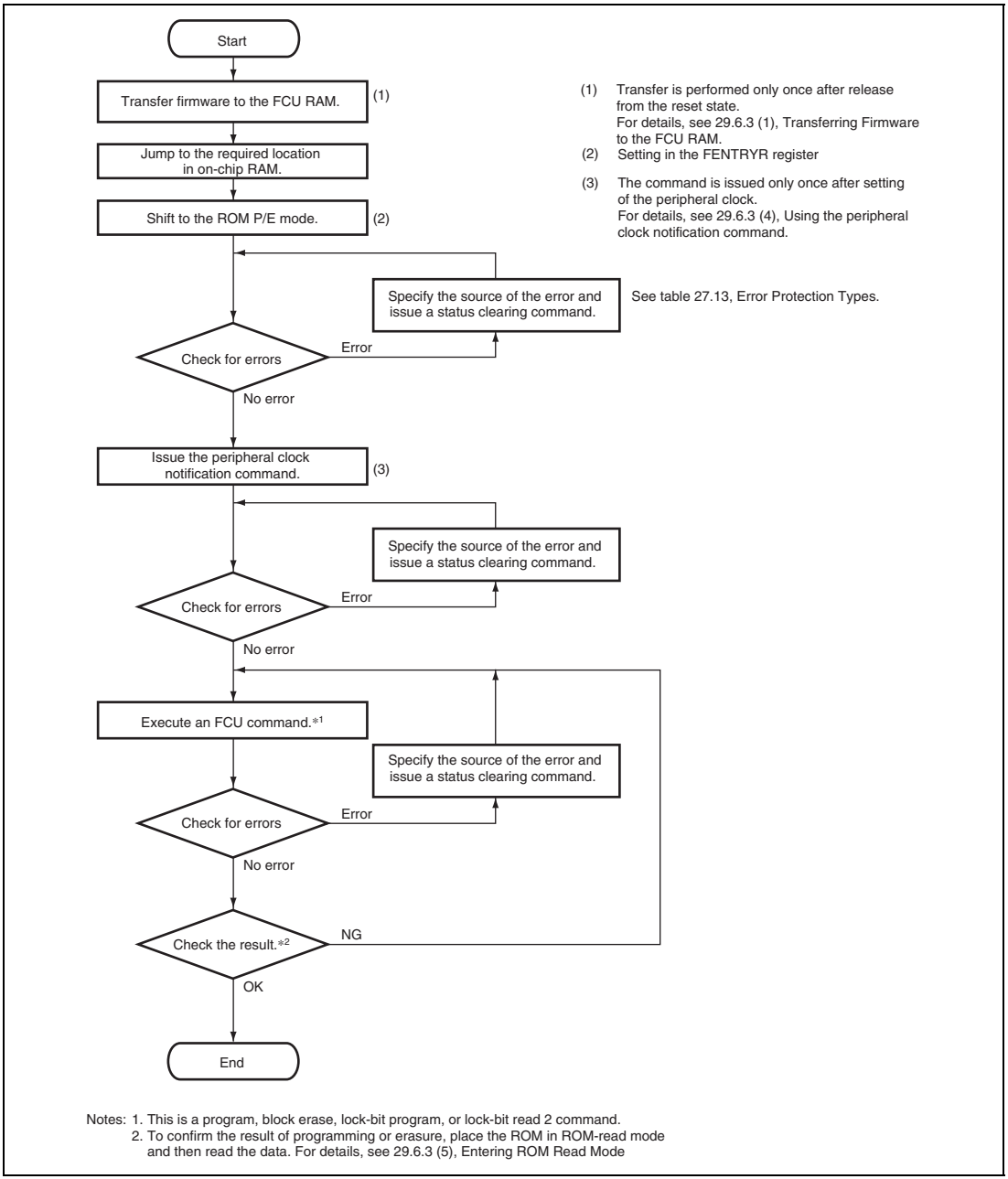


Figure 29.14 Overview of the Flow of Processing for Programming and Erasure

(1) Transferring Firmware to the FCU RAM

To use FCU commands, the FCU firmware must be stored in the FCU RAM. When this LSI is started, the FCU firmware is not stored in the FCU RAM; copy the firmware stored in the FCU firmware area to the FCU RAM. If the FCUERR bit in FSTATR1 is 1, the firmware stored in the FCU RAM may have been damaged; reset the FCU and copy the FCU firmware again in this case.

Figure 29.15 shows the procedure for firmware transfer to the FCU RAM. Before writing data to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. Transfer firmware to FCU RAM by the CPU or DMAC. For details on the DMAC settings, refer to section 11, Direct Memory Access Controller (DMAC).

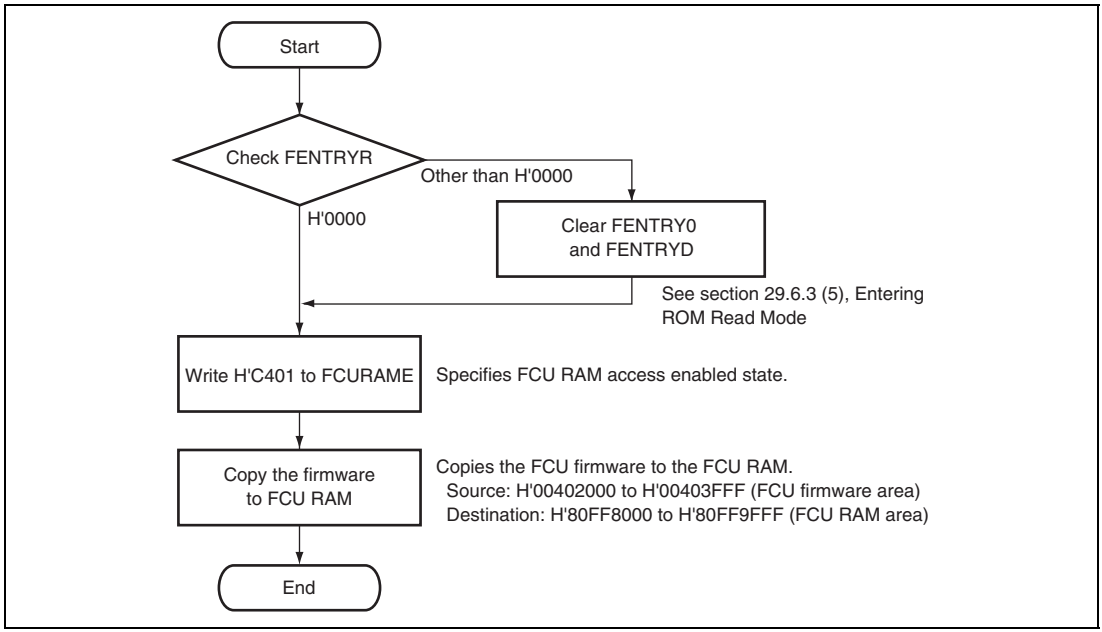


Figure 29.15 Procedure for Firmware Transfer to FCU RAM

(2) Jumping to On-Chip RAM

To prevent the fetching of instructions from the flash memory while it is being programmed or erased, execution must be shifted to an area other than the flash memory (ROM). Copy the required program code to on-chip RAM and then have execution jump to the location of the code in the on-chip RAM.

(3) Entering ROM P/E Mode

Figure 29.16 shows the procedure for a transition to ROM P/E mode.

To execute ROM-related FCU commands, set the FENTRY0 bit in FENTRYR appropriately to make the FCU enter ROM P/E mode (see section 29.6.2, Conditions for FCU Command Acceptance). For the conditions for writing to the FENTRY0 bit, refer to section 29.3.10, Flash Protect Register (FPROTR).

After a transition from ROM read mode to ROM P/E mode, the FCU is in ROM P/E normal mode.

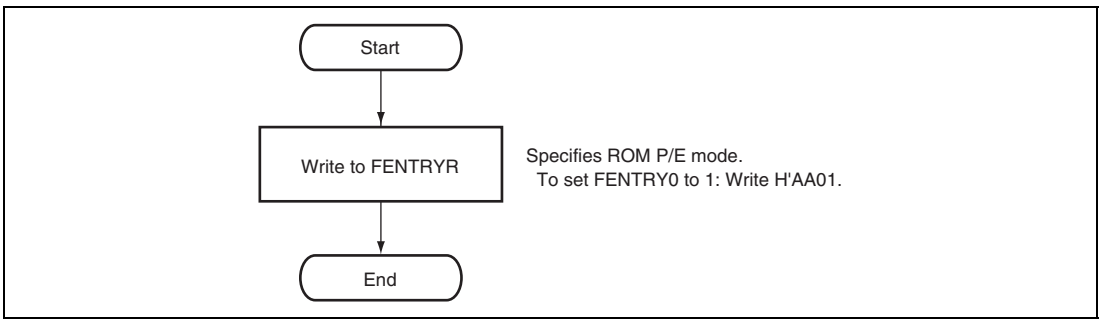


Figure 29.16 Procedure for Transition to ROM P/E Mode

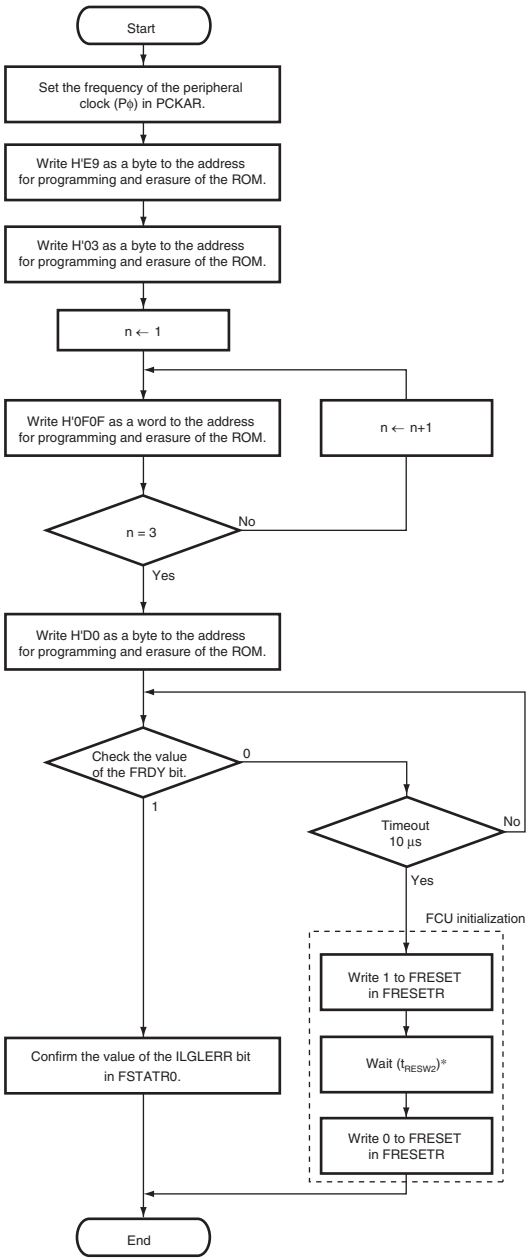
(4) Using the Peripheral Clock Notification Command

Figure 29.17 shows the flow of using the peripheral clock notification command.

The frequency of the peripheral clock to be used before programming or erasure of the flash memory (ROM) must be set in the PCKAR. Selectable values are in the range from 5 to 50 MHz. If the setting is not in this range, the FCU detects an error and enters the command-locked state (see section 29.9.3, Error Protection).

The peripheral clock notification command is used after setting the PCKAR register. For a peripheral clock notification command, H'E9 and H'03 are written in byte units in the first and second cycles, respectively, to the address for programming or erasure of the ROM. In the third to fifth cycles of the command, writing is executed in word units. As the first address, use an address that is aligned with a four-byte boundary. After H'0F0F has been written as a word unit three times to the address for programming or erasure of the ROM, when H'D0 is written as a byte unit to the address for programming or erasure of the ROM, the FCU starts processing for setting the frequency of the peripheral clock. Completion of the setting can be confirmed by checking the value of the FRDY bit in the FSTATR0 register.

After release from the reset state, if the peripheral clock settings in use are not changed, execution once makes the setting valid for subsequent FCU commands.



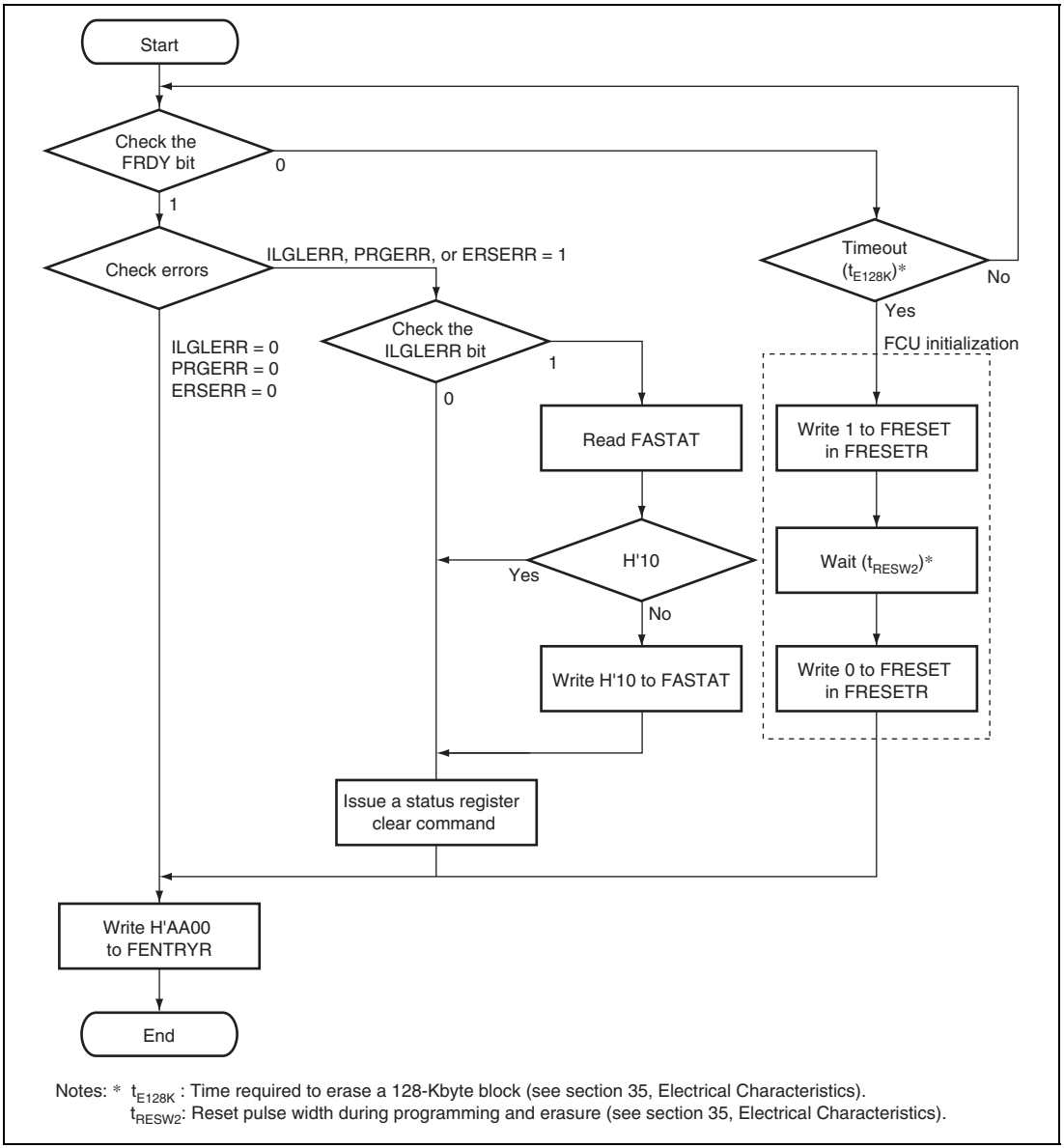
Note: * t_{RESW2} denotes the width of a reset pulse during programming or erasure (see section 35, Electrical Characteristics).

Figure 29.17 Flow for Using the Peripheral Clock Notification Command

(5) Entering ROM Read Mode

Figure 29.18 shows the procedure for a transition to ROM read mode.

To enable high-speed ROM read access through the ROM cache, clear the FENTRY0 bit in FENTRYR to make the FCU enter ROM read mode (see section 29.6.2, Conditions for FCU Command Acceptance). A transition from ROM P/E mode to ROM read mode must be made while no FCU error has been detected since FCU command processing is completed.



Notes: * t_{E128K} : Time required to erase a 128-Kbyte block (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).

Figure 29.18 Procedure for Transition to ROM Read Mode

(6) Using ROM P/E Normal Mode Transition Command

The FCU can be moved to ROM P/E normal mode in two ways: one is to set FENTRYR appropriately in ROM read mode (see section 29.6.3 (1), Transferring Firmware to the FCU RAM) and the other is to issue a normal mode transition command in ROM P/E mode (figure 29.19). The status read mode transition command and the lock bit read mode transition command can be used in the same way as the normal mode transition command.

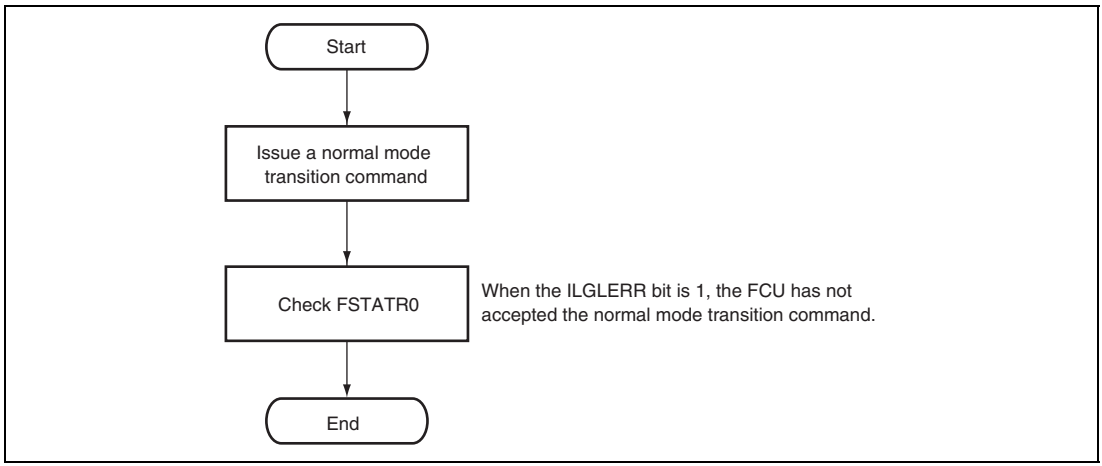


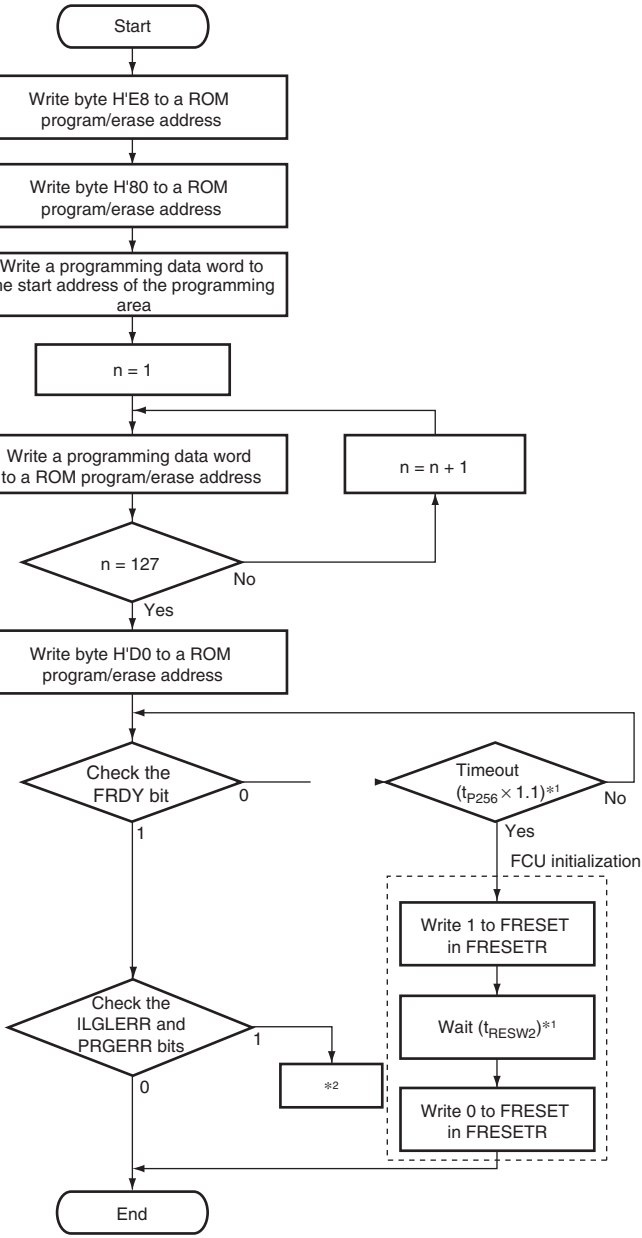
Figure 29.19 Procedure to Use ROM P/E Normal Mode Transition Command

(7) Programming

Figure 29.20 shows the procedure for ROM programming.

To program the ROM, use the program command. Write byte H'E8 to a ROM program/erase address in the first cycle of the program command and byte H'80 in the second cycle. Access the peripheral bus in words from the third to 130th cycles of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be a 256-byte boundary address. After writing words to ROM program/erase addresses 127 times, write byte H'D0 to a ROM program/erase address in the 131st cycle; the FCU then starts ROM programming. Read the FRDY bit in FSTATR0 to confirm that ROM programming is completed.

If the area accessed in the third to 130th cycles includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming.



Notes: 1. t_{P256}: Time required to write 256-byte data (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. See figure 29.24 when ILGLERR is 1 or PRGERR is 1.

Figure 29.20 Procedure for ROM Programming

(8) Erasure

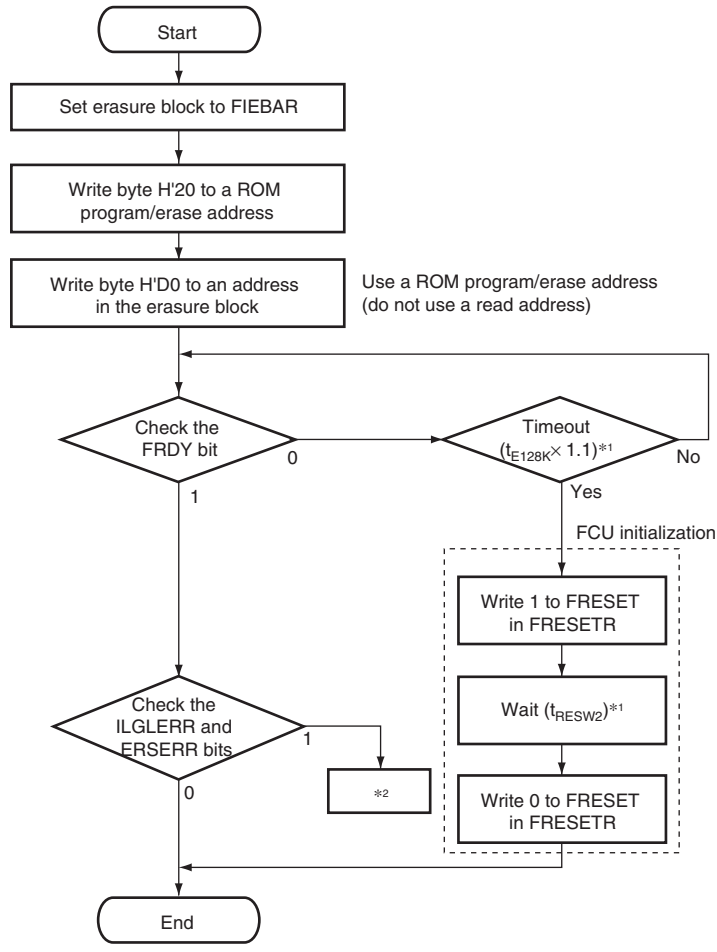
Figure 20.21 shows the procedure for ROM erasure.

To erase the ROM, use the block erase command. Write byte H'20 to a ROM program/erase address in the first cycle of the block erase command. Write byte H'D0 to an address in the target erasure block in the second cycle; the FCU then starts ROM erasure. Read the FRDY bit in FSTATR0 to confirm that ROM erasure is completed.

Set the information corresponding to the erasure block you want to erase in FIEBAR before issuing the first cycle of the block erase command (H'20). There are several points to note regarding setting of the erasure block information in FIEBAR.

To ignore the protection provided by the lock bit during erasure, set the FPROTCN bit in FPROTR to 1 before starting erasure.

1. When a block erase command is issued at an address in EB8 with FIEBAR set to H'08, EB8 is erased and the ILGLERR bit is set to 1.
2. When a block erase command is issued at an address in EB9 with FIEBAR set to H'09, EB9 is erased and the ILGLERR bit is set to 1.
3. In cases other than 1 and 2, erasure is not performed and the ILGLERR bit is set to 1. When a block erase command is issued with FIEBAR set to any value other than erasure block information (H'00 to H'0F), ILGLERR is set to 1 and erasure does not proceed.



Notes: 1. t_{E128K} : Time required to erase a 128-Kbyte block (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. See figure 29.24 when ILGLERR is 1 or ERSERR is 1.

Figure 29.21 Procedure for ROM Erasure

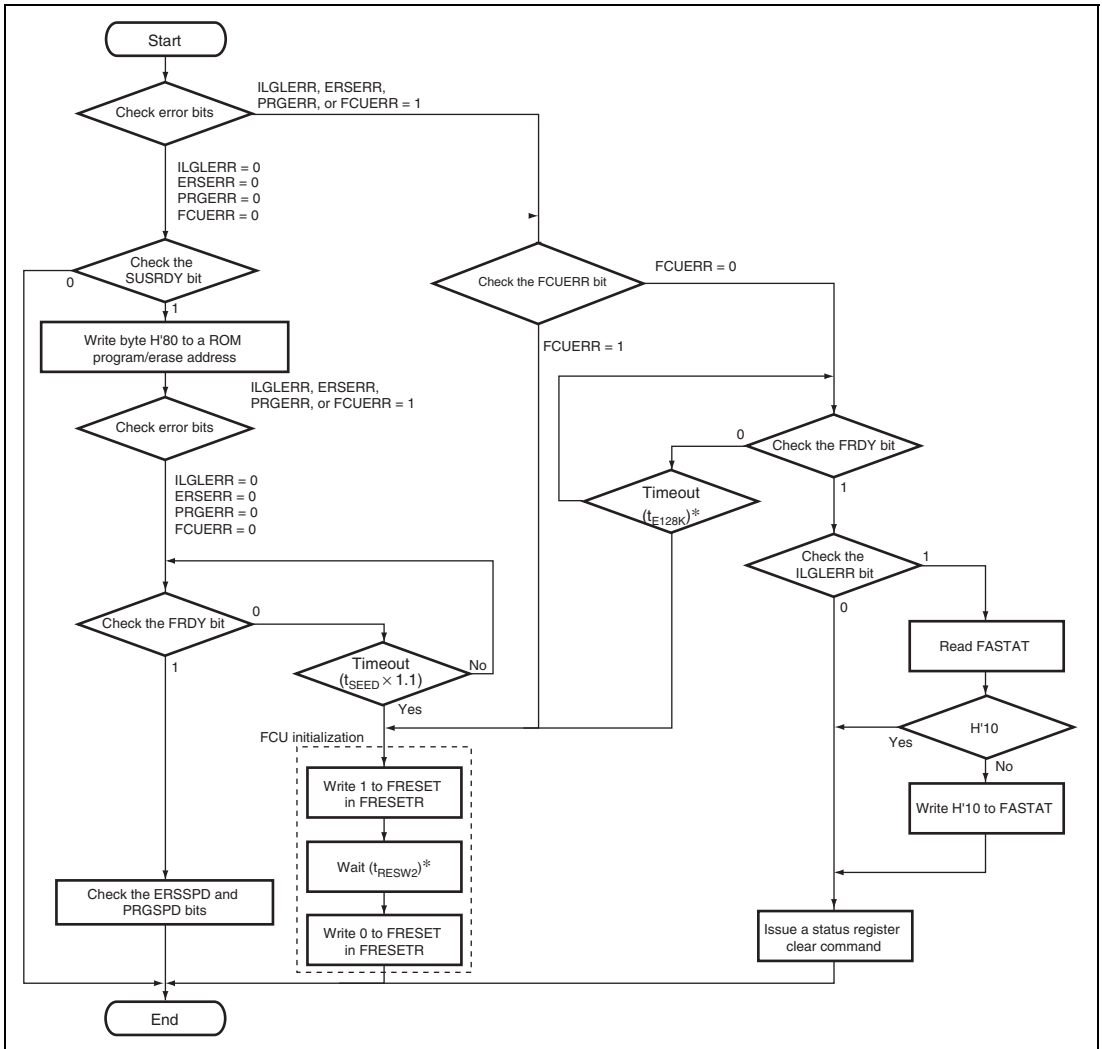
(9) Suspending Programming or Erasure

Figure 29.22 shows the procedure for suspending programming or erasure.

To suspend programming or erasure of the ROM, use the P/E suspend command. Before issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1 are 0; that is, to ensure that programming or erasure processing is being performed correctly. Also, check that the SUSRDY bit in FSTATR1 is 1 to ensure that a suspend command is acceptable.

After issuing a P/E suspend command, read both FSTATR0 and FSTATR1 to ensure no error has occurred. If an error has occurred, at least one of the ILGLERR, PRGERR, ERSERR, and FCUERR bits is set to 1. If programming/erasure is complete within the period from when the SUSRDY bit is ensured to be 1 until a P/E suspend command is accepted, the ILGLERR bit is set to 1 as the issued command is detected as illegal. If a P/E suspend command is accepted when programming/erasure is complete, no error occurs, hence no transition to a suspended state (the RDY bit is 1 and both the ERSSPD and PRGSPD bits are 0).

Once a P/E suspend command is accepted and programming/erasure is normally suspended, the FCU enters a suspended state and that the FRDY bit is 1 and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend and ensuring that the FCU has entered a suspend state, determine which operation to perform in the succeeding process. If a P/E resume command is issued in the succeeding process while the FCU has not entered a suspended state, an illegal command error occurs and the FCU enters a command-locked state (see section 29.9.3, Error Protection).



Notes: * t_{E128K} : Time required to erase a 128-Kbyte block (see section 35, Electrical Characteristics).
t_{RESW2} : Reset pulse width during programming/erasing (see section 35, Electrical Characteristics).

Figure 29.22 Procedure for Programming/Erasure Suspension

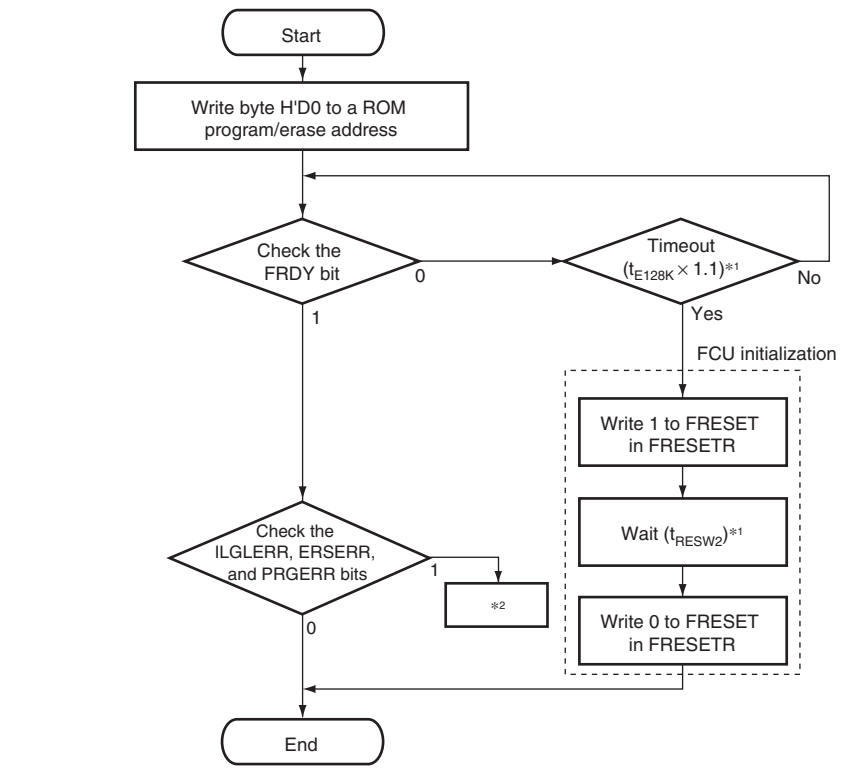
Once the FCU has entered the erasure-suspended state, blocks not for erasing can be written to. In both programming-suspended and erasure-suspended states, the FCU can be moved to ROM read mode by clearing FENTRYR.

For the operation when the FCU accepts a P/E suspend command, see section 29.6.4, Suspending Operation.

(10) Resuming Programming or Erasure

Figure 29.23 shows the procedure for resuming programming or erasure.

To resume programming or erasure that has been suspended, use the P/E resume command. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspension command was issued.



Notes: 1. t_{E128K} : Time required to erase a 128-Kbyte block (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. See figure 29.24 when ILGLERR is 1, ERSERR is 1 or PRGERR is 1.

Figure 29.23 Procedure for Resuming Programming or Erasure

(11) Clearing Status Register 0 (FSTATR0)

Figure 29.24 shows the procedure for clearing the status register.

To clear the ILGLERR, PRGERR, and ERSERR bits in FSTATR0, use the status register clear command. When any one of the ILGLERR, PRGER, and ERSERR bits is 1, the FCU is in command-locked state, in which the FCU only accepts the status register clear command and does not accept other commands. When the ILGLERR bit is 1, check also the value of the ROMAE, EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTAT. If a status register clear command is issued without clearing these bits, the ILGLERR bit is not cleared.

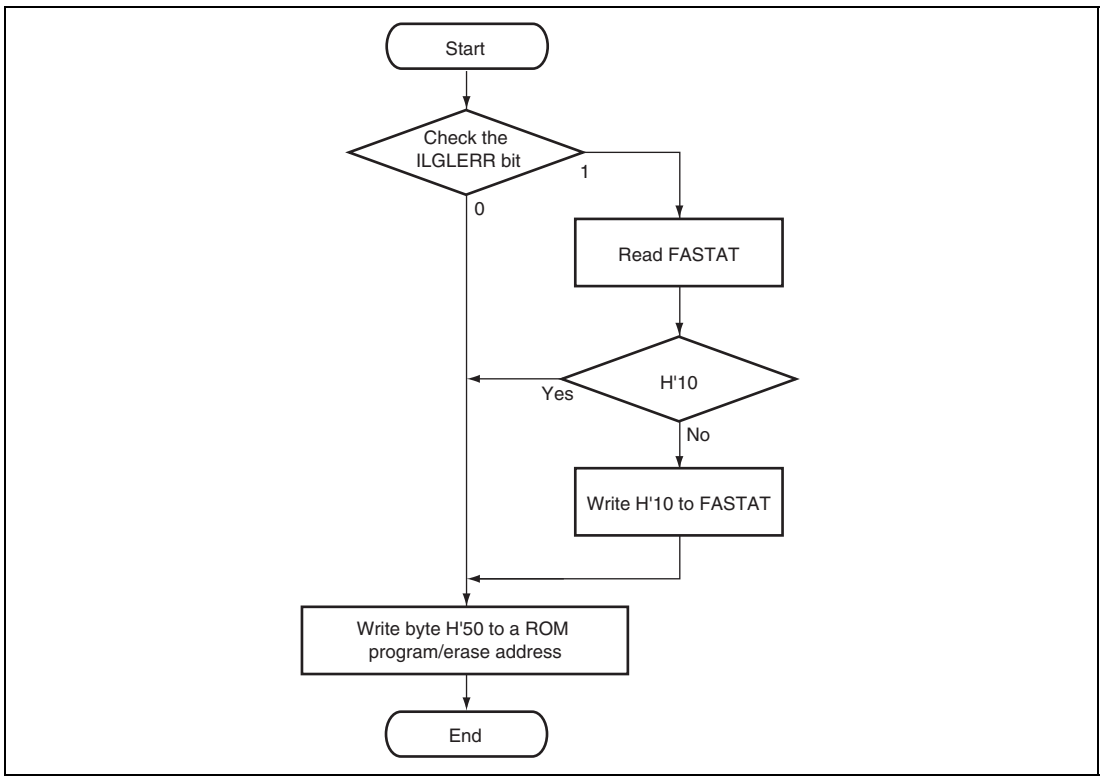


Figure 29.24 Procedure for Clearing Status Register 0

(12) Checking Status Register 0 (FSTATR0)

Figure 29.25 shows the procedure for checking the status register.

The FSTATR0 value can be checked in two ways: one is to directly read FSTATR0 and the other is to read a ROM program/erase address in ROM status read mode. After an FCU command is issued that is neither a normal mode transition command nor a lock bit read mode transition command, the FCU is in ROM status read mode. In the example shown in figure 29.25, a status read mode transition command is issued to enter ROM status read mode, and then a ROM program/erase address is read to check the FSTATR0 value.

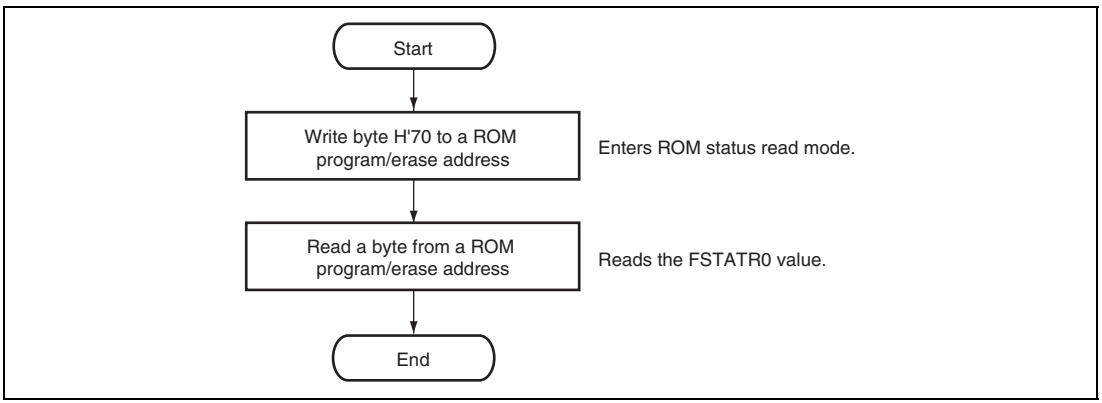


Figure 29.25 Procedure for Checking Status Register 0

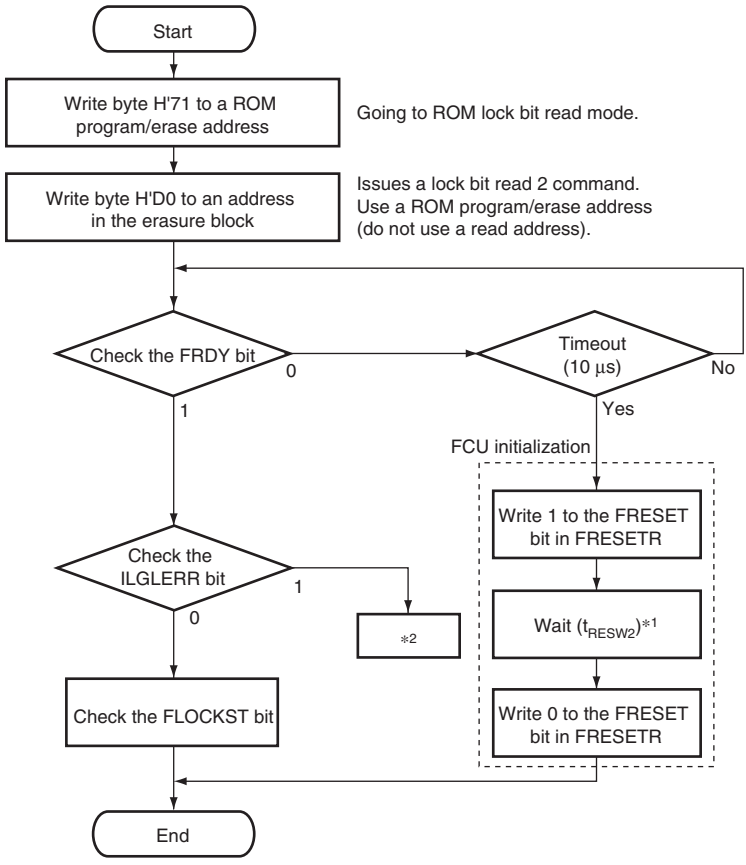
(13) Reading Lock Bit

Each erasure block in the user MAT has a lock bit. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased.

The lock bit status can be checked in register read mode. Set FRDMD in FMODR to 1 and set the erasure block number for which you want to read the lock bit in FIEBAR. After that, issue the lock bit read 2 command at an address for ROM programming and erasure within the target erasure block. The lock bit in the specified erasure block will then be copied to the FLOCKST bit in register FSTATR1.

When the lock bit read mode transition command is issued at an address that is not within the erasure block set in FIEBAR, the ILGLERR bit is set to 1. The ILGLERR bit is also set to 1 if a value other than erase block information (H'00 to H'0F) is set in FIEBAR when lock bit read mode transition command is issued.

Figure 29.26 shows the Procedure for Reading Lock Bits in Register-Read Mode.



Note: 1. t_{RESW2} : Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. See figure 29.24 when ILGLERR is 1.

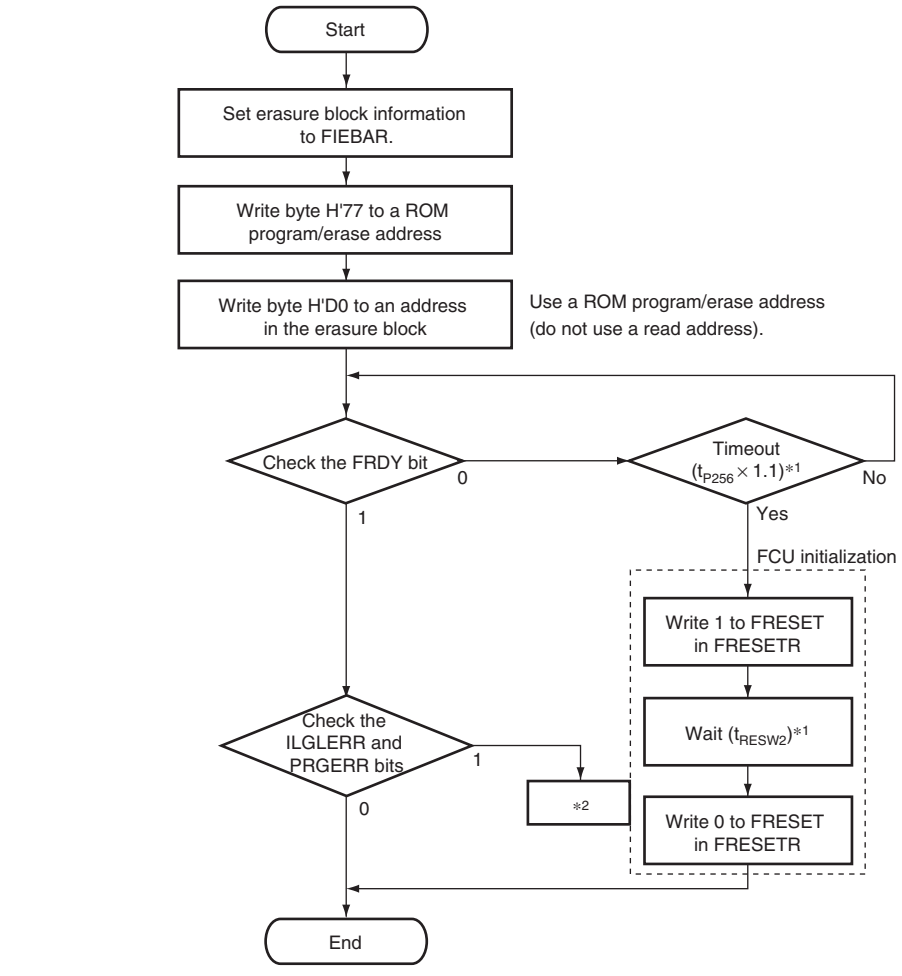
Figure 29.26 Procedure for Reading Lock Bit in Register Read Mode

(14) Writing to Lock Bit

Each erasure block in the user MAT has a lock bit. To write to a lock bit, use the lock bit program command. Write byte H'77 to a ROM program/erase address in the first cycle of the lock bit program command. Write byte H'D0 to an address in the target erasure block whose lock bit is to be written to in the second cycle; the FCU then starts writing to the lock bit. Read the FRDY bit in FSTATR0 to confirm that writing is completed.

Before issuing the first cycle of the lock bit program command (H’77), set the erasure block information for the target block in FIEBAR. When the lock bit programming command is issued at an address that is not within the erasure block set in FIEBAR, the ILGLERR bit is set to 1 and writing to the lock bit does not proceed. The ILGLERR bit is also set to 1 if a value other than erase block information (H’00 to H’0F) is set in FIEBAR when the lock bit programming command is issued.

Figure 29.27 shows the Procedure for Writing to Lock Bits.



Notes: 1. t_{P256}: Time required to write 256-byte data (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. See figure 29.24 when ILGLERR is 1 or PRGERR is 1.

Figure 29.27 Procedure for Writing to the Lock Bit

To erase a lock bit, use the block erase command. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be erased. Set the FPROTCN bit to 1, and then issue a block erase command to erase a lock bit. The block erase command erases all data in the specified erasure block; it is not possible to erase only the lock bit.

29.6.4 Suspending Operation

When a P/E suspend command is issued while ROM is being programmed or erased, the FCU suspends the programming or erasure processing. Figure 29.28 gives an overview of operation for suspending programming. Upon accepting a programming command, the FCU clears the FRDY bit in FSTATR0 to 0 and starts programming. Once the FCU enters a state where it is ready to accept a command after the start of programming, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command while reapplying a write pulse, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse, suspends programming, and sets the PRGSPD bit to 1. Once the process completes, the FCU sets the FRDY bit to 1 and enters a programming suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts programming.

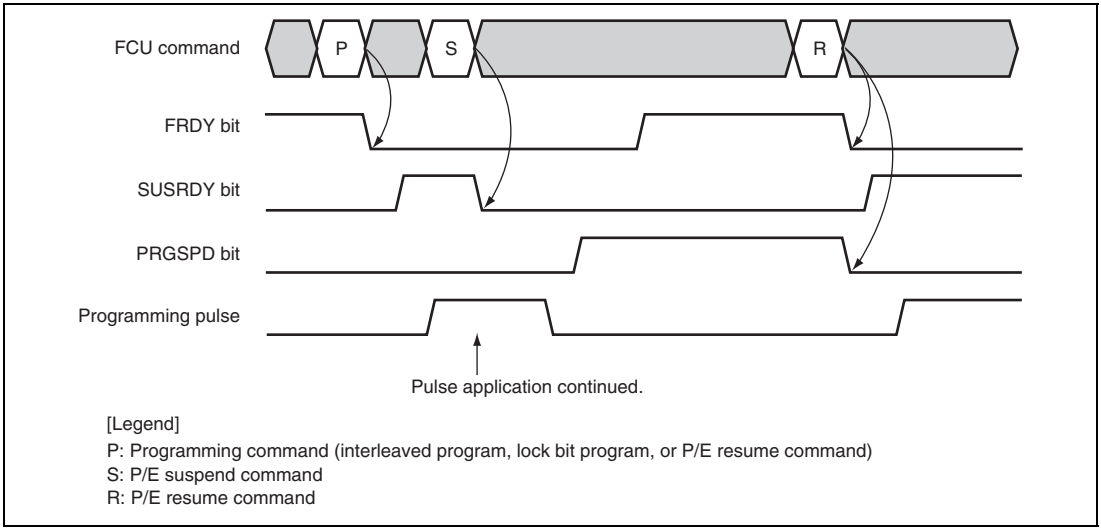


Figure 29.28 Suspending Programming Processing

Figure 29.29 shows the operation for suspending erasure processing in suspension-priority mode (the ESUSPMD bit in FCPSR is 0). Upon accepting an erasing command, the FCU clears the FRDY bit to 0 and starts erasing. Once the FCU enters a state where it is ready to accept a command after the start of erasing, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command during its erasing operation, the FCU starts a suspending process even while applying a pulse and sets the ERSSPD bit to 1. Once the suspending process completes, the FCU sets the FRDY bit to 1 and enters an erasing suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts erasing. The operations of the FRDY, SUSRDY, and ERSSPD bits are independent of the erasure-suspended mode.

The setting for the erasure-suspended mode affects the control methods for erasure pulse. In suspend-priority mode, if the FCU accepts a P/E suspend command while applying erasure pulse A, which has not been suspended previously, the FCU suspends the pulse application and enters an erasure-suspended state. After the FCU resumes erasing by accepting a P/E resume command, if the FCU accepts a P/E suspend command while applying erasing pulse A, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse and enters an erasure-suspended state. Next, after the FCU accepts a P/E resume command and starts applying a new pulse B, if the FCU accepts a P/E suspend command, the FCU suspends the pulse application. In suspense-priority mode, the suspense process is given priority by suspending once every pulse application.

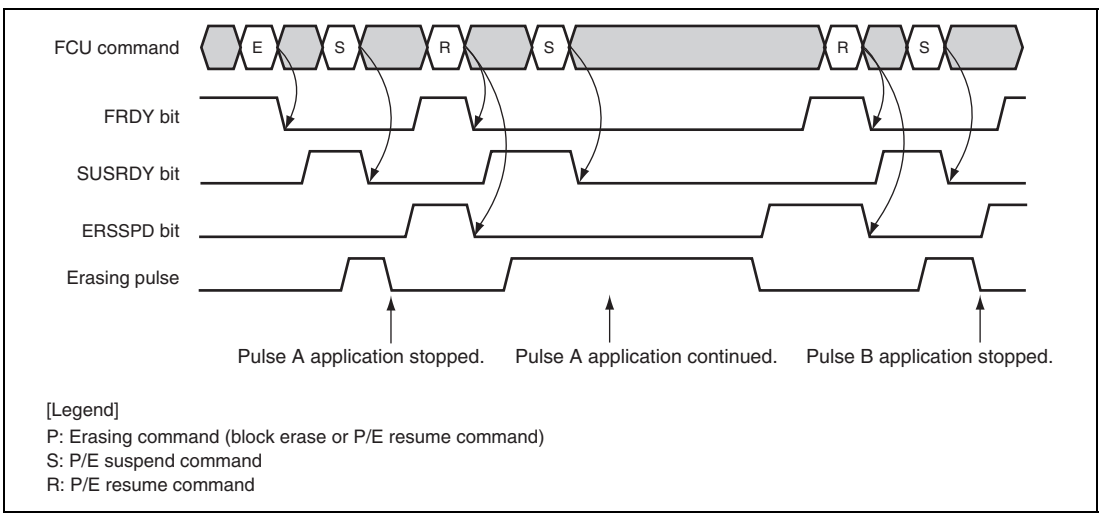


Figure 29.29 Suspending Erasure Processing (Suspension-Priority Mode)

Figure 29.30 shows how erasure processing is suspended in erasure-priority mode (with the ESUSPMD bit in FCPSR being 1). The operation for suspending erasure processing in erasure-priority mode (the ESUSPMD bit in FCPSR is 1) is equivalent to that for suspending programming processing.

In erasure-priority mode, if the FCU accepts a P/E suspend command while applying an erasing pulse, the FCU always continues applying the pulse. As processing to reapply an erasing pulse never takes place in this mode, the total time required for erasure processing is shorter than in suspension-priority mode.

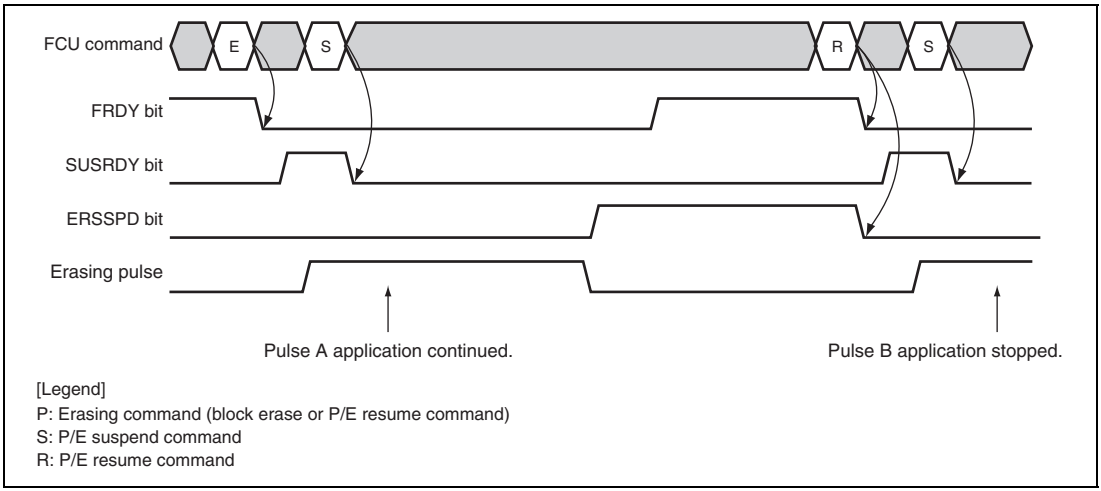


Figure 29.30 Suspending Erasure Processing (Erasure-Priority Mode)

29.7 User Boot Mode

To program or erase the user MAT in user boot mode, issue FCU commands to the FCU. A user-defined boot mode can be implemented by writing to the user boot MAT a ROM programming/erasing routine that uses a desired communications interface; when this LSI is started in user boot mode after that, the user-defined boot mode is initiated. Programming/erasure of the user boot MAT is only enabled in boot mode.

29.7.1 User Boot Mode Initiation

When this LSI is started in user boot mode, execution starts in the embedded program stored MAT, necessary processing such as FCU firmware transfer to the FCU RAM is performed, and then execution jumps to the location indicated by the reset vector of the user boot MAT. Figure 29.31 gives an overview of the boot sequence in user boot mode.

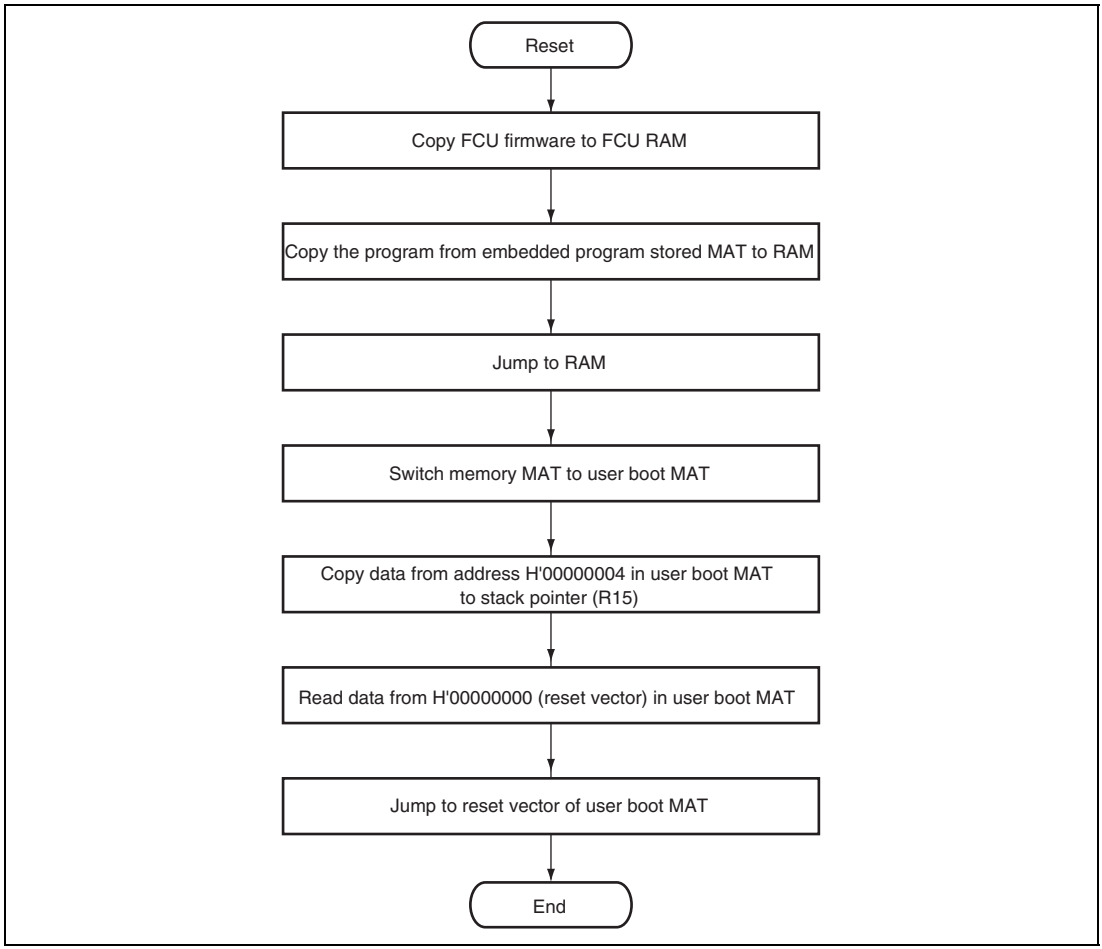


Figure 29.31 Overview of Boot Sequence in User Boot Mode

29.7.2 User MAT Programming

The user MAT can be programmed by starting this LSI in user boot mode while the user MAT programming/erasing routine created by the user is stored in the user boot MAT. Be sure to copy the user MAT programming/erasing routine to the RAM and execute it in the RAM. The user boot MAT is selected in the initial state in user boot mode; be sure to switch the memory MAT to the user MAT before starting programming. If an FCU command for ROM programming or erasure is issued while the user boot MAT is selected, the FCU does not program or erase the ROM. Figure 29.32 shows an example of the user MAT programming procedure.

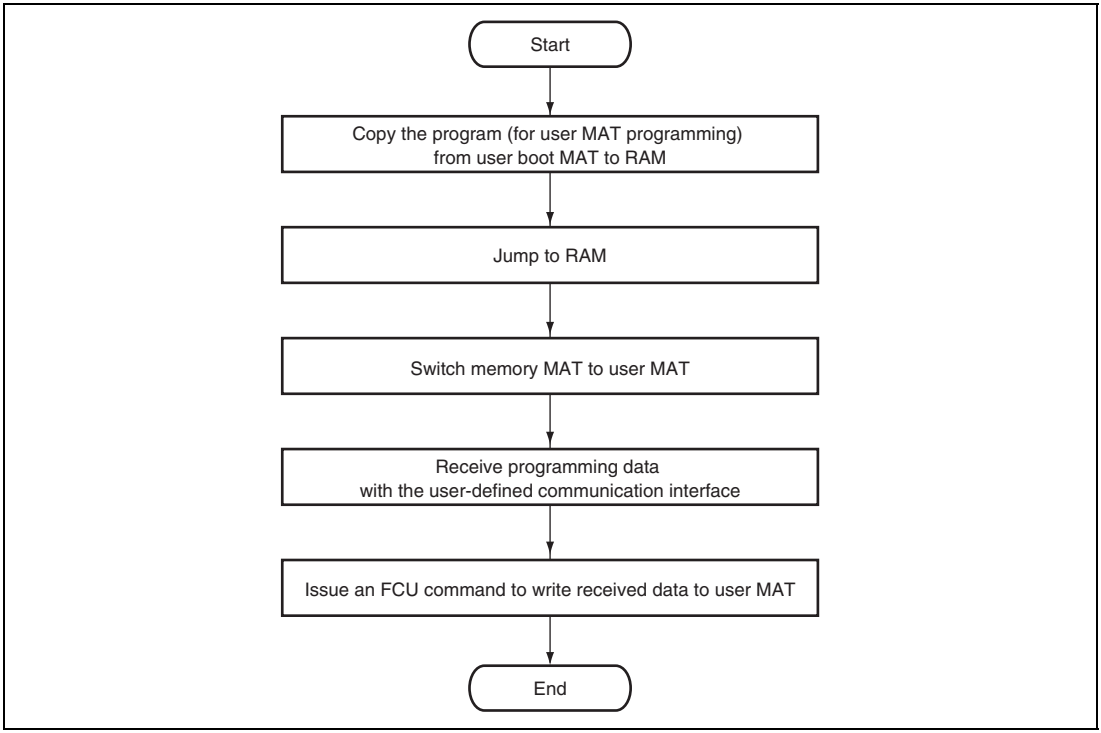


Figure 29.32 Example of User MAT Programming

29.8 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as for a discrete flash memory. Use a PROM programmer that supports the MCU device type (FZTAT1024DV3A) having on-chip Renesas 1-Mbyte flash memory.

29.9 Protection

There are three types of ROM programming/erasure protection: hardware, software, and error protection.

29.9.1 Hardware Protection

The hardware protection function disables ROM programming and erasure according to the LSI pin settings.

(1) Protection through FWE Pin

When a low level is applied to the FWE pin, the FWE bit in FPMON becomes 0. In this state, a 1 cannot be written to the FENTRY0 bit in FENTRYR; that is, ROM P/E mode cannot be entered, which prevents the ROM from being programmed or erased.

When the FRDY bit is 1 and the FWE pin is driven low, the FCU clears the FENTRY0 bit to disable ROM programming and erasure. If the FRDY bit in FSTATR0 has already been set to 0 before the FWE pin is driven low, the FCU continues command processing. Even while processing a command, the FCU can accept a P/E suspend command. To resume programming or erasing the ROM, reset the FENTRY0 bit to the value that was set before being cleared, and then issue a P/E resume command.

If an attempt is made to issue a programming or erasing command to the ROM against the protection through the FWE pin, the FCU detects an error and enters command-locked state.

(2) Protection through Mode Pins

While the on-chip ROM is disabled, ROM programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 4, MCU Operating Modes. In user boot mode or user program mode, the user boot MAT cannot be programmed or erased.

29.9.2 Software Protection

The software protection function disables ROM programming and erasure according to the control register settings or the lock bit settings in the user MAT. If an attempt is made to issue a programming or erasing command to the ROM against software protection, the FCU detects an error and enters command-locked state.

(1) Protection through FENTRYR

When the FENTRY0 bit is 0, the 1-Mbyte ROM (read addresses: H'00000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF) is set to ROM read mode. In ROM read mode In ROM read mode, the FCU does not accept commands, so ROM programming and erasure are disabled. If an attempt is made to issue an FCU command in ROM read mode, the FCU detects an illegal command error and enters command-locked state (see section 29.9.3, Error Protection).

(2) Protection through Lock Bits

Each erasure block in the user MAT has a lock bit. When the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased. To program or erase the erasure block whose lock bit is 0, set the FPROTCN bit to 1. If an attempt is made to issue a programming or erasing command against protection by lock bits, the FCU detects an programming/erasure error and enters command-locked state (see section 29.9.3, Error Protection).

29.9.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the ROM cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While the ROMAEINT bit in FAEINT is 1, an FIFE interrupt is generated if the ROMAE bit in FASTAT becomes 1.

Table 29.13 shows the error protection types dedicated for the ROM, those used in common by the ROM and the FLD, and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FCUERR bit in FSTATR1, and the ROMAE bit in FASTST) after each error detection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the ROM. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

Table 29.13 Error Protection Types

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
FENTRYR setting error	The value set in FENTRYR is not H'0001, H'0002, H'0008, H'0010, or H'0080.	1	0	0	0	0
	The FENTRYR setting for resuming operation does not match that for suspending operation.	1	0	0	0	0

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
Illegal command error	An undefined code has been specified in the first cycle of an FCU command.	1	0	0	0	0
	The value specified in the last of the multiple cycles of an FCU command is not H'D0.	1	0	0	0	0
	The peripheral clock specified in PCKAR is not in the range from 1 to 100 MHz.	1	0	0	0	0
	The command issued during programming or erasure is not a suspend command.	1	0	0	0	0
	A suspend command has been issued during operation that is neither programming nor erasure.	1	0	0	0	0
	A suspend command has been issued in suspended state.	1	0	0	0	0
	A resume command has been issued in a state that is not a suspended state.	1	0	0	0	0
	A programming or erasing command (program, lock bit program, block erase) has been issued in programming-suspended state.	1	0	0	0	0
	A block erase command has been issued in erasure-suspended state.	1	0	0	0	0
	A program, lock bit program, or non-interleaved program command has been issued for an erasure-suspended area in erasure-suspended state.	1	0	0	0	0
	The value specified in the second cycle of a program command is not H'80.	1	0	0	0	0
	A block erasure command, lock bit read 2 command, or lock bit programming command was issued at an address that is not within the erasure block set in FIEBAR.	1	0	0	0	0
	A command has been issued in command-locked state.	1	0/1	0/1	0/1	0/1

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
Erasure error	An error has occurred during erasure processing.	0	1	0	0	0
	A block erase command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.	0	1	0	0	0
Programming error	An error has occurred during programming processing.	0	0	1	0	0
	A program, lock bit program, or program command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.	0	0	1	0	0
FCU error	An error has occurred during CPU processing in the FCU.	0	0	0	1	0
ROM access error	A read access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 1 in ROM P/E normal mode.	1	0	0	0	1
	An access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 0	1	0	0	0	1
	A read access command has been issued to addresses H'00000000 to H'001FFFFF while the FENTRYR register value is not H'0000	1	0	0	0	1
	A ROM programming or erasing command (interleaved program, lock bit program, or block erase command) has been issued while the user boot MAT is selected.	1	0	0	0	1
	An access command has been issued to an address other than the addresses for ROM programming/erasure H'80800000 to H'80807FFF while the user boot MAT is selected.	1	0	0	0	1

29.10 Interrupt Sources

Table 29.14 shows the interrupt sources.

The interrupt sources are enabled or disabled by the ROM access violation interrupt enable bit (ROMAEIE) or FCU command lock interrupt enable bit (CMDLKIE) in the flash-access error-interrupt enabling register (FAEINT). When the access violation bit (ROMAE) in the flash access status register (FASTAT) or FCU command lock bit (CMDLK) is set to 1, , a flash interface error interrupt is generated if the ROM access violation interrupt enable bit (ROMAEIE) or FCU command lock interrupt enable bit (CMDLKIE), respectively, is set to 1.

The interrupt request is cancelled by clearing the interrupt flag bit to 0.

Table 29.14 Types and conditions of interrupt requests

Interrupt sources	Interrupt enable bit	Interrupt flag	Condition
Interrupt due to Flash Interface Errors	ROMAEIE, CMDLKIE	ROMAE, CMDLK	ROMAEIE • ROMAE + CMDLKIE • CMDLK

29.11 Usage Notes

29.11.1 Switching between User MAT and User Boot MAT

The user MAT and user boot MAT are allocated to the same address area. If the ROM area is accessed during switching between the user MAT and user boot MAT, an unexpected MAT may be accessed because the number of cycles required to access the ROM area depends on the internal bus status. To avoid such unexpected behavior, take the following steps before and after MAT switching.

1. Modifying interrupt settings before MAT switching

   There are two ways to avoid ROM area access due to an interrupt during MAT switching: one is to specify the interrupt vector fetch destination outside the ROM area through the vector base register (VBR) setting in the CPU, and the other is to mask interrupts. Note that NMI interrupts cannot be masked in this LSI; when masking interrupts to avoid ROM area access in this LSI, design the system so that no NMI is generated during MAT switching.
2. Switching between MATs through a program outside the ROM area

   To avoid CPU instruction fetch in the ROM area during MAT switching, execute the MAT switching processing outside the ROM area.
3. Performing dummy read of ROMMAT

   After writing to ROMMAT to switch between MATs, perform a dummy read of ROMMAT to ensure that the register write is completed.

Figure 29.33 shows an example of the MAT switching process.

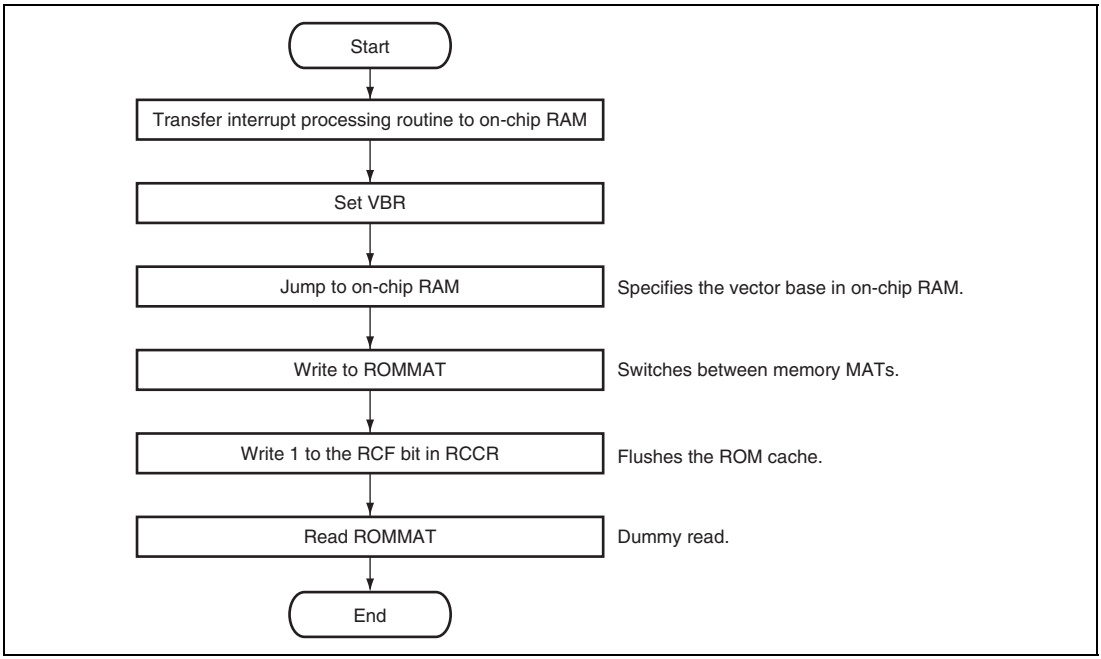


Figure 29.33 Example of MAT Switching Steps

29.11.2 State in which Interrupts are Ignored

In the following mode or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

29.11.3 Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no instruction is executed or no data is read from the programming-suspended or erasure-suspended area.

29.11.4 Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcomputers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

29.11.5 FWE Pin State

Ensure that the FWE pin level does not change during programming or erasure. If the FWE level goes low, the current programming or erasure terminates abnormally and the FRDY bit is set to 1 (the erasure or programming error bit in FASTATR0 is set), and then FENTRYR is cleared. To reprogram ROM, do it after erasing data with the FWE pin at the high level.

In a transition from single-chip mode to user program mode, issue an FCU command after driving the FWE pin high, making sure that the FWE bit in FPMON is set to 1, and setting the FENTRYR register.

In a transition from user program mode to single-chip mode, drive the FWE pin low after ROM programming is completed, making sure that the FRDY bit in FSTATR0 is set to 1, and clearing the FENTRYR register.

For ROM protection in a mode that begins with the FWE pin at the high level, drive the FWE pin low tMDH1 after the reset is cleared.

Cancel ROM protection using the same steps as the transition from single-chip mode to user program mode, and set ROM protection using the same steps as the transition from user program mode to single-chip mode.

29.11.6 Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t_{RESW2} (see section 35, Electrical Characteristics). Since a high voltage is applied to the ROM during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the ROM while the FCU is in the reset state.

When a power-on reset is generated by asserting the $\overline{\text{RES}}$ pin during programming or erasure of the flash memory, hold the reset state for a period of t_{RESW2} (see section 35, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the ROM and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

When executing a power-on reset by asserting the $\overline{\text{RES}}$ pin or the FCU reset with the FRESET bit set in FRESETR during programming/erasure, all data including a lock bit of a programming/erasure target area are undefined.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the ROM, or initialization of its internal circuit.

29.11.7 Suspension by Programming/Erasure Suspension

When suspending programming/erasure processing with the programming/erasure suspend command, make sure to complete the operations with the resume command.

29.11.8 Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

29.11.9 Allocation of Interrupt Vectors during Programming and Erasure

Generation of an interrupt during programming and erasure can lead to fetching from the vector in the flash memory (ROM). For this reason, prepare the interrupt vector table and the interrupt processing routines in areas other than the flash memory (ROM).

29.11.10 Items Prohibited during Programming and Erasure

High voltages are applied within the flash memory (ROM) during programming and erasure. To prevent destruction of the chip, ensure that the following operations do not proceed during programming and erasure.

- Cutting off the power supply
- Transitions to software standby mode, deep software standby mode, or module standby for the flash memory (ROM). Confirm that the FRDY bit in FSTSTRO register is 1 before allowing any of these transitions.
- Rewriting the value of the FRQCR register
- Read access to the flash memory by the CPU, DMAC or DTC
- Setting the PCKAR register for a different frequency from that of Pϕ.

29.11.11 Prohibition after the Flash Memory (ROM) Returns from Module Standby

Refrain from writing, erasing and reading flash memory over a period of 500 μ s after the flash memory (ROM) has returned from module standby.

29.11.12 Abnormal Ending of Programming or Erasure

A lock bit may be set to 0 (in the protected state) due to a reset, an FCU reset by the FRESET bit in the FRESETR register, a transition to the command-locked state because an error has been detected, or programming or erasure not being completed normally.

If this is the case, issue a block erase command to erase the lock bit while the FPROTR.FPROTCN bit is set to 1. After that, repeat the programming until it is finished.

Section 30 Data Flash (FLD)

This LSI includes 32 Kbytes of flash memory (FLD) for storing data. The FLD has the following features.

30.1 Features

- Flash-memory MATs
  Data MAT: 32 Kbytes (8 Kbytes × 4 blocks)

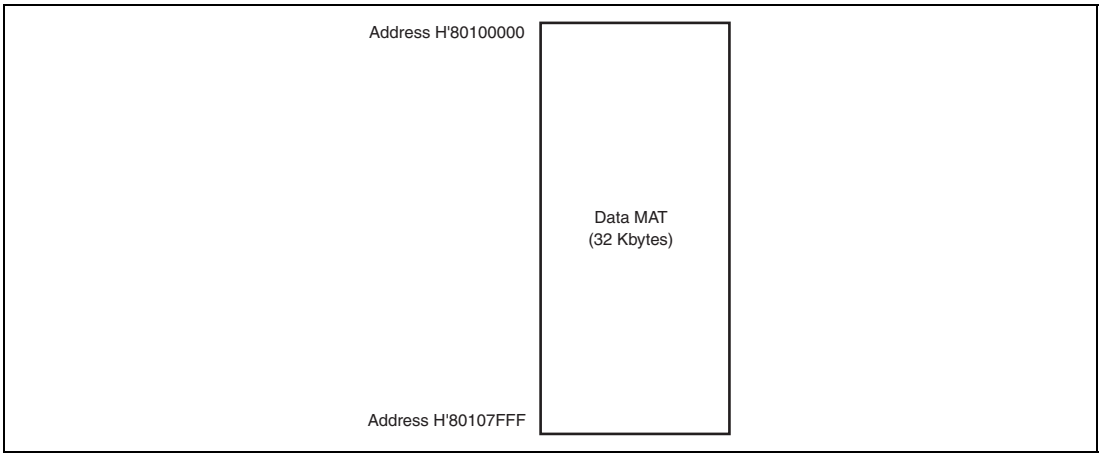


Figure 30.1 Memory MAT Configuration in FLD

- Accesses through the peripheral bus

  The data MAT can be accessed through the peripheral bus.

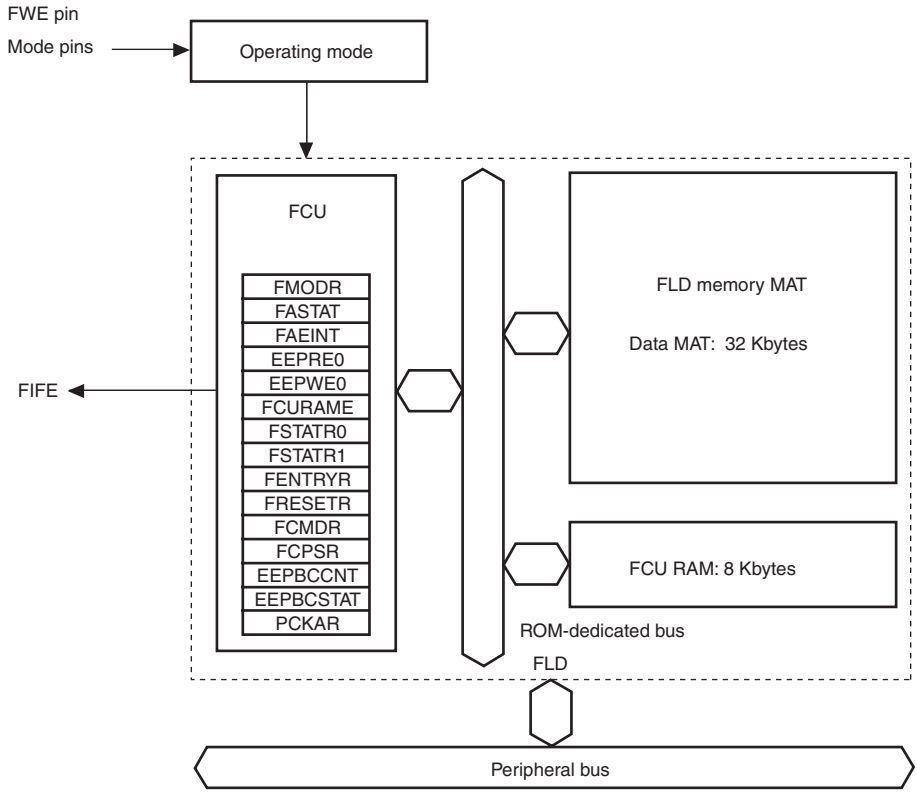
  The program used for reading can be executed from the on-chip RAM or ROM.
- Programming and erasing methods

  The FLD has a dedicated sequencer (FCU) for reprogramming of the flash-memory MATs. The FLD is programmed and erased by issuing commands to the FCU.
- BGO (background operation) function

  The CPU can execute programs located in areas other than the ROM while the FCU is programming or erasing the ROM.

  A program located in ROM can be executed while the FCU is programming or erasing the FLD.
- Suspending and resuming operation

  After the FCU has suspended programming or erasing the FLD, and the CPU has read data in the FLD, the FCU can resume programming or erasure of the FLD. These operations are called suspension (suspend processing) and resumption (resume processing).



[Legend]

FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
EEPRE0:	FLD read enable register 0
EEPWE0:	FLD program/erase enable register 0
FCURAME:	FCU RAM enable register
FSTATR0, FSTATR1:	Flash status registers 0 and 1
FENTRYR:	Flash P/E mode entry register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FCPSR:	FCU processing switch register
EEPBCCNT:	FLD blank check control register
EEPBCSTAT:	FLD blank check status register
PCKAR:	Peripheral clock notification register

Figure 30.2 Block Diagram of FLD

- Programming/erasing unit

  The data MAT is programmed in 8-byte or 128-byte units and erased in block units (8 Kbytes) in user mode, user program mode, and user boot mode. In boot mode, the data MAT is programmed in 256-byte units and erased in block units (8 Kbytes).

  Figure 30.3 shows the block configuration of the data MAT of this LSI. The data MAT is divided into four 8-Kbyte blocks (DB00 to DB03).

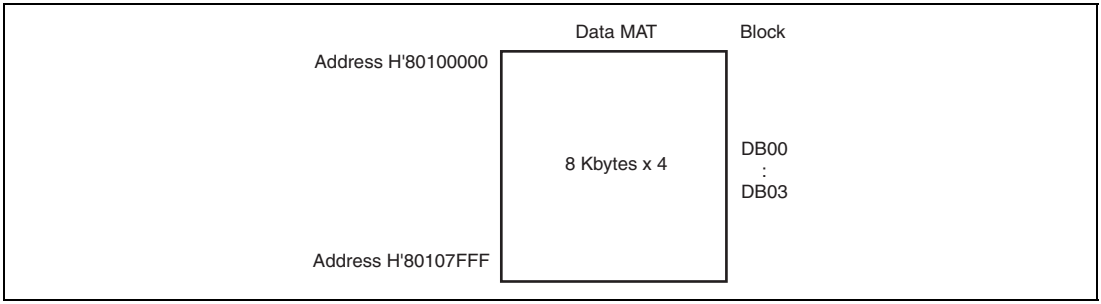


Figure 30.3 Block Configuration of Data MAT

- Blank check function

If data is read from erased FLD by the CPU, undefined values are read. Using blank check command of the FCU allows checking of whether the FLD is erased (in a blank state). Either an 8 Kbytes (1 erasure block) or 8 bytes of area can be checked by a single execution of the blank check command.

- Three types of on-board programming modes
  - — Boot mode

    The FLD can be programmed using the SCI. The bit rate for SCI communications between the host and the LSI can be automatically adjusted.
  - — User mode/user program mode

    The FLD can be programmed with a desired interface. The user mode includes the MCU extended mode and MCU single-chip mode (modes 2 and 3) in which the on-chip ROM is enabled.
  - — User boot mode

    The FLD can be programmed with a desired interface. To make a transition to this mode, a reset is needed.

- Protection modes

  This LSI supports two modes to protect memory against programming, erasing, or reading: hardware protection by the levels on the mode pins and software protection by the setting of the FENTRYD bit, EEPRE0 register, or EEPWE0 register. The FENTRYD bit enables or disables FLD programming or erasure by the FCU. EEPRE0 controls protection of each data MAT block against reading, and EEPWE0 controls protection of each data MAT block against programming and erasure.

  The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure. In addition, the LSI provides a function to protect the FLD against instruction fetch attempted by the CPU.

- Programming and erasing time and count

  Refer to section 35, Electrical Characteristics.

30.2 Input/Output Pins

Table 30.1 shows the input/output pins used for the FLD. The combination of FWE, MD1 and MD0 pin levels determines the FLD programming mode (see section 30.4, Overview of FLD-Related Modes). In boot mode, programming and erasing the FLD can be performed by the host via the PA0/RXD0 and PA1/TXD0 pins (refer to section 30.5, Boot Mode).

Table 30.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	This LSI enters the power-on reset state when this signal goes low.
Test reset	$\overline{\text{TRST}}$	Input	Initialization-signal input pin.
Flash write enable	FWE	Input	This pin has hardware protection function for flash memory rewriting.
Mode 1	MD1	Input	This pin specifies the operating mode for this LSI.
Mode 0	MD0	Input	This pin specifies the operating mode for this LSI.
SCI transmit data	TXD0 (PA1)	Output	This pin outputs serial transmit data (used in boot mode)
SCI receive data	RXD0 (PA0)	Input	This pin inputs serial receive data (used in boot mode)

30.3 Register Descriptions

Table 30.2 shows the FLD-related registers. Some of these registers have ROM-related bits, but this section only describes the FLD-related bits. For the registers consisting of bits used by the ROM and FLD in common (FCURAME, FSTATR0, FSTATR1, FRESETR, FCMDR, and FCPSR) and the ROM-dedicated bits, refer to section 29.3, Register Descriptions. The FLD-related registers are initialized by a power-on reset. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 30.2 Register Configuration

Register Name	Symbol	R/W*1	Initial Value	Address	Access Size
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*2	H'00	H'FFFFA810	8
Flash access error interrupt enable register	FAEINT	R/W	H'9F	H'FFFFA811	8
FLD read enable register 0	EEPRE0	R/(W)*3	H'0000	H'FFFFA840	8, 16
FLD program/erase enable register 0	EEPWE0	R/(W)*3	H'0000	H'FFFFA850	8, 16
FCU RAM enable register	FCURAME	R/(W)*3	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*5	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*5	H'FFFFA901	16
Flash P/E mode entry register	FENTRYR	R/(W)*4	H'0000*5	H'FFFFA902	8, 16
Flash reset register	FRESETR	R/(W)*3	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*5	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W	H'0000*5	H'FFFFA918	8, 16
FLD blank check control register	EEPBCCNT	R/W	H'0000*5	H'FFFFA91A	8, 16
FLD blank check status register	EEPBCSTAT	R	H'0000*5	H'FFFFA91E	8, 16
Peripheral clock notification register	PCKAR	R/W	H'0000*5	H'FFFFA938	8, 16

Notes: 1. In on-chip ROM disabled mode, the bits of the FLD-related registers are always read as 0 and writing to them is ignored.
2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.
4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.
5. This register can be initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

30.3.1 Flash Mode Register (FMODR)

FMODR specifies an operating mode for the FCU. In on-chip ROM disabled mode, the FMODR bits are always read as H'00, and writing to them is ignored. FMODR can be initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FRDMD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	FRDMD	0	R/W	FCU Read Mode Select Bit Selects the read mode to read the ROM or FLD using FCU. This bit specifies the FLD lock bit read mode transition or blank check processing in the FLD (see section 30.6.1, FCU Command List, 30.6.3, FCU Command Usage), whereas this bit must be set to specify the read method for the lock bits in the ROM (see section 29, Flash Memory (ROM)). 0: Memory area read mode This mode is selected to enter the FLD lock bit read mode. Since the FLD has no lock bits, reading an FLD area results in an undefined value. 1: Register read mode To make the blank check command available for use, register read mode is set.
3 to 0	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.

30.3.2 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and FLD. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 30.7.3, Error Protection). To cancel command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	ROM Access Error Refer to section 29, Flash Memory (ROM).
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLK	0	R	FCU Command Lock Indicates whether the FCU is in command-locked state (see section 30.7.3, Error Protection). 0: The FCU is not in command-locked state 1: The FCU is in command-locked state [Setting condition] • When the FCU detects an error and enters command-locked state [Clearing condition] • When the FCU completes the status-clear command processing

Bit	Bit Name	Initial Value	R/W	Description
3	EEPAE	0	R/(W)*	FLD Access Error Indicates whether an access error has been generated for the FLD. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters command-locked state. 0: No FLD access error has occurred 1: An FLD access error has occurred [Setting conditions] • When a read access command is issued to the FLD area while the FENTRYD bit in FENTRYR is 1 in FLD P/E normal mode • When a write access command is issued to the FLD area while the FENTRYD bit is 0 • When an access command is issued to the FLD area while the FENTRY0 bit in FENTRYR is 1 [Clearing condition] • 0 is written to this bit after reading EEPAE = 1.
2	EEPIFE	0	R/(W)*	FLD Instruction Fetch Error Indicates whether an instruction fetch error has been generated for the FLD. 0: No FLD instruction fetch error has occurred 1: An FLD instruction fetch error has occurred [Setting condition] • When an attempt is made to fetch an instruction from the FLD [Clearing condition] • When 0 is written to this bit after reading EEPIFE = 1.

Bit	Bit Name	Initial Value	R/W	Description
1	EEPRPE	0	R/(W)*	FLD Read Protect Error Indicates whether an error has been generated against the FLD read protection provided by the EEPRE0 and EEPRE1 settings. 0: The FLD has not been read against the EEPRE0 setting 1: An attempt has been made to read data from the FLD against the EEPRE0 setting [Setting condition] • When an attempt is made to read data from the FLD area that has been read-protected through the EEPRE0 setting [Clearing condition] • When 0 is written to this bit after reading EEPRPE = 1
0	EEPWPE	0	R/(W)*	FLD Program/Erase Protect Error Indicates whether an error has been generated against the FLD program/erasure protection provided by the EEPWE0 setting. 0: No programming or erasing command has been issued to the FLD against the EEPWE0 setting 1: A programming or erasing command has been issued to the FLD against the EEPWE0 setting [Setting condition] • When a programming or erasing command is issued to the FLD area that has been program/erase-protected through the EEPWE0 setting [Clearing condition] • When 0 is written to this bit after reading EEPWPE = 1

Note: * Only 0 can be written to clear the flag after 1 is read.

30.3.3 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupt requests. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
Initial value:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAEIE	1	R/W	ROM Access Error Interrupt Enable Refer to section 29, Flash Memory (ROM).
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLKIE	1	R/W	FCU Command Lock Interrupt Enable Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. 0: Disables an FIFE interrupt request when CMDLK = 1. 1: Enables an FIFE interrupt request when CMDLK = 1.
3	EEPAEIE	1	R/W	FLD Access Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD access error occurs and the EEPAE bit in FASTAT becomes 1. 0: Disables an FIFE interrupt request when EEPAE = 1. 1: Enables an FIFE interrupt request when EEPAE = 1.

Bit	Bit Name	Initial Value	R/W	Description
2	EEPIFEIE	1	R/W	FLD Instruction Fetch Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD instruction fetch error occurs and the EEPIFE bit in FASTAT becomes 1. 0: Disables an FIFE interrupt request when EEPIFE = 1. 1: Enables an FIFE interrupt request when EEPIFE = 1.
1	EEPRPEIE	1	R/W	FLD Read Protect Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD read protect error occurs and the EEPRPE bit in FASTAT becomes 1. 0: Disables an FIFE interrupt request when EEPRPE = 1. 1: Enables an FIFE interrupt request when EEPRPE = 1.
0	EEPWPEIE	1	R/W	FLD Program/Erase Protect Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD program/erase protect error occurs and the EEPWPE bit in FASTAT becomes 1. 0: Disables an FIFE interrupt request when EEPWPE = 1 1: Enables an FIFE interrupt request when EEPWPE = 1

30.3.4 FLD Read Enable Register 0 (EEPRE0)

EEPRE0 enables or disables read access to blocks DB00 to DB03 (see figure 30.3) in the data MAT. In on-chip ROM disabled mode, EEPRE0 is read as H'0000 and writing to it is ignored. EEPRE0 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	DBRE03	DBRE02	DBRE01	DBRE00
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R/W	R/W	R/W	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	All 0	R/(W)*	Key Code These bits enable or disable DBRE03 to DBRE00 bit modification. The data written to these bits are not stored.
7 to 4	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
3	DBRE03	0	R/W	DB03 to DB00 Block Read Enable Enables or disables read access to blocks DB03 to DB00 in the data MAT. The DBREi bit (i = 03 to 00) controls read access to block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'2D is written to the KEY bits. 0: Disables read access 1: Enables read access
2	DBRE02	0	R/W	
1	DBRE01	0	R/W	
0	DBRE00	0	R/W	

Note: * Write data is not retained.

30.3.5 FLD Program/Erase Enable Register 0 (EEPWE0)

EEPWE0 enables or disables programming and erasure of blocks DB00 to DB03 (see figure 30.3) in the data MAT. In on-chip ROM disabled mode, EEPWE0 is read as H'0000 and writing to it is ignored. EEPWE0 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	DBWE03	DBWE02	DBWE01	DBWE00
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R/W	R/W	R/W	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	All 0	R/(W)*	Key Code These bits enable or disable DBWE03 to DBWE00 bit modification. The data written to these bits are not stored.
7 to 4	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
3	DBWE03	0	R/W	DB03 to DB00 Block Program/Erase Enable Enables or disables programming and erasure of blocks DB03 to DB00 in the data MAT. The DBWEi bit (i = 03 to 00) controls programming and erasure of block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'1E is written to the KEY bits. 0: Disables programming and erasure 1: Enables programming and erasure
2	DBWE02	0	R/W	
1	DBWE01	0	R/W	
0	DBWE00	0	R/W	

Note: * Write data is not retained.

30.3.6 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or FLD. To specify the P/E mode for the ROM or FLD so that the FCU can accept commands, set either FENTRYD or FENTRY0 to 1. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR is initialized by a power-on reset, or setting the FRESET bit in FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FEKEY								FENTRYD	—	—	—	—	—	—	FENTRY0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FEKEY	H'00	R/(W)*	Key Code These bits enable or disable the FENTRYD and FENTRY0 bit modification. The data written to these bits are not retained.

Bit	Bit Name	Initial Value	R/W	Description
7	FENTRYD	0	R/W	FLD P/E Mode Entry This bit specifies the P/E mode for the FLD. 00: The FLD is in read mode 11: The FLD is in P/E mode [Write enabling conditions] When the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRYD while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRYD while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.
6 to 1	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
0	FENTRY0	0	R/W	ROM P/E Mode Entry 1, 0 Refer to section 29, Flash Memory (ROM).

Note: * Write data is not retained.

30.3.7 FLD Blank Check Control Register (EEPBCCNT)

EEPBCCNT specifies the addresses and sizes of the target areas to be checked by the blank check command. In on-chip ROM disabled mode, EEPBCCNT is read as H'0000, and writing to it is ignored. EEPBCCNT is initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BCADR										-	-	BC SIZE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved The write value must always be 0; otherwise operation is not guaranteed.
12 to 3	BCADR	All 0	R/W	Blank Check Address Setting Bit Use these bits to specify the address of the target area when the size of the target area to be checked by the blank check command is 8 bytes (the BCSIZE bit is set to 0). When the BCSIZE bit is set to 0, the start address of the target area is the value obtained by summing the EEPBCCNT value (the value obtained by shifting the set BCADR value by 3 bits) and the start address of an erased block specified when a blank check command is issued.

Bit	Bit Name	Initial Value	R/W	Description
2, 1	—	All 0	R	Reserved The write value must always be 0; otherwise operation is not guaranteed.
0	BCSIZE	0	R/W	Blank Check Size Setting Bit This bit selects the size of the target area to be checked by the blank check command. 0: Selects 8 bytes as the size of a blank check target area. 1: Selects 8 Kbytes as the size of a blank check target area.

30.3.8 FLD Blank Check Status Register (EEPBCSTAT)

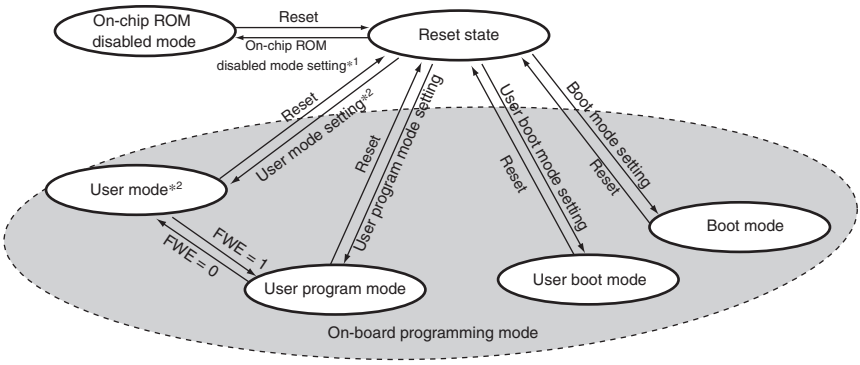
EEPBCSTAT stores check results by executing the blank check command. In on-chip ROM disabled mode, EEPBCSTAT is read as H'0000, and writing to it is ignored. EEPBCSTAT is initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BCST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved The write value must be 0; otherwise operation is not guaranteed.
0	BCST	0	R	Blank Check Status Bit Indicates the result of a blank check. 0: The target area is erased (blank). 1: The target area is filled with 0s and/or 1s.

30.4 Overview of FLD-Related Modes

Figure 30.4 shows the FLD-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD1 and MD0 pin settings, refer to section 4, MCU Operating Modes.



Notes: 1. Indicates the MCU extended modes 0 and 1.
2. Indicates the MCU extended mode 2 and single chip mode.

Figure 30.4 FLD-Related Mode Transition

- The FLD cannot be read, programmed, or erased in on-chip ROM disabled mode.
- The FLD can be read, programmed, and erased on the board in user mode, user program mode, user boot mode, and boot mode.
- In user mode, the ROM cannot be programmed or erased but the FLD can be programmed and erased. While the FLD is being programmed or erased, the ROM can be read. Therefore, the user can program the FLD while executing an application program in the ROM protected against programming and erasure.

Table 30.3 compares programming- and erasure-related items for the boot mode, user mode, user program mode, and user boot mode.

Table 30.3 Comparison of Programming Modes

Item	Boot Mode	User Mode	User Program Mode	User Boot Mode
Programming/ erasure environment	On-board programming			
Programming/ erasure enabled MAT	FLD	FLD	FLD	FLD
Programming/ erasure control	Host	FCU	FCU	FCU
Entire area erasure	Available (automatic)	Available	Available	Available
Block erasure	Available*1	Available	Available	Available
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM	From any device via RAM
Reset-start MAT	Embedded program stored MAT	User MAT	User MAT	User boot MAT*2

Notes: 1. The entire area is erased when the LSI is started. After that, a specified block can be erased.
2. After the LSI is started in the embedded program stored MAT and the boot program provided by Renesas Corp. is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- In boot mode, the user MAT and user boot MAT in the ROM and the FLD are all erased immediately after the LSI is started. The FLD can then be programmed from the host via the SCI. The FLD can also be read after this entire area erasure.
- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user mode or user program mode.

30.5 Boot Mode

To program or erase the FLD in boot mode, send control commands and programming data from the host. For the system configuration and settings in boot mode, refer to section 29, Flash Memory (ROM). This section describes only the commands dedicated for the FLD.

30.5.1 Inquiry/Selection Host Commands

Table 30.4 shows the inquiry/selection host commands dedicated to the FLD. The data MAT inquiry and data MAT information inquiry commands are used in the step for inquiry regarding the MAT programming information shown in figure 29.11 in section 29.5.4, Inquiry/Selection Host Command Wait State.

Table 30.4 Inquiry/Selection Host Commands (for FLD only)

Host Command Name	Function
Data MAT inquiry	Inquires regarding the availability of the data MAT
Data MAT information inquiry	Inquires regarding the number of data MAT areas and the start and end addresses

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Data MAT Inquiry

In response to the data MAT inquiry command sent from the host, this LSI returns the information concerning the availability of the data MAT.

Command	H'2A

Response	H'3A	Size	Availability	SUM

[Legend]

Size (1 byte): Total number of characters in the availability field (fixed at 1)

Availability (1 byte): Availability of the data MAT (fixed at H'01)
H'00: No data MAT is available
H'01: Data MAT is available

SUM (1 byte): Checksum

(2) Data MAT Information Inquiry

In response to the data MAT information inquiry command sent from the host, this LSI returns the number of data MAT areas and their addresses.

Command	H'2B

Response			
	H'3B	Size	Area count
	Area start address		
	Area end address		
	Area start address		
	Area end address		
	:		
	Area start address		
	Area end address		
	SUM		

[Legend]

Size (1 byte): Total number of bytes in the area count, area start address, and area end address fields

Area count (1 byte): Number of data MAT areas (consecutive areas are counted as one area)

Area start address (4 bytes): Start address of the data MAT area

Area end address (4 bytes): End address of the data MAT area

SUM (1 byte): Checksum

The information concerning the block configuration in the data MAT is included in the response to the erasure block information inquiry command (refer to section 29.5.4, Inquiry/Selection Host Command Wait State).

30.5.2 Programming/Erasing Host Commands

Table 30.5 shows the programming/erasing host commands dedicated to the FLD. FLD-dedicated host commands are provided only for data MAT checksum and blank check; the programming, erasing, and reading commands are used in common for the ROM and data MAT.

To program the FLD, issue from the host a user MAT programming selection command and then a 256-byte programming command specifying the data MAT address as the programming address. To erase the data MAT, issue an erasure selection command and then a block erasure command specifying an erasure block in the data MAT. The information concerning the erasure block configuration in the data MAT is included in the response to the erasure block information inquiry command. To read data from the data MAT, select the user MAT through a memory read command specifying the data MAT address as the read address.

For the user MAT programming selection, user boot MAT programming selection, 256-byte programming, erasure selection, block erasure selection, and memory read commands, refer to section 29.5.5, Programming/Erasing Host Command Wait State. For the erasure block information inquiry command, refer to section 29.5.4, Inquiry/Selection Host Command Wait State.

Table 30.5 Programming/Erasure Host Commands (for FLD)

Host Command Name	Function
Data MAT checksum	Performs checksum verification for the data MAT
Data MAT blank check	Checks whether the data MAT is blank

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Data MAT Checksum

In response to the data MAT checksum command sent from the host, this LSI sums the FLD data in byte units and returns the result (checksum).

Command	H'61			
Response	H'71	Size	MAT checksum	SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the data MAT data

SUM (4 bytes): Checksum (for the response data)

(2) Data MAT Blank Check

In response to the data MAT blank check command sent from the host, this LSI checks whether the data MAT is completely erased. When the data MAT is completely erased, this LSI returns a response (H'06). If the data MAT has an unerased area, this LSI returns an error response (sends H'E2 and H'52 in that order).

Command	H'62	
Response	H'06	
Error response	H'E2	H'52

30.6 User Mode, User Program Mode, and User Boot Mode

30.6.1 FCU Command List

To program or erase the FLD in user mode, user program mode, or user boot mode, issue FCU commands to the FCU. Table 30.6 is a list of FCU commands for FLD programming and erasure.

Table 30.6 FCU Command List (FLD-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 30.6.2, Conditions for FCU Command Acceptance).
Status read mode transition	Moves to the status read mode (see section 30.6.2, Conditions for FCU Command Acceptance).
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 30.6.2, Conditions for FCU Command Acceptance).
Program	Programs FLD (in 8-byte or 128-byte units).
Block erase	Erases FLD (in block units).
P/E suspend	Suspends programming or erasure.
P/E resume	Resumes programming or erasure.
Status register clear	Clears the IRGERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state.
Blank check	Checks if a specified area is erased (blank).
Peripheral clock notification	Specifies the peripheral clock frequency

FCU commands other than the program command and blank check command are also used for ROM programming and erasure. When the blank check command is issued to the ROM, the lock bits in the ROM are read out.

To issue a command to the FCU, access the FLD area through the peripheral bus. Table 30.7 shows the FCU command formats for the program command and blank check command. For the other command formats, refer to section 29.6.1, FCU Command List. When a peripheral-bus access, as shown in table 30.7, is made under specified conditions, the FCU performs processing specified by a selected command. For the conditions for the FCU command acceptance, refer to section 30.6.2, Conditions for FCU Command Acceptance. For details of command usage, refer to section 30.6.3, FCU Command Usage.

When the FRDMD bit is set to 0 (memory area read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU accepts the lock bit read mode transition command. Since the FLD has no lock bits, making peripheral-bus read access to FLD areas after a transition to the lock bit read mode results in undefined read data. The FCU detects no access violation error when the undefined data is read. When the FRDMD bit is set to 1 (register read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU enters a waiting state to wait for the command in the second cycle (H'D0) of the blank check command. At this stage, if H'D0 is written into an FLD area by a peripheral-bus write access, the FCU detects it and starts performing the blank check processes specified by the set values in the EEPBCCNT register, and once the check completes the FCU writes check results into the EEPBCSTAT register.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 29.6.4, Suspending Operation.

Table 30.7 FCU Command Formats (for FLD only)

Command	Number of Command Cycles*	First Cycle		Second Cycle		Third Cycle		Fourth Cycle to Cycle N + 2		Cycle N + 3	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Program (8-byte programming: N = 4)	7	EA	H'E8	EA	H'04	WA	WD1	EA	WDn	EA	H'D0
Program (128-byte programming: N = 64)	67	EA	H'E8	EA	H'40	WA	WD1	EA	WDn	EA	H'D0
Blank check	2	EA	H'71	BA	H'D0	—	—	—	—	—	—

[Legend]

EA: FLD area address
An arbitrary address within the range of H'8010000 to H’80107FFF

WA: The start address of write data

BA: The address of an FLD erasure block
(An arbitrary address in the erase target block)

WDn: n-th word of programming data (n = 1 to N)

Note: * The numbers of cycles for commands are numbers of cycles of write access to programming/erasure addresses over the peripheral bus (P bus) by the CPU.

30.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 30.5 is an FCU mode transition diagram.

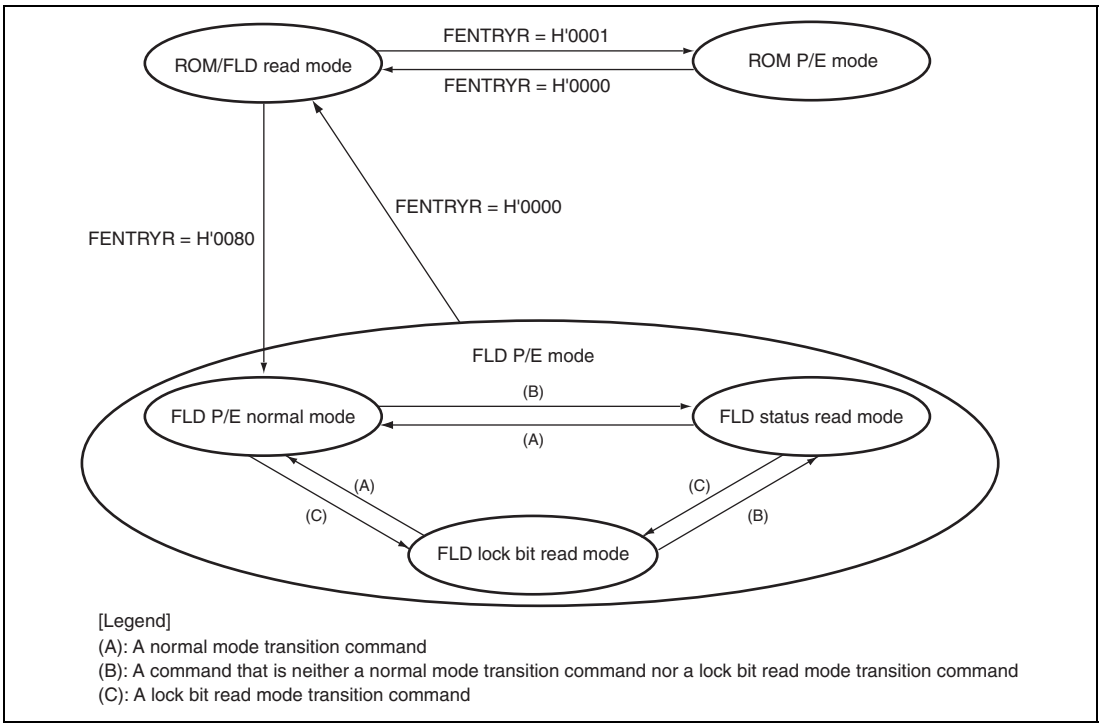


Figure 30.5 FCU Mode Transition Diagram (FLD-Related Modes)

(1) ROM P/E Mode

The FCU can accept ROM programming and erasing commands in this mode. The FLD cannot be read. The FCU enters this mode when the FENTRYD bit is set to 0 and the FENTRY0 bit is set to 1 in FENTRYR. For details of this mode, refer to section 29.6.2, Conditions for FCU Command Acceptance.

(2) ROM/FLD Read Mode

The FLD can be read through the peripheral bus at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY0 bit is set to 0 and the FENTRYD bit in FENTRYR is set to 0.

(3) FLD P/E Mode

- FLD P/E normal mode

  The FCU enters this mode when the FENTRYD bit is set to 1 and the FENTRY0 bit is set to 0 in ROM/FLD read mode or ROM P/E mode, or when a normal mode transition command is accepted in FLD P/E mode. Table 30.8 shows the commands that can be accepted in this mode. If the FLD area is read through the peripheral bus, an FLD access error occurs and the FCU enters the command-locked state. High-speed read operation is available for ROM.
- FLD status read mode

  The FCU enters this mode when the FCU accepts a command that is neither the normal mode transition command nor the lock bit read mode transition command in FLD P/E mode. The FLD status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 30.8 shows the commands that can be accepted in this mode. If the FLD area is read through the peripheral bus, the FSTATR0 value is read. High-speed read operation is available for ROM.
- FLD lock bit read mode

  The FCU enters this mode when the FCU accepts a lock bit read mode transition command in FLD P/E mode. Table 30.8 shows the commands that can be accepted in this mode. Since the FLD has no lock bits, reading an FLD area via the peripheral bus to FLD areas results in an undefined value. However, no access violation occurs in this case. High-speed read operation is available for ROM.

Table 30.8 shows the correlation between each FCU mode/state and its acceptable commands. When an unacceptable command is issued, the FCU enters the command-locked state (see section 30.7.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR bit values in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR bit values in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 30.8, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and blank check processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 30.8 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 30.8 FCU Modes/States and Acceptable Commands

	P/E Normal Mode			Status Read Mode							Lock Bit Read Mode		
Item	Programming-Suspended	Erasure-Suspended	Other State	Programming/Erasure Processing	Programming/Erasure Suspension Processing	Blank Check Processing	Programming-Suspended	Erasure-Suspended	Command-Locked	Other State	Programming-Suspended	Erasure-Suspended	Other State
FRDY bit in FSTATR0	1	1	1	0	0	0	1	1	0/1	1	1	1	1
SUSRDY bit in FSTATR0	0	0	0	1	0	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0	0	0/1	0	0	1	0	0	0	1	0
PRGSPD bit in FSTATR0	1	0	0	0	0/1	0	1	0	0	0	1	0	0
CMDLK bit in FASTAT	0	0	0	0	0	0	0	0	1	0	0	0	0
Normal mode transition	A	A	A	×	×	×	A	A	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	A	A	×	A	A	A	A
Lock bit read mode transition (lock bit read 1)	A	A	A	×	×	×	A	A	×	A	A	A	A
Program	×	*	A	×	×	×	×	*	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	A	A	×	×	A	A	×
Status register clear	A	A	A	×	×	×	A	A	A	A	A	A	A
Blank check	A	A	A	×	×	×	A	A	×	A	A	A	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	A	×	×	A

[Legend]

A: Acceptable

*: Only programming is acceptable for the areas other than the erasure-suspended block

×: Not acceptable

30.6.3 FCU Command Usage

This section shows how to program and erase the FLD using the program command and block erase command, respectively, and how to check the erasure status of the FLD using the blank check command. For the firmware transfer to the FCU RAM and the other FCU command usage, refer to section 29.6.3, FCU Command Usage.

If the FCU enters the command lock state in the middle of its handling of commands by setting the FCUERR bit in FSTATR1 to 1, the FRDY bit in FSTATR0 retains 0. Since the FCU halts its operation in the command lock state, the FRDY bit is not set to 1 from 0.

If the FRDY bit retains 0 for longer than the programming/erasure time or suspend delay time (see section 35, Electrical Characteristics), an abnormal operation may have occurred. In such case, initialize the FCU by issuing an FCU reset.

If the FRDY bit is set to 1 upon the termination of an FCU command operation, the FCUERR bit is cleared to 0. On the other hand, it can be checked via the ILGLERR, ERSERR, or PRGERR bit whether or not an error has occurred after a command operation terminates.

(1) Using the Peripheral Clock Notification Command

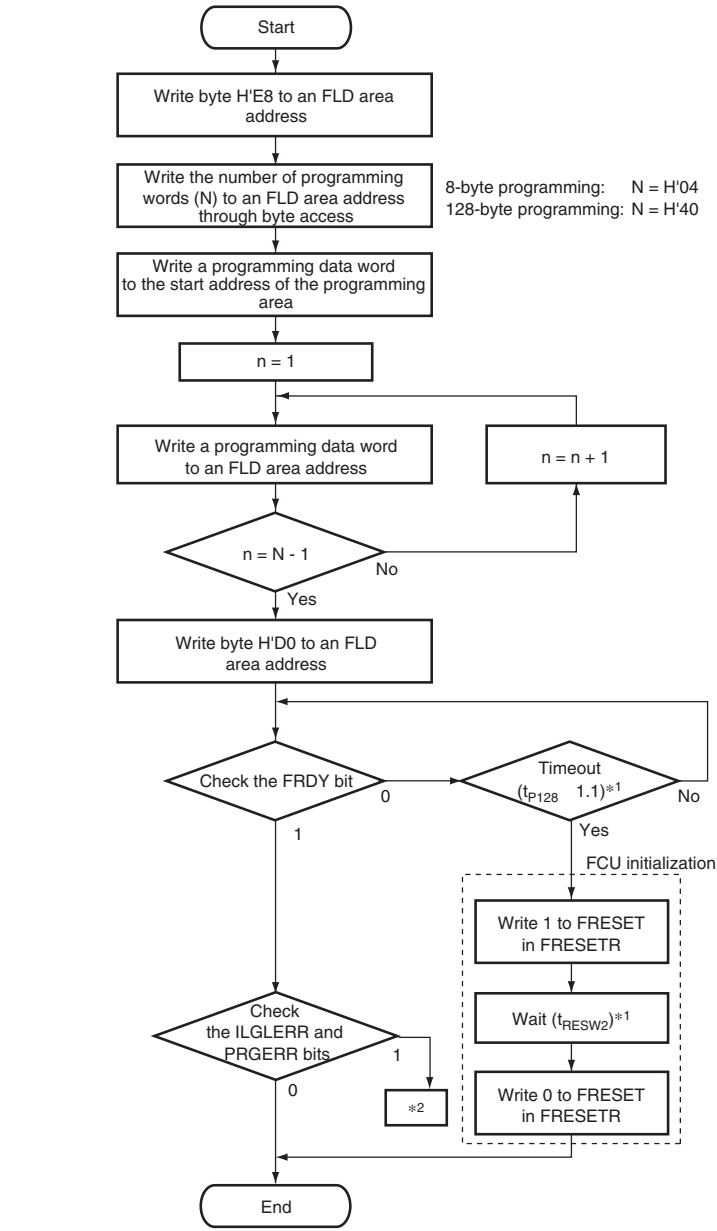
The command is used for notification of the peripheral clock frequency. For details, see section 29.6.3, FCU Command Usage, in section 29, Flash Memory (ROM). Proceed by setting the FENTRYD bit in FENTRYR to 1 and specifying the address as an address within the region corresponding to the data flash (FLD).

(2) Programming

To program the FLD, use the program command. Write byte H'E8 to an FLD area address in the first cycle of the program command and the number of words (N)* to be programmed through byte access in the second cycle. Access the peripheral bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be an 8-byte boundary address for 8-byte programming or a 128-byte boundary address for 128-byte programming. After writing words to FLD area addresses N times, write byte H'D0 to an FLD area address in cycle N + 3; the FCU then starts FLD programming. Read the FRDY bit in FSTATR0 to confirm that FLD programming is completed.

If the area accessed in the third cycle to cycle N + 2 includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the programming and erasure protection provided by the EEPWE0 and EEPWE1 settings, set the program/erase enable bit for the target block to 1 before starting programming. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming. Figure 30.6 shows the procedure for FLD programming

Note: * N = H'04 for 8-byte programming or N = H'40 for 128-byte programming.



Notes: 1. t_{P128} : Time required for programming 128-byte data (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. When ILGLERR = 1 or PRGERR = 1, see figure 29.24 in section 29, Flash Memory (ROM).

Figure 30.6 Procedure for FLD Programming

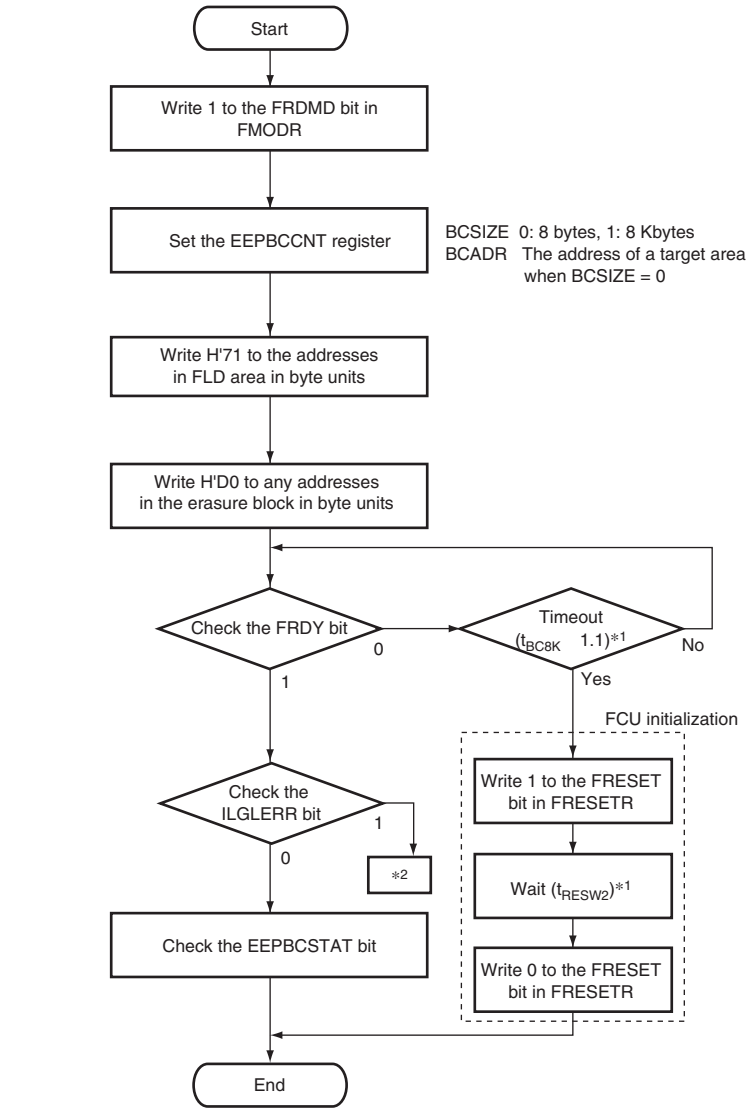
(3) Erasure

To erase the ROM, use the block erase command. The FLD can be erased in the same way as ROM erasure (refer to section 29, Flash Memory (ROM)). Note that the FLD has a programming and erasure protection function through a register. To ignore the programming and erasure protection provided by the EEPWE0 setting, set the program/erase enable bit for the target block to 1 before starting erasure.

(4) Checking of the Erased State

Since reading the FLD erased by the CPU results in undefined values, the blank check command should be used to check the erased state of the FLD. To make the blank check command available for use, set the FRDMD bit in FMODR to 1 to enable the command first, and then specify the size and start address of a target area via the EEPBCCNT register. When the BCSIZE bit of the EEPBCCNT register is set to 1, a check can be performed on the entire erased block (8 Kbytes) specified in the second cycle of the command. When the BCSIZE bit is set to 0, a check can be performed on an 8-byte area starting from the address obtained by summing the start address of the erased area specified in the second cycle of the command and the value held by the EEPBCCNT register. In the first cycle of the command, a value of H'71 is written in byte into an address of the FLD. In the second cycle, once a value of H'D0 is written into a specified address included in the target area, the FCU starts the blank check on the FLD. It can be checked whether or not the check is complete via the FRDY bit in the FSTATR0. After the blank check is complete, it can be checked whether the target area is erased or filled with 0s and/or 1s via the BCST bit of the EEPBCSTAT register.

Figure 30.7 shows the procedure of the FLD blank check.



Notes: 1. t_{BC8K}: Time required for blank check (see section 35, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 35, Electrical Characteristics).
2. When ILGLERR = 1, see figure 29.24 in section 29, Flash Memory (ROM).

Figure 30.7 Procedure of the FLD Blank Check

30.7 Protection

There are three types of FLD programming/erasure protection: hardware, software, and error protection.

30.7.1 Hardware Protection

The hardware protection function disables FLD programming and erasure according to the mode pin settings in this LSI.

While the on-chip ROM is disabled, FLD programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 4, MCU Operating Modes.

30.7.2 Software Protection

The software protection function disables FLD programming and erasure according to the control register settings. If an attempt is made to issue a programming or erasing command to the FLD against software protection, the FCU detects an error and enters command-locked state.

(1) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the FCU does not accept commands for the FLD, so FLD programming and erasure are disabled. If an attempt is made to issue an FCU command for the FLD while the FENTRYD bit is 0, the FCU detects an illegal command error and enters command-locked state (see section 30.7.3, Error Protection).

(2) Protection through EEPWE0

When the DBWEi (i = 00 to 03) bit in EEPWE0 is 0, programming and erasure of block DBi in the data MAT is disabled. If an attempt is made to program or erasure block DBi while the DBWEi bit is 0, the FCU detects a program/erase protect error and enters command-locked state (see section 30.7.3, Error Protection).

30.7.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the FLD cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While an FLD-related interrupt enable bit (EEPAEIE, EEPIFEIE, EEPRPEIE, or EEPWPEIE) in FAEINT is 1, an FIFE interrupt is generated if the corresponding status bit (EEPAE, EEPIFE, EEPRPE, or EEPWPE) in FASTAT becomes 1.

Table 30.9 shows the error protection types for the FLD and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTST) after each error detection. For the error protection types used in common by the ROM and FLD (FENTRYR setting error, most of illegal command errors, erasing error, programming error, and FCU error), refer to section 29.9.3, Error Protection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the FLD. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

Table 30.9 Error Protection Types (for FLD only

Error	Description	ILGLERR	ERSERR	PRGERR	EEPAE	EEPIFE	EEPRPE	EEPWPE
Illegal command error	The value specified in the second cycle of a program command is neither H'04 nor H'40.	1	0	0	0	0	0	0
	A lock bit program command has been issued to an area in the FLD while the FENTRYD bit of FENTRYR register is set to 1.	1	0	0	0	0	0	0
FLD access error	A read access command has been issued to the FLD area while FENTRYD = 1 in FENTRYR in FLD P/E normal mode.	1	0	0	1	0	0	0
	A write access command has been issued to the FLD area while FENTRYD = 0.	1	0	0	1	0	0	0
	An access command has been issued to the FLD area while the FENTRY0 bit in FENTRYR is 1.	1	0	0	1	0	0	0
FLD instruction fetch error	An instruction fetch has been made in the FLD area.	1	0	0	0	1	0	0
FLD read protect error	A read access command has been issued to the FLD area protected against reading through EEPRE0.	1	0	0	0	0	1	0
FLD program protect error	A program command or block erase command has been issued to the FLD area protected against programming and erasure through EEPWE0.	1	0	0	0	0	0	1

30.8 Interrupt Sources

The interrupt request and corresponding sources are listed in table 30.10.

The interrupt sources are enabled or disabled by the FCU command lock interrupt enable bit (CMDLKIE), FLD access error interrupt enable bit (EEPAEIE), FLD instruction fetch error interrupt enable bit (EEPIFEIE), FLD read protect error interrupt enable bit (EEPRPEIE), and FLD program/erase protect error interrupt enable bit (EEPWPEIE) in the flash access error interrupt enable register (FAEINT).

Setting of the FCU command lock bit (CMDLK), FLD instruction fetch error bit (EEPIFE), FLD read protect error bit (EEPRPE), or FLD program/erase protect error bit (EEPWPE) in the flash access status register (FASTAT) to 1 while the corresponding bit from among the FCU command lock interrupt enable bit (CMDLKIE), FLD access error interrupt enable bit (EEPAEIE), FLD instruction fetch error interrupt enable bit (EEPIFEIE), FLD read protect error interrupt enable bit (EEPRPEIE), and FLD program/erase protect error interrupt enable bit (EEPWPEIE), respectively, in the flash access error interrupt enable register (FAEINT) is set to 1, leads to the generation of a flash interface error interrupt (FIFE). An interrupt request is cancelled by clearing the interrupt flag bit to 0.

Table 30.10 Interrupt Request and Corresponding Conditions

Interrupt Request	Interrupt Enable Bits	Interrupt Flags	Condition
Flash interface error interrupt	CMDLKIE, EEPIFEIE, EEPRPEIE, EEPWPEIE	CMDLK, EEPIFE, EEPRPE, EEPWPE	CMDLKIE • CMDLK + EEPAEIE • EEPIFE + EEPRPEIE • EEPRPE + EEPWPEIE • EEPWPE

30.9 Usage Notes

30.9.1 Protection of data MAT Immediately after a Reset is Cancelled

As the initial value of EEPRE0 and EEPWE0 is H'0000, the data MAT programming, erasure, and reading are disabled immediately after a reset is cancelled. To read data from the data MAT, set EEPRE0 appropriately before accessing the FLD. To program or erase the data MAT, set EEPWE0 appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data MAT without setting the registers, the FCU detects an error and enters command-locked state.

30.9.2 State in which Interrupts are Ignored

In the following modes or period, the NMI or maskable interrupt requests are ignored.

- Boot mode
- Programmer mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

30.9.3 Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no data is read from the programming-suspended or erasure-suspended area.

30.9.4 Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcontrollers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

30.9.5 Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t_{RESW2} (see section 35, Electrical Characteristics). Since a high voltage is applied to the FLD during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the FLD while the FCU is in the reset state.

When a power-on reset is generated by asserting the $\overline{\text{RES}}$ pin during programming or erasure of the flash memory, hold the reset state for a period of t_{RESW2} (see section 35, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the FLD and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

When executing a power-on reset by asserting the $\overline{\text{RES}}$ pin or the FCU reset with the FRESET bit set in FRESETR during programming/erasure, all data including a lock bit of a programming/erasure target area are undefined.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the FLD, or initialization of its internal circuit.

30.9.6 Suspension by Programming/Erasure Suspension

When suspending programming/erasure processing with the programming/erasure suspend command, make sure to complete the operations with the resume command.

30.9.7 Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

30.9.8 Items Prohibited During Programming and Erasure

High voltages are applied within the data memory (FLD) during programming and erasure. To prevent destruction of the chip, ensure that the following operations are not performed during programming and erasure.

- Cutting off the power supply
- Transitions to software standby mode, deep software standby mode, and of the flash memory (ROM) to module standby: ensure that the value of the FRDY bit in FSTSTR0 has become 1 when any of these transitions is to be made.
- Read access to the flash memory by the CPU, DMAC, or DTC
- Altering the value of register FRQCR
- Setting the PCKAR register for a different frequency from that of Pϕ.

30.9.9 Items Prohibited for Data Flash (FLD) after Return from Module Standby

Do not program, erase, or read the data flash (FLD) for 500 μs after it has returned from module standby.

30.9.10 Abnormal Ending of Programming or Erasure

A lock bit may be set to 0 (in the protected state) due to a reset, an FCU reset by the FRESET bit in the FRESETR register, a transition to the command-locked state because an error has been detected, or programming or erasure not being completed normally.

If this is the case, issue a block erase command to erase the lock bit while the FPROTR.FPROTCN bit is set to 1. After that, repeat the programming until it is finished.

Section 31 On-Chip RAM

This LSI has a high-speed RAM which is accessible by the CPU in a single cycle, and a RAM for retention which can retain data even in deep software standby mode, enabling the storage of instructions and data.

General enabling and enabling of writing are available for the high-speed on-chip RAM, so all memory operations or write operations can be enabled or disabled.

Retention or non-retention of data while the on-chip RAM for retention is in deep software standby mode is also selectable. Moreover, the on-chip RAM for retention is connected to a peripheral bus. For numbers of cycles for access to the on-chip RAM for retention, see section 10, Bus State Controller (BSC).

31.1 Features

- Memory mapping
  The on-chip RAM is allocated to the address apace as shown in tables 31.1 and 31.2.

Table 31.1 On-Chip RAM (High Speed) Address Space

Page	Address
Page 0 (16 Kbytes)	H'FFF80000 to H'FFF83FFF
Page 1 (8 Kbytes)	H'FFF84000 to H'FFF85FFF
Page 2 (4 Kbytes)	H'FFF86000 to H'FFF86FFF
Page 3 (4 Kbytes)	H'FFF87000 to H'FFF87FFF

Table 31.2 On-Chip RAM (for Retention) Address Space

Page	Address
Page 0 (12 Kbytes)	H'FFFD8000 to H'FFFDAFFF

- Ports

  Each page in the high-speed on-chip RAM has two independent ports, one each for reading and writing. The ports are connected to the internal bus (I bus), CPU instruction fetch bus (F bus), and CPU memory access bus (M bus); however, the F bus is only connected to the read port.

  The F and M buses are used for access from the CPU. The I bus is used for access from the DMAC or DTC.

  The on-chip RAM for retention has a single port for reading and writing, and this is connected to the peripheral bus.

- Priority

  Requests for simultaneous access to the same page of the high-speed on-chip RAM over multiple buses are processed in order of bus priority. The bus priority is as follows: I bus (highest), M bus (middle), F bus (lowest).

31.2 Notes on Usage

31.2.1 Page Conflict

Simultaneous access to the same page of the high-speed on-chip RAM over different buses leads to a page conflict. In this case, each access is completed normally but the conflict lowers the efficiency of memory access. To avoid such conflict, we recommend preventative measures in software. For example, ensuring that access over different buses is to different pages can avoid page conflict.

31.2.2 RAME and RAMWE Bits

When setting the RAME and RAMWE bits for the high-speed on-chip RAM to “disabled”, be sure to execute an instruction to read from or write to the same arbitrary address on each page before making the setting. If such an instruction is not executed, the last data to have been written to each page may not actually be written to the RAM.

```
//For page 0
  MOV.L   #H'FFF80000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 1
  MOV.L   #H'FFF84000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 2
  MOV.L   #H'FFF86000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0

//For page 3
  MOV.L   #H'FFF87000, R0
  MOV.L   @R0, R1
  MOV.L   R1, @R0
```

Figure 31.1 Example of Instruction Execution

31.2.3 Area Where Instruction Allocation is Prohibited

Do not place instructions in the 16 bytes from the last address (addresses H'FFFDAFF0 to H'FFFDAFFF) of the on-chip RAM for retention. Placing an instruction at an address in this range may lead to fetching by the CPU overrunning into the on-chip peripheral module space above address H'FFFDB000, thus generating an address error.

Section 32 Power-Down Modes

This LSI supports sleep mode, software standby mode, deep software standby mode, and module standby function. In power-down modes, functions of CPU, clocks, on-chip memory, or part of on-chip peripheral modules are halted or the internal power-supply is turned off, through which low power consumption is achieved. These power-down modes are canceled by a reset or interrupt.

32.1 Features

32.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

1. Sleep mode
2. Software standby mode
3. Deep software standby mode
4. Module standby function

Figure 32.1 shows a block diagram of the power-down modes.

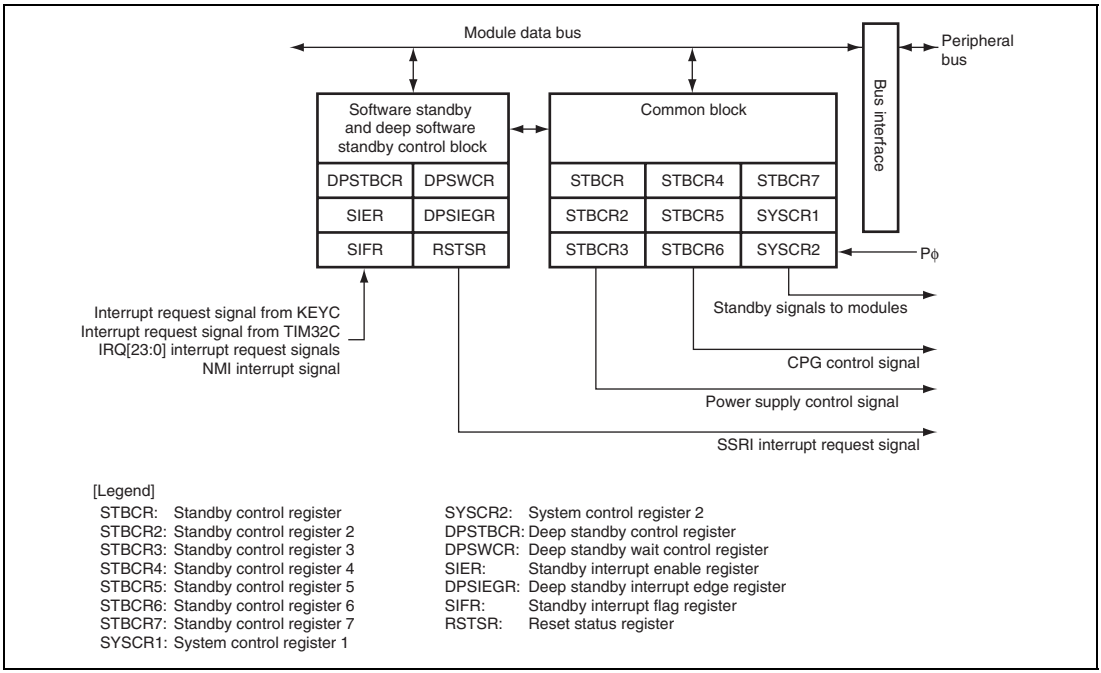


Figure 32.1 Power-Down Mode Block Diagram

Table 32.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module states in each mode and the procedures for canceling each mode.

Table 32.1 States of Power-Down Modes

		State									
Power-Down Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip RAM (High-Speed)**	**On-Chip RAM (for Data Retention)**	**On-Chip Peripheral Modules**	**KEYC, TIM32C**	**Power Supply**	**External Memory Refresh**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bit in STBCR cleared to 0	Running	Halted	Held	Running	Running	Running	Running	Running	Auto-refresh	• Interrupt • Manual reset • Power-on reset • DMA address error
Software standby mode	Execute SLEEP instruction with STBY bit in STBCR set to 1 and DPSTBY bit in DPSTBCR set to 0	Halted	Halted	Held	Halted (contents are held)	Halted (contents are held)	Halted	Running	Running	Self-refresh	• NMI interrupt • IRQ interrupt • KEYC/TIM32C interrupt • Manual reset • Power-on reset
Deep software standby mode	Execute SLEEP instruction with STBY bit in STBCR and DPSTBY bit in DPSTBCR set to 1	Halted	Halted	Un-defined	Halted (contents are not held)	Halted (contents are held※3)	Halted	Running	Halted	Self-refresh	• NMI interrupt※1 • IRQ interrupt※1※2 • KEYC/TIM32C interrupt※1 • Power-on reset
Module standby function	Set the MSTP bits in STBCR2 to STBCR7 to 1	Running	Running	Held	Held	Held	Specified module halted	Specified module halted	Running	Auto-refresh	• Clear MSTP bit to 0 • Power-on reset (only the modules of which the MSTP bit is 0 in the initial state.)

Notes: 1. When deep software standby mode is canceled by the interrupt, power-on reset exception handling is executed instead of interrupt exception handling. It can be checked by reading the DPSRSTF bit in RSTSR whether the power-on reset exception handling is provided by the usual power-on reset or by the deep software standby mode cancellation.

2. Deep software standby mode can be canceled by the interrupts IRQ0 to IRQ9.

32.2 Register Descriptions

The following registers are used in power-down modes. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 32.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	STBCR	R/W	H'00	H'FFFE0014	8
Standby control register 2	STBCR2	R/W	H'63	H'FFFE0018	8
Standby control register 3	STBCR3	R/W	H'68	H'FFFE0408	8
Standby control register 4	STBCR4	R/W	H'F7	H'FFFE040C	8
Standby control register 5	STBCR5	R/W	H'FF	H'FFFE0418	8
Standby control register 6	STBCR6	R/W	H'E0	H'FFFE041C	8
Standby control register 7	STBCR7	R/W	H'F0	H'FFFE0500	8
System control register 1	SYSCR1	R/W	H'FF	H'FFFE0402	8
System control register 2	SYSCR2	R/W	H'FF	H'FFFE0404	8
Deep standby control register	DPSTBCR	R/W	H'00	H'FFFEC510	8
Deep standby wait control register	DPSWCR	R/W	H'00	H'FFFEC512	8
Standby interrupt enable register	SIER	R/W	H'0000	H'FFFEC514	16
Standby interrupt flag register	SIER	R/W	H'0000	H'FFFEC516	16
Deep standby interrupt edge register	DPSIEGR	R/W	H'0000	H'FFFEC518	16
Reset status register	RSTSR	R/W	H'00	H'FFFEC51A	8

32.2.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of the power-down mode. STBCR is initialized to H'00 by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	STBY	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction puts chip into sleep mode. 1: Executing SLEEP instruction puts chip into software standby mode.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

32.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR2 is initialized to H'63 by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	MSTP27	MSTP26	MSTP25	MSTP24	-	-	MSTP21	-
Initial value:	0	1	1	0	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP27	0	R/W	Module Stop 27 When the MSTP27 bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI runs. 1: Clock supply to H-UDI halted.
6	MSTP26	1	R/W	Module Stop 26 When the MSTP26 bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC runs. 1: Clock supply to UBC halted.
5	MSTP25	1	R/W	Module Stop 25 When the MSTP25 bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC runs. 1: Clock supply to DMAC halted.

Bit	Bit Name	Initial Value	R/W	Description
4	MSTP24	0	R/W	Module Stop 24 When the MSTP24 bit is set to 1, the supply of the clock to the FPU is halted. After setting the MSTP24 bit to 1, the MSTP24 bit cannot be cleared by writing 0. This means that, after the supply of the clock to the FPU is halted by setting the MSTP24 bit to 1, the supply cannot be restarted by clearing the MSTP24 bit to 0. To restart the supply of the clock to the FPU after it was halted, reset the LSI by a power-on reset. 0: FPU runs. 1: Clock supply to FPU is halted.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MSTP21	1	R/W	Module Stop 21 When the MSTP21 bit is set to 1, the supply of the clock to the DTC is halted. 0: DTC runs. 1: Clock supply to DTC halted.
0	—	1	R	Reserved These bits are always read as 1. The write value should always be 1.

32.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR3 is initialized to H'68 by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	HIZ	MSTP 36	MSTP 35	-	MSTP 33	MSTP 32	MSTP 31	MSTP 30
Initial value:	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	HIZ	0	R/W	Port High Impedance Selects whether the state of specific output pin is retained or high impedance in software standby mode. As to which pins are controlled, see appendix A, Pin States. This bit must not be set while the TME bit in WTSCR of the WDT is 1. To set the output pin to high-impedance, set the HIZ bit to 1 only while the TME bit is 0. 0: The pin state is retained in software standby mode. 1: The pin is set to high-impedance in software standby mode.
6	MSTP36	1	R/W	Module Stop 36 When the MSTP36 bit is set to 1, the supply of the clock to the MTU2S is halted. 0: MTU2S runs. 1: Clock supply to MTU2S is halted.
5	MSTP35	1	R/W	Module Stop 35 When the MSTP35 bit is set to 1, the supply of the clock to the MTU2 is halted. 0: MTU2 runs. 1: Clock supply to MTU2 is halted.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the supply of the clock to the IIC3 is halted. 0: IIC3 runs. 1: Clock supply to IIC3 is halted.
2	MSTP32	0	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the supply of the clock to the internal RAM (high speed) is halted. 0: Internal RAM (high speed) runs. 1: Clock supply to internal RAM (high speed) is halted.
1	MSTP31	1	R/W	Module Stop 31 When the MSTP31 bit is set to 1, the supply of the clock to the internal RAM (for data retention) is halted. 0: Internal RAM (for data retention) runs. 1: Clock supply to internal RAM (for data retention) is halted.
0	MSTP30	0	R/W	Module Stop 30 When the MSTP30 bit is set to 1, the supply of the clock to the ROM and FLD is halted. 0: ROM and FLD run. 1: Clock supply to ROM and FLD is halted.

32.2.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR4 is initialized to H'F7 by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	-	MSTP42	MSTP41	-
Initial value:	1	1	1	1	0	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP47	1	R/W	Module Stop 47 When the MSTP47 bit is set to 1, the supply of the clock to the SCIF4 is halted. 0: SCIF4 runs. 1: Clock supply to SCIF4 is halted.
6	MSTP46	1	R/W	Module Stop 46 When the MSTP46 bit is set to 1, the supply of the clock to the SCIF5 is halted. 0: SCIF5 runs. 1: Clock supply to SCIF5 is halted.
5	MSTP45	1	R/W	Module Stop 45 When the MSTP45 bit is set to 1, the supply of the clock to the SCIF6 is halted. 0: SCIF6 runs. 1: Clock supply to SCIF6 is halted.
4	MSTP44	1	R/W	Module Stop 44 When the MSTP44 bit is set to 1, the supply of the clock to the SCIF7 is halted. 0: SCIF7 runs. 1: Clock supply to SCIF7 is halted.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP42	1	R/W	Module Stop 42 When the MSTP42 bit is set to 1, the supply of the clock to the CMT is halted. 0: CMT runs. 1: Clock supply to CMT is halted.
1	MSTP41	1	R/W	Module Stop 41 When the MSTP41 bit is set to 1, the supply of the clock to the CMT2 is halted. 0: CMT2 runs. 1: Clock supply to CMT2 is halted.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

32.2.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR5 is initialized to H'FF by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	MSTP51	MSTP50
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP57	1	R/W	Module Stop 57 When the MSTP57 bit is set to 1, the supply of the clock to the SCI0 is halted. 0: SCI0 runs. 1: Clock supply to SCI0 is halted.
6	MSTP56	1	R/W	Module Stop 56 When the MSTP56 bit is set to 1, the supply of the clock to the SCI1 is halted. 0: SCI1 runs. 1: Clock supply to SCI1 is halted.
5	MSTP55	1	R/W	Module Stop 55 When the MSTP55 bit is set to 1, the supply of the clock to the SCI2 is halted. 0: SCI2 runs. 1: Clock supply to SCI2 is halted.
4	MSTP54	1	R/W	Module Stop 54 When the MSTP54 bit is set to 1, the supply of the clock to the SCI3 is halted. 0: SCI3 runs. 1: Clock supply to SCI3 is halted.

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP53	1	R/W	Module Stop 53 When the MSTP53 bit is set to 1, the supply of the clock to the RSPI is halted. 0: RSPI runs. 1: Clock supply to RSPI is halted.
2	MSTP52	1	R/W	Module Stop 52 When the MSTP52 bit is set to 1, the supply of the clock to the ADC0 is halted. 0: ADC0 runs. 1: Clock supply to ADC0 is halted.
1	MSTP51	1	R/W	Module Stop 51 When the MSTP51 bit is set to 1, the supply of the clock to the ADC1 is halted. 0: ADC1 runs. 1: Clock supply to ADC1 is halted.
0	MSTP50	1	R/W	Module Stop 50 When the MSTP50 bit is set to 1, the supply of the clock to the RCAN-ET is halted. 0: RCAN-ET runs. 1: Clock supply to RCAN-ET is halted.

32.2.6 Standby Control Register 6 (STBCR6)

STBCR6 is an 8-bit readable/writable register that controls the operation of each module in power-down modes. STBCR6 is initialized to H'E0 by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	MSTP67	-	-	-	-	-	-	-
Initial value:	1	1	1	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP67	1	R/W	Module Stop 67 When the MSTP67 bit is set to 1, the supply of the clock to the LVDS is halted. 0: LVDS runs. 1: Clock supply to LVDS is halted. Note: This bit is only valid in the SH72315A. In the SH72315L/SH72314L, this bit is reserved and is always read as 1. The write value should always be 1.
6, 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

32.2.7 Standby Control Register 7 (STBCR7)

STBCR7 is an 8-bit readable/writable register that controls the operation of each module in power-down modes. STBCR7 is initialized to H'F0 by a power-on reset input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal used to return from deep software standby mode or by the internal reset at an overflow of the WDT. STBCR7 retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

Bit:	7	6	5	4	3	2	1	0
	MSTP77	MSTP76	MSTP75	MSTP74	-	-	-	-
Initial value:	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP77	1	R/W	Module Stop 77 When the MSTP77 bit is set to 1, the supply of the clock to the TIM32C is halted. 0: TIM32C runs. 1: Clock supply to TIM32C is halted.
6	MSTP76	1	R/W	Module Stop 76 When the MSTP76 bit is set to 1, the supply of the clock to the KEYC is halted. 0: KEYC runs. 1: Clock supply to KEYC is halted.
5	MSTP75	1	R/W	Module Stop 75 The MSTP75 bit controls the TIM32C/KEYC clock in combination with the MSTP74 bit. For details, see the description of the MSTP74 bit.

Bit	Bit Name	Initial Value	R/W	Description
4	MSTP74	1	R/W	Module Stop 74 The MSTP74 bit controls the TIM32C/KEYC clock in combination with the MSTP75 bit to reduce unnecessary power consumption. The MSTP75 bit stops the control block of the KEYC/TIM32C clock and the MSTP74 bit stops the crystal oscillator of the KEYC/TIM32C clock. Set these bits according to the type of the clock connected to EXTAL32/XTAL32, as follows: 00: A crystal oscillator is connected to EXTAL32/XTAL32. 01: An external clock is input to EXTAL32. (EXTAL32 is connected to the external clock and XTAL32 is left open.) 10: Setting prohibited 11: The KEYC/TIM32C clock is not used. (EXTAL32 is connected to Vss and XTAL32 is left open.)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

32.2.8 System Control Register 1 (SYSCR1)

SYSCR1 is an 8-bit readable/writable register that enables or disables access to the on-chip RAM (high-speed). SYSCR1 is initialized to H'FF by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

When an RAME bit is set to 1, the corresponding on-chip RAM (high-speed) area is enabled. When an RAME bit is cleared to 0, the corresponding on-chip RAM (high-speed) area cannot be accessed. In this case, an undefined value is returned when reading data or fetching an instruction from the on-chip RAM (high-speed), and writing to the on-chip RAM (high-speed) is ignored. The initial value of an RAME bit is 1.

Note that when clearing the RAME bit to 0 to disable the on-chip RAM (high-speed), be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAME bit. If such an instruction is not executed, the data last written to each page may not be written to the on-chip RAM (high-speed). Furthermore, an instruction to access the on-chip RAM (high-speed) should not be located immediately after the instruction to write to SYSCR1. If an on-chip RAM (high-speed) access instruction is set, normal access is not guaranteed.

When setting the RAME bit to 1 to enable the on-chip RAM (high-speed), an instruction to read SYSCR1 should be located immediately after the instruction to write to SYSCR1. If an instruction to access the on-chip RAM (high-speed) is located immediately after the instruction to write to SYSCR1, normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAME3	RAME2	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	RAME3	1	R/W	RAM Enable 3 (corresponding RAM area: H'FFF87000 to H'FFF87FFF (page 3)) 0: Access to on-chip RAM (high-speed) disabled 1: Access to on-chip RAM (high-speed) enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RAME2	1	R/W	RAM Enable 2 (corresponding RAM area: H'FFF86000 to H'FFF86FFF (page 2)) 0: Access to on-chip RAM (high-speed) disabled 1: Access to on-chip RAM (high-speed) enabled
1	RAME1	1	R/W	RAM Enable 1 (corresponding RAM area: H'FFF84000 to H'FFF85FFF (page 1)) 0: Access to on-chip RAM (high-speed) disabled 1: Access to on-chip RAM (high-speed) enabled
0	RAME0	1	R/W	RAM Enable 0 (corresponding RAM area: H'FFF80000 to H'FFF83FFF (page 0)) 0: Access to on-chip RAM (high-speed) disabled 1: Access to on-chip RAM (high-speed) enabled

32.2.9 System Control Register 2 (SYSCR2)

SYSCR2 is an 8-bit readable/writable register that enables or disables write to the on-chip RAM (high-speed). SYSCR2 is initialized to H'FF by a power-on reset, but it retains the previous value on a manual reset or in software standby mode. Only byte access is valid.

When an RAMWE bit is set to 1, the corresponding on-chip RAM (high-speed) area is enabled. When an RAMWE bit is cleared to 0, the corresponding on-chip RAM (high-speed) area cannot be written to. In this case, writing to the on-chip RAM (high-speed) is ignored. The initial value of an RAMWE bit is 1.

Note that when clearing the RAME bit to 0 to disable the on-chip RAM (high-speed), be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAMWE bit. If such an instruction is not executed, the data last written to each page may not be written to the on-chip RAM (high-speed). Furthermore, an instruction to access the on-chip RAM (high-speed) should not be located immediately after the instruction to write to SYSCR2. If an on-chip RAM (high-speed) access instruction is set, normal access is not guaranteed.

When setting the RAME bit to 1 to enable write to the on-chip RAM (high-speed), an instruction to read SYSCR2 should be located immediately after the instruction to write to SYSCR2. If an instruction to access the on-chip RAM (high-speed) is located immediately after the instruction to write to SYSCR2, normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAM WE3	RAM WE2	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	RAMWE3	1	R/W	RAM Write Enable 3 (corresponding RAM area: H'FFF87000 to H'FFF87FFF (page 3)) 0: Write to on-chip RAM (high-speed) disabled 1: Write to on-chip RAM (high-speed) enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RAMWE2	1	R/W	RAM Write Enable 2 (corresponding RAM area: H'FFF86000 to H'FFF86FFF (page 2)) 0: Write to on-chip RAM (high-speed) disabled 1: Write to on-chip RAM (high-speed) enabled
1	RAMWE1	1	R/W	RAM Write Enable 1 (corresponding RAM area: H'FFF84000 to H'FFF85FFF (page 1)) 0: Write to on-chip RAM (high-speed) disabled 1: Write to on-chip RAM (high-speed) enabled
0	RAMWE0	1	R/W	RAM Write Enable 0 (corresponding RAM area: H'FFF80000 to H'FFF83FFF (page 0)) 0: Write to on-chip RAM (high-speed) disabled 1: Write to on-chip RAM (high-speed) enabled

32.2.10 Deep Standby Control Register (DPSTBCR)

DPSTBCR is an 8-bit readable/writable register that controls deep software standby mode.

DPSTBCR is initialized by a power-on reset signal input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal at a restoration from deep software standby mode or by the internal reset at an overflow of the WDT.

Bit:	7	6	5	4	3	2	1	0
	DPSTBY	-	-	-	-	RAMCUT2	RAMCUT1	RAMCUT0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	DPSTBY	0	R/W	Deep Software Standby Executing the SLEEP instruction when the STBY bit in STBCR is 1 causes a transition to software standby mode. At this time, if this bit is 1 and there is no software standby mode cancellation source, a transition is made to deep software standby mode. STBY DPSTBY 0 x : A transition is made to sleep mode after SLEEP instruction execution. 1 0 : A transition is made to software standby mode after SLEEP instruction execution. 1 1 : A transition is made to deep software standby mode after SLEEP instruction execution. When deep software standby mode is canceled by the interrupt, this bit remains 1. To clear this bit, write 0 to the bit.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	RAMCUT2	0	R/W	On-Chip RAM Off 2 Controls the internal power supply to the on-chip RAM (for data retention) during deep software standby mode. For details, see the description of the RAMCUT0 bit.
1	RAMCUT1	0	R/W	On-Chip RAM Off 1 Controls the internal power supply to the on-chip RAM (for data retention) during deep software standby mode. For details, see the description of the RAMCUT0 bit.
0	RAMCUT0	0	R/W	On-Chip RAM Off 0 Controls the internal power supply to the on-chip RAM (for data retention) during deep software standby mode in combination with RAMCUT2 and RAMCUT1. RAMCUT2 to 0 000: The power is supplied to the on-chip RAM (for data retention) during deep software standby mode. (Data in the on-chip RAM (for data retention) is retained.) 111: The power is not supplied to the on-chip RAM (for data retention) during deep software standby mode. (Data in the on-chip RAM (for data retention) is not retained.) Other than above: Setting prohibited

32.2.11 Deep Standby Wait Control Register (DPSWCR)

DPSWCR is an 8-bit readable/writable register that selects the oscillation settling time (internal reset time) when returning from deep standby mode.

DPSWCR is initialized by a power-on reset signal input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal used to return from deep software standby mode or by the internal reset at an overflow of the WDT.

Bit:	7	6	5	4	3	2	1	0
	WTSEL	WTSTS[6:0]						
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	WTSEL	0	R/W	Wait Control Select Selects the method for determining the oscillation settling time period. When this bit is 0, the WTSTS[6:0] bits are used as the oscillation settling time setting register. When this bit is 1, the WTSTS[6:0] bits are used as the compare register to the oscillation settling time counter. 0: Deep software standby mode is canceled when the time specified by WTSTS[6:0] has elapsed. 1: Deep software standby mode is canceled when the value specified by WTSTS[6:0] matches the counter value.

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	WTSTS[6:0]	All 0	R/W	Deep Software Standby Wait Time Setting These bits select the oscillation settling time when the WTSEL bit is 0. The time period is determined by counting the EXTAL input clock pulses. The values in parentheses below indicate the time period when the EXTAL input clock frequency is 12.5 MHz. 0000101: 64 states (5.12 μs) 0000110: 512 states (41 μs) 0000111: 1024 states (81.9 μs) 0001000: 2048 states (164 μs) 0001001: 4096 states (328 μs) 0001010: 16348 states (1.31 ms) 0001011: 32768 states (2.62 ms) 0001100: 65536 states (5.24 ms) 0001101: 131072 states (10.5 ms) 0001110: 262144 states (21 ms) 0001111: 524288 states (41.9 mss) Other than above: Setting prohibited When WTSEL = 1, deep software standby mode is canceled when the oscillation settling time counter value matches the WTSTS[6:0] setting.

32.2.12 Standby Interrupt Enable Register (SIER)

SIER is a 16-bit readable/writable register that enables or disables the interrupt to cancel software standby mode or deep software standby mode.

SIER is initialized by a power-on reset signal input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal used to return from deep software standby mode or by the internal reset at an overflow of the WDT.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	DKEYCE	DTIM32CE	-	DIRQ9E	DIRQ8E	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	DKEYCE	0	R/W	KEYC Interrupt Enable Enables or disables cancellation of deep software standby mode when the KSF bits in KSSR are set to 1. 0: Cancellation of deep software standby mode by the KEYC interrupt is disabled. 1: Cancellation of deep software standby mode by the KEYC interrupt is enabled.
11	DTIM32CE	0	R/W	TIM32C Interrupt Enable Enables or disables cancellation of deep software standby mode when any of the CH2F, CH1UF, CH1DF, CH0UF, and CH0DF bits in TI32SR is set to 1. 0: Cancellation of deep software standby mode by the TIM32C interrupt is disabled. 1: Cancellation of deep software standby mode by the TIM32C interrupt is enabled.
10	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	DIRQ9E	0	R/W	IRQ9 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ9EG bit in DPSIEGR is input to the IRQ9 (PG9/IRQ9 pin). 0: Cancellation of deep software standby mode by the IRQ9 is disabled. 1: Cancellation of deep software standby mode by the IRQ9 is enabled.
8	DIRQ8E	0	R/W	IRQ8 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ8EG bit in DPSIEGR is input to the IRQ8 (PG8/IRQ8 pin). 0: Cancellation of deep software standby mode by the IRQ8 is disabled. 1: Cancellation of deep software standby mode by the IRQ8 is enabled.
7	DIRQ7E	0	R/W	IRQ7 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ7EG bit in DPSIEGR is input to the IRQ7 (PG7/IRQ7 pin). 0: Cancellation of deep software standby mode by the IRQ7 is disabled. 1: Cancellation of deep software standby mode by the IRQ7 is enabled.
6	DIRQ6E	0	R/W	IRQ6 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ6EG bit in DPSIEGR is input to the IRQ6 (PG6/IRQ6 pin). 0: Cancellation of deep software standby mode by the IRQ6 is disabled. 1: Cancellation of deep software standby mode by the IRQ6 is enabled.

Bit	Bit Name	Initial Value	R/W	Description
5	DIRQ5E	0	R/W	IRQ5 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ5EG bit in DPSIEGR is input to the IRQ5 (PG5/IRQ5 pin). 0: Cancellation of deep software standby mode by the IRQ5 is disabled. 1: Cancellation of deep software standby mode by the IRQ5 is enabled.
4	DIRQ4E	0	R/W	IRQ4 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ4EG bit in DPSIEGR is input to the IRQ4 (PG4/IRQ4 pin). 0: Cancellation of deep software standby mode by the IRQ4 is disabled. 1: Cancellation of deep software standby mode by the IRQ4 is enabled.
3	DIRQ3E	0	R/W	IRQ3 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ3EG bit in DPSIEGR is input to the IRQ3 (PG3/IRQ3 pin). 0: Cancellation of deep software standby mode by the IRQ3 is disabled. 1: Cancellation of deep software standby mode by the IRQ3 is enabled.
2	DIRQ2E	0	R/W	IRQ2 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ2EG bit in DPSIEGR is input to the IRQ2 (PG2/IRQ2 pin). 0: Cancellation of deep software standby mode by the IRQ2 is disabled. 1: Cancellation of deep software standby mode by the IRQ2 is enabled.

Bit	Bit Name	Initial Value	R/W	Description
1	DIRQ1E	0	R/W	IRQ1 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ1EG bit in DPSIEGR is input to the IRQ1 (PG1/IRQ1 pin). 0: Cancellation of deep software standby mode by the IRQ1 is disabled. 1: Cancellation of deep software standby mode by the IRQ1 is enabled.
0	DIRQ0E	0	R/W	IRQ0 Interrupt Enable Enables or disables cancellation of deep software standby mode when the edge selected by the DIRQ0EG bit in DPSIEGR is input to the IRQ0 (PG0/IRQ0 pin). 0: Cancellation of deep software standby mode by the IRQ0 is disabled. 1: Cancellation of deep software standby mode by the IRQ0 is enabled.

32.2.13 Standby Interrupt Flag Register (SIFR)

SIFR is a 16-bit readable/writable register that indicates the interrupt which has canceled software standby mode or deep software standby mode. When the interrupt edge specified in DPSIEGR is generated, the corresponding bit in this register is set to 1. Note that the bits are set when interrupts are generated even though the system is not in software standby mode or deep software standby mode. Therefore, clear the register bits immediately before a transition to software standby or deep software standby mode.

SIFR is initialized by a power-on reset signal input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal used to return from deep software standby mode or by the internal reset at an overflow of the WDT.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNMIF	-	-	-	SSRF	-	DIRQ 9F	DIRQ 8F	DIRQ 7F	DIRQ 6F	DIRQ 5F	DIRQ 4F	DIRQ 3F	DIRQ 2F	DIRQ 1F	DIRQ 0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

Bit	Bit Name	Initial Value	R/W	Description
15	DNMIF	0	R/(W)*	NMI Interrupt Flag Indicates whether the NMI input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] NMI input specified by DPSIEGR is generated.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11	SSRF	0	R/(W)*	SSR Interrupt Flag Indicates whether the system has been returned from software standby mode by the TIM32C/KEYC interrupt or not. 0: Return from software standby mode has not been generated. [Clearing condition] Writing 0 to SSRF after having read SSRF = 1 within the processing routine for the low power consumption interrupt. 1: Return from software standby mode has been generated. [Setting condition] A software standby return request generated by the TIM32C/KEYC interrupt is detected in software standby mode.
10	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
9	DIRQ9F	0	R/(W)*	IRQ9 Interrupt Flag Indicates whether the IRQ9 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ9 interrupt specified in DPSIEGR is generated.
8	DIRQ8F	0	R/(W)*	IRQ8 Interrupt Flag Indicates whether the IRQ8 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ8 interrupt specified in DPSIEGR is generated.

Bit	Bit Name	Initial Value	R/W	Description
7	DIRQ7F	0	R/(W)*	IRQ7 Interrupt Flag Indicates whether the IRQ7 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ7 interrupt specified in DPSIEGR is generated.
6	DIRQ6F	0	R/(W)*	IRQ6 Interrupt Flag Indicates whether the IRQ6 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ6 interrupt specified in DPSIEGR is generated.
5	DIRQ5F	0	R/(W)*	IRQ5 Interrupt Flag Indicates whether the IRQ5 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ5 interrupt specified in DPSIEGR is generated.
4	DIRQ4F	0	R/(W)*	IRQ4 Interrupt Flag Indicates whether the IRQ4 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ4 interrupt specified in DPSIEGR is generated.

Bit	Bit Name	Initial Value	R/W	Description
3	DIRQ3F	0	R/(W)*	IRQ3 Interrupt Flag Indicates whether the IRQ3 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ3 interrupt specified in DPSIEGR is generated.
2	DIRQ2F	0	R/(W)*	IRQ2 Interrupt Flag Indicates whether the IRQ2 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ2 interrupt specified in DPSIEGR is generated.
1	DIRQ1F	0	R/(W)*	IRQ1 Interrupt Flag Indicates whether the IRQ1 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ1 interrupt specified in DPSIEGR is generated.
0	DIRQ0F	0	R/(W)*	IRQ0 Interrupt Flag Indicates whether the IRQ0 input specified in DPSIEGR has been generated or not. [Clearing condition] Writing 0 after reading 1 [Setting condition] IRQ0 interrupt specified in DPSIEGR is generated.

Note: * Only 0 can be written to clear the flag.

32.2.14 Deep Standby Interrupt Edge Register (DPSIEGR)

DPSIEGR is a 16-bit readable/writable register that selects the edges for the interrupt input pins to cancel deep software standby mode.

DPSIEGR is initialized by a power-on reset signal input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal used to return from deep software standby mode or by the internal reset at an overflow of the WDT.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNMIEG	-	-	-	-	-	DIRQ9EG	DIRQ8EG	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	DNMIEG	0	R/W	NMI Interrupt Edge Selects the edge of the NMI input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	DIRQ9EG	0	R/W	IRQ9 Interrupt Edge Selects the edge of the IRQ9 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
8	DIRQ8EG	0	R/W	IRQ8 Interrupt Edge Selects the edge of the IRQ8 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
7	DIRQ7EG	0	R/W	IRQ7 Interrupt Edge Selects the edge of the IRQ7 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.

Bit	Bit Name	Initial Value	R/W	Description
6	DIRQ6EG	0	R/W	IRQ6 Interrupt Edge Selects the edge of the IRQ6 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
5	DIRQ5EG	0	R/W	IRQ5 Interrupt Edge Selects the edge of the IRQ5 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
4	DIRQ4EG	0	R/W	IRQ4 Interrupt Edge Selects the edge of the IRQ4 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
3	DIRQ3EG	0	R/W	IRQ3 Interrupt Edge Selects the edge of the IRQ3 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
2	DIRQ2EG	0	R/W	IRQ2 Interrupt Edge Selects the edge of the IRQ2 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
1	DIRQ1EG	0	R/W	IRQ1 Interrupt Edge Selects the edge of the IRQ1 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.
0	DIRQ0EG	0	R/W	IRQ0 Interrupt Edge Selects the edge of the IRQ0 input pin. 0: An interrupt request generated at the falling edge. 1: An interrupt request generated at the rising edge.

32.2.15 Reset Status Register (RSTSR)

RSTSR is an 8-bit readable/writable register that indicates that deep software standby mode has been canceled by an interrupt.

RSTSR is initialized by a power-on reset signal input from the $\overline{\text{RES}}$ pin but is not initialized by the internal reset signal used to return from deep software standby mode or by the internal reset at an overflow of the WDT.

Bit:	7	6	5	4	3	2	1	0
	DPSRSTF	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
7	DPSRSTF	0	R/(W)*	Deep Software Standby Reset Flag Indicates that deep software standby mode has been canceled by the interrupt specified in SIER and DPSIEGR and that the internal reset has been generated. [Clearing condition] Writing 0 after reading 1 [Setting condition] Deep software standby mode is canceled by the specified interrupt.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Only 0 can be written to clear the flag.

32.3 Operation

32.3.1 Sleep Mode

(1) Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run and the clock signal continues to be output to the CK pin.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, and on-chip peripheral module), a DMA/DTC address error, or a reset (manual reset or power-on reset).

- Canceling by an interrupt
  When an NMI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. When the priority level of the generated interrupt is equal to or lower than the interrupt mask level that is set in the status register (SR) of the CPU, or the interrupt by the on-chip peripheral module is disabled on the module side, the interrupt request is not accepted and sleep mode is not canceled.
- Canceling by a DMA/DTC address error
  When a DMA/DTC address error occurs, sleep mode is canceled and DMA/DTC address error exception handling is executed.
- Canceling by a reset
  Sleep mode is canceled by a power-on reset or a manual reset.

(3) Notes on Transition to Sleep Mode

If a sleep mode cancellation interrupt is input during a transition to sleep mode, the following operation is performed according to the timing of the interrupt acceptance.

- Interrupt exception handing is executed and then a transition is made to sleep mode.
- A transition is made to sleep mode, sleep mode is canceled by the interrupt, and then the interrupt exception handling is executed.
- A transition is not made to sleep mode and the interrupt exception handling is executed (a transition to sleep mode is canceled).

32.3.2 Software Standby Mode

(1) Transition to Software Standby Mode

The LSI switches from a program execution state to software standby mode by executing the SLEEP instruction with the STBY bit in STBCR being 1 and DPSTBY bit in DPSTBCR being 0. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules other than the KEYC and TIM32C halt. The clock output from the CK pin also stops. The KEYC and TIM32C operation can be performed.

The contents of the CPU registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. As for the states of on-chip peripheral module registers in software standby mode, see section 34.3, Register States in Each Operating Mode. For the states of the pins in software standby mode, see appendix A, Pin States.

The CPU takes one cycle to finish writing to STBCR or DPSTBCR, and then executes processing for the next instruction. However, it takes one or more cycles to actually write. Therefore, execute a SLEEP instruction after reading STBCR or DPSTBCR to have the values written to STBCR or DPSTBCR by the CPU to be definitely reflected in the SLEEP instruction.

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT.
2. Select the count clock type with the CKS[2:0] bits in WTCSR to secure appropriate oscillation settling time and set a value in WTCNT to start counting.
3. After setting the STBY in STBCR to 1 and the DPSTBY bit in DPSTBCR to 0, read STBCR and DPSTBCR.
4. Clear the flags in SIFR, KSSR, and TI32SR and the DPSRSTF bit in RSTSR, and read SIFR and RSTSR. Then, execute the SLEEP instruction.

Figure 32.2 shows the software standby mode setting flow.

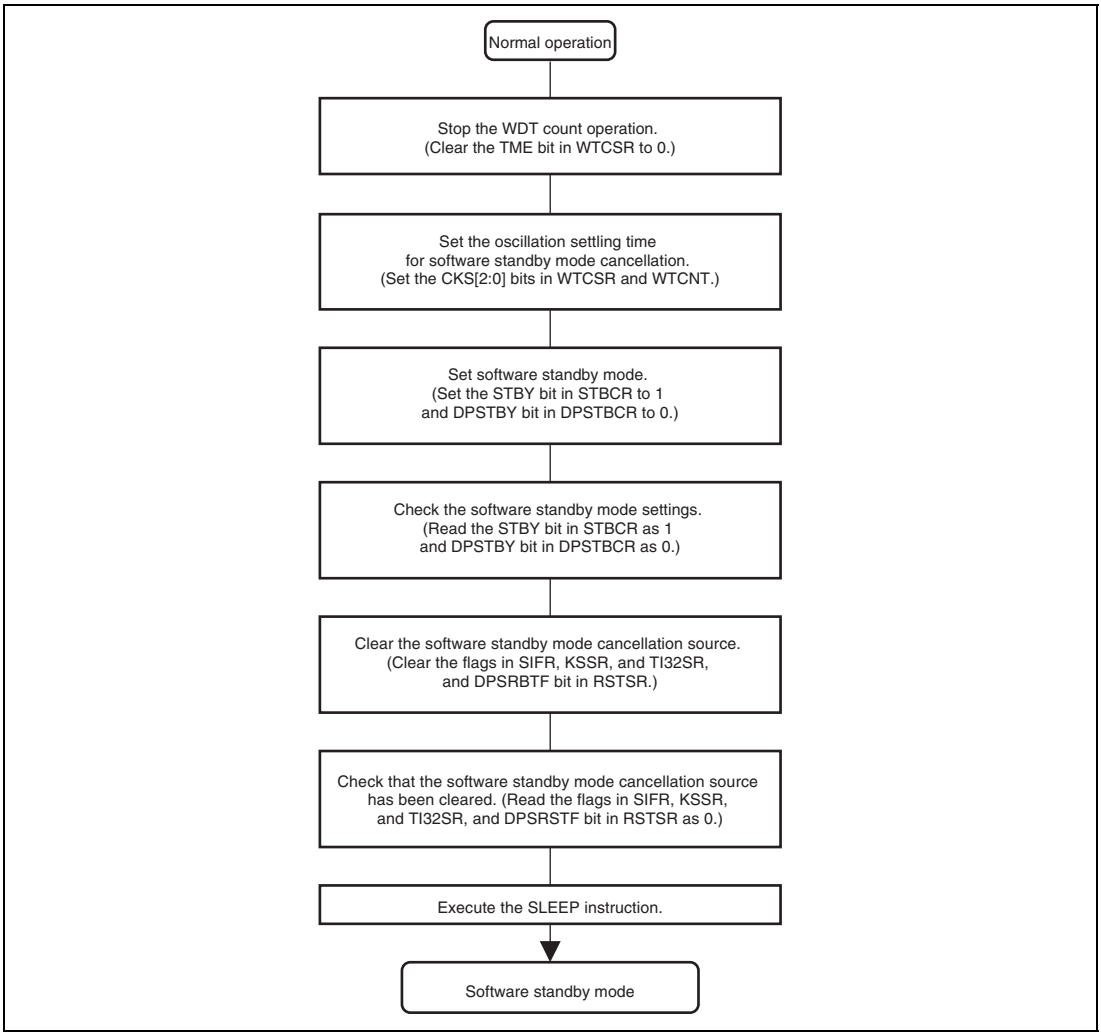


Figure 32.2 Software Standby Mode Setting Flow

(2) Canceling Software Standby Mode

Software standby mode is canceled by the NMI or IRQ interrupt, an interrupt from the KEYC or TIM32C, or a reset (manual reset or power-on reset).

- Canceling by an interrupt

  When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) of the interrupt controller (INTC)), the falling edge or rising edge of an IRQ pin (IRQ23 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)), or an interrupt from the key scan controller (KEYC) or 32-kHz timer (TIM32C) is detected, clock oscillation is started. This clock pulse is supplied only to the oscillation settling counter (WDT) used to count the oscillation settling time.

  After the elapse of the time set in the clock select bits (CKS[2:0]) in the watchdog timer control/status register (WTCSR) of the WDT and the timer counter (WTCNT) before the transition to software standby mode, the WDT overflow occurs. Since this overflow indicates that the clock has been stabilized, the clock pulse will be supplied to the entire chip after this overflow. Software standby mode is thus cleared and interrupt exception handling is started.

  Since all the TIM32C/KEYC interrupt sources of the return from software standby are handled collectively as one SSRI interrupt, check the pertinent flag to determine the specific software standby cancellation source. For details of the flags, see sections 27.3.2, Timer32 Status Register (TI32SR), 28.3.5, Key Scan Status Register (KSSR), and 32.2.13, Standby Interrupt Flag Register (SIFR).

  When canceling software standby mode by the interrupt, set the CKS[2:0] bits and WTCNT so that the WDT overflow period will be equal to or longer than the oscillation settling time.

  The clock output phase of the CKIO pin may be unstable immediately after detecting an interrupt and until software standby mode is canceled.

- Canceling by a reset

  When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin is driven low, software standby mode is canceled and the LSI enters the power-on reset or manual reset state.

  Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation settles.

(3) Notes on Transition to Software Standby Mode

If a software standby mode cancellation source is input during a transition to software standby mode, the following operation is performed according to the timing of the interrupt acceptance.

- Interrupt exception handing is executed and then a transition is made to software standby mode.
- A transition is made to software standby mode, software standby mode is canceled by the interrupt, and then the interrupt exception handling is executed.
- A transition is not made to software standby mode and the interrupt exception handling is executed (a transition to software standby mode is canceled).
- If an interrupt of which mask level is equal to or higher than the level set in the CPU status register is suspended or is newly generated, a transition is not made to software standby mode and interrupt exception handling is executed (a transition to software standby mode is canceled).

32.3.3 Software Standby Mode Application Example

This example describes a transition to software standby mode on the falling edge of the NMI signal, and cancellation on the rising edge of the NMI signal. The timing is shown in figure 32.3.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in the interrupt control register (ICR) is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is 1 (rising edge detection) by the NMI exception service routine and the SLEEP instruction is executed with the STBY bit in STBCR being 1 and DPSTBY bit in DPSTBCR being 0, software standby mode is entered. Thereafter, software standby mode is canceled when the NMI pin is changed from low to high level.

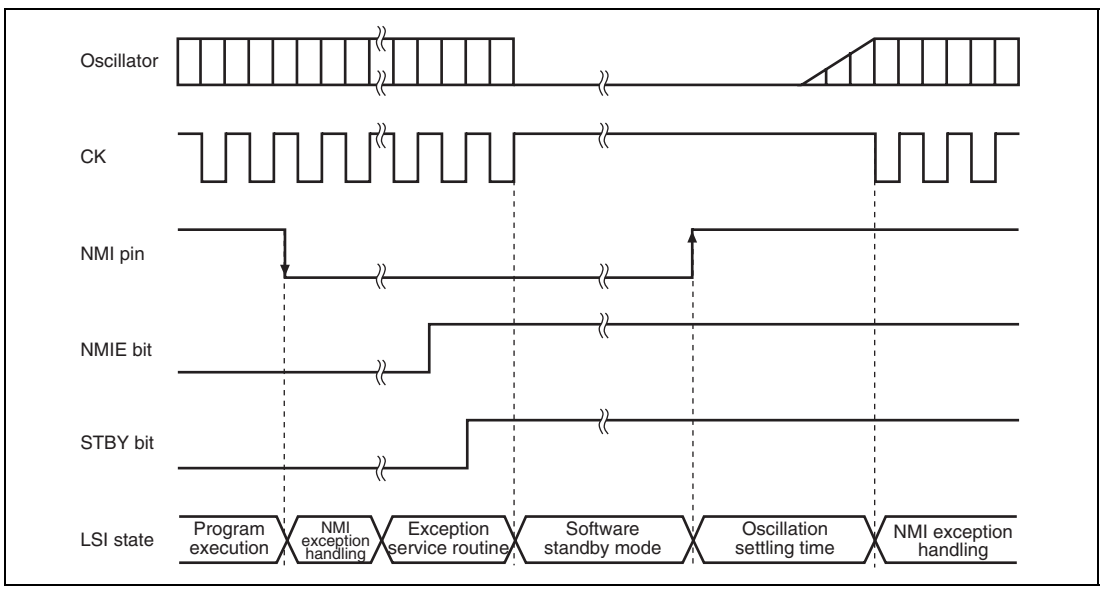


Figure 32.3 NMI Timing in Software Standby Mode (Application Example)

32.3.4 Deep Software Standby Mode

(1) Transition to Deep Software Standby Mode

The LSI enters software standby mode when the SLEEP instruction is executed with the STBY bit in STBCR being 1. At this time, if the DPSTBY bit in DPSTBCR is 1, the LSI enters deep software standby mode.

In deep software standby mode, the CPU, on-chip peripheral modules other than KEYC and TIM32C, on-chip RAM (high-speed), and oscillators are stopped and the internal power supply to these modules is turned off. This can significantly reduce power consumption. In deep software standby mode, data in the registers of the CPU, on-chip peripheral modules other than KEYC and TIM32C, and on-chip RAM is undefined.

Data in the on-chip RAM (for data retention) can be retained in deep software standby mode when all the RAMCUT2 to RAMCUT0 bits in DPSTBCR are set to 0. When all of the RAMCUT2 to RAMCUT0 are set to 1, the internal power supply to the on-chip RAM (for data retention) is also stopped, which can additionally reduce the power consumption. At this time, data in the on-chip RAM (for data retention) is undefined.

The KEYC and TIM32C operation can be performed in deep software standby mode.

As for the states of on-chip peripheral module registers in deep software standby mode, see section 34.3, Register States in Each Operating Mode. For the states of the pins in software standby mode, see appendix A, Pin States.

The procedure for switching to deep software standby mode is as follows. Figure 32.4 also shows its flowchart.

1. To retain data in the on-chip RAM (for data retention) in deep software standby mode, clear the MSTP31 bit in STBCR3 (cancel the module standby state of the on-chip RAM (for data retention)) and transfer data to be retained to the on-chip RAM area (for data retention). After the data transfer, execute read and write of an arbitrary but the same address in the on-chip RAM (for data retention). When this is not executed, data last written may not be written to the on-chip RAM (for data retention). If there is a write to the on-chip RAM (for data retention) after this time, execute this processing after the last write to the on-chip RAM (for data retention).
2. To retain data in the on-chip RAM (for data retention) in deep software standby mode, set the RAMCUT2 to RAMCUT0 bits in DPSTBCR to 000. Not to retain data in the on-chip RAM (for data retention) for reducing power consumption during deep software standby mode, set the RAMCUT2 to RAMCUT0 bits in DPSTBCR to 111.

3. Set the oscillation settling time (internal reset time) for return from deep software standby mode with the WTSEL and WTSTS[6:0] bits in DPSWCR.
4. To cancel deep standby mode by an interrupt, set the corresponding bit in SIER and DPSIEGR.
5. After setting the STBY bit in STBCR and DPSTBY bit in DPSTBCR to 1, read STBCR and DPSTBCR.
6. Clear the flags in SIFR, KSSR, and TI32SR and the DPSRSTF bit in RSTSR, and read SIFR and RSTSR. Then, execute the SLEEP instruction.

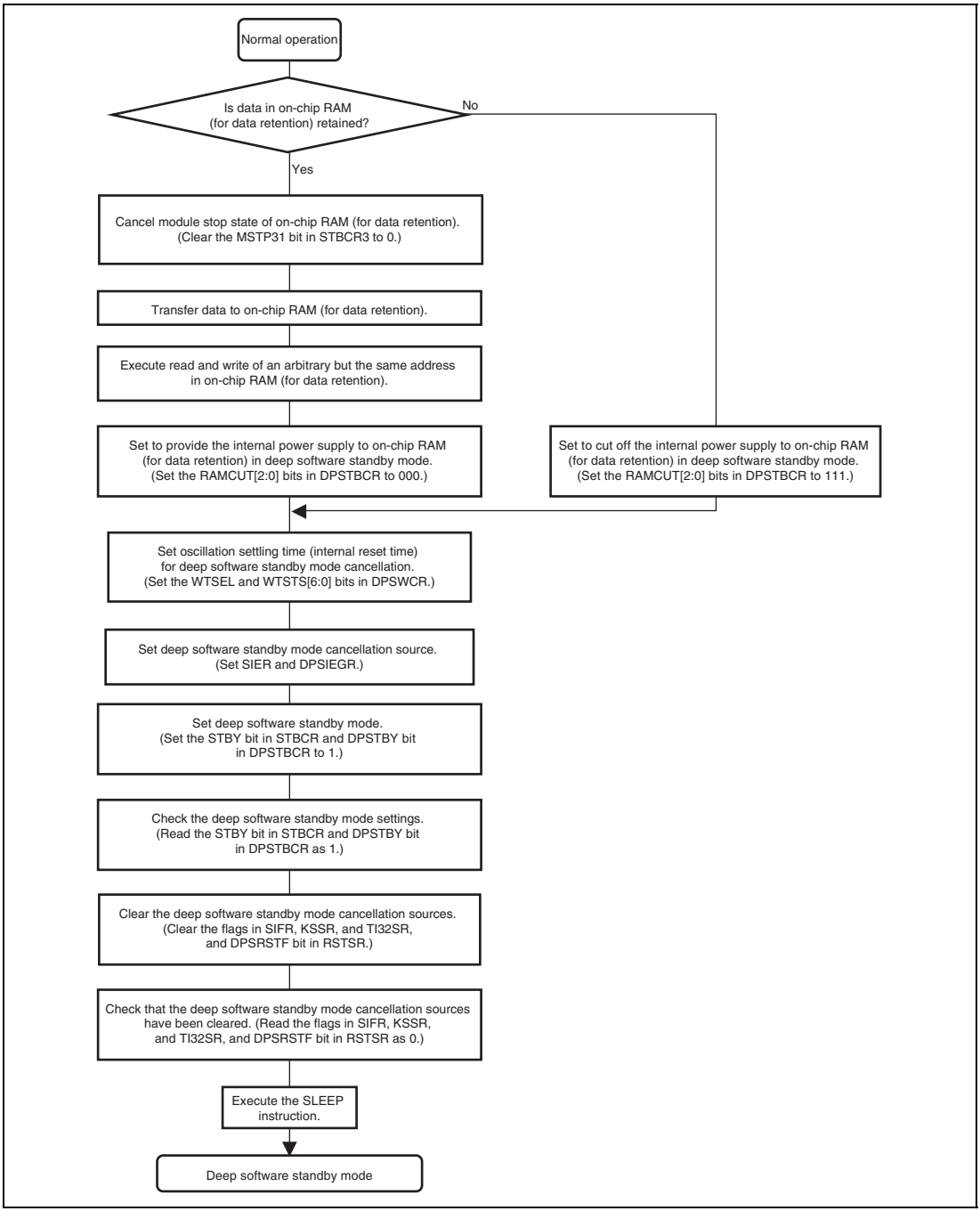


Figure 32.4 Flowchart of Transition to Deep Software Standby Mode

(2) Canceling Deep Software Standby Mode

Deep software standby mode is canceled by interrupts from external interrupt pins (NMI and IRQ0 to IRQ9), KEYC, and TIM32C, and by a reset from the $\overline{\text{RES}}$ pin.

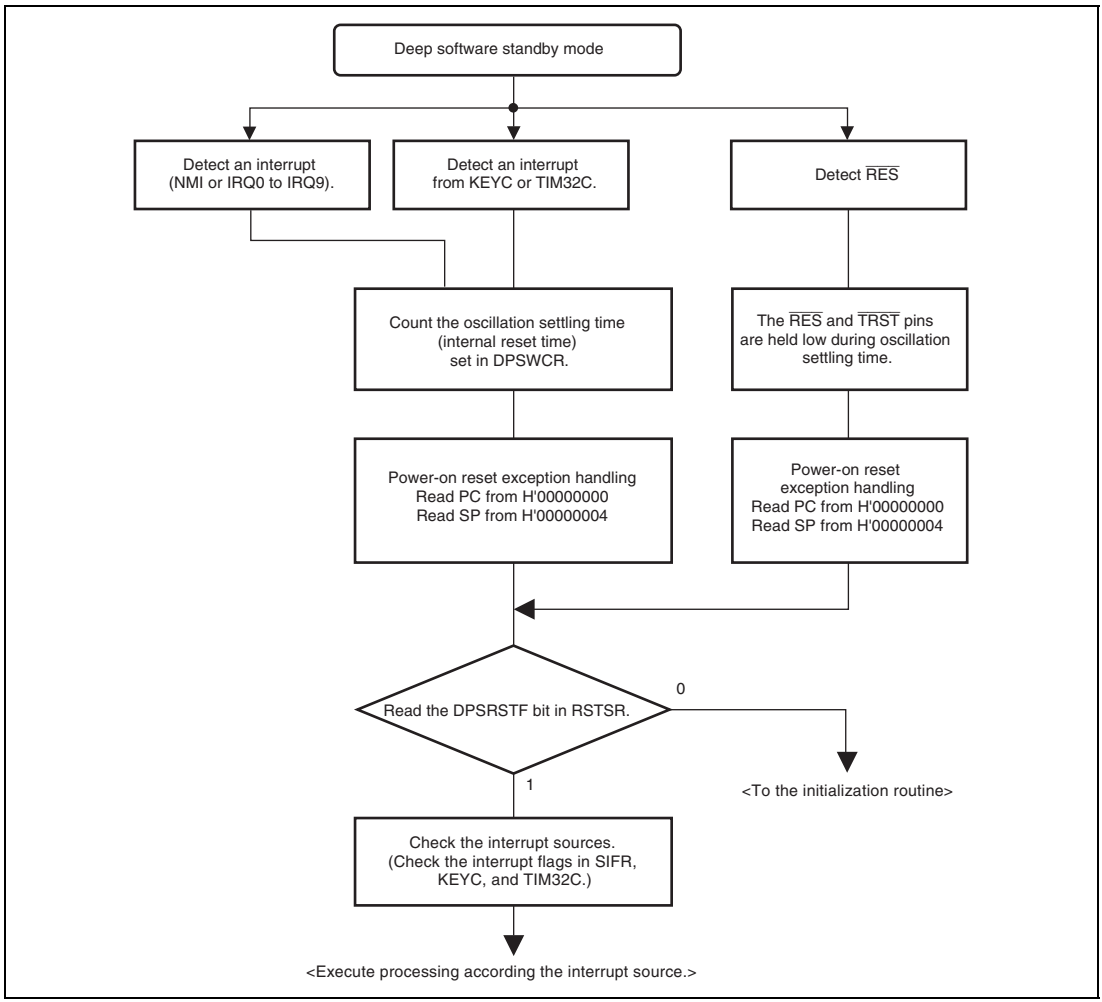


Figure 32.5 Flowchart of Canceling Deep Software Standby Mode

- Canceling by an interrupt

  Deep software standby mode is canceled when any of the DNMIF and DIRQnF (n = 0 to 9) bits in SIFR, the KSF bit in KSSR, and the CH2F, CH1UF, CH1DF, CH0UF, and CH0DF bits in TI32SR is set to 1.

  When a deep software standby mode cancellation source is generated, the clock oscillation is started, internal power supply is provided, and internal reset signal is generated to the entire LSI. After the oscillation settling time specified by DPSWCR has elapsed, stabilized clock pulses are supplied to the entire LSI and the internal reset is canceled. Then, deep software standby mode is canceled and the reset exception handling is executed. The DPSRSTF bit in RSTSR is set to 1 on cancellation of deep software standby mode.
- Canceling by $\overline{\text{RES}}$

  Driving the $\overline{\text{RES}}$ pin low starts the clock oscillation and internal power supply. The clock supply to the LSI is also started. At this time, the $\overline{\text{RES}}$ pin should be held low until the clock oscillation is stabilized. When the $\overline{\text{RES}}$ pin is driven high, the CPU starts the reset exception handling. Keep the $\overline{\text{TRST}}$ pin low during the oscillation settling time regardless whether the H-UDI function is used or not.

(3) Notes on Transition to Deep Software Standby Mode

If a software standby mode cancellation interrupt or deep software standby mode cancellation interrupt is input during a transition to deep software standby mode, the following operation is performed according to the timing of the interrupt acceptance.

- Interrupt exception handing is executed and then a transition is made to deep software standby mode.
- A transition is made to deep software standby mode, deep software standby mode is canceled by the interrupt, and then the interrupt exception handling is executed.
- A transition is made to software standby mode, software standby mode is canceled, and then the interrupt exception handling is executed (a transition to deep software standby mode is canceled).

If there is a possibility that a software standby mode cancellation interrupt or deep software standby mode cancellation interrupt will be input during a transition to deep software standby mode, also set the WDT count stop (set TME in WTCSR to 0) and oscillation settling time following the software standby mode cancellation (set CKS[2:0] in WTCSR and WTCNT), considering cancellation of software standby mode before a transition to deep software standby mode.

(4) Operation after Canceling Deep Software Standby Mode

The oscillation settling time following cancellation of deep software standby mode is set with the WTSEL and WTSTS[6:0] bits in DPSWCR.

- When WTSEL = 0

  The WTSTS[6:0] bits are used as an oscillation settling time setting register.

  Table 32.3 shows the WTSTS[6:0] settings and oscillation settling time with the WTSEL bit being 0.

Table 32.3 Oscillation Settling Time Settings (WTSEL = 0)

								Oscillation Settling Time*		
	WTSTS								Time	
WTSEL	[6]	[5]	[4]	[3]	[2]	[1]	[0]	Number of States	EXTAL Input = 12.5 MHz	EXTAL Input = 10 MHz
0	0	0	0	0	1	0	1	64	5.12 µs	6.4 µs
						1	0	512	41 µs	51.2 µs
							1	1,024	81.9 µs	102 µs
				1	0	0	0	2,048	164 µs	205 µs
							1	4,096	328 µs	410 µs
						1	0	16,384	1.31 ms	1.64 ms
							1	32,768	2.62 ms	3.28 ms
					1	0	0	65,536	5.24 ms	6.55 ms
							1	131,072	10.5 ms	13.1 ms
						1	0	262,144	21 ms	26.2 ms
							1	524,288	41.9 ms	52.4 ms
	Other than above							Reserved	—	—

Note: * Since the oscillation settling time includes the period when the oscillator is unstable, the length of the time differs depending on the oscillator characteristics. Therefore, use the above values as reference values.

- When WTSEL = 1

  The WTSTS[6:0] bits are used as a compare register to the oscillation settling time counter. The oscillation settling time counter counts the EXTAL/2048 clock pulses and the oscillation settling time state is canceled when the counter value matches the WTSTS[6:0] bits.

  Table 32.4 shows the WTSTS[6:0] settings and oscillation settling time with the WTSEL bit being 1.

Table 32.4 Oscillation Settling Time Settings (WTSEL = 1)

									Oscillation Settling Time*	
	WTSTS								Time	
WTSEL	[6]	[5]	[4]	[3]	[2]	[1]	[0]	Number of States	EXTAL Input = 12.5 MHz	EXTAL Input = 10 MHz
0	0	1	1	0	0	0	1	49	8.03 ms	10 ms
	0	1	1	1	1	1	0	62	10.2 ms	12.7 ms
	1	0	0	1	0	0	1	73	12 ms	15 ms
	1	0	1	1	1	0	0	92	15.1 ms	18.8 ms
	1	1	0	0	0	1	0	98	16.1 ms	20.1 ms
	1	1	1	1	0	1	0	122	20 ms	25 ms

Note: * Since the oscillation settling time includes the period when the oscillator is unstable, the length of the time differs depending on the oscillator characteristics. Therefore, use the above values as reference values.

32.3.5 Module Standby Function

(1) Transition to Module Standby Function

Setting the standby control register MSTP bits to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in the program execution state and sleep mode. Disable a module before placing it in the module standby mode. In addition, do not access the module's registers while it is in the module standby state.

For details on the states of on-chip peripheral module registers in module standby mode, see section 34.3, Register States in Each Operating Mode.

(2) Canceling Module Standby Function

The module standby function can be canceled by clearing each MSTP bit to 0, or by a power-on reset (only possible for modules of which initial value is 0). When taking a module out of the module standby state by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm that it has been cleared to 0.

32.4 Usage Notes

32.4.1 Power Consumption during Oscillation Settling Time

Power consumption is increased during oscillation settling time.

32.4.2 Write to Registers

When writing to the register related to power-down modes, the CPU, after executing a write instruction, executes the next instruction without waiting for the write operation to complete.

Therefore, to reflect the change specified by writing to the register while the next instruction is executed, insert a dummy read of the same register between the register write instruction and the next instruction.

32.4.3 Note on Using IRQx Interrupt Requests to Initiate Release from Software Standby

When an IRQx interrupt request is to drive release from software standby, automatically clear the IRQx flag in IRQRRx through execution of interrupt processing for IRQx while the IRQ sense selection setting in ICRx has been changed such that this will not lead to the generation of an IRQx interrupt request.

The IRQxF flag in IRQ interrupt register x (IRQRRx) having the value 1 ensures that the signal requesting release from software standby is not cleared even when the IRQ sense selection setting in IRQx is changed or the IRQxF flag in IRQRRx for the given interrupt request is cleared to 0.

32.4.4 Note on Using TIM32C and KEYC Interrupts to Initiate Release from Software Standby

When a TIM32C or KEYC interrupt is to drive release from software standby, set the low-power consumption interrupt source (SSRI) higher in the order of priority than the interrupt sources of the TIM32C module (CH0D, CH0U, CH1D, CH1U, and CH2) and of the KEYC module (KSI).

After that, clear the SSRF flag in the standby interrupt flag register (SIFR) to 0 during processing for the low power consumption interrupt, and then execute processing for the TIM32C or KEYC interrupt.

Unless the SSRF flag in SIFR register is cleared to 0 during processing for the low power consumption interrupt, a subsequent low power consumption interrupt will not be able to initiate waking up software standby mode.

Section 33 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for the boundary scan function and emulator support.

33.1 Features

The user debugging interface (H-UDI) is a serial input/output interface that supports JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

The H-UDI of this LSI incorporates a boundary scan TAP controller and an emulation TAP controller for controlling the H-UDI interrupt function. When the $\overline{\text{TRST}}$ pin is asserted, including the case of power-on, the boundary scan TAP controller is selected. By inputting the emulation TAP controller switching command, the emulation TAP controller is selected. To switch from the emulation TAP controller to the boundary scan TAP controller, assert the $\overline{\text{TRST}}$ pin.

In ASE mode, the emulation TAP controller is selected. For connection with the emulator, see the manual for the emulator.

Figure 33.1 shows a block diagram of the H-UDI.

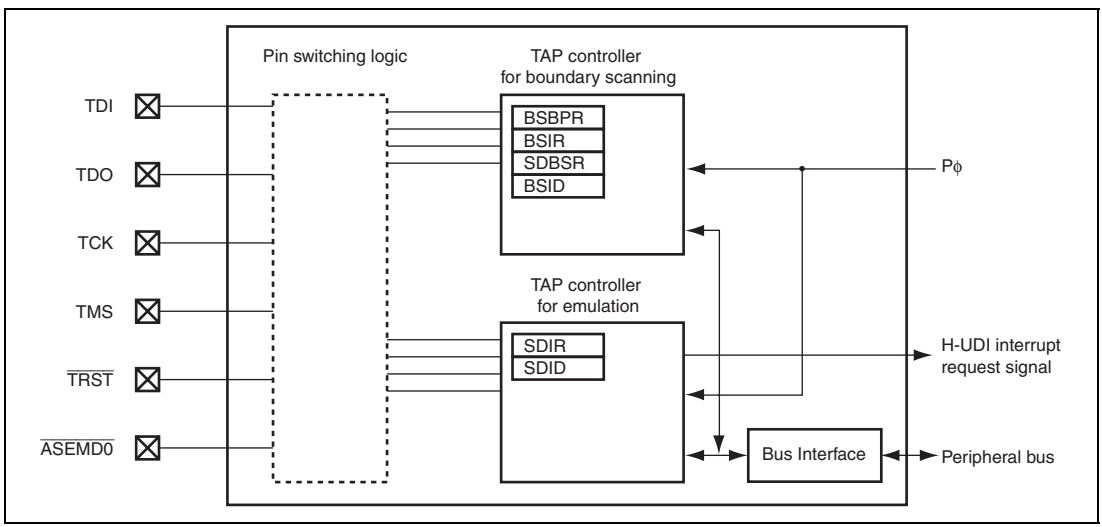


Figure 33.1 Block Diagram of H-UDI

33.2 Input/Output Pins

Table 33.1 Pin Configuration

Pin Name	Symbol	I/O	Function	Handling When Not Used
H-UDI serial data input/output clock pin	TCK	Input	Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.	Open*1
Mode select input pin	TMS	Input	The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol complies with the JTAG standard (IEEE Std.1149.1).	Open*1
H-UDI reset input pin	$\overline{\text{TRST}}$*2	Input	Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a period when power is turned on regardless of using the H-UDI function. See section 33.5.2, Reset Configuration, for more information.	Fix to ground level or connect to $\overline{\text{RES}}$*3
H-UDI serial data input pin	TDI	Input	Data is transferred to the H-UDI by changing this signal in synchronization with TCK.	Open*1
H-UDI serial data output pin	TDO	Output	Data is read from the H-UDI by reading this pin in synchronization with TCK. The initial value of the data output timing is the TCK falling edge, but this initial value can be changed to the TCK rising edge by inputting the TDO transition timing switching command to SDIR. See section 33.5.4, TDO Output Timing, for more information.	Open
ASE mode select pin	$\overline{\text{ASEMD0}}$*4	Input	If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RES}}$ pin is asserted, ASE mode is entered; if a high level is input, product chip mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RES}}$ negation.	Fix to Vcc

Notes: 1. This pin is pulled up in the chip. When designing a board to be used with an emulator or using the interrupts or resets via the H-UDI, external pull-up resistors can also be used.

2. When designing a board to be used with an emulator or using the interrupts or resets via the H-UDI, keep the $\overline{\text{TRST}}$ low while the $\overline{\text{RES}}$ is low at a power-on. At this time, the $\overline{\text{TRST}}$ should be controlled separately from the $\overline{\text{RES}}$. Execute a power-on reset before using the H-UDI.
3. Fix this pin to the ground level or connect this pin to the same signal as the $\overline{\text{RES}}$ (a signal behaving in the same way as the $\overline{\text{RES}}$). Note the following point when it is fixed to the ground level:

   If this pin is externally fixed to the ground, micro-current flows since the $\overline{\text{TRST}}$ is pulled up inside the chip. The current value depends on the input pull-up MOS current specification. Connecting to the ground level does not affect the chip operation but causes unnecessary power consumption.
4. When the emulator is not in use, fix this pin to the high level.

The TCK frequency should be lower than the on-chip peripheral clock frequency.

33.3 Description of the Boundary Scan TAP Controller

The boundary scan TAP controller has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 33.2 Register Configuration of the Boundary Scan TAP Controller

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	BSBPR	—	—	—	—
Instruction register	BSIR	—	—	—	—
Boundary scan register	SDBSR	—	—	—	—
ID register	BSID	R	H'080B1447	—	—

33.3.1 Bypass Register (BSBPR)

BSBPR is a 1-bit register that cannot be accessed by the CPU. When BSIR is set to BYPASS mode, BSBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined.

33.3.2 Instruction Register (BSIR)

BSIR is a 4-bit register and initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state. This register cannot be accessed by the CPU.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	TI[3:0]	0100	—	Test Instruction The H-UDI instruction is transferred to BSIR as a serial input from TDI. For commands, see table 33.3.

Table 33.3 Supported Commands for Boundary Scan TAP Controller

Bits 3 to 0				
TI3	**TI2**	**TI1**	**TI0**	**Description**
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	0	1	1	Emulation TAP controller switching command
0	1	0	0	IDCODE (initial value)
0	1	1	0	CLAMP
0	1	1	1	HIGHZ
Other than the above				Reserved

33.3.3 Boundary Scan Register (SDBSR)

SDBSR is a shift register located on the PAD to control input/output pins of this LSI. This register cannot be accessed by the CPU. The initial value is undefined.

The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands can be used to perform the boundary scan test that conforms to the JTAG standard. Table 33.4 shows the correspondence between the LSI pins and the bits of the boundary scan register.

Table 33.4 Correspondence between the LSI Pins and the Bits of the Boundary Scan Register (for SH7231)

Bit Number	Pin Name*	Type
519	NMI	INPUT
518	PH2	OUTPUT
517	PH2	CONTROL
516	PH2	INPUT
515	PH1	OUTPUT
514	PH1	CONTROL
513	PH1	INPUT
512	PH0	OUTPUT
511	PH0	CONTROL
510	PH0	INPUT
509	PC15	OUTPUT
508	PC15	CONTROL
507	PC15	INPUT
506	PC14	OUTPUT
505	PC14	CONTROL
504	PC14	INPUT
503	PC13	OUTPUT
502	PC13	CONTROL
501	PC13	INPUT
500	PC11	OUTPUT
499	PC11	CONTROL
498	PC11	INPUT
497	PC12	OUTPUT
496	PC12	CONTROL
495	PC12	INPUT
494	PA15	OUTPUT
493	PA15	CONTROL
492	PA15	INPUT

Bit Number	Pin Name*	Type
491	PC10	OUTPUT
490	PC10	CONTROL
489	PC10	INPUT
488	PC6	OUTPUT
487	PC6	CONTROL
485	PC2	OUTPUT
484	PC2	CONTROL
483	PC2	INPUT
482	PC5	OUTPUT
481	PC5	CONTROL
480	PC5	INPUT
479	PC1	OUTPUT
478	PC1	CONTROL
477	PC1	INPUT
476	PC9	OUTPUT
475	PC9	CONTROL
474	PC9	INPUT
473	PC0	OUTPUT
472	PC0	CONTROL
471	PC0	INPUT
470	PC4	OUTPUT
469	PC4	CONTROL
468	PC4	INPUT
467	PC8	OUTPUT
466	PC8	CONTROL
465	PC8	INPUT
464	PA7	OUTPUT
463	PA7	CONTROL
462	PA7	INPUT

Bit Number	Pin Name*	Type
461	PC3	OUTPUT
460	PC3	CONTROL
459	PC3	INPUT
458	PC7	OUTPUT
457	PC7	CONTROL
456	PC7	INPUT
455	PA16	OUTPUT
454	PA16	CONTROL
453	PA16	INPUT
452	PA9	OUTPUT
451	PA9	CONTROL
450	PA9	INPUT
449	PA8	OUTPUT
448	PA8	CONTROL
447	PA8	INPUT
446	PA6	OUTPUT
445	PA6	CONTROL
444	PA6	INPUT
443	PA5	OUTPUT
442	PA5	CONTROL
441	PA5	INPUT
440	PA18	OUTPUT
439	PA18	CONTROL
438	PA18	INPUT
437	PA19	OUTPUT
436	PA19	CONTROL
435	PA19	INPUT
434	PA4	OUTPUT
433	PA4	CONTROL
432	PA4	INPUT
431	PA3	OUTPUT
430	PA3	CONTROL

Bit Number	Pin Name*	Type
429	PA3	INPUT
428	PA17	OUTPUT
427	PA17	CONTROL
426	PA17	INPUT
425	PA14	OUTPUT
424	PA14	CONTROL
423	PA14	INPUT
422	PA13	OUTPUT
421	PA13	CONTROL
420	PA13	INPUT
419	PA12	OUTPUT
418	PA12	CONTROL
417	PA12	INPUT
416	PA11	OUTPUT
415	PA11	CONTROL
414	PA11	INPUT
413	PA10	OUTPUT
412	PA10	CONTROL
411	PA10	INPUT
410	PA2	OUTPUT
409	PA2	CONTROL
408	PA2	INPUT
407	PA1	OUTPUT
406	PA1	CONTROL
405	PA1	INPUT
404	PA0	OUTPUT
403	PA0	CONTROL
402	PA0	INPUT
401	PD31	OUTPUT
400	PD31	CONTROL
399	PD31	INPUT
398	PD30	OUTPUT

Bit Number	Pin Name*	Type
397	PD30	CONTROL
396	PD30	INPUT
395	PD15	OUTPUT
394	PD15	CONTROL
393	PD15	INPUT
392	PD14	OUTPUT
391	PD14	CONTROL
390	PD14	INPUT
389	PD29	OUTPUT
388	PD29	CONTROL
387	PD29	INPUT
386	PD13	OUTPUT
385	PD13	CONTROL
384	PD13	INPUT
383	PD12	OUTPUT
382	PD12	CONTROL
381	PD12	INPUT
380	PD28	OUTPUT
379	PD28	CONTROL
378	PD28	INPUT
377	PD11	OUTPUT
376	PD11	CONTROL
375	PD11	INPUT
374	PD27	OUTPUT
373	PD27	CONTROL
372	PD27	INPUT
371	PD10	OUTPUT
370	PD10	CONTROL
369	PD10	INPUT
368	PD26	OUTPUT
367	PD26	CONTROL
366	PD26	INPUT

Bit Number	Pin Name*	Type
365	PD9	OUTPUT
364	PD9	CONTROL
363	PD9	INPUT
362	PD25	OUTPUT
361	PD25	CONTROL
360	PD25	INPUT
359	PD8	OUTPUT
358	PD8	CONTROL
357	PD8	INPUT
356	PD7	OUTPUT
355	PD7	CONTROL
354	PD7	INPUT
353	PD24	OUTPUT
352	PD24	CONTROL
351	PD24	INPUT
350	PD23	OUTPUT
349	PD23	CONTROL
348	PD23	INPUT
347	PD6	OUTPUT
346	PD6	CONTROL
345	PD6	INPUT
344	PD5	OUTPUT
343	PD5	CONTROL
342	PD5	INPUT
341	PD22	OUTPUT
340	PD22	CONTROL
339	PD22	INPUT
338	PD4	OUTPUT
337	PD4	CONTROL
336	PD4	INPUT
335	PD21	OUTPUT
334	PD21	CONTROL

Bit Number	Pin Name*	Type
333	PD21	INPUT
332	PD20	OUTPUT
331	PD20	CONTROL
330	PD20	INPUT
329	PD3	OUTPUT
328	PD3	CONTROL
327	PD3	INPUT
326	PD19	OUTPUT
325	PD19	CONTROL
324	PD19	INPUT
323	PD18	OUTPUT
322	PD18	CONTROL
321	PD18	INPUT
320	PD2	OUTPUT
319	PD2	CONTROL
318	PD2	INPUT
317	PD1	OUTPUT
316	PD1	CONTROL
315	PD1	INPUT
314	PD17	OUTPUT
313	PD17	CONTROL
312	PD17	INPUT
311	PD16	OUTPUT
310	PD16	CONTROL
309	PD16	INPUT
308	PD0	OUTPUT
307	PD0	CONTROL
306	PD0	INPUT
305	PB9	OUTPUT
304	PB9	CONTROL
303	PB9	INPUT
302	PB1	OUTPUT
301	PB1	CONTROL
300	PB1	INPUT
299	PB7	OUTPUT
298	PB7	CONTROL
297	PB7	INPUT
296	PB3	OUTPUT
295	PB3	CONTROL
294	PB3	INPUT
293	PB2	OUTPUT
292	PB2	CONTROL
291	PB2	INPUT
290	PB8	OUTPUT
289	PB8	CONTROL
288	PB8	INPUT
287	PB0	OUTPUT
286	PB0	CONTROL
285	PB0	INPUT
284	PB13	OUTPUT
283	PB13	CONTROL
282	PB13	INPUT
281	PB11	OUTPUT
280	PB11	CONTROL
279	PB11	INPUT
278	PB12	OUTPUT
277	PB12	CONTROL
276	PB12	INPUT
275	PB10	OUTPUT
274	PB10	CONTROL
273	PB10	INPUT
272	PB5	OUTPUT
271	PB5	CONTROL
270	PB5	INPUT

Bit Number	Pin Name*	Type
269	PB4	OUTPUT
268	PB4	CONTROL
267	PB4	INPUT
266	PB6	OUTPUT
265	PB6	CONTROL
264	PB6	INPUT
263	PG14	OUTPUT
262	PG14	CONTROL
261	PG14	INPUT
260	PG15	OUTPUT
259	PG15	CONTROL
258	PG15	INPUT
257	PG13	OUTPUT
256	PG13	CONTROL
255	PG13	INPUT
254	PG12	OUTPUT
253	PG12	CONTROL
252	PG12	INPUT
251	PG11	OUTPUT
250	PG11	CONTROL
249	PG11	INPUT
248	PG10	OUTPUT
247	PG10	CONTROL
246	PG10	INPUT
245	PG9	OUTPUT
244	PG9	CONTROL
243	PG9	INPUT
242	PG8	OUTPUT
241	PG8	CONTROL
240	PG8	INPUT
239	PG7	OUTPUT
238	PG7	CONTROL
237	PG7	INPUT
236	PG6	OUTPUT
235	PG6	CONTROL
234	PG6	INPUT
233	PG5	OUTPUT
232	PG5	CONTROL
231	PG5	INPUT
230	PG4	OUTPUT
229	PG4	CONTROL
228	PG4	INPUT
227	PG3	OUTPUT
226	PG3	CONTROL
225	PG3	INPUT
224	PG2	OUTPUT
223	PG2	CONTROL
222	PG2	INPUT
221	PG1	OUTPUT
220	PG1	CONTROL
219	PG1	INPUT
218	PG0	OUTPUT
217	PG0	CONTROL
216	PG0	INPUT
215	PK7	OUTPUT
214	PK7	CONTROL
213	PK7	INPUT
212	PK6	OUTPUT
211	PK6	CONTROL
210	PK6	INPUT
209	PK5	OUTPUT
208	PK5	CONTROL
207	PK5	INPUT
206	PK4	OUTPUT

Bit Number	Pin Name*	Type
205	PK4	CONTROL
204	PK4	INPUT
203	PK3	OUTPUT
202	PK3	CONTROL
201	PK3	INPUT
200	PK2	OUTPUT
199	PK2	CONTROL
198	PK2	INPUT
197	PK1	OUTPUT
196	PK1	CONTROL
195	PK1	INPUT
194	PK0	OUTPUT
193	PK0	CONTROL
192	PK0	INPUT
191	PJ15	OUTPUT
190	PJ15	CONTROL
189	PJ15	INPUT
188	PJ14	OUTPUT
187	PJ14	CONTROL
186	PJ14	INPUT
185	PJ13	OUTPUT
184	PJ13	CONTROL
183	PJ13	INPUT
182	PJ12	OUTPUT
181	PJ12	CONTROL
180	PJ12	INPUT
179	PJ11	OUTPUT
178	PJ11	CONTROL
177	PJ11	INPUT
176	PJ10	OUTPUT
175	PJ10	CONTROL
174	PJ10	INPUT

Bit Number	Pin Name*	Type
173	PJ9	OUTPUT
172	PJ9	CONTROL
171	PJ9	INPUT
170	PJ8	OUTPUT
169	PJ8	CONTROL
168	PJ8	INPUT
167	PJ7	OUTPUT
166	PJ7	CONTROL
165	PJ7	INPUT
164	PJ6	OUTPUT
163	PJ6	CONTROL
162	PJ6	INPUT
161	PJ5	OUTPUT
160	PJ5	CONTROL
159	PJ5	INPUT
158	PJ4	OUTPUT
157	PJ4	CONTROL
156	PJ4	INPUT
155	PJ3	OUTPUT
154	PJ3	CONTROL
153	PJ3	INPUT
152	PJ2	OUTPUT
151	PJ2	CONTROL
150	PJ2	INPUT
149	PJ1	OUTPUT
148	PJ1	CONTROL
147	PJ1	INPUT
146	PJ0	OUTPUT
145	PJ0	CONTROL
144	PJ0	INPUT
143	PE18	OUTPUT
142	PE18	CONTROL

Bit Number	Pin Name*	Type
141	PE18	INPUT
140	PE20	OUTPUT
139	PE20	CONTROL
138	PE20	INPUT
137	PE19	OUTPUT
136	PE19	CONTROL
135	PE19	INPUT
134	PE23	OUTPUT
133	PE23	CONTROL
132	PE23	INPUT
131	PE22	OUTPUT
130	PE22	CONTROL
129	PE22	INPUT
128	PE21	OUTPUT
127	PE21	CONTROL
126	PE21	INPUT
125	PE14	OUTPUT
124	PE14	CONTROL
123	PE14	INPUT
122	PE13	OUTPUT
121	PE13	CONTROL
120	PE13	INPUT
119	PE12	OUTPUT
118	PE12	CONTROL
117	PE12	INPUT
116	PE16	OUTPUT
115	PE16	CONTROL
114	PE16	INPUT
113	PE17	OUTPUT
112	PE17	CONTROL
111	PE17	INPUT
110	PE15	OUTPUT
109	PE15	CONTROL
108	PE15	INPUT
107	PE11	OUTPUT
106	PE11	CONTROL
105	PE11	INPUT
104	PE10	OUTPUT
103	PE10	CONTROL
102	PE10	INPUT
101	PE9	OUTPUT
100	PE9	CONTROL
99	PE9	INPUT
98	PE8	OUTPUT
97	PE8	CONTROL
96	PE8	INPUT
95	PE7	OUTPUT
94	PE7	CONTROL
93	PE7	INPUT
92	PE6	OUTPUT
91	PE6	CONTROL
90	PE6	INPUT
89	PE5	OUTPUT
88	PE5	CONTROL
87	PE5	INPUT
86	PE4	OUTPUT
85	PE4	CONTROL
84	PE4	INPUT
83	PE3	OUTPUT
82	PE3	CONTROL
81	PE3	INPUT
80	PE2	OUTPUT
79	PE2	CONTROL
78	PE2	INPUT

Bit Number	Pin Name*	Type
77	PE1	OUTPUT
76	PE1	CONTROL
75	PE1	INPUT
74	PE0	OUTPUT
73	PE0	CONTROL
72	PE0	INPUT
71	MD1	INPUT
70	PF15	INPUT
69	PF14	INPUT
68	PF13	INPUT
67	PF12	INPUT
66	PF11	INPUT
65	PF10	INPUT
64	PF9	INPUT
63	PF8	INPUT
62	PF7	INPUT
61	PF6	INPUT
60	PF5	INPUT
59	PF4	INPUT
58	PF3	INPUT
57	PF2	INPUT
56	PF1	INPUT
55	PF0	INPUT
54	MD0	INPUT
53	PH15	OUTPUT
52	PH15	CONTROL
51	PH15	INPUT
50	PH14	OUTPUT
49	PH14	CONTROL
48	PH14	INPUT
47	PH13	OUTPUT
46	PH13	CONTROL
45	PH13	INPUT
44	PH12	OUTPUT
43	PH12	CONTROL
42	PH12	INPUT
41	AUDATA0	OUTPUT
40	AUDATA0	CONTROL
39	AUDCK	OUTPUT
38	AUDCK	CONTROL
37	AUDATA1	OUTPUT
36	AUDATA1	CONTROL
35	AUDATA2	OUTPUT
34	AUDATA2	CONTROL
33	AUDATA3	OUTPUT
32	AUDATA3	CONTROL
31	$\overline{\text{AUDSYNC}}$	OUTPUT
30	$\overline{\text{AUDSYNC}}$	CONTROL
29	PH11	OUTPUT
28	PH11	CONTROL
27	PH11	INPUT
26	PH10	OUTPUT
25	PH10	CONTROL
24	PH10	INPUT
23	PH9	OUTPUT
22	PH9	CONTROL
21	PH9	INPUT
20	PH8	OUTPUT
19	PH8	CONTROL
18	PH8	INPUT
17	PH7	OUTPUT
16	PH7	CONTROL
15	PH7	INPUT
14	PH6	OUTPUT

Bit Number	Pin Name*	Type
13	PH6	CONTROL
12	PH6	INPUT
11	PH5	OUTPUT
10	PH5	CONTROL
9	PH5	INPUT
8	PH4	OUTPUT
7	PH4	CONTROL
6	PH4	INPUT
5	PH3	OUTPUT
4	PH3	CONTROL
3	PH3	INPUT
2	FWE/$\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$	OUTPUT
1	FWE/$\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$	CONTROL
0	FWE/$\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$	INPUT

Note: * The pin of CONTROL is active-low. When this pin is driven low, the state of the corresponding pin is output.

33.3.4 ID Register (BSID)

BSID is a 32-bit register that cannot be accessed by the CPU. The register can be read from H-UDI pins when the IDCODE command is set, but is not writable.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DID[31:16]															
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DID[15:0]															
Initial value:	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID[31:0]	H'080B1447	R	Device This is an ID register defined by JTAG. The upper four bits may be changed for different chip versions.

33.4 Description of the Emulation TAP Controller

To use the emulation TAP controller, enter the emulation TAP controller switching command in the BSIR register of the boundary scan TAP controller. The emulation TAP controller has the following registers. For the states of these registers in each processing status, refer to section 34, List of Registers.

Table 33.5 Register Configuration of the Emulation TAP Controller

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Instruction register	SDIR	R	H'EFFD	H'FFFE2000	16
ID register	SDID	R	H'080B1447	H'FFFE1810	32

33.4.1 Instruction Register (SDIR)

SDIR is a 16-bit read-only register and initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state. H-UDI can write to this register regardless of the CPU mode. When a reserved command is set in this register, the operation is not guaranteed. The initial value is H'EFFD.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI[7:0]								-	-	-	-	-	-	-	-
Initial value:	1*	1*	1*	0*	1*	1*	1*	1*	1	1	1	1	1	1	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value of TI[7:0] is a reserved value, but replace it with a non-reserved value when setting a command.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI[7:0]	11101111*	R	Test Instruction Instruction for the H-UDI is transferred to SDIR as a serial input from TDI. For commands, see table 33.6.
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved These bits are always read as 0.
0	—	1	R	Reserved These bits are always read as 1.

Table 33.6 Supported Commands for Emulation TAP Controller

Bits 15 to 8								
TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	Description
0	1	1	0	—	—	—	—	H-UDI reset negation
0	1	1	1	—	—	—	—	H-UDI reset assertion
1	0	0	1	1	1	0	0	TDO transition timing switch
1	0	1	1	—	—	—	—	H-UDI interrupt
Other than the above								Reserved

33.4.2 ID Register (SDID)

SDID is a 32-bit register that can be accessed by the CPU and is used to identify the particular ID code of this LSI. The register cannot be read from H-UDI pins.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SDID[31:16]															
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDID[15:0]															
Initial value:	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SDID[31:0]	H'080B1447	R	Device ID code of this LSI

33.5 Operation

33.5.1 TAP Controller

Figure 33.2 shows the internal states of the TAP controller. This state machine conforms to the state transitions defined by JTAG.

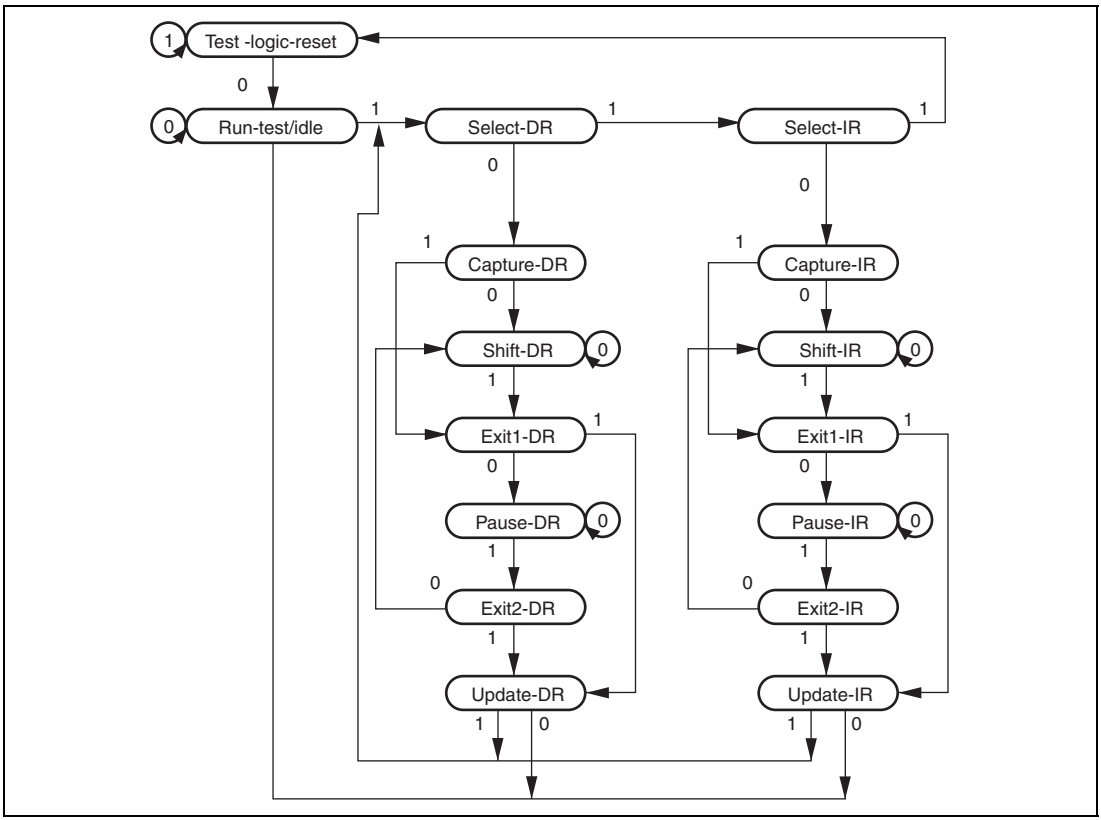


Figure 33.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. For details on transition timing of the TDO value, see section 33.5.4, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. By assertion of the $\overline{\text{TRST}}$, a transition is made to test-logic-reset asynchronously with TCK.

33.5.2 Reset Configuration

Table 33.7 Reset Configuration

$\overline{\text{ASEMD0}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Power-on reset and H-UDI reset
		H	Power-on reset
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*2
		H	Power-on reset
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs product chip mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode

2. In ASE mode, reset hold is entered if the $\overline{\text{TRST}}$ pin is driven low while the $\overline{\text{RES}}$ pin is negated. In this state, the CPU does not start up. When $\overline{\text{TRST}}$ is driven high, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is cancelled by a power-on reset.

33.5.3 TAP Controller Switching

In the H-UDI of this LSI, the boundary scan TAP controller and emulation TAP controller for controlling the H-UDI reset and interrupt functions are separated. When the $\overline{\text{TRST}}$ pin is asserted, including the case of power-on, the boundary scan TAP controller becomes valid and boundary scan functions specified by JTAG are enabled. By inputting the emulation TAP controller switching command, the H-UDI reset and interrupt functions are enabled. Figure 33.3 shows the sequence of switching from boundary scan TAP controller to emulation TAP controller.

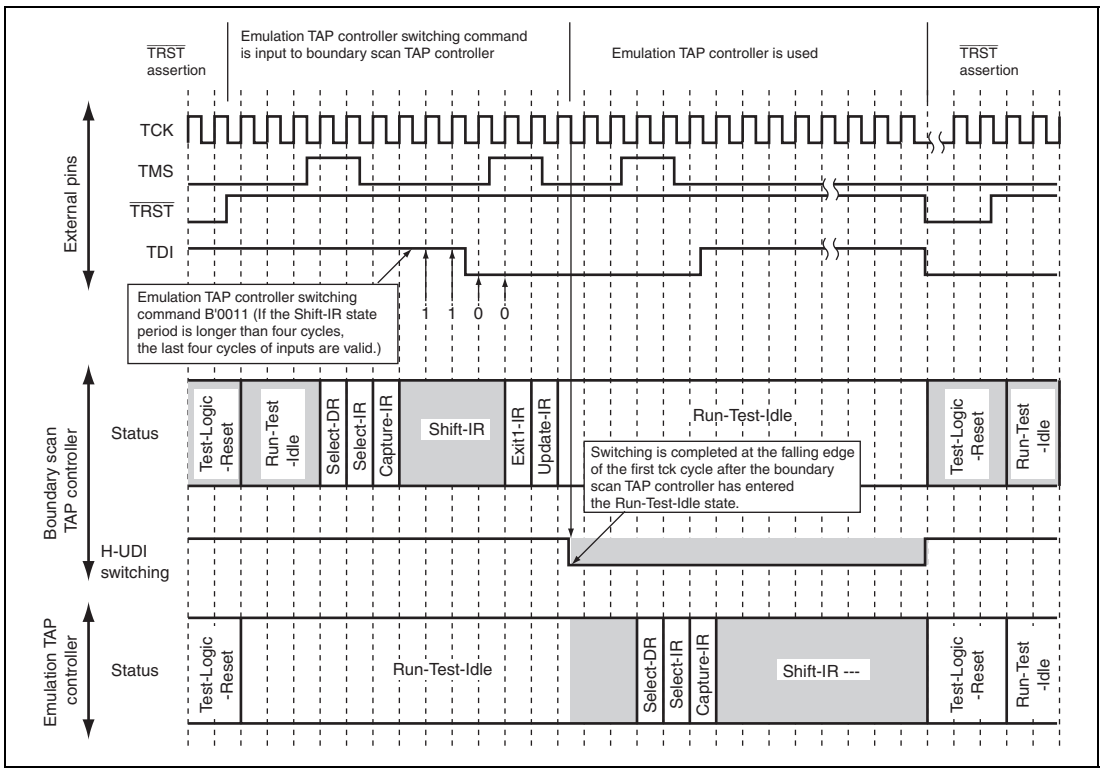


Figure 33.3 Sequence of Switching from Boundary Scan TAP Controller to Emulation TAP Controller

33.5.4 TDO Output Timing

When the emulation TAP controller is selected, a transition on the TDO pin is output on the falling edge of TCK with the initial value. However, setting a TDO transition timing switching command in SDIR via the H-UDI pin and passing the Update-IR state synchronizes the TDO transition with the rising edge of TCK. This command does not affect the output timing of the boundary scan TAP controller.

To synchronize the transition of TDO with the falling edge of TCK after setting the TDO transition timing switching command, the $\overline{\text{TRST}}$ pin must be asserted simultaneously with the power-on reset. In the case of power-on reset by the $\overline{\text{RES}}$ pin, the sync reset is still in operation for a certain period in the LSI even after the $\overline{\text{RES}}$ pin is negated. Thus, if the $\overline{\text{TRST}}$ pin is asserted immediately after the negation of the $\overline{\text{RES}}$ pin, the TDO transition timing switching command is cleared, resulting in TDO transitions synchronized with the falling edges of TCK. To prevent this, make sure to allow a period of 20 tcyc or longer between the signal transitions of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins.

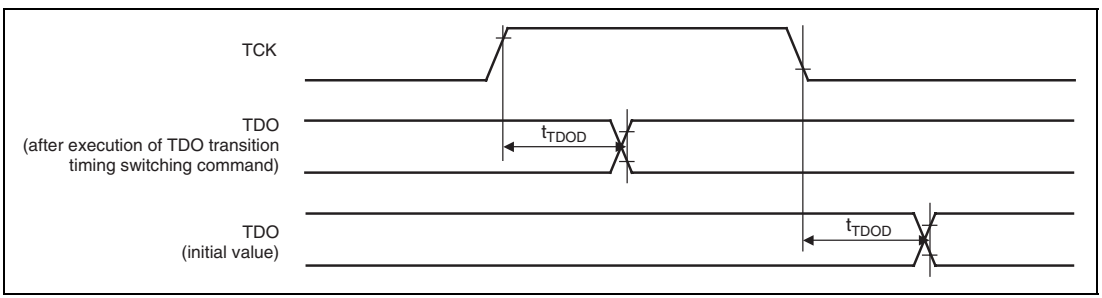


Figure 33.4 H-UDI Data Transfer Timing

33.5.5 H-UDI Reset

An H-UDI reset occurs when an H-UDI reset assert command is set in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is cleared by setting an H-UDI reset negate command. The required time between the H-UDI reset assert command and H-UDI reset negate command is the same as time for keeping the $\overline{\text{RES}}$ pin low to apply a power-on reset.

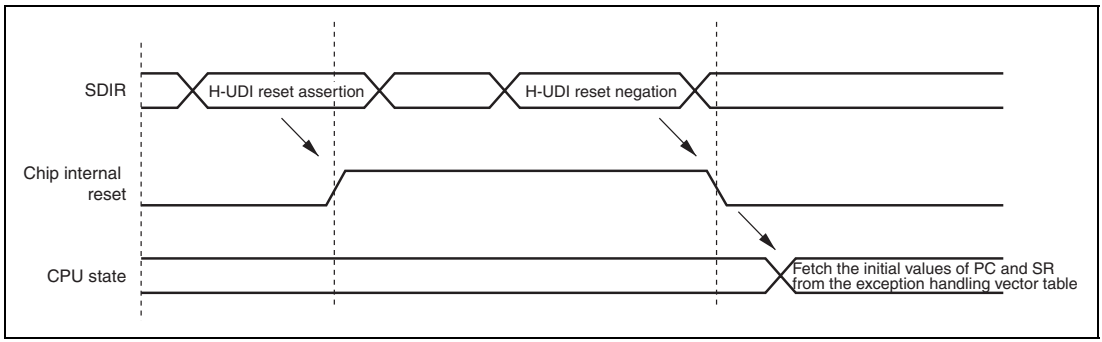


Figure 33.5 H-UDI Reset

33.5.6 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in fetching the exception service routine start address from the exception handling vector table, jumping to that address, and starting program execution from that address. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in software standby mode.

33.6 Boundary Scan

By setting the commands in BSIR by the H-UDI, the H-UDI pins can be configured for boundary scan mode defined by JTAG.

33.6.1 Supported Instructions

This LSI supports three required instructions (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three optional instructions (IDCODE, CLAMP, and HIGHZ) defined by JTAG.

(1) BYPASS

The BYPASS instruction is a required standard instruction to operate the bypass register. This instruction is used to increase the transfer speed of serial data of other LSIs on the printed circuit board by reducing the shift path. During execution of this instruction, the test circuit does not affect the system circuit.

(2) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction inputs a value from the internal circuit of the LSI to the boundary scan register, and output the data from scan path or load the data to the scan path. During execution of the instruction, the value on the input pin of the LSI is transferred to the internal circuit and the value of the internal circuit is output externally from the output pin. Execution of the instruction does not affect the system circuit of the LSI.

In SAMPLE operation, the snapshots of the value transferred from the input pin to the internal circuit and the value transferred from the internal circuit to the output pin are captured in the boundary scan register and then read from the scan path. Capturing of the snapshots is performed in synchronization with the rising edge of TCK in the capture-DR state. The capturing is performed without interfering with normal operation of the LSI.

In PRELOAD operation, an initial value is set in the output latch of the boundary scan register from the scan path before execution of the EXTEST instruction. Without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transferred to the output latch) during execution of the EXTEST instruction (the parallel output latch is always output to the output pin with the EXTEST instruction).

(3) EXTEST

The EXTEST instruction tests the external circuit when this LSI is mounted on the printed circuit board. During execution of this instruction, the output pin is used to output the test data (set in advance by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board and the input pin is used to capture the test result from the printed circuit board to the boundary scan register. When a test is performed using the EXTEST instruction N times, the N-th test data is scanned-in during (N-1)-th scan-out.

The data loaded in the boundary scan register of the output pin in the capture-DR state of this instruction is not used in testing of the external circuit (an exchange is made in shift operation).

(4) IDCODE

When the IDCODE instruction is selected, the TDO pin outputs the ID register value from the LSB first while the TAP controller is in the Shift-DR state. During execution of this instruction, the test circuit does not affect the system circuit.

When the TAP controller is in the test-logic-reset state, the instruction register is initialized to hold the IDCODE instruction.

(5) CLAMP

When the CLAMP instruction is selected, the output pin outputs the value in the boundary scan register preset with the SAMPLE/PRELOAD command. The boundary scan register retains the previous value while the CLAMP instruction is selected, regardless of the TAP controller state.

The bypass register is connected between the TDI and TDO pins to perform the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all the output pins are set to the high impedance state. The boundary scan register retains the previous value while the HIGHZ instruction is selected, regardless of the TAP controller state.

The bypass register is connected between the TDI and TDO pins to perform the same operation as when the BYPASS instruction is selected.

33.6.2 Notes

1. The following pins are inapplicable to the boundary scan:
   - Clock related pins: EXTAL, XTAL, EXTAL32, and XTAL32
   - System and E10A related pins: $\overline{\text{RES}}$, $\overline{\text{ASEMD0}}$, and $\overline{\text{WDTOVF}}$
   - H-UDI related pins: TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$
   - LVDS related pins: PL0/RXCLKINP, PL1/RXCLKINM, PL2/RXIN0P, PL3/RXIN0M, PL4/RXIN1P, and PL5/RXIN1M

   PL0 to PL5 in the SH72315L/SH72314L are also inapplicable to the boundary scan.
2. Fix the $\overline{\text{ASEMD0}}$ pin at high while executing the boundary scan.
3. The maximum frequency of the TCK is 2 MHz when the boundary scan commands (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, CLAMP, HIGHZ, and emulation TAP controller switching command) are executed.

33.7 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is not set again from the H-UDI. If the same command is to be set continuously, the command must be set after a command (BYPASS mode, etc.) that does not affect chip operations is once set.
2. In software standby mode and H-UDI module standby state, none of the functions in the H-UDI can be used. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.
3. Regardless of whether the H-UDI is used, make sure to keep the $\overline{\text{TRST}}$ pin low to initialize the H-UDI at power-on or in recovery from deep standby by the $\overline{\text{RES}}$ pin assertion.
4. If the $\overline{\text{TRST}}$ pin is asserted immediately after the setting of the TDO transition timing switching command and the negation of the $\overline{\text{RES}}$ pin, the TDO transition timing switching command is cleared. To avoid this case, make sure to put 20 tcyc or longer between the signal transition timing of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins. For details, see section 33.5.4, TDO Output Timing.
5. When starting the TAP controller after the negation of the $\overline{\text{TRST}}$ pin, make sure to allow 200 ns or longer after the negation.
6. The H-UDI is used for emulator connection and therefore the boundary scan and H-UDI functions described in this section cannot be used when an emulator is used.
7. The TCK pin frequency should be lower than the peripheral clock frequency (Pϕ).
8. Data is input or output with the LSB first during serial data transfer.
9. For the power-on reset signal applied to the $\overline{\text{TRST}}$ pin, the following cautions should be taken:
   - When the power supply is turned on or deep software standby mode is canceled by the assertion of the $\overline{\text{RES}}$ pin, be sure to apply the reset signal.
   - To prevent the $\overline{\text{TRST}}$ pin on the board tester from affecting the LSI system operation, separate the circuits.
   - To prevent the LSI system reset from affecting the board tester $\overline{\text{TRST}}$ pin, separate the circuits.

Figure 33.6 shows a sample design of reset-related signals which do not cause interference.

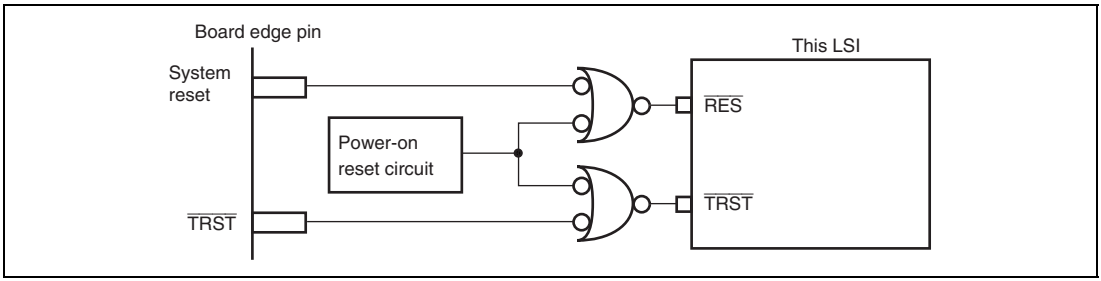


Figure 33.6 Sample Design of Reset-Related Signals Which Do Not Cause Interference

Section 34 List of Registers

This section gives information on the on-chip I/O registers of this LSI in the following structures.

1. Register Addresses (by functional module, in order of the corresponding section numbers)

- Registers are described by functional module, in order of the corresponding section numbers.
- Access to reserved addresses which are not described in this register address list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

4. Notes when Writing to the On-Chip Peripheral Modules

To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design. When the CPU writes data to the internal peripheral registers, the CPU performs the succeeding instructions without waiting for the completion of writing to registers. For example, a case is described here in which the system is transferring to the software standby mode for power savings. To make this transition, the SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR register is indispensable to complete writing to the STBY bit. To reflect the change by internal peripheral registers while performing the succeeding instructions, execute a dummy read of registers to which write instruction is given and then perform the succeeding instructions.

34.1 Register Addresses (by Functional Module, in Order of the Corresponding Section Numbers)

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CPG	Frequency control register	FRQCR	16	H'FFFE0010	16
	MTU2S clock frequency control register	MCLKCR	8	H'FFFE0410	8
	AD clock frequency control register	ACLKCR	8	H'FFFE0414	8
	Oscillation stop detection control register	OSCCR	8	H'FFFE001C	8
INTC	Interrupt control register 0	ICR0	16	H'FFFE0800	16, 32
	Interrupt control register 1	ICR1	16	H'FFFE0802	16
	Interrupt control register 2	ICR2	16	H'FFFE0804	16, 32
	Interrupt control register 3	ICR3	16	H'FFFE0806	16
	IRQ interrupt request register 0	IRQRR0	16	H'FFFE0808	16, 32
	IRQ interrupt request register 1	IRQRR1	16	H'FFFE080A	16
	Bank control register	IBCR	16	H'FFFE080C	16, 32
	Bank number register	IBNR	16	H'FFFE080E	16
	Interrupt priority register 01	IPR01	16	H'FFFE0818	16, 32
	Interrupt priority register 02	IPR02	16	H'FFFE081A	16
	Interrupt priority register 03	IPR03	16	H'FFFE081C	16, 32
	Interrupt priority register 04	IPR04	16	H'FFFE081E	16
	Interrupt priority register 06	IPR06	16	H'FFFE0C00	16, 32
	Interrupt priority register 08	IPR08	16	H'FFFE0C04	16, 32
	Interrupt priority register 09	IPR09	16	H'FFFE0C06	16
	Interrupt priority register 10	IPR10	16	H'FFFE0C08	16, 32
	Interrupt priority register 11	IPR11	16	H'FFFE0C0A	16
	Interrupt priority register 12	IPR12	16	H'FFFE0C0C	16, 32
	Interrupt priority register 13	IPR13	16	H'FFFE0C0E	16
	Interrupt priority register 14	IPR14	16	H'FFFE0C10	16, 32
	Interrupt priority register 15	IPR15	16	H'FFFE0C12	16
	Interrupt priority register 16	IPR16	16	H'FFFE0C14	16, 32
	Interrupt priority register 17	IPR17	16	H'FFFE0C16	16
	Interrupt priority register 18	IPR18	16	H'FFFE0C18	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
UBC	Break address register_0	BAR_0	32	H'FFFC0400	32
	Break address mask register_0	BAMR_0	32	H'FFFC0404	32
	Break bus cycle register_0	BBR_0	16	H'FFFC04A0	16
	Break address register_1	BAR_1	32	H'FFFC0410	32
	Break address mask register_1	BAMR_1	32	H'FFFC0414	32
	Break bus cycle register_1	BBR_1	16	H'FFFC04B0	16
	Break address register_2	BAR_2	32	H'FFFC0420	32
	Break bus cycle register_2	BBR_2	16	H'FFFC04A8	16
	Break address register_3	BAR_3	32	H'FFFC0430	32
	Break bus cycle register_3	BBR_3	16	H'FFFC04AC	16
	Break address register_4	BAR_4	32	H'FFFC0440	32
	Break bus cycle register_4	BBR_4	16	H'FFFC04B0	16
	Break address register_5	BAR_5	32	H'FFFC0450	32
	Break bus cycle register_5	BBR_5	16	H'FFFC04B4	16
	Break address register_6	BAR_6	32	H'FFFC0460	32
	Break bus cycle register_6	BBR_6	16	H'FFFC04B8	16
	Break address register_7	BAR_7	32	H'FFFC0470	32
	Break bus cycle register_7	BBR_7	16	H'FFFC04BC	16
	Break control register	BRCR	32	H'FFFC04C0	32
DTC	DTC enable register A	DTCERA	16	H'FFFE6000	8, 16
	DTC enable register B	DTCERB	16	H'FFFE6002	8, 16
	DTC enable register C	DTCERC	16	H'FFFE6004	8, 16
	DTC enable register D	DTCERD	16	H'FFFE6006	8, 16
	DTC enable register E	DTCERE	16	H'FFFE6008	8, 16
	DTC enable register F	DTCERF	16	H'FFFE600A	8, 16
	DTC enable register G	DTCERG	16	H'FFFE600C	8, 16
	DTC control register	DTCCR	8	H'FFFE6010	8
	DTC vector base register	DTCVBR	32	H'FFFE6014	8, 16, 32
BSC	Common control register	CMNCR	32	H'FFFC0000	32
	CS0 space bus control register	CS0BCR	32	H'FFFC0004	32
	CS1 space bus control register	CS1BCR	32	H'FFFC0008	32
	CS2 space bus control register	CS2BCR	32	H'FFFC000C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
BSC	CS3 space bus control register	CS3BCR	32	H'FFFC0010	32
	CS4 space bus control register	CS4BCR	32	H'FFFC0014	32
	CS5 space bus control register	CS5BCR	32	H'FFFC0018	32
	CS6 space bus control register	CS6BCR	32	H'FFFC001C	32
	CS7 space bus control register	CS7BCR	32	H'FFFC0020	32
	CS0 space wait control register	CS0WCR	32	H'FFFC0028	32
	CS1 space wait control register	CS1WCR	32	H'FFFC002C	32
	CS2 space wait control register	CS2WCR	32	H'FFFC0030	32
	CS3 space wait control register	CS3WCR	32	H'FFFC0034	32
	CS4 space wait control register	CS4WCR	32	H'FFFC0038	32
	CS5 space wait control register	CS5WCR	32	H'FFFC003C	32
	CS6 space wait control register	CS6WCR	32	H'FFFC0040	32
	CS7 space wait control register	CS7WCR	32	H'FFFC0044	32
	SDRAM control register	SDCR	32	H'FFFC004C	32
	Refresh timer control/status register	RTCSR	32	H'FFFC0050	32
	Refresh timer counter	RTCNT	32	H'FFFC0054	32
	Refresh time constant register	RTCOR	32	H'FFFC0058	32
	Bus function extending register	BSCEHR	16	H'FFFE3C1A	16
DMAC	DMA source address register_0	SAR_0	32	H'FFFE1000	16, 32
	DMA destination address register_0	DAR_0	32	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR_0	32	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR_0	32	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR_0	32	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR_0	32	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR_0	32	H'FFFE1108	16, 32
	DMA source address register_1	SAR_1	32	H'FFFE1010	16, 32
	DMA destination address register_1	DAR_1	32	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR_1	32	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR_1	32	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR_1	32	H'FFFE1110	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload destination address register_1	RDAR_1	32	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR_1	32	H'FFFE1118	16, 32
	DMA source address register_2	SAR_2	32	H'FFFE1020	16, 32
	DMA destination address register_2	DAR_2	32	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR_2	32	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR_2	32	H'FFFE102C	8, 16, 32
	DMA reload source address register_2	RSAR_2	32	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR_2	32	H'FFFE1124	16, 32
	DMA reload transfer count register_2	RDMATCR_2	32	H'FFFE1128	16, 32
	DMA source address register_3	SAR_3	32	H'FFFE1030	16, 32
	DMA destination address register_3	DAR_3	32	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR_3	32	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR_3	32	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR_3	32	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR_3	32	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR_3	32	H'FFFE1138	16, 32
	DMA operation register	DMAOR	16	H'FFFE1200	8, 16
	DMA extension resource selector 0	DMARS0	16	H'FFFE1300	16
	DMA extension resource selector 1	DMARS1	16	H'FFFE1304	16
MTU2	Timer control register_0	TCR_0	8	H'FFFE4300	8, 16, 32
	Timer mode register_0	TMDR_0	8	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	8	H'FFFE4302	8, 16
	Timer I/O control register L_0	TIORL_0	8	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	8	H'FFFE4304	8, 16, 32
	Timer status register_0	TSR_0	8	H'FFFE4305	8
	Timer counter_0	TCNT_0	16	H'FFFE4306	16
	Timer general register A_0	TGRA_0	16	H'FFFE4308	16, 32
	Timer general register B_0	TGRB_0	16	H'FFFE430A	16
	Timer general register C_0	TGRC_0	16	H'FFFE430C	16, 32
	Timer general register D_0	TGRD_0	16	H'FFFE430E	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer general register E_0	TGRE_0	16	H'FFFE4320	16, 32
	Timer general register F_0	TGRF_0	16	H'FFFE4322	16
	Timer interrupt enable register2_0	TIER2_0	8	H'FFFE4324	8, 16
	Timer status register2_0	TSR2_0	8	H'FFFE4325	8
	Timer buffer operation transfer mode register_0	TBTM_0	8	H'FFFE4326	8
	Timer control register_1	TCR_1	8	H'FFFE4380	8, 16
	Timer mode register_1	TMDR_1	8	H'FFFE4381	8
	Timer I/O control register_1	TIOR_1	8	H'FFFE4382	8
	Timer interrupt enable register_1	TIER_1	8	H'FFFE4384	8, 16, 32
	Timer status register_1	TSR_1	8	H'FFFE4385	8
	Timer counter_1	TCNT_1	16	H'FFFE4386	16
	Timer general register A_1	TGRA_1	16	H'FFFE4388	16, 32
	Timer general register B_1	TGRB_1	16	H'FFFE438A	16
	Timer input capture control register	TICCR	8	H'FFFE4390	8
	Timer control register_2	TCR_2	8	H'FFFE4000	8, 16
	Timer mode register_2	TMDR_2	8	H'FFFE4001	8
	Timer I/O control register_2	TIOR_2	8	H'FFFE4002	8
	Timer interrupt enable register_2	TIER_2	8	H'FFFE4004	8, 16, 32
	Timer status register_2	TSR_2	8	H'FFFE4005	8
	Timer counter_2	TCNT_2	16	H'FFFE4006	16
	Timer general register A_2	TGRA_2	16	H'FFFE4008	16, 32
	Timer general register B_2	TGRB_2	16	H'FFFE400A	16
	Timer control register_3	TCR_3	8	H'FFFE4200	8, 16, 32
	Timer mode register_3	TMDR_3	8	H'FFFE4202	8, 16
	Timer I/O control register H_3	TIORH_3	8	H'FFFE4204	8, 16, 32
	Timer I/O control register L_3	TIORL_3	8	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	8	H'FFFE4208	8, 16
	Timer status register_3	TSR_3	8	H'FFFE422C	8, 16
	Timer counter_3	TCNT_3	16	H'FFFE4210	16, 32
	Timer general register A_3	TGRA_3	16	H'FFFE4218	16, 32
	Timer general register B_3	TGRB_3	16	H'FFFE421A	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer general register C_3	TGRC_3	16	H'FFFE4224	16, 32
	Timer general register D_3	TGRD_3	16	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238	8, 16
	Timer control register_4	TCR_4	8	H'FFFE4201	8
	Timer mode register_4	TMDR_4	8	H'FFFE4203	8
	Timer I/O control register H_4	TIORH_4	8	H'FFFE4206	8, 16
	Timer I/O control register L_4	TIORL_4	8	H'FFFE4207	8
	Timer interrupt enable register_4	TIER_4	8	H'FFFE4209	8
	Timer status register_4	TSR_4	8	H'FFFE422D	8
	Timer counter_4	TCNT_4	16	H'FFFE4212	16
	Timer general register A_4	TGRA_4	16	H'FFFE421C	16, 32
	Timer general register B_4	TGRB_4	16	H'FFFE421E	16
	Timer general register C_4	TGRC_4	16	H'FFFE4228	16, 32
	Timer general register D_4	TGRD_4	16	H'FFFE422A	16
	Timer buffer operation transfer mode register_4	TBTM_4	8	H'FFFE4239	8
	Timer A/D converter start request control register	TADCR	16	H'FFFE4240	16
	Timer A/D converter start request cycle set register A	TADCORA_4	16	H'FFFE4244	16, 32
	Timer A/D converter start request cycle set register B_4	TADCORB_4	16	H'FFFE4246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	16	H'FFFE4248	16, 32
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	16	H'FFFE424A	16
	Timer control register U_5	TCRU_5	8	H'FFFE4084	8
	Timer control register V_5	TCRV_5	8	H'FFFE4094	8
	Timer control register W_5	TCRW_5	8	H'FFFE40A4	8
	Timer I/O control register U_5	TIORU_5	8	H'FFFE4086	8
	Timer I/O control register V_5	TIORV_5	8	H'FFFE4096	8
	Timer I/O control register W_5	TIORW_5	8	H'FFFE40A6	8
	Timer interrupt enable register_5	TIER_5	8	H'FFFE40B2	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer status register_5	TSR_5	8	H'FFFE40B0	8
	Timer start register_5	TSTR_5	8	H'FFFE40B4	8
	Timer counter U_5	TCNTU_5	16	H'FFFE4080	16, 32
	Timer counter V_5	TCNTV_5	16	H'FFFE4090	16, 32
	Timer counter W_5	TCNTW_5	16	H'FFFE40A0	16, 32
	Timer general register U_5	TGRU_5	16	H'FFFE4082	16
	Timer general register V_5	TGRV_5	16	H'FFFE4092	16
	Timer general register W_5	TGRW_5	16	H'FFFE40A2	16
	Timer compare match clear register	TCNTCMPCLR	8	H'FFFE40B6	8
	Timer start register	TSTR	8	H'FFFE4280	8, 16
	Timer synchronous register	TSYR	8	H'FFFE4281	8
	Timer counter synchronous start register	TCSYSTR	8	H'FFFE4282	8
	Timer read/write enable register	TRWER	8	H'FFFE4284	8
	Timer output master enable register	TOER	8	H'FFFE420A	8
	Timer output control register 1	TOCR1	8	H'FFFE420E	8, 16
	Timer output control register 2	TOCR2	8	H'FFFE420F	8
	Timer gate control register	TGCR	8	H'FFFE420D	8
	Timer cycle control register	TCDR	16	H'FFFE4214	16, 32
	Timer dead time data register	TDDR	16	H'FFFE4216	16
	Timer subcounter	TCNTS	16	H'FFFE4220	16, 32
	Timer cycle buffer register	TCBR	16	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	8	H'FFFE4230	8, 16
	Timer interrupt skipping counter	TITCNT	8	H'FFFE4231	8
	Timer buffer transfer set register	TBTER	8	H'FFFE4232	8
	Timer dead time enable register	TDER	8	H'FFFE4234	8
	Timer waveform control register	TWCR	8	H'FFFE4260	8
	Timer output level buffer register	TOLBR	8	H'FFFE4236	8
MTU2S	Timer control register_3S	TCR_3S	8	H'FFFE4A00	8, 16, 32
	Timer mode register_3S	TMDR_3S	8	H'FFFE4A02	8, 16
	Timer I/O control register H_3S	TIORH_3S	8	H'FFFE4A04	8, 16, 32
	Timer I/O control register L_3S	TIORL_3S	8	H'FFFE4A05	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2S	Timer interrupt enable register_3S	TIER_3S	8	H'FFFE4A08	8, 16
	Timer status register_3S	TSR_3S	8	H'FFFE4A2C	8, 16
	Timer counter_3S	TCNT_3S	16	H'FFFE4A10	16, 32
	Timer general register A_3S	TGRA_3S	16	H'FFFE4A18	16, 32
	Timer general register B_3S	TGRB_3S	16	H'FFFE4A1A	16
	Timer general register C_3S	TGRC_3S	16	H'FFFE4A24	16, 32
	Timer general register D_3S	TGRD_3S	16	H'FFFE4A26	16
	Timer buffer operation transfer mode register_3S	TBTM_3S	8	H'FFFE4A38	8, 16
	Timer control register_4S	TCR_4S	8	H'FFFE4A01	8
	Timer mode register_4S	TMDR_4S	8	H'FFFE4A03	8
	Timer I/O control register H_4S	TIORH_4S	8	H'FFFE4A06	8, 16
	Timer I/O control register L_4S	TIORL_4S	8	H'FFFE4A07	8
	Timer interrupt enable register_4S	TIER_4S	8	H'FFFE4A09	8
	Timer status register_4S	TSR_4S	8	H'FFFE4A2D	8
	Timer counter_4S	TCNT_4S	16	H'FFFE4A12	16
	Timer general register A_4S	TGRA_4S	16	H'FFFE4A1C	16, 32
	Timer general register B_4S	TGRB_4S	16	H'FFFE4A1E	16
	Timer general register C_4S	TGRC_4S	16	H'FFFE4A28	16, 32
	Timer general register D_4S	TGRD_4S	16	H'FFFE4A2A	16
	Timer buffer operation transfer mode register_4S	TBTM_4S	8	H'FFFE4A39	8
	Timer A/D converter start request control register S	TADCRS	16	H'FFFE4A40	16
	Timer A/D converter start request cycle set register A_4S	TADCORA_4S	16	H'FFFE4A44	16, 32
	Timer A/D converter start request cycle set register B_4S	TADCORB_4S	16	H'FFFE4A46	16
	Timer A/D converter start request cycle set buffer register A_4S	TADCOBRA_4S	16	H'FFFE4A48	16, 32
	Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	16	H'FFFE4A4A	16
	Timer control register U_5S	TCRU_5S	8	H'FFFE4884	8
	Timer control register V_5S	TCRV_5S	8	H'FFFE4894	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2S	Timer control register W_5S	TCRW_5S	8	H'FFFE48A4	8
	Timer I/O control register U_5S	TIORU_5S	8	H'FFFE4886	8
	Timer I/O control register V_5S	TIORV_5S	8	H'FFFE4896	8
	Timer I/O control register W_5S	TIORW_5S	8	H'FFFE48A6	8
	Timer interrupt enable register_5S	TIER_5S	8	H'FFFE48B2	8
	Timer status register_5S	TSR_5S	8	H'FFFE48B0	8
	Timer start register_5S	TSTR_5S	8	H'FFFE48B4	8
	Timer counter U_5S	TCNTU_5S	16	H'FFFE4880	16, 32
	Timer counter V_5S	TCNTV_5S	16	H'FFFE4890	16, 32
	Timer counter W_5S	TCNTW_5S	16	H'FFFE48A0	16, 32
	Timer general register U_5S	TGRU_5S	16	H'FFFE4882	16
	Timer general register V_5S	TGRV_5S	16	H'FFFE4892	16
	Timer general register W_5S	TGRW_5S	16	H'FFFE48A2	16
	Timer compare match clear register S	TCNTCMPCLRS	8	H'FFFE48B6	8
	Timer start register S	TSTRS	8	H'FFFE4A80	8, 16
	Timer synchronous register S	TSYRS	8	H'FFFE4A81	8
	Timer read/write enable register S	TRWERS	8	H'FFFE4A84	8
	Timer output master enable register S	TOERS	8	H'FFFE4A0A	8
	Timer output control register 1S	TOCR1S	8	H'FFFE4A0E	8, 16
	Timer output control register 2S	TOCR2S	8	H'FFFE4A0F	8
	Timer gate control register S	TGCRS	8	H'FFFE4A0D	8
	Timer cycle data register S	TCDRS	16	H'FFFE4A14	16, 32
	Timer dead time data register S	TDDRS	16	H'FFFE4A16	16
	Timer subcounter S	TCNTSS	16	H'FFFE4A20	16, 32
	Timer cycle buffer register S	TCBRS	16	H'FFFE4A22	16
	Timer interrupt skipping set register S	TITCRS	8	H'FFFE4A30	8, 16
	Timer interrupt skipping counter S	TITCNTS	8	H'FFFE4A31	8
	Timer buffer transfer set register S	TBTERS	8	H'FFFE4A32	8
	Timer dead time enable register S	TDERS	8	H'FFFE4A34	8
	Timer synchronous clear register S	TSYCRS	8	H'FFFE4A50	8
	Timer waveform control register S	TWCRS	8	H'FFFE4A60	8
	Timer output level buffer register S	TOLBRS	8	H'FFFE4A36	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
POE2	Input level control/status register 1	ICSR1	16	H'FFFE5000	16
	Output level control/status register 1	OCSR1	16	H'FFFE5002	16
	Input level control/status register 2	ICSR2	16	H'FFFE5004	16
	Output level control/status register 2	OCSR2	16	H'FFFE5006	16
	Input level control/status register 3	ICSR3	16	H'FFFE5008	16
	Software port output enable register	SPOER	8	H'FFFE500A	8
	Port output enable control register 1	POECR1	8	H'FFFE500B	8
	Port output enable control register 2	POECR2	16	H'FFFE500C	16
CMT	Compare match timer start register	CMSTR	16	H'FFFEC000	16
	Compare match timer control/ status register_0	CMCSR_0	16	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	16	H'FFFEC004	16
	Compare match constant register_0	CMCOR_0	16	H'FFFEC006	16
	Compare match timer control/ status register_1	CMCSR_1	16	H'FFFEC008	16
	Compare match counter_1	CMCNT_1	16	H'FFFEC00A	16
	Compare match constant register_1	CMCOR_1	16	H'FFFEC00C	16
CMT2	Timer start register	CM2STR	16	H'FFFEC100	16
	Timer control register	CM2CR	16	H'FFFEC104	16
	Timer I/O control register	CM2IOR	16	H'FFFEC108	16
	Timer status register	CM2SR	16	H'FFFEC10C	16
	Timer counter	CM2CNT	32	H'FFFEC110	32
	Compare match constant register	CM2COR	32	H'FFFEC114	32
	Input capture register 0	CM2ICR0	32	H'FFFEC118	32
	Input capture register 1	CM2ICR1	32	H'FFFEC11C	32
	Output compare register 0	CM2OCR0	32	H'FFFEC120	32
	Output compare register 1	CM2OCR1	32	H'FFFEC124	32
WDT	Watchdog timer control/status register	WTCSR	16	H'FFFE0000	16*
	Watchdog timer counter	WTCNT	16	H'FFFE0002	16*
	Watchdog reset control/status register	WRCSR	16	H'FFFE0004	16*
SCI (channel 0)	Serial mode register_0	SCSMR_0	8	H'FFFF8000	8
	Bit rate register_0	SCBRR_0	8	H'FFFF8002	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCI (channel 0)	Serial control register_0	SCSCR_0	8	H'FFFF8004	8
	Transmit data register_0	SCTDR_0	8	H'FFFF8006	8
	Serial status register_0	SCSSR_0	8	H'FFFF8008	8
	Receive data register_0	SCRDR_0	8	H'FFFF800A	8
	Serial direction control register_0	SCSDCR_0	8	H'FFFF800C	8
	Serial port register_0	SCSPTR_0	8	H'FFFF800E	8
	Serial mode register 2_0	SCSMR2_0	8	H'FFFF8010	8
	Transmit bit rate adjust counter_0	SCTBACNT_0	8	H'FFFF8018	8
	Receive bit rate adjust counter_0	SCRBACNT_0	8	H'FFFF801A	8
	Bit rate adjust compare register_0	SCBACOR_0	8	H'FFFF8014	8
SCI (channel 1)	Serial mode register_1	SCSMR_1	8	H'FFFF8800	8
	Bit rate register_1	SCBRR_1	8	H'FFFF8802	8
	Serial control register_1	SCSCR_1	8	H'FFFF8804	8
	Transmit data register_1	SCTDR_1	8	H'FFFF8806	8
	Serial status register_1	SCSSR_1	8	H'FFFF8808	8
	Receive data register_1	SCRDR_1	8	H'FFFF880A	8
	Serial direction control register_1	SCSDCR_1	8	H'FFFF880C	8
	Serial port register_1	SCSPTR_1	8	H'FFFF880E	8
	Serial mode register 2_1	SCSMR2_1	8	H'FFFF8810	8
	Transmit bit rate adjust counter_1	SCTBACNT_1	8	H'FFFF8818	8
	Receive bit rate adjust counter_1	SCRBACNT_1	8	H'FFFF881A	8
	Bit rate adjust compare register_1	SCBACOR_1	8	H'FFFF8814	8
SCI (channel 2)	Serial mode register_2	SCSMR_2	8	H'FFFF9000	8
	Bit rate register_2	SCBRR_2	8	H'FFFF9002	8
	Serial control register_2	SCSCR_2	8	H'FFFF9004	8
	Transmit data register_2	SCTDR_2	8	H'FFFF9006	8
	Serial status register_2	SCSSR_2	8	H'FFFF9008	8
	Receive data register_2	SCRDR_2	8	H'FFFF900A	8
	Serial direction control register_2	SCSDCR_2	8	H'FFFF900C	8
	Serial port register_2	SCSPTR_2	8	H'FFFF900E	8
	Serial mode register 2_2	SCSMR2_2	8	H'FFFF9010	8
	Transmit bit rate adjust counter_2	SCTBACNT_2	8	H'FFFF9018	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCI (channel 2)	Receive bit rate adjust counter_2	SCRBACNT_2	8	H'FFFF901A	8
	Bit rate adjust compare register_2	SCBACOR_2	8	H'FFFF9014	8
SCI (channel 3)	Serial mode register_3	SCSMR_3	8	H'FFFF9800	8
	Bit rate register_3	SCBRR_3	8	H'FFFF9802	8
	Serial control register_3	SCSCR_3	8	H'FFFF9804	8
	Transmit data register_3	SCTDR_3	8	H'FFFF9806	8
	Serial status register_3	SCSSR_3	8	H'FFFF9808	8
	Receive data register_3	SCRDR_3	8	H'FFFF980A	8
	Serial direction control register_3	SCSDCR_3	8	H'FFFF980C	8
	Serial port register_3	SCSPTR_3	8	H'FFFF980E	8
	Serial mode register 2_3	SCSMR2_3	8	H'FFFF9810	8
	Transmit bit rate adjust counter_3	SCTBACNT_3	8	H'FFFF9818	8
	Receive bit rate adjust counter_3	SCRBACNT_3	8	H'FFFF981A	8
	Bit rate adjust compare register_3	SCBACOR_3	8	H'FFFF9814	8
SCIF (channel 4)	Serial mode register_4	SCSMR_4	16	H'FFFE8000	16
	Bit rate register_4	SCBRR_4	8	H'FFFE8004	8
	Serial control register_4	SCSCR_4	16	H'FFFE8008	16
	Transmit data register_4	SCFTDR_4	8	H'FFFE800C	8
	Serial status register_4	SCFSR_4	16	H'FFFE8010	16
	Receive FIFO data register_4	SCFRDR_4	8	H'FFFE8014	8
	FIFO control register_4	SCFCR_4	16	H'FFFE8018	16
	FIFO data count register_4	SCFDR_4	16	H'FFFE801C	16
	Serial port register_4	SCSPTR_4	16	H'FFFE8020	16
	Line status register_4	SCLSR_4	16	H'FFFE8024	16
	Serial direction control register_4	SCSDCR_4	8	H'FFFE8102	8
	FIFO trigger control register_4	SCFTCR_4	16	H'FFFE8104	16
SCIF (channel 5)	SCSMR_5	SCSMR_5	16	H'FFFE8800	16
	Bit rate register_5	SCBRR_5	8	H'FFFE8804	8
	Serial control register_5	SCSCR_5	16	H'FFFE8808	16
	Transmit FIFO data register_5	SCFTDR_5	8	H'FFFE880C	8
	Serial status register_5	SCFSR_5	16	H'FFFE8810	16
	Receive FIFO data register_5	SCFRDR_5	8	H'FFFE8814	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCIF (channel 5)	FIFO control register_5	SCFCR_5	16	H'FFFE8818	16
	FIFO data count register_5	SCFDR_5	16	H'FFFE881C	16
	Serial port register_5	SCSPTR_5	16	H'FFFE8820	16
	Line status register_5	SCLSR_5	16	H'FFFE8824	16
	Serial direction control register_5	SCSDCR_5	8	H'FFFE8902	8
	FIFO trigger control register_5	SCFTCR_5	16	H'FFFE8904	16
SCIF (channel 6)	Serial mode register_6	SCSMR_6	16	H'FFFE9000	16
	Bit rate register_6	SCBRR_6	8	H'FFFE9004	8
	Serial control register_6	SCSCR_6	16	H'FFFE9008	16
	Transmit FIFO data register_6	SCFTDR_6	8	H'FFFE900C	8
	Serial status register_6	SCFSR_6	16	H'FFFE9010	16
	Receive FIFO data register_6	SCFRDR_6	8	H'FFFE9014	8
	FIFO control register_6	SCFCR_6	16	H'FFFE9018	16
	FIFO data count register_6	SCFDR_6	16	H'FFFE901C	16
	Serial port register_6	SCSPTR_6	16	H'FFFE9020	16
	Line status register_6	SCLSR_6	16	H'FFFE9024	16
	Serial direction control register_6	SCSDCR_6	8	H'FFFE9102	8
	FIFO trigger control register_6	SCFTCR_6	16	H'FFFE9104	16
SCIF (channel 7)	Serial mode register_7	SCSMR_7	16	H'FFFE9800	16
	Bit rate register_7	SCBRR_7	8	H'FFFE9804	8
	Serial control register_7	SCSCR_7	16	H'FFFE9808	16
	Transmit FIFO data register_7	SCFTDR_7	8	H'FFFE980C	8
	Serial status register_7	SCFSR_7	16	H'FFFE9810	16
	Receive FIFO data register_7	SCFRDR_7	8	H'FFFE9814	8
	FIFO control register_7	SCFCR_7	16	H'FFFE9818	16
	FIFO data count register_7	SCFDR_7	16	H'FFFE981C	16
	Serial port register_7	SCSPTR_7	16	H'FFFE9820	16
	Line status register_7	SCLSR_7	16	H'FFFE9824	16
	Serial direction control register_7	SCSDCR_7	8	H'FFFE9902	8
	FIFO trigger control register_7	SCFTCR_7	16	H'FFFE9904	16
IIC3	I^2C bus control register 1	ICCR1	8	H'FFFEE000	8
	I^2C bus control register 2	ICCR2	8	H'FFFEE001	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
IIC3	I²C bus mode register	ICMR	8	H'FFFEE002	8
	I²C bus interrupt enable register	ICIER	8	H'FFFEE003	8
	I²C bus status register	ICSR	8	H'FFFEE004	8
	Slave address register	SAR	8	H'FFFEE005	8
	I²C bus transmit data register	ICDRT	8	H'FFFEE006	8
	I²C bus receive data register	ICDRR	8	H'FFFEE007	8
	NF2CYC register	NF2CYC	8	H'FFFEE008	8
ADC	A/D data register 0	ADDR0	16	H'FFFE5800	16
	A/D data register 1	ADDR1	16	H'FFFE5802	16
	A/D data register 2	ADDR2	16	H'FFFE5804	16
	A/D data register 3	ADDR3	16	H'FFFE5806	16
	A/D data register 4	ADDR4	16	H'FFFE5808	16
	A/D data register 5	ADDR5	16	H'FFFE580A	16
	A/D data register 6	ADDR6	16	H'FFFE580C	16
	A/D data register 7	ADDR7	16	H'FFFE580E	16
	A/D control/status register_0	ADCSR_0	16	H'FFFE5810	16
	A/D control register_0	ADCR_0	16	H'FFFE5812	16
	A/D data register 8	ADDR8	16	H'FFFE5900	16
	A/D data register 9	ADDR9	16	H'FFFE5902	16
	A/D data register 10	ADDR10	16	H'FFFE5904	16
	A/D data register 11	ADDR11	16	H'FFFE5906	16
	A/D data register 12	ADDR12	16	H'FFFE5908	16
	A/D data register 13	ADDR13	16	H'FFFE590A	16
	A/D data register 14	ADDR14	16	H'FFFE590C	16
	A/D data register 15	ADDR15	16	H'FFFE590E	16
	A/D control/status register_1	ADCSR_1	16	H'FFFE5910	16
	A/D control register_1	ADCR_1	16	H'FFFE5912	16
	A/D shadow data register	ADSDR	32	H'FFFE5B00	32
	A/D shadow select register	ADSSR	16	H'FFFE5B04	16
	A/D trigger select register_0	ADTSR_0	16	H'FFFE5B10	16
PFC	Port A IO register H	PAIORH	16	H'FFFE3804	8, 16
	Port A IO register L	PAIORL	16	H'FFFE3806	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port A control register H1	PACRH1	16	H'FFFE380A	8, 16
	Port A control register L2	PACRL2	16	H'FFFE380C	8, 16
	Port A control register L1	PACRL1	16	H'FFFE380E	8, 16
	Port A pull-up MOS control register H	PAPCRH	16	H'FFFE3820	8, 16
	Port A pull-up MOS control register L	PAPCRL	16	H'FFFE3822	8, 16
	Port B IO register L	PBIORL	16	H'FFFE3886	8, 16
	Port B control register L2	PBCRL2	16	H'FFFE388C	8, 16
	Port B control register L1	PBCRL1	16	H'FFFE388E	8, 16
	Port B pull-up MOS control register L	PBPCRL	16	H'FFFE38A2	8, 16
	Port C IO register L	PCIORL	16	H'FFFE3906	8, 16
	Port C control register L2	PCCRL2	16	H'FFFE390C	8, 16
	Port C control register L1	PCCRL1	16	H'FFFE390E	8, 16
	Port C pull-up MOS control register L	PCPCRL	16	H'FFFE3922	8, 16
	Port D IO register H	PDIORH	16	H'FFFE3984	8, 16
	Port D IO register L	PDIORL	16	H'FFFE3986	8, 16
	Port D control register H2	PDCRH2	16	H'FFFE3988	8, 16
	Port D control register H1	PDCRH1	16	H'FFFE398A	8, 16
	Port D control register L2	PDCRL2	16	H'FFFE398C	8, 16
	Port D control register L1	PDCRL1	16	H'FFFE398E	8, 16
	Port D pull-up MOS control register H	PDPCRH	16	H'FFFE39A0	8, 16
	Port D pull-up MOS control register L	PDPCRL	16	H'FFFE39A2	8, 16
	Port E IO register H	PEIORH	16	H'FFFE3A04	8, 16
	Port E IO register L	PEIORL	16	H'FFFE3A06	8, 16
	Port E control register H1	PECRH1	16	H'FFFE3A0A	8, 16
	Port E control register L2	PECRL2	16	H'FFFE3A0C	8, 16
	Port E control register L1	PECRL1	16	H'FFFE3A0E	8, 16
	Port E pull-up MOS control register H	PEPCRH	16	H'FFFE3A20	8, 16
	Port E pull-up MOS control register L	PEPCRL	16	H'FFFE3A22	8, 16
	Port G I/O register L	PGIORL	16	H'FFFE3B06	8, 16
	Port G control register L1	PGCRL1	16	H'FFFE3B0E	8, 16
	Port G pull-up MOS control register L	PGPCRL	16	H'FFFE3B22	8, 16
	Port H I/O register L	PHIORL	16	H'FFFE3B86	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port H control register L1	PHCRL1	16	H'FFFE3B8E	8, 16
	Port H pull-up MOS control register L	PHPCRL	16	H'FFFE3BA2	8, 16
	Port J I/O register L	PJIORL	16	H'FFFE3C86	8, 16
	Port J control register L1	PJCRL1	16	H'FFFE3C8E	8, 16
	Port J pull-up MOS control register L	PJPCRL	16	H'FFFE3CA2	8, 16
	Port K I/O register L	PKIORL	16	H'FFFE3D06	8, 16
	Port K control register L1	PKCRL1	16	H'FFFE3D0E	8, 16
	Port K pull-up MOS control register L	PKPCRL	16	H'FFFE3D22	8, 16
	Port L pull-up MOS control register L	PLPCRL	16	H'FFFE3DA2	8, 16
	Large current port control register	HCPCR	16	H'FFFE3A14	8, 16
	I/O buffer driver control register	DRVCR	16	H'FFFE39A8	8, 16
	Port function extension register	PFEXCR	16	H'FFFE3BA8	8, 16
I/O port	Port A data register H	PADRH	16	H'FFFE3800	8, 16
	Port A data register L	PADRL	16	H'FFFE3802	8, 16
	Port A port register H	PAPRH	16	H'FFFE381C	8, 16
	Port A port register L	PAPRL	16	H'FFFE381E	8, 16
	Port B data register L	PBDRL	16	H'FFFE3882	8, 16
	Port B port register L	PBPRL	16	H'FFFE389E	8, 16
	Port C data register L	PCDRL	16	H'FFFE3902	8, 16
	Port C port register L	PCPRL	16	H'FFFE391E	8, 16
	Port D data register H	PDDRH	16	H'FFFE3980	8, 16
	Port D data register L	PDDRL	16	H'FFFE3982	8, 16
	Port D port register H	PDPRH	16	H'FFFE399C	8, 16
	Port D port register L	PDPRL	16	H'FFFE399E	8, 16
	Port E data register H	PEDRH	16	H'FFFE3A00	8, 16
	Port E data register L	PEDRL	16	H'FFFE3A02	8, 16
	Port E port register L	PEPRL	16	H'FFFE3A1E	8, 16
	Port F data register L	PFDRL	16	H'FFFE3A82	8, 16
	Port G data register L	PGDRL	16	H'FFFE3B02	8, 16
	Port G port register L	PGPRL	16	H'FFFE3B12	8, 16
	Port H data register L	PHDRL	16	H'FFFE3B82	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
I/O port	Port H port register L	PHPRL	16	H'FFFE3B92	8, 16
	Port J data register L	PJDRL	16	H'FFFE3C82	8, 16
	Port J port register L	PJPRL	16	H'FFFE3C92	8, 16
	Port K data register L	PKDR	16	H'FFFE3D02	8, 16
	Port K port register L	PKPRL	16	H'FFFE3D12	8, 16
	Port L data register L	PLDRL	16	H'FFFE3D82	8, 16
LVDS (SH72315A only)	LVDS receive FIFO data register	LVFRDR	32	H'FFFEB000	32
	LVDS control register	LVCR	16	H'FFFEB004	16
	LVDS status register	LVSR	16	H'FFFEB006	16
	FIFO control register	LVFCR	16	H'FFFEB008	16
	FIFO data count register	LVFDR	16	H'FFFEB00A	16
	Control code register 1	LVCCR1	16	H'FFFEB00C	16
	Control code register 2	LVCCR2	16	H'FFFEB00E	16
	Line number count register	LVLCNT	16	H'FFFEB010	16
	Frame number count register	LVDCNT	16	H'FFFEB012	16
RSPI	RSPI control register	SPCR	8	H'FFFFB800	8, 16
	RSPI slave select polarity register	SSLP	8	H'FFFFB801	8
	RSPI pin control register	SPPCR	8	H'FFFFB802	8, 16
	RSPI status register	SPSR	8	H'FFFFB803	8
	RSPI data register	SPDR	32	H'FFFFB804	16, 32
	RSPI sequence control register	SPSCR	8	H'FFFFB808	8, 16
	RSPI sequence status register	SPSSR	8	H'FFFFB809	8
	RSPI bit rate register	SPBR	8	H'FFFFB80A	8, 16
	RSPI data control register	SPDCR	8	H'FFFFB80B	8
	RSPI clock delay register	SPCKD	8	H'FFFFB80C	8, 16
	RSPI slave select negation delay register	SSLND	8	H'FFFFB80D	8
	RSPI next-access delay register	SPND	8	H'FFFFB80E	8
	RSPI command register 0	SPCMD0	16	H'FFFFB810	16
	RSPI command register 1	SPCMD1	16	H'FFFFB812	16
	RSPI command register 2	SPCMD2	16	H'FFFFB814	16
	RSPI command register 3	SPCMD3	16	H'FFFFB816	16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	Master control register		MCR	16	H'FFFFD000	16
	General status register		GSR	16	H'FFFFD002	16
	Bit configuration register 1		BCR1	16	H'FFFFD004	16
	Bit configuration register 0		BCR0	16	H'FFFFD006	16
	Interrupt request register		IRR	16	H'FFFFD008	16
	Interrupt mask register		IMR	16	H'FFFFD00A	16
	Error counter register		TEC/REC	16	H'FFFFD00C	16
	Transmit pending 1, 0		TXPR1, 0	32	H'FFFFD020	32
	Transmit cancel 0		TXCR0	16	H'FFFFD02A	16
	Transmit acknowledge 0		TXACK0	16	H'FFFFD032	16
	Abort acknowledge 0		ABACK0	16	H'FFFFD03A	16
	Data frame receive pending 0		RXPR0	16	H'FFFFD042	16
	Remote frame receive pending 0		RFPR0	16	H'FFFFD04A	16
	Mailbox interrupt mask register 0		MBIMR0	16	H'FFFFD052	16
	Unread message status register 0		UMSR0	16	H'FFFFD05A	16
	MB[0].	CONTROL0H	—	16	H'FFFFD100	16, 32
		CONTROL0L	—	16	H'FFFFD102	16
		LAFMH	—	16	H'FFFFD104	16, 32
		LAFML	—	16	H'FFFFD106	16
		MSG_DATA[0]	—	8	H'FFFFD108	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD109	8
		MSG_DATA[2]	—	8	H'FFFFD10A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD10B	8
		MSG_DATA[4]	—	8	H'FFFFD10C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD10D	8
		MSG_DATA[6]	—	8	H'FFFFD10E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD10F	8
		CONTROL1H	—	8	H'FFFFD110	8, 16
		CONTROL1L	—	8	H'FFFFD111	8
	MB[1].	CONTROL0H	—	16	H'FFFFD120	16, 32
		CONTROL0L	—	16	H'FFFFD122	16
		LAFMH	—	16	H'FFFFD124	16, 32

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[1].	LAFML	—	16	H'FFFFD126	16
		MSG_DATA[0]	—	8	H'FFFFD128	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD129	8
		MSG_DATA[2]	—	8	H'FFFFD12A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD12B	8
		MSG_DATA[4]	—	8	H'FFFFD12C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD12D	8
		MSG_DATA[6]	—	8	H'FFFFD12E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD12F	8
		CONTROL1H	—	8	H'FFFFD130	8, 16
		CONTROL1L	—	8	H'FFFFD131	8
	MB[2].	CONTROL0H	—	16	H'FFFFD140	16, 32
		CONTROL0L	—	16	H'FFFFD142	16
		LAFMH	—	16	H'FFFFD144	16, 32
		LAFML	—	16	H'FFFFD146	16
	MB[2].	MSG_DATA[0]	—	8	H'FFFFD148	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD149	8
		MSG_DATA[2]	—	8	H'FFFFD14A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD14B	8
		MSG_DATA[4]	—	8	H'FFFFD14C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD14D	8
		MSG_DATA[6]	—	8	H'FFFFD14E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD14F	8
		CONTROL1H	—	8	H'FFFFD150	8, 16
		CONTROL1L	—	8	H'FFFFD151	8
	MB[3].	CONTROL0H	—	16	H'FFFFD160	16, 32
		CONTROL0L	—	16	H'FFFFD162	16
		LAFMH	—	16	H'FFFFD164	16, 32
		LAFML	—	16	H'FFFFD166	16
		MSG_DATA[0]	—	8	H'FFFFD168	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD169	8
		MSG_DATA[2]	—	8	H'FFFFD16A	8, 16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[3].	MSG_DATA[3]	—	8	H'FFFFD16B	8
		MSG_DATA[4]	—	8	H'FFFFD16C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD16D	8
		MSG_DATA[6]	—	8	H'FFFFD16E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD16F	8
		CONTROL1H	—	8	H'FFFFD170	8, 16
		CONTROL1L	—	8	H'FFFFD171	8
	MB[4].	CONTROL0H	—	16	H'FFFFD180	16, 32
		CONTROL0L	—	16	H'FFFFD182	16
		LAFMH	—	16	H'FFFFD184	16, 32
		LAFML	—	16	H'FFFFD186	16
		MSG_DATA[0]	—	8	H'FFFFD188	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD189	8
		MSG_DATA[2]	—	8	H'FFFFD18A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD18B	8
		MSG_DATA[4]	—	8	H'FFFFD18C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD18D	8
		MSG_DATA[6]	—	8	H'FFFFD18E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD18F	8
		CONTROL1H	—	8	H'FFFFD190	8, 16
		CONTROL1L	—	8	H'FFFFD191	8
	MB[5].	CONTROL0H	—	16	H'FFFFD1A0	16, 32
		CONTROL0L	—	16	H'FFFFD1A2	16
		LAFMH	—	16	H'FFFFD1A4	16, 32
		LAFML	—	16	H'FFFFD1A6	16
		MSG_DATA[0]	—	8	H'FFFFD1A8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD1A9	8
		MSG_DATA[2]	—	8	H'FFFFD1AA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD1AB	8
		MSG_DATA[4]	—	8	H'FFFFD1AC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD1AD	8
		MSG_DATA[6]	—	8	H'FFFFD1AE	8, 16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[5].	MSG_DATA[7]	—	8	H'FFFFD1AF	8
		CONTROL1H	—	8	H'FFFFD1B0	8, 16
		CONTROL1L	—	8	H'FFFFD1B1	8
	MB[6].	CONTROL0H	—	16	H'FFFFD1C0	16, 32
		CONTROL0L	—	16	H'FFFFD1C2	16
		LAFMH	—	16	H'FFFFD1C4	16, 32
		LAFML	—	16	H'FFFFD1C6	16
		MSG_DATA[0]	—	8	H'FFFFD1C8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD1C9	8
		MSG_DATA[2]	—	8	H'FFFFD1CA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD1CB	8
		MSG_DATA[4]	—	8	H'FFFFD1CC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD1CD	8
		MSG_DATA[6]	—	8	H'FFFFD1CE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD1CF	8
		CONTROL1H	—	8	H'FFFFD1D0	8, 16
		CONTROL1L	—	8	H'FFFFD1D1	8
	MB[7].	CONTROL0H	—	16	H'FFFFD1E0	16, 32
		CONTROL0L	—	16	H'FFFFD1E2	16
		LAFMH	—	16	H'FFFFD1E4	16, 32
		LAFML	—	16	H'FFFFD1E6	16
		MSG_DATA[0]	—	8	H'FFFFD1E8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD1E9	8
		MSG_DATA[2]	—	8	H'FFFFD1EA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD1EB	8
		MSG_DATA[4]	—	8	H'FFFFD1EC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD1ED	8
		MSG_DATA[6]	—	8	H'FFFFD1EE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD1EF	8
		CONTROL1H	—	8	H'FFFFD1F0	8, 16
		CONTROL1L	—	8	H'FFFFD1F1	8

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[8].	CONTROL0H	—	16	H'FFFFD200	16, 32
		CONTROL0L	—	16	H'FFFFD202	16
		LAFMH	—	16	H'FFFFD204	16, 32
		LAFML	—	16	H'FFFFD206	16
		MSG_DATA[0]	—	8	H'FFFFD208	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD209	8
		MSG_DATA[2]	—	8	H'FFFFD20A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD20B	8
		MSG_DATA[4]	—	8	H'FFFFD20C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD20D	8
		MSG_DATA[6]	—	8	H'FFFFD20E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD20F	8
		CONTROL1H	—	8	H'FFFFD210	8, 16
		CONTROL1L	—	8	H'FFFFD211	8
	MB[9].	CONTROL0H	—	16	H'FFFFD220	16, 32
		CONTROL0L	—	16	H'FFFFD222	16
		LAFMH	—	16	H'FFFFD224	16, 32
		LAFML	—	16	H'FFFFD226	16
		MSG_DATA[0]	—	8	H'FFFFD228	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD229	8
		MSG_DATA[2]	—	8	H'FFFFD22A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD22B	8
		MSG_DATA[4]	—	8	H'FFFFD22C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD22D	8
		MSG_DATA[6]	—	8	H'FFFFD22E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD22F	8
		CONTROL1H	—	8	H'FFFFD230	8, 16
		CONTROL1L	—	8	H'FFFFD231	8
	MB[10].	CONTROL0H	—	16	H'FFFFD240	16, 32
		CONTROL0L	—	16	H'FFFFD242	16
		LAFMH	—	16	H'FFFFD244	16, 32
		LAFML	—	16	H'FFFFD246	16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[10].	MSG_DATA[0]	—	8	H'FFFFD248	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD249	8
		MSG_DATA[2]	—	8	H'FFFFD24A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD24B	8
		MSG_DATA[4]	—	8	H'FFFFD24C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD24D	8
		MSG_DATA[6]	—	8	H'FFFFD24E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD24F	8
		CONTROL1H	—	8	H'FFFFD250	8, 16
		CONTROL1L	—	8	H'FFFFD251	8
	MB[11].	CONTROL0H	—	16	H'FFFFD260	16, 32
		CONTROL0L	—	16	H'FFFFD262	16
		LAFMH	—	16	H'FFFFD264	16, 32
		LAFML	—	16	H'FFFFD266	16
		MSG_DATA[0]	—	8	H'FFFFD268	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD269	8
		MSG_DATA[2]	—	8	H'FFFFD26A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD26B	8
		MSG_DATA[4]	—	8	H'FFFFD26C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD26D	8
		MSG_DATA[6]	—	8	H'FFFFD26E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD26F	8
		CONTROL1H	—	8	H'FFFFD270	8, 16
		CONTROL1L	—	8	H'FFFFD271	8
	MB[12].	CONTROL0H	—	16	H'FFFFD280	16, 32
		CONTROL0L	—	16	H'FFFFD282	16
		LAFMH	—	16	H'FFFFD284	16, 32
		LAFML	—	16	H'FFFFD286	16
		MSG_DATA[0]	—	8	H'FFFFD288	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD289	8
		MSG_DATA[2]	—	8	H'FFFFD28A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD28B	8

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[12].	MSG_DATA[4]	—	8	H'FFFFD28C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD28D	8
		MSG_DATA[6]	—	8	H'FFFFD28E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD28F	8
		CONTROL1H	—	8	H'FFFFD290	8, 16
		CONTROL1L	—	8	H'FFFFD291	8
	MB[13].	CONTROL0H	—	16	H'FFFFD2A0	16, 32
		CONTROL0L	—	16	H'FFFFD2A2	16
		LAFMH	—	16	H'FFFFD2A4	16, 32
		LAFML	—	16	H'FFFFD2A6	16
		MSG_DATA[0]	—	8	H'FFFFD2A8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD2A9	8
		MSG_DATA[2]	—	8	H'FFFFD2AA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD2AB	8
		MSG_DATA[4]	—	8	H'FFFFD2AC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD2AD	8
		MSG_DATA[6]	—	8	H'FFFFD2AE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD2AF	8
		CONTROL1H	—	8	H'FFFFD2B0	8, 16
		CONTROL1L	—	8	H'FFFFD2B1	8
	MB[14].	CONTROL0H	—	16	H'FFFFD2C0	16, 32
		CONTROL0L	—	16	H'FFFFD2C2	16
		LAFMH	—	16	H'FFFFD2C4	16, 32
		LAFML	—	16	H'FFFFD2C6	16
		MSG_DATA[0]	—	8	H'FFFFD2C8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD2C9	8
		MSG_DATA[2]	—	8	H'FFFFD2CA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD2CB	8
		MSG_DATA[4]	—	8	H'FFFFD2CC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD2CD	8
		MSG_DATA[6]	—	8	H'FFFFD2CE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD2CF	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[14]. CONTROL1H	—	8	H'FFFFD2D0	8, 16
	CONTROL1L	—	8	H'FFFFD2D1	8
	MB[15]. CONTROL0H	—	16	H'FFFFD2E0	16, 32
	CONTROL0L	—	16	H'FFFFD2E2	16
	LAFMH	—	16	H'FFFFD2E4	16, 32
	LAFML	—	16	H'FFFFD2E6	16
	MSG_DATA[0]	—	8	H'FFFFD2E8	8, 16, 32
	MSG_DATA[1]	—	8	H'FFFFD2E9	8
	MSG_DATA[2]	—	8	H'FFFFD2EA	8, 16
	MSG_DATA[3]	—	8	H'FFFFD2EB	8
	MSG_DATA[4]	—	8	H'FFFFD2EC	8, 16, 32
	MSG_DATA[5]	—	8	H'FFFFD2ED	8
	MSG_DATA[6]	—	8	H'FFFFD2EE	8, 16
	MSG_DATA[7]	—	8	H'FFFFD2EF	8
	CONTROL1H	—	8	H'FFFFD2F0	8, 16
	CONTROL1L	—	8	H'FFFFD2F1	8
TIM32C	Timer 32 control register_0	TI32CR_0	8	H'FFFEC200	8
	Timer 32 control register_1	TI32CR_1	8	H'FFFEC204	8
	Timer 32 control register_2	TI32CR_2	8	H'FFFEC208	8
	Timer 32 status register	TI32SR	8	H'FFFEC20C	8
	Timer 32 interrupt enable register	TI32IER	8	H'FFFEC210	8
	Timer 32 counter 8_0	TI32CNT8_0	8	H'FFFEC214	8
	Timer 32 counter 8_1	TI32CNT8_1	8	H'FFFEC218	8
	Timer 32 counter 16	TI32CNT16	16	H'FFFEC21C	16
	Timer 32 constant register_2	TI32COR_2	16	H'FFFEC220	16
	Timer 32 start register	TI32STR	8	H'FFFEC224	8
	Timer 32 state monitor register	TI32SMR	8	H'FFFEC280	8
KEYC	Key scan control register 1	KSCR1	16	H'FFFEC300	16
	Key scan control register 2	KSCR2	16	H'FFFEC302	16
	Key scan function control register	KSFCR	8	H'FFFEC304	8
	Key scan data register_0	KSDR_0	8	H'FFFEC310	8
	Key scan data register_1	KSDR_1	8	H'FFFEC311	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
KEYC	Key scan data register_2	KSDR_2	8	H'FFFEC312	8
	Key scan data register_3	KSDR_3	8	H'FFFEC313	8
	Key scan data register_4	KSDR_4	8	H'FFFEC314	8
	Key scan data register_5	KSDR_5	8	H'FFFEC315	8
	Key scan data register_6	KSDR_6	8	H'FFFEC316	8
	Key scan data register_7	KSDR_7	8	H'FFFEC317	8
	Key scan data register_8	KSDR_8	16	H'FFFEC318	16
	Key scan interrupt enable register	KSIER	8	H'FFFEC320	8
	Key scan status register	KSSR	8	H'FFFEC322	8
	Key scan counter control register	KSCCR	8	H'FFFEC324	8
	Key scan counter	KSCNT	8	H'FFFEC326	8
	Key scan counter start register	KSCSR	8	H'FFFEC328	8
ROM/FLD	Flash pin monitor register	FPMON	8	H'FFFFA800	8
	Flash mode register	FMODR	8	H'FFFFA802	8
	Flash access status register	FASTAT	8	H'FFFFA810	8
	Flash access error interrupt enable register	FAEINT	8	H'FFFFA811	8
	ROM MAT select register	ROMMAT	16	H'FFFFA820	8, 16
	FCU RAM enable register	FCURAME	16	H'FFFFA854	8, 16
	Flash status register 0	FSTATR0	8	H'FFFFA900	8, 16
	Flash status register 1	FSTATR1	8	H'FFFFA901	8
	Flash P/E mode entry register	FENTRYR	16	H'FFFFA902	8, 16
	Flash protect register	FPROTR	16	H'FFFFA904	8, 16
	Flash reset register	FRESETR	16	H'FFFFA906	8, 16
	FCU command register	FCMDR	16	H'FFFFA90A	8, 16
	FCU processing switch register	FCPSR	16	H'FFFFA918	8, 16
	FLD blank check control register	EEPBCCNT	16	H'FFFFA91A	8, 16
	Flash P/E status register	FPESTAT	16	H'FFFFA91C	8, 16
	FLD blank check status register	EEPBCSTAT	16	H'FFFFA91E	8, 16
	FLD read enable register 0	EEPRE0	16	H'FFFFA840	8, 16
	FLD program/erase enable register 0	EEPWE0	16	H'FFFFA850	8, 16
	Peripheral clock notification register	PCKAR	16	H'FFFFA938	8, 16
	Erasure block notification register	FIEBAR	16	H'FFFFA93A	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Power-down mode	Standby control register	STBCR	8	H'FFFE0014	8
	Standby control register 2	STBCR2	8	H'FFFE0018	8
	Standby control register 3	STBCR3	8	H'FFFE0408	8
	Standby control register 4	STBCR4	8	H'FFFE040C	8
	Standby control register 5	STBCR5	8	H'FFFE0418	8
	Standby control register 6	STBCR6	8	H'FFFE041C	8
	Standby control register 7	STBCR7	8	H'FFFEC500	8
	System control register 1	SYSCR1	8	H'FFFE0402	8
	System control register 2	SYSCR2	8	H'FFFE0404	8
	Deep standby control register	DPSTBCR	8	H'FFFEC510	8
	Deep standby wait control register	DPSWCR	8	H'FFFEC512	8
	Standby interrupt enable register	SIER	16	H'FFFEC514	16
	Standby interrupt flag register	SIFR	16	H'FFFEC516	16
	Deep standby interrupt edge register	DPSIEGR	16	H'FFFEC518	16
	Reset status register	RSTSR	8	H'FFFEC51A	8
H-UDI	Instruction register	SDIR	16	H'FFFE2000	16
	ID register	SDID	32	H'FFFE1810	32

Note: * The access sizes of the WDT registers are different between the read and write to prevent incorrect writing.

34.2 Register Bits

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CPG	FRQCR	—	—	—	—	—	STC2	STC1	STC0
		—	IFC2	IFC1	IFC0	—	PFC2	PFC1	PFC0
	MCLKCR	MSSCS1	MSSCS0	—	—	—	MSDIVS2	MSDIVS1	MSDIVS0
	ACLKCR	ASSCS1	ASSCS0	—	—	—	ASDIVS2	ASDIVS1	ASDIVS0
	OSCCR	—	—	—	—	—	OSCSTOP	—	OSCERS
INTC	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	ICR2	IRQ151S	IRQ150S	IRQ141S	IRQ140S	IRQ131S	IRQ130S	IRQ121S	IRQ120S
		IRQ111S	IRQ110S	IRQ101S	IRQ100S	IRQ91S	IRQ90S	IRQ81S	IRQ80S
	ICR3	IRQ231S	IRQ230S	IRQ221S	IRQ220S	IRQ211S	IRQ210S	IRQ201S	IRQ200S
		IRQ191S	IRQ190S	IRQ181S	IRQ180S	IRQ171S	IRQ170S	IRQ161S	IRQ160S
	IRQRR0	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	IRQRR1	—	—	—	—	—	—	—	—
		IRQ23F	IRQ22F	IRQ21F	IRQ20F	IRQ19F	IRQ18F	IRQ17F	IRQ16F
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE1	BE0	BOVE	—	—	—	—	—
		—	—	—	—	BN3	BN2	BN1	BN0
	IPR01								
	IPR02								
	IPR03								
	IPR04								
	IPR06								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
INTC	IPR08								
	IPR09								
	IPR10								
	IPR11								
	IPR12								
	IPR13								
	IPR14								
	IPR15								
	IPR16								
	IPR17								
	IPR18								
UBC	BAR_0	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BAMR_0	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24
		BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
		BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8
		BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
	BBR_0	—	—	UBID	—	—	CP2	CP1	CP0
		CD1	CD0	ID1	ID0	RW1	RW0	SZ1	SZ0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
UBC	BAR_1	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BAMR_1	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24
		BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
		BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8
		BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
	BBR_1	—	—	UBID	—	—	CP2	CP1	CP0
		CD1	CD0	ID1	ID0	RW1	RW0	SZ1	SZ0
	BAR_2	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BBR_2	—	—	UBID	—	—	—	—	—
		CD1	CD0	ID1	ID0	—	—	—	—
	BAR_3	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BBR_3	—	—	UBID	—	—	—	—	—
		CD1	CD0	ID1	ID0	—	—	—	—
	BAR_4	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BBR_4	—	—	UBID	—	—	—	—	—
		CD1	CD0	ID1	ID0	—	—	—	—
	BAR_5	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BBR_5	—	—	UBID	—	—	—	—	—
		CD1	CD0	ID1	ID0	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
UBC	BAR_6	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BBR_6	—	—	UBID	—	—	—	—	—
		CD1	CD0	ID1	ID0	—	—	—	—
	BAR_7	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BBR_7	—	—	UBID	—	—	—	—	—
		CD1	CD0	ID1	ID0	—	—	—	—
	BRCR	—	—	—	—	—	—	SCMFD1	SCMFD0
		—	—	—	—	—	—	CKS1	CKS0
		SCMFC7	SCMFC6	SCMFC5	SCMFC4	SCMFC3	SCMFC2	SCMFC1	SCMFC0
		PCB7	PCB6	PCB5	PCB4	PCB3	PCB2	PCB1	PCB0
DTC	DTCERA	DTCERA15	DTCERA14	DTCERA13	DTCERA12	DTCERA11	DTCERA10	DTCERA9	DTCERA8
		DTCERA7	DTCERA6	—	—	DTCERA3	DTCERA2	—	—
	DTCERB	DTCERB15	DTCERB14	DTCERB13	DTCERB12	DTCERB11	DTCERB10	DTCERB9	DTCERB8
		DTCERB7	DTCERB6	DTCERB5	DTCERB4	DTCERB3	DTCERB2	DTCERB1	DTCERB0
	DTCERC	DTCERC15	DTCERC14	DTCERC13	DTCERC12	—	—	—	—
		—	—	—	—	DTCERC3	DTCERC2	DTCERC1	DTCERC0
	DTCERD	DTCERD15	DTCERD14	DTCERD13	DTCERD12	DTCERD11	DTCERD10	DTCERD9	DTCERD8
		DTCERD7	DTCERD6	—	—	—	—	—	—
	DTCERE	DTCERE15	DTCERE14	DTCERE13	DTCERE12	DTCERE11	DTCERE10	DTCERE9	DTCERE8
		—	—	—	DTCERE4	DTCERE3	DTCERE2	DTCERE1	DTCERE0
	DTCERF	DTCERF15	DTCERF14	DTCERF13	DTCERF12	DTCERF11	DTCERF10	DTCERF9	DTCERF8
		DTCERF7	DTCERF6	DTCERF5	DTCERF4	DTCERF3	DTCERF2	DTCERF1	DTCERF0
	DTCERG	DTCERG15	DTCERG14	DTCERG13	DTCERG12	DTCERG11	DTCERG10	DTCERG9	DTCERG8
		—	—	DTCERG5	—	—	—	—	—
	DTCCR	—	—	—	RRS	RCHNE	—	—	ERR

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DTC	DTCVBR								
						—	—	—	—
		—	—	—	—	—	—	—	—
BSC	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY1	DPRTY0	DMAIW2
		DMAIW1	DMAIW0	DMAIWA	—	—	HIZCK	HIZMEM	HIZCNT
	CS0BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS1BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS2BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS3BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS4BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS5BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	CS6BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS7BCR	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2
		IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
		—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
		—	—	—	—	—	—	—	—
	CS0WCR (Normal space)	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	SW1	SW0	WR3	WR2	WR1
		WR0	WM	—	—	—	—	HW1	HW0
	CS0WCR (burst ROM (clock asynchronous))	—	—	—	—	—	—	—	—
		—	—	BST1	BST0	—	—	BW1	BW0
		—	—	—	—	—	W3	W2	W1
		W0	WM	—	—	—	—	—	—
	CS0WCR (burst ROM (clock synchronous))	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	BW1	BW0
		—	—	—	—	—	W3	W2	W1
		W0	WM	—	—	—	—	—	—
	CS1WCR (Normal space, SRAM with byte selection)	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW2	WW1	WW0
		—	—	—	SW1	SW0	WR3	WR2	WR1
		WR0	WM	—	—	—	—	HW1	HW0
	CS2WCR (Normal space, SRAM with byte selection)	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	WR3	WR2	WR1
		WR0	WM	—	—	—	—	—	—
	CS2WCR (SDRAM)	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	A2CL1
		A2CL0	—	—	—	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	CS3WCR (Normal space, SRAM with byte selection)	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR3	WR2	WR1
		WR0	WM	—	—	—	—	—	—
	CS3WCR (SDRAM)	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	WTRP1	WTRP0	—	WTRCD1	WTRCD0	—	A3CL1
		A3CL0	—	—	TRWL1	TRWL0	—	WTRC1	WTRC0
	CS4WCR (Normal space, SRAM with byte selection)	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW2	WW1	WW0
		—	—	—	SW1	SW0	WR3	WR2	WR1
		WR0	WM	—	—	—	—	HW1	HW0
	CS4WCR (burst ROM (clock asynchronous))	—	—	—	—	—	—	—	—
		—	—	BST1	BST0	—	—	BW1	BW0
		—	—	—	SW1	SW0	W3	W2	W1
		W0	WM	—	—	—	—	HW1	HW0
	CS5WCR (Normal space, SRAM with byte selection, MPX-I/O)	—	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW/BAS	—	WW2	WW1	WW0
		—	—	—	SW1	SW0	WR3	WR2	WR1
		WR0	WM	—	—	—	—	HW1	HW0
	CS6WCR (Normal space, SRAM with byte selection)	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW1	SW0	WR3	WR2	WR1
		WR0	WM	—	—	—	—	HW1	HW0
	CS7WCR (Normal space, SRAM with byte selction)	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW2	WW1	WW0
		—	—	—	SW1	SW0	WR3	WR2	WR1
		WR0	WM	—	—	—	—	HW1	HW0
	SDCR	—	—	—	—	—	—	—	—
		—	—	—	A2ROW1	A2ROW0	—	A2COL1	A2COL0
		—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV
		—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3COL0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	RTCSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0
	RTCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	RTCOR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	BSCEHR	DTLOCK	—	—	—	DTBST	DTSA	—	DTPR
		—	—	—	—	—	—	—	—
DMAC	SAR_0								
	DAR_0								
	DMATCR_0	—	—	—	—	—	—	—	—
	CHCR_0	TC	—	—	RLD	SARE	DARE	TCRE	—
		DO	TL	—	—	HE	HIE	AM	AL
		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
		DL	DS	TB	TS1	TS0	IE	TE	DE

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	RSAR_0								
	RDAR_0								
	RDMATCR_0	—	—	—	—	—	—	—	—
	SAR_1								
	DAR_1								
	DMATCR_1	—	—	—	—	—	—	—	—
	CHCR_1	TC	—	—	RLD	SARE	DARE	TCRE	—
		DO	TL	—	—	HE	HIE	AM	AL
		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
		DL	DS	TB	TS1	TS0	IE	TE	DE
	RSAR_1								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	RDAR_1								
	RDMATCR_1	—	—	—	—	—	—	—	—
	SAR_2								
	DAR_2								
	DMATCR_2	—	—	—	—	—	—	—	—
	CHCR_2	TC	—	—	RLD	SARE	DARE	TCRE	—
		—	—	—	—	HE	HIE	—	—
		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
		—	—	TB	TS1	TS0	IE	TE	DE
	RSAR_2								
	RDAR_2								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	RDMATCR_2	—	—	—	—	—	—	—	—
	SAR_3								
	DAR_3								
	DMATCR_3	—	—	—	—	—	—	—	—
	CHCR_3	TC	—	—	RLD	SARE	DARE	TCRE	—
		—	—	—	—	HE	HIE	—	—
		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
		—	—	TB	TS1	TS0	IE	TE	DE
	RSAR_3								
	RDAR_3								
	RDMATCR_3	—	—	—	—	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	DMAOR	—	—	CMS1	CMS0	—	—	PR1	PR0
		—	—	—	—	—	AE	NMIF	DME
	DMARS0	CH1 MID5	CH1 MID4	CH1 MID3	CH1 MID2	CH1 MID1	CH1 MID0	CH1 RID1	CH1 RID0
		CH0 MID5	CH0 MID4	CH0 MID3	CH0 MID2	CH0 MID1	CH0 MID0	CH0 RID1	CH0 RID0
	DMARS1	CH3 MID5	CH3 MID4	CH3 MID3	CH3 MID2	CH3 MID1	CH3 MID0	CH3 RID1	CH3 RID0
		CH2 MID5	CH2 MID4	CH2 MID3	CH2 MID2	CH2 MID1	CH2 MID0	CH2 RID1	CH2 RID0
MTU2	TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_0	—	BFE	BFB	BFA	MD3	MD2	MD1	MD0
	TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
	TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
	TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_0								
	TGRA_0								
	TGRB_0								
	TGRC_0								
	TGRD_0								
	TGRE_0								
	TGRF_0								
	TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE
	TSR2_0	—	—	—	—	—	—	TGFF	TGFE
	TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA
	TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0
	TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2	TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
	TCNT_1								
	TGRA_1								
	TGRB_1								
	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
	TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0
	TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
	TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
	TCNT_2								
	TGRA_2								
	TGRB_2								
	TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0
	TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
	TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
	TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3								
	TGRA_3								
	TGRB_3								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2	TGRC_3								
	TGRD_3								
	TBTM_3	—	—	—	—	—	—	TTSB	TTSA
	TCR_4	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_4	—	—	BFB	BFA	MD3	MD2	MD1	MD0
	TIORH_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
	TIORL_4	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
	TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4								
	TGRA_4								
	TGRB_4								
	TGRC_4								
	TGRD_4								
	TBTM_4	—	—	—	—	—	—	TTSB	TTSA
	TADCR	BF1	BF0	—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4								
	TADCORB_4								
	TADCOBRA_4								
	TADCOBRB_4								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2	TCRU_5	—	—	—	—	—	—	TPSC1	TPSC0
	TCRV_5	—	—	—	—	—	—	TPSC1	TPSC0
	TCRW_5	—	—	—	—	—	—	TPSC1	TPSC0
	TIORU_5	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0
	TIORV_5	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0
	TIORW_5	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0
	TIER_5	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
	TSR_5	—	—	—	—	—	CMFU5	CMFV5	CMFW5
	TSTR_5	—	—	—	—	—	CSTU5	CSTV5	CSTW5
	TCNTU_5								
	TCNTV_5								
	TCNTW_5								
	TGRU_5								
	TGRV_5								
	TGRW_5								
	TCNTCMPCLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S	SCH4S
	TRWER	—	—	—	—	—	—	—	RWE
	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
	TOCR2	BF1	BF0	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCR	—	BDC	N	P	FB	WF	VF	UF
	TCDR								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2	TDDR								
	TCNTS								
	TCBR								
	TITCR	T3AEN	3ACOR2	3ACOR1	3ACOR0	T4VEN	4VCOR2	4VCOR1	4VCOR0
	TITCNT	—	3ACNT2	3ACNT1	3ACNT0	—	4VCNT2	4VCNT1	4VCNT0
	TBTER	—	—	—	—	—	—	BTE1	BTE0
	TDER	—	—	—	—	—	—	—	TDER
	TWCR	CCE	—	—	—	—	—	—	WRE
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU2S	TCR_3S	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_3S	—	—	BFB	BFA	MD3	MD2	MD1	MD0
	TIORH_3S	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
	TIORL_3S	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
	TIER_3S	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3S								
	TGRA_3S								
	TGRB_3S								
	TGRC_3S								
	TGRD_3S								
	TBTM_3S	—	—	—	—	—	—	TTSB	TTSA
	TCR_4S	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
	TMDR_4S	—	—	BFB	BFA	MD3	MD2	MD1	MD0
	TIORH_4S	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2S	TIORL_4S	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
	TIER_4S	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4S								
	TGRA_4S								
	TGRB_4S								
	TGRC_4S								
	TGRD_4S								
	TBTM_4S	—	—	—	—	—	—	TTSB	TTSA
	TADCRS	BF1	BF0	—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4S								
	TADCORB_4S								
	TADCOBRA_4S								
	TADCOBRB_4S								
	TCRU_5S	—	—	—	—	—	—	TPSC1	TPSC0
	TCRV_5S	—	—	—	—	—	—	TPSC1	TPSC0
	TCRW_5S	—	—	—	—	—	—	TPSC1	TPSC0
	TIORU_5S	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0
	TIORV_5S	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0
	TIORW_5S	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0
	TIER_5S	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
	TSR_5S	—	—	—	—	—	CMFU5	CMFV5	CMFW5

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2S	TSTR_5S	—	—	—	—	—	CSTU5	CSTV5	CSTW5
	TCNTU_5S								
	TCNTV_5S								
	TCNTW_5S								
	TGRU_5S								
	TGRV_5S								
	TGRW_5S								
	TCNTCMPCLRS	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
	TSTRS	CST4	CST3	—	—	—	—	—	—
	TSYRS	SYNC4	SYNC3	—	—	—	—	—	—
	TRWERS	—	—	—	—	—	—	—	RWE
	TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1S	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
	TOCR2S	BF1	BF0	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCRS	—	BDC	N	P	FB	WF	VF	UF
	TCDRS								
	TDDRS								
	TCNTSS								
	TCBRS								
	TITCRS	T3AEN	3ACOR2	3ACOR1	3ACOR0	T4VEN	4VCOR2	4VCOR1	4VCOR0
	TITCNTS	—	3ACNT2	3ACNT1	3ACNT0	—	4VCNT2	4VCNT1	4VCNT0
	TBTERS	—	—	—	—	—	—	BTE1	BTE0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU2S	TDERS	—	—	—	—	—	—	—	TDER
	TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
	TWCRS	CCE	—	—	—	—	—	SCC	WRE
	TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
POE2	ICSR1	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1
		POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0
	OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1
		—	—	—	—	—	—	—	—
	ICSR2	POE7F	POE6F	POE5F	POE4F	—	—	—	PIE2
		POE7M1	POE7M0	POE6M1	POE6M0	POE5M1	POE5M0	POE4M1	POE4M0
	OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2
		—	—	—	—	—	—	—	—
	ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3
		—	—	—	—	—	—	POE8M1	POE8M0
	SPOER	—	—	—	—	—	MTU2SHIZ	MTU2CH0HIZ	MTU2CH34HIZ
	POECR1	—	—	—	—	MTU2PE3ZE	MTU2PE2ZE	MTU2PE1ZE	MTU2PE0ZE
	POECR2	—	MTU2P1CZE	MTU2P2CZE	MTU2P3CZE	—	MTU2SP1CZE	MTU2SP2CZE	MTU2SP3CZE
		—	—	—	—	—	—	—	—
CMT	CMSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCSR_0	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS1	CKS0
	CMCNT_0								
	CMCOR_0								
	CMCSR_1	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS1	CKS0
	CMCNT_1								
	CMCOR_1								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CMT2	CM2STR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	STR
	CM2CR	CCLR2	CCLR1	CCLR0	—	—	—	CMS	—
		OC1IE	OC0IE	IC1IE	IC0IE	CM2IE	—	CKS1	CKS0
	CM2IOR	CM2E	—	OC1E	OC0E	OC1[1]	OC1[0]	OC0[1]	OC0[0]
		—	—	IC1E	IC0E	IC1[1]	IC1[0]	IC0[1]	IC0[0]
	CM2SR	CM2F	OVF	—	—	OCF1	OCF0	ICF1	ICF0
		—	—	—	—	—	—	—	—
	CM2CNT								
	CM2COR								
	CM2ICR0								
	CM2ICR1								
	CM2OCR0								
	CM2OCR1								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
WDT	WTCSR	IOVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
	WTCNT								
	WRCSR	WOVF	RSTE	RSTS	—	—	—	—	—
SCI (channel 0)	SCSMR_0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
	SCBRR_0								
	SCSCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
	SCTDR_0								
	SCSSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_0								
	SCSDCR_0	—	—	—	—	DIR	—	—	—
	SCSPTR_0	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
	SCSMR2_0	—	BAE	SPSEL1	SPSEL0	—	—	—	—
	SCTBACNT_0	TBACNT7	TBACNT6	TBACNT5	TBACNT4	TBACNT3	TBACNT2	TBACNT1	TBACNT0
	SCRBACNT_0	RBACNT7	RBACNT6	RBACNT5	RBACNT4	RBACNT3	RBACNT2	RBACNT1	RBACNT0
	SCBACOR_0	BACOR7	BACOR6	BACOR5	BACOR4	BACOR3	BACOR2	BACOR1	BACOR0
SCI (channel 1)	SCSMR_1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
	SCBRR_1								
	SCSCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
	SCTDR_1								
	SCSSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_1								
	SCSDCR_1	—	—	—	—	DIR	—	—	—
	SCSPTR_1	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
	SCSMR2_1	—	BAE	SPSEL1	SPSEL0	—	—	—	—
	SCTBACNT_1	TBACNT7	TBACNT6	TBACNT5	TBACNT4	TBACNT3	TBACNT2	TBACNT1	TBACNT0
	SCRBACNT_1	RBACNT7	RBACNT6	RBACNT5	RBACNT4	RBACNT3	RBACNT2	RBACNT1	RBACNT0
	SCBACOR_1	BACOR7	BACOR6	BACOR5	BACOR4	BACOR3	BACOR2	BACOR1	BACOR0
SCI (channel 2)	SCSMR_2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
	SCBRR_2								
	SCSCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
	SCTDR_2								
	SCSSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SCI (channel 2)	SCRDR_2								
	SCSDCR_2	—	—	—	—	DIR	—	—	—
	SCSPTR_2	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
	SCSMR2_2	—	BAE	SPSEL1	SPSEL0	—	—	—	—
	SCTBACNT_2	TBACNT7	TBACNT6	TBACNT5	TBACNT4	TBACNT3	TBACNT2	TBACNT1	TBACNT0
	SCRBACNT_2	RBACNT7	RBACNT6	RBACNT5	RBACNT4	RBACNT3	RBACNT2	RBACNT1	RBACNT0
	SCBACOR_2	BACOR7	BACOR6	BACOR5	BACOR4	BACOR3	BACOR2	BACOR1	BACOR0
SCI (channel 3)	SCSMR_3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
	SCBRR_3								
	SCSCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
	SCTDR_3								
	SCSSR_3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_3								
	SCSDCR_3	—	—	—	—	DIR	—	—	—
	SCSPTR_3	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
	SCSMR2_3	—	BAE	SPSEL1	SPSEL0	—	—	—	—
	SCTBACNT_3	TBACNT7	TBACNT6	TBACNT5	TBACNT4	TBACNT3	TBACNT2	TBACNT1	TBACNT0
	SCRBACNT_3	RBACNT7	RBACNT6	RBACNT5	RBACNT4	RBACNT3	RBACNT2	RBACNT1	RBACNT0
	SCBACOR_3	BACOR7	BACOR6	BACOR5	BACOR4	BACOR3	BACOR2	BACOR1	BACOR0
SCIF (channel 4)	SCSMR_4	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
	SCBRR_4								
	SCSCR_4	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
	SCFTDR_4								
	SCFSR_4	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_4								
	SCFCR_4	—	—	—	—	—	—	—	—
		RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP
	SCFDR_4	—	—	—	T4	T3	T2	T1	T0
		—	—	—	R4	R3	R2	R1	R0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SCIF (channel 4)	SCSPTR_4	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSDCR_4	—	—	—	—	DIR	—	—	—
	SCFTCR_4	RTRGS	—	—	RFTC4	RFTC3	RFTC2	RFTC1	RFTC0
		TTRGS	—	—	TFTC4	TFTC3	TFTC2	TFTC1	TFTC0
SCIF (channel 5)	SCSMR_5	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
	SCBRR_5								
	SCSCR_5	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
	SCFTDR_5								
	SCFSR_5	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_5								
	SCFCR_5	—	—	—	—	—	—	—	—
		RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP
	SCFDR_5	—	—	—	T4	T3	T2	T1	T0
		—	—	—	R4	R3	R2	R1	R0
	SCSPTR_5	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSDCR_5	—	—	—	—	DIR	—	—	—
	SCFTCR_5	RTRGS	—	—	RFTC4	RFTC3	RFTC2	RFTC1	RFTC0
		TTRGS	—	—	TFTC4	TFTC3	TFTC2	TFTC1	TFTC0
SCIF (channel 6)	SCSMR_6	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
	SCBRR_6								
	SCSCR_6	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE1	CKE0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SCIF (channel 6)	SCFTDR_6								
	SCFSR_6	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_6								
	SCFCR_6	—	—	—	—	—	—	—	—
		RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP
	SCFDR_6	—	—	—	T4	T3	T2	T1	T0
		—	—	—	R4	R3	R2	R1	R0
	SCSPTR_6	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSDCR_6	—	—	—	—	DIR	—	—	—
	SCFTCR_6	RTRGS	—	—	RFTC4	RFTC3	RFTC2	RFTC1	RFTC0
		TTRGS	—	—	TFTC4	TFTC3	TFTC2	TFTC1	TFTC0
SCIF (channel 7)	SCSMR_7	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
	SCBRR_7								
	SCSCR_7	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
	SCFTDR_7								
	SCFSR_7	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_7								
	SCFCR_7	—	—	—	—	—	—	—	—
		RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP
	SCFDR_7	—	—	—	T4	T3	T2	T1	T0
		—	—	—	R4	R3	R2	R1	R0
	SCSPTR_7	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SCIF (channel 7)	SCSDCR_7	—	—	—	—	DIR	—	—	—
	SCFTCR_7	RTRGS	—	—	RFTC4	RFTC3	RFTC2	RFTC1	RFTC0
		TTRGS	—	—	TFTC4	TFTC3	TFTC2	TFTC1	TFTC0
IIC3	ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
	ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	RXTIMES	IICRST	—
	ICMR	MLS	—	—	—	BCWP	BC2	BC1	BC0
	ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
	ICDRT								
	ICDRR								
	NF2CYC	—	—	—	—	—	—	—	NF2CYC
ADC	ADDR0	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR1	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR2	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR3	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR4	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR5	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR6	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR7	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADCSR_0	ADF	ADIE	TRGE	—	CONADF	STC2	STC1	STC0
		CKS1	CKS0	ADM1	ADM0	ADCS	CH2	CH1	CH0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ADC	ADCR_0	—	—	ADST	—	—	—	—	—
		—	—	—	—	—	—	—	—
	ADDR8	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR9	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR10	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR11	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR12	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR13	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR14	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADDR15	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2
		ADD1	ADD0	—	—	—	—	—	—
	ADCSR_1	ADF	ADIE	TRGE	—	CONADF	STC2	STC1	STC0
		CKS1	CKS0	ADM1	ADM0	ADCS	CH2	CH1	CH0
	ADCR_1	—	—	ADST	—	—	—	—	—
		—	—	—	—	—	—	—	—
	ADSDR	ADSD0[15]	ADSD0[14]	ADSD0[13]	ADSD0[12]	ADSD0[11]	ADSD0[10]	ADSD0[9]	ADSD0[8]
		ADSD0[7]	ADSD0[6]	ADSD0[5]	ADSD0[4]	ADSD0[3]	ADSD0[2]	ADSD0[1]	ADSD0[0]
		ADSD1[15]	ADSD1[14]	ADSD1[13]	ADSD1[12]	ADSD1[11]	ADSD1[10]	ADSD1[9]	ADSD1[8]
		ADSD1[7]	ADSD1[6]	ADSD1[5]	ADSD1[4]	ADSD1[3]	ADSD1[2]	ADSD1[1]	ADSD1[0]
	ADSSR	—	—	AD0FCE	AD1FCE	—	—	—	ADDALS
		AD0SE	AD0SS2	AD0SS1	AD0SS0	AD1SE	AD1SS2	AD1SS1	AD1SS0
	ADTSR_0	TRG11S3	TRG11S2	TRG11S1	TRG11S0	TRG01S3	TRG01S2	TRG01S1	TRG01S0
		TRG1S3	TRG1S2	TRG1S1	TRG1S0	TRG0S3	TRG0S2	TRG0S1	TRG0S0
PFC	PAIORH	—	—	—	—	—	—	—	—
		—	—	—	—	PA19IOR	PA18IOR	PA17IOR	PA16IOR

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PFC	PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR
		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR
	PACRH1	—	—	—	—	—	—	—	—
		PA19MD1	PA19MD0	PA18MD1	PA18MD0	PA17MD1	PA17MD0	PA16MD1	PA16MD0
	PACRL2	PA15MD1	PA15MD0	PA14MD1	PA14MD0	PA13MD1	PA13MD0	PA12MD1	PA12MD0
		PA11MD1	PA11MD0	PA10MD1	PA10MD0	PA9MD1	PA9MD0	PA8MD1	PA8MD0
	PACRL1	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0
		PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0
	PAPCRH	—	—	—	—	—	—	—	—
		—	—	—	—	PA19PCR	PA18PCR	PA17PCR	PA16PCR
	PAPCRL	PA15PCR	PA14PCR	PA13PCR	PA12PCR	PA11PCR	PA10PCR	PA9PCR	PA8PCR
		PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR
	PBIORL	—	—	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
	PBCRL2	—	—	—	—	PB13MD1	PB13MD0	PB12MD1	PB12MD0
		PB11MD1	PB11MD0	PB10MD1	PB10MD0	PB9MD1	PB9MD0	PB8MD1	PB8MD0
	PBCRL1	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
		PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0
	PBPCRL	—	—	PB13PCR	PB12PCR	PB11PCR	PB10PCR	PB9PCR	PB8PCR
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
	PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCCRL2	PC15MD1	PC15MD0	PC14MD1	PC14MD0	PC13MD1	PC13MD0	PC12MD1	PC12MD0
		PC11MD1	PC11MD0	PC10MD1	PC10MD0	PC9MD1	PC9MD0	PC8MD1	PC8MD0
	PCCRL1	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0
		PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0
	PCPCRL	PC15PCR	PC14PCR	PC13PCR	PC12PCR	PC11PCR	PC10PCR	PC9PCR	PC8PCR
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
	PDIORH	PD31IOR	PD30IOR	PD29IOR	PD28IOR	PD27IOR	PD26IOR	PD25IOR	PD24IOR
		PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IOR
	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PFC	PDCRH2	PD31MD1	PD31MD0	PD30MD1	PD30MD0	PD29MD1	PD29MD0	PD28MD1	PD28MD0
		PD27MD1	PD27MD0	PD26MD1	PD26MD0	PD25MD1	PD25MD0	PD24MD1	PD24MD0
	PDCRH1	PD23MD1	PD23MD0	PD22MD1	PD22MD0	PD21MD1	PD21MD0	PD20MD1	PD20MD0
		PD19MD1	PD19MD0	PD18MD1	PD18MD0	PD17MD1	PD17MD0	PD16MD1	PD16MD0
	PDCRL2	PD15MD1	PD15MD0	PD14MD1	PD14MD0	PD13MD1	PD13MD0	PD12MD1	PD12MD0
		PD11MD1	PD11MD0	PD10MD1	PD10MD0	PD9MD1	PD9MD0	PD8MD1	PD8MD0
	PDCRL1	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0
		PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PDM1D0	PD0MD1	PD0MD0
	PDPCRH	PD31PCR	PD30PCR	PD29PCR	PD28PCR	PD27PCR	PD26PCR	PD25PCR	PD24PCR
		PD23PCR	PD22PCR	PD21PCR	PD20PCR	PD19PCR	PD18PCR	PD17PCR	PD16PCR
	PDPCRL	PD15PCR	PD14PCR	PD13PCR	PD12PCR	PD11PCR	PD10PCR	PD9PCR	PD8PCR
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
	PEIORH	—	—	—	—	—	—	—	—
		PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR
	PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR
		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
	PECRH1	PE23MD1	PE23MD0	PE22MD1	PE22MD0	PE21MD1	PE21MD0	PE20MD1	PE20MD0
		PE19MD1	PE19MD0	PE18MD1	PE18MD0	PE17MD1	PE17MD0	PE16MD1	PE16MD0
	PECRL2	PE15MD1	PE15MD0	PE14MD1	PE14MD0	PE13MD1	PE13MD0	PE12MD1	PE12MD0
		PE11MD1	PE11MD0	PE10MD1	PE10MD0	PE9MD1	PE9MD0	PE8MD1	PE8MD0
	PECRL1	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0
		PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0
	PEPCRH	—	—	—	—	—	—	—	—
		PE23PCR	PE22PCR	PE21PCR	PE20PCR	PE19PCR	PE18PCR	PE17PCR	PE16PCR
	PEPCRL	PE15PCR	PE14PCR	PE13PCR	PE12PCR	PE11PCR	PE10PCR	PE9PCR	PE8PCR
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
	PGIORL	PG15IOR	PG14IOR	PG13IOR	PG12IOR	PG11IOR	PG10IOR	PG9IOR	PG8IOR
		PG7IOR	PG6IOR	PG5IOR	PG4IOR	PG3IOR	PG2IOR	PG1IOR	PG0IOR
	PGCRL1	PG15MD	PG14MD	PG13MD	PG12MD	PG11MD	PG10MD	PG9MD	PG8MD
		PG7MD	PG6MD	PG5MD	PG4MD	PG3MD	PG2MD	PG1MD	PG0MD
	PGPCRL	PG15PCR	PG14PCR	PG13PCR	PG12PCR	PG11PCR	PG10PCR	PG9PCR	PG8PCR
		PG7PCR	PG6PCR	PG5PCR	PG4PCR	PG3PCR	PG2PCR	PG1PCR	PG0PCR

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PFC	PHIORL	PH15IOR	PH14IOR	PH13IOR	PH12IOR	PH11IOR	PH10IOR	PH9IOR	PH8IOR
		PH7IOR	PH6IOR	PH5IOR	PH4IOR	PH3IOR	PH2IOR	PH1IOR	PH0IOR
	PHCRL1	PH15MD	PH14MD	PH13MD	PH12MD	PH11MD	PH10MD	PH9MD	PH8MD
		PH7MD	PH6MD	PH5MD	PH4MD	PH3MD	PH2MD	PH1MD	PH0MD
	PHPCRL	PH15PCR	PH14PCR	PH13PCR	PH12PCR	PH11PCR	PH10PCR	PH9PCR	PH8PCR
		PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PCR
	PJIORL	PJ15IOR	PJ14IOR	PJ13IOR	PJ12IOR	PJ11IOR	PJ10IOR	PJ9IOR	PJ8IOR
		PJ7IOR	PJ6IOR	PJ5IOR	PJ4IOR	PJ3IOR	PJ2IOR	PJ1IOR	PJ0IOR
	PJCRL1	PJ15MD	PJ14MD	PJ13MD	PJ12MD	PJ11MD	PJ10MD	PJ9MD	PJ8MD
		PJ7MD	PJ6MD	PJ5MD	PJ4MD	PJ3MD	PJ2MD	PJ1MD	PJ0MD
	PJPCRL	PJ15PCR	PJ14PCR	PJ13PCR	PJ12PCR	PJ11PCR	PJ10PCR	PJ9PCR	PJ8PCR
		PJ7PCR	PJ6PCR	PJ5PCR	PJ4PCR	PJ3PCR	PJ2PCR	PJ1PCR	PJ0PCR
	PKIORL	—	—	—	—	—	—	—	—
		PK7IOR	PK6IOR	PK5IOR	PK4IOR	PK3IOR	PK2IOR	PK1IOR	PK0IOR
	PKCRL1	—	—	—	—	—	—	—	—
		PK7MD	PK6MD	PK5MD	PK4MD	PK3MD	PK2MD	PK1MD	PK0MD
	PKPCRL	—	—	—	—	—	—	—	—
		PK7PCR	PK6PCR	PK5PCR	PK4PCR	PK3PCR	PK2PCR	PK1PCR	PK0PCR
	PLPCRL	—	—	—	—	—	—	—	—
		—	—	PL5PCR	PL4PCR	PL3PCR	PL2PCR	PL1PCR	PL0PCR
	HCPCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	MZIZEH	MZIZEL
	DRVCR	PJDRV	—	PKDRV	—	—	—	—	—
		—	—	—	—	—	—	—	—
	PFEXCR	—	—	—	—	—	—	—	GPIE
		—	—	—	—	—	—	—	—
I/O port	PADRH	—	—	—	—	—	—	—	—
		—	—	—	—	PA19DR	PA18DR	PA17DR	PA16DR
	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PAPRH	—	—	—	—	—	—	—	—
		—	—	—	—	PA19PR	PA18PR	PA17PR	PA16PR

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
I/O port	PAPRL	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR
		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
	PBDRL	—	—	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPRL	—	—	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPRL	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PDDRH	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR
		PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR
	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
	PDPRH	PD31PR	PD30PR	PD29PR	PD28PR	PD27PR	PD26PR	PD25PR	PD24PR
		PD23PR	PD22PR	PD21PR	PD20PR	PD19PR	PD18PR	PD17PR	PD16PR
	PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
	PEDRH	—	—	—	—	—	—	—	—
		PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR
	PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPRH	—	—	—	—	—	—	—	—
		PE23PR	PE22PR	PE21PR	PE20PR	PE19PR	PE18PR	PE17PR	PE16PR
	PEPRL	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR
		PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PFDRL	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
	PGDRL	PG15DR	PG14DR	PG13DR	PG12DR	PG11DR	PG10DR	PG9DR	PG8DR
		PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
	PGPRL	PG15PR	PG14PR	PG13PR	PG12PR	PG11PR	PG10PR	PG9PR	PG8PR
		PG7PR	PG6PR	PG5PR	PG4PR	PG3PR	PG2PR	PG1PR	PG0PR

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
I/O port	PHDRL	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR
		PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
	PHPRL	PH15PR	PH14PR	PH13PR	PH12PR	PH11PR	PH10PR	PH9PR	PH8PR
		PH7PR	PH6PR	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
	PJDRL	PJ15DR	PJ14DR	PJ13DR	PJ12DR	PJ11DR	PJ10DR	PJ9DR	PJ8DR
		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
	PJPRL	PJ15PR	PJ14PR	PJ13PR	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR
		PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
	PKDRL	—	—	—	—	—	—	—	—
		PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR
	PKPRL	—	—	—	—	—	—	—	—
		PK7PR	PK6PR	PK5PR	PK4PR	PK3PR	PK2PR	PK1PR	PK0PR
	PLDRL	—	—	—	—	—	—	—	—
		—	—	PL5DR	PL4DR	PL3DR	PL2DR	PL1DR	PL0DR
LVDS (SH72315A only)	LVFRDR	LVD0[15]	LVD0[14]	LVD0[13]	LVD0[12]	LVD0[11]	LVD0[10]	LVD0[9]	LVD0[8]
		LVD0[7]	LVD0[6]	LVD0[5]	LVD0[4]	LVD0[3]	LVD0[2]	LVD0[1]	LVD0[0]
		LVD1[15]	LVD1[14]	LVD1[13]	LVD1[12]	LVD1[11]	LVD1[10]	LVD1[9]	LVD1[8]
		LVD1[7]	LVD1[6]	LVD1[5]	LVD1[4]	LVD1[3]	LVD1[2]	LVD1[1]	LVD1[0]
	LVCR	—	—	—	—	LVECIE	LVRXIE	LVUREIE	LVOREIE
		—	—	—	—	ENDC	—	CHSEL	RE
	LVSR	—	—	—	—	EOF	SOF	EOL	SOL
		—	—	—	—	ECDET	RDRF	URER	ORER
	LVFCR	—	—	—	—	—	—	—	—
		—	LRTRG2	LRTRG1	LRTRG0	—	—	—	FRST
	LVFDR	—	—	—	—	—	—	—	—
		—	—	—	R4	R3	R2	R1	R0
	LVCCR1	SOL7	SOL6	SOL5	SOL4	SOL3	SOL2	SOL1	SOL0
		EOL7	EOL6	EOL5	EOL4	EOL3	EOL2	EOL1	EOL0
	LVCCR2	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
		EOF7	EOF6	EOF5	EOF4	EOF3	EOF2	EOF1	EOF0
	LVLCNT	LCNT15	LCNT14	LCNT13	LCNT12	LCNT11	LCNT10	LCNT9	LCNT8
		LCNT7	LCNT6	LCNT5	LCNT4	LCNT3	LCNT2	LCNT1	LCNT0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
LVDS (SH72315A only)	LVDCNT	DCNT15	DCNT14	DCNT13	DCNT12	DCNT11	DCNT10	DCNT9	DCNT8
		DCNT7	DCNT6	DCNT5	DCNT4	DCNT3	DCNT2	DCNT1	DCNT0
RSPI	SPCR	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	SPMS
	SSLP	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
	SPPCR	—	—	MOIFE	MOIFV	—	SPOM	—	SPLP
	SPSR	SPRF	—	SPTEF	—	—	MODF	MIDLE	OVRF
	SPDR	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCR	—	—	—	—	—	SPSLN2	SPSLN1	SPSLN0
	SPSSR	—	—	SPECM1	SPECM0	—	—	SPCP1	SPCP0
	SPBR	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPDCR	—	—	SPLW	SPRDTD	—	—	SPFC1	SPFC0
	SPCKD	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0
	SSLND	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0
	SPND	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0
	SPCMD0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
	SPCMD1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
	SPCMD2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
	SPCMD3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
RCAN-ET	MCR	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
RCAN-ET	BCR0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR	—	—	IRR13	IRR12	—	—	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	TXPR1□0	TXPR1[15:8]							
		TXPR1[7:0]							
		TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR0	UMSR0[15:8]							
		UMSR0[7:0]							
RCAN-ET (MCR15 = 1)	MB[0]. CONTROL0H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
RCAN-ET (MCR15 = 0)	MB[0]. CONTROL0H	—	STDID[10:4]						
		STDID[3:0]				RTR	IDE	EXTID[17:16]	
RCAN-ET	MB[0]. CONTROL0L	EXTID[15:8]							
		EXTID[7:0]							

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
RCAN-ET (MCR15 = 1)	MB[0]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
RCAN-ET (MCR15 = 0)	MB[0]. LAFMH	—	STDID_LAFM[10:4]						
		STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]	
RCAN-ET	MB[0]. LAFML	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[0]. MSG_DATA[0]	MSG_DATA_0							
	MB[0]. MSG_DATA[1]	MSG_DATA_1							
	MB[0]. MSG_DATA[2]	MSG_DATA_2							
	MB[0]. MSG_DATA[3]	MSG_DATA_3							
	MB[0]. MSG_DATA[4]	MSG_DATA_4							
	MB[0]. MSG_DATA[5]	MSG_DATA_5							
	MB[0]. MSG_DATA[6]	MSG_DATA_6							
	MB[0]. MSG_DATA[7]	MSG_DATA_7							
	MB[0]. CONTROL1H	—	—	NMC	—	—	MBC[2:0]		
	MB[0]. CONTROL1L	—	—	—	—	DLC[3:0]			
RCAN-ET (MCR15 = 1)	MB[1]. CONTROL0H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
RCAN-ET (MCR15 = 0)	MB[1]. CONTROL0H	—	STDID[10:4]						
		STDID[3:0]				RTR	IDE	EXTID[17:16]	
RCAN-ET	MB[1]. CONTROL0L	EXTID[15:8]							
		EXTID[7:0]							
RCAN-ET (MCR15 = 1)	MB[1]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
RCAN-ET	MB[1].	—	STDID_LAFM[10:4]						
(MCR15 = 0)	LAFMH	STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]	
RCAN-ET	MB[1].	EXTID_LAFM[15:8]							
	LAFML	EXTID_LAFM[7:0]							
	MB[1]. MSG_DATA[0]	MSG_DATA0							
	MB[1]. MSG_DATA[1]	MSG_DATA1							
	MB[1]. MSG_DATA[2]	MSG_DATA2							
	MB[1]. MSG_DATA[3]	MSG_DATA3							
	MB[1]. MSG_DATA[4]	MSG_DATA4							
	MB[1]. MSG_DATA[5]	MSG_DATA5							
	MB[1]. MSG_DATA[6]	MSG_DATA6							
	MB[1]. MSG_DATA[7]	MSG_DATA7							
	MB[1]. CONTROL1H	—	—	NMC	ATX	DART	MBC[2:0]		
	MB[1]. CONTROL1L	—	—	—	—	DLC[3:0]			
	MB[2].	Same bit configuration as MB[1]							
	MB[3].	Same bit configuration as MB[1]							
	↓	(Ditto)							
	MB[13].	Same bit configuration as MB[1]							
	MB[14].	Same bit configuration as MB[1]							
	MB[15].	Same bit configuration as MB[1]							

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
TIM32C	TI32CR_0	MS	—	—	—	—	—	CKS1	CKS0
	TI32CR_1	MS	—	—	—	—	—	CKS1	CKS0
	TI32CR_2	—	—	—	—	CCLR	—	CKS1	CKS0
	TI32SR	CH2F	—	—	—	CH1UF	CH1DF	CH0UF	CH0DF
	TI32IER	CH2IE	—	—	—	CH1UIE	CH1DIE	CH0UIE	CH0DIE
	TI32CNT8_0	TI32CNT8_0 [7]	TI32CNT8_0 [6]	TI32CNT8_0 [5]	TI32CNT8_0 [4]	TI32CNT8_0 [3]	TI32CNT8_0 [2]	TI32CNT8_0 [1]	TI32CNT8_0 [0]
	TI32CNT8_1	TI32CNT8_1 [7]	TI32CNT8_1 [6]	TI32CNT8_1 [5]	TI32CNT8_1 [4]	TI32CNT8_1 [3]	TI32CNT8_1 [2]	TI32CNT8_1 [1]	TI32CNT8_1 [0]
	TI32CNT16	TI32CNT16 [15]	TI32CNT16 [14]	TI32CNT16 [13]	TI32CNT16 [12]	TI32CNT16 [11]	TI32CNT16 [10]	TI32CNT16 [9]	TI32CNT16 [8]
		TI32CNT16 [7]	TI32CNT16 [6]	TI32CNT16 [5]	TI32CNT16 [4]	TI32CNT16 [3]	TI32CNT16 [2]	TI32CNT16 [1]	TI32CNT16 [0]
	TI32COR_2								
	TI32STR	—	—	—	—	—	CST2	CST1	CST0
	TI32SMR	UP1	DWN1	UP0	DWN0	—	—	—	—
KEYC	KSCR1	DRRST	—	—	—	—	—	—	—
		—	—	—	—	—	KMS	KSE17	KSE16
	KSCR2	KSE15	KSE14	KSE13	KSE12	KSE11	KSE10	KSE9	KSE8
		KSE7	KSE6	KSE5	KSE4	KSE3	KSE2	KSE1	KSE0
	KSFCR	—	—	—	—	—	KSFCR2	KSFCR1	KSFCR0
	KSDR_0	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_1	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_2	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_3	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_4	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_5	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_6	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_7	KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSDR_8	KSDR15	KSDR14	KSDR13	KSDR12	KSDR11	KSDR10	KSDR9	KSDR8
		KSDR7	KSDR6	KSDR5	KSDR4	KSDR3	KSDR2	KSDR1	KSDR0
	KSIER	—	—	—	—	—	KSIE2	KSIE1	KSIE0

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
KEYC	KSSR	—	—	—	—	—	KSF2	KSF1	KSF0
	KSCCR	—	—	KSOC	KSPC	—	—	CKS1	CKS0
	KSCNT								
	KSCSR	—	—	—	—	—	—	—	CST
ROM/FLD	FPMON	FWE	—	—	—	—	—	—	—
	FMODR	—	—	—	FRDMD	—	—	—	—
	FASTAT	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
	FAEINT	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
	ROMMAT	KEY							
		—	—	—	—	—	—	—	ROMSEL
	FCURAME	KEY							
		—	—	—	—	—	—	—	FCRME
	FSTATR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	•	ERSSPD	PRGSPD
	FSTATR1	FCUERR	—	—	FLOCKST	—	—	—	—
	FENTRYR	FKEY							
		FENTRYD	—	—	—	—	—	—	FENTRY0
	FPROTR	FPKEY							
		—	—	—	—	—	—	—	FPROTCN
	FRESETR	FRKEY							
		—	—	—	—	—	—	—	FRESET
	FCMDR	CMDR							
		PCMDR							
	FCPSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ESUSPMD
	EEPBCCNT	—	—	—	BCADR				
		BCADR					—	—	BCSIZE
	FPESTAT	—	—	—	—	—	—	—	—
		PEERRST							
	EEPBCSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BCST
	EEPRE0	KEY							
		—	—	—	—	DBRE03	DBRE02	DBRE01	DBRE00

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ROM/FLD	EEPWE0	KEY							
		—	—	—	—	DBWE03	DBWE02	DBWE01	DBWE00
	PCKAR	—	—	—	—	—	—	—	—
		PCKA							
	FIEBAR	—	—	—	—	—	—	—	—
		FIEBAR[7:0]							
Power-down mode	STBCR	STBY	—	—	—	—	—	—	—
	STBCR2	MSTP27	MSTP26	MSTP25	MSTP24	—	—	MSTP21	—
	STBCR3	HIZ	MSTP36	MSTP35	—	MSTP33	MSTP32	MSTP31	MSTP30
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	—	MSTP42	MSTP41	—
	STBCR5	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	MSTP51	MSTP50
	STBCR6	MSTP67	—	—	—	—	—	—	—
	STBCR7	MSTP77	MSTP76	MSTP75	MSTP74	—	—	—	—
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	DPSTBCR	DPSTBY	—	—	—	—	RAMCUT2	RAMCUT1	RAMCUT0
	DPSWCR	WTSEL	WTSTS6	WTSTS5	WTSTS4	WTSTS3	WTSTS2	WTSTS1	WTSTS0
	SIER	—	—	—	DKEYCE	DTIM32CE	—	DIRQ9E	DIRQ8E
		DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
	SIFR	DNMIF	—	—	—	SSRF	—	DIRQ9F	DIRQ8F
		DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
	DPSIEGR	DNMIEG	—	—	—	—	—	DIRQ9EG	DIRQ8EG
		DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
	RSTSR	DPSRSTF	—	—	—	—	—	—	—
H-UDI	SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0
		—	—	—	—	—	—	—	—
	SDID	SDID31	SDID30	SDID29	SDID28	SDID27	SDID26	SDID25	SDID24
		SDID23	SDID22	SDID21	SDID20	SDID19	SDID18	SDID17	SDID16
		SDID15	SDID14	SDID13	SDID12	SDID11	SDID10	SDID9	SDID8
		SDID7	SDID6	SDID5	SDID4	SDID3	SDID2	SDID1	SDID0

34.3 Register States in Each Operating Mode

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
CPG	FRQCR	Initialized*1	Retained	Retained	Initialized	—	Retained
	MCLKCR	Initialized*1	Retained	Retained	Initialized	—	Retained
	ACLKCR	Initialized*1	Retained	Retained	Initialized	—	Retained
	OSCCR	Initialized	Retained	Retained	Initialized	—	Retained
INTC	ICR0	Initialized (Other than NMIL)	Retained	Retained	Initialized	—	Retained
	ICR1	Initialized	Retained	Retained	Initialized	—	Retained
	ICR2	Initialized	Retained	Retained	Initialized	—	Retained
	ICR3	Initialized	Retained	Retained	Initialized	—	Retained
	IRQRR0	Initialized	Retained	Retained	Initialized	—	Retained
	IRQRR1	Initialized	Retained	Retained	Initialized	—	Retained
	IBCR	Initialized	Retained	Retained	Initialized	—	Retained
	IBNR	Initialized	Retained*2	Retained	Initialized	—	Retained
	IPR01	Initialized	Retained	Retained	Initialized	—	Retained
	IPR02	Initialized	Retained	Retained	Initialized	—	Retained
	IPR03	Initialized	Retained	Retained	Initialized	—	Retained
	IPR04	Initialized	Retained	Retained	Initialized	—	Retained
	IPR06	Initialized	Retained	Retained	Initialized	—	Retained
	IPR08	Initialized	Retained	Retained	Initialized	—	Retained
	IPR09	Initialized	Retained	Retained	Initialized	—	Retained
	IPR10	Initialized	Retained	Retained	Initialized	—	Retained
	IPR11	Initialized	Retained	Retained	Initialized	—	Retained
	IPR12	Initialized	Retained	Retained	Initialized	—	Retained
	IPR13	Initialized	Retained	Retained	Initialized	—	Retained
	IPR14	Initialized	Retained	Retained	Initialized	—	Retained
	IPR15	Initialized	Retained	Retained	Initialized	—	Retained
	IPR16	Initialized	Retained	Retained	Initialized	—	Retained
	IPR17	Initialized	Retained	Retained	Initialized	—	Retained
	IPR18	Initialized	Retained	Retained	Initialized	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
UBC	BAR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAMR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAMR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	BAR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	BBR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	BRCR	Initialized	Retained	Retained	Initialized	Retained	Retained
DTC	DTCERA	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCERB	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCERC	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCERD	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCERE	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCERF	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCERG	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCCR	Initialized	Retained	Retained	Initialized	Retained	Retained
	DTCVBR	Initialized	Retained	Retained	Initialized	Retained	Retained
BSC	CMNCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS0BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS1BCR	Initialized	Retained	Retained	Initialized	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
BSC	CS2BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS3BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS4BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS5BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS6BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS7BCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS0WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS1WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS2WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS3WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS4WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS5WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS6WCR	Initialized	Retained	Retained	Initialized	—	Retained
	CS7WCR	Initialized	Retained	Retained	Initialized	—	Retained
	SDCR	Initialized	Retained	Retained	Initialized	—	Retained
	RTCSR	Initialized	Retained	Retained	Initialized	—	Retained
	RTCNT	Initialized	Retained	Retained	Initialized	—	Retained
	RTCOR	Initialized	Retained	Retained	Initialized	—	Retained
	BSCEHR	Initialized	Retained	Retained	Initialized	—	Retained
DMAC	SAR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	DAR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMATCR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	CHCR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	RSAR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDAR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDMATCR_0	Initialized	Retained	Retained	Initialized	Retained	Retained
	SAR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	DAR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMATCR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	CHCR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	RSAR_1	Initialized	Retained	Retained	Initialized	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
DMAC	RDAR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDMATCR_1	Initialized	Retained	Retained	Initialized	Retained	Retained
	SAR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	DAR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMATCR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	CHCR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	RSAR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDAR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDMATCR_2	Initialized	Retained	Retained	Initialized	Retained	Retained
	SAR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	DAR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMATCR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	CHCR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	RSAR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDAR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	RDMATCR_3	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMAOR	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMARS0	Initialized	Retained	Retained	Initialized	Retained	Retained
	DMARS1	Initialized	Retained	Retained	Initialized	Retained	Retained
MTU2	TCR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORH_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORL_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRC_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRD_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRE_0	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
MTU2	TGRF_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER2_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR2_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTM_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIOR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TICCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIOR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORH_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORL_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRC_3	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
MTU2	TGRD_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTM_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCR_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORH_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORL_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRC_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRD_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTM_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCORA_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCORB_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCOBRA_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCOBRB_4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCRU_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCRV_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCRW_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORU_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORV_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORW_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSTR_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTU_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTV_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTW_5	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
MTU2	TGRU_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRV_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRW_5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTCMPCLR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSTR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSYR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCSYSTR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TRWER	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOER	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOCR1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOCR2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCDR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TDDR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCBR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TITCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TITCNT	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTER	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TDER	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TWCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOLBR	Initialized	Retained	Retained	Initialized	Initialized	Retained
MTU2S	TCR_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORH_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORL_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
MTU2S	TGRC_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRD_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTM_3S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCR_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TMDR_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORH_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORL_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNT_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRA_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRB_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRC_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRD_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTM_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCORA_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCORB_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCOBRA_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TADCOBRB_4S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCRU_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCRV_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCRW_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORU_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORV_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIORW_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TIER_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSR_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSTR_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTU_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTV_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
MTU2S	TCNTW_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRU_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRV_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGRW_5S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTCMPCLRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSTRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSYRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TRWERS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOERS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOCR1S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOCR2S	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TGCRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCDRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TDDRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCNTSS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TCBRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TITCRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TITCNTS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TBTERS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TDERS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TSYCRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TWCRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TOLBRS	Initialized	Retained	Retained	Initialized	Initialized	Retained
POE2	ICSR1	Initialized	Retained	Retained	Initialized	—	Retained
	OCSR1	Initialized	Retained	Retained	Initialized	—	Retained
	ICSR2	Initialized	Retained	Retained	Initialized	—	Retained
	OCSR2	Initialized	Retained	Retained	Initialized	—	Retained
	ICSR3	Initialized	Retained	Retained	Initialized	—	Retained
	SPOER	Initialized	Retained	Retained	Initialized	—	Retained
	POECR1	Initialized	Retained	Retained	Initialized	—	Retained
	POECR2	Initialized	Retained	Retained	Initialized	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
CMT	CMSTR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CMCSR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CMCNT_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CMCOR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CMCSR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CMCNT_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CMCOR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
CMT2	CM2STR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2CR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2IOR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2SR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2CNT	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2COR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2ICR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2ICR1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2OCR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	CM2OCR1	Initialized	Retained	Retained	Initialized	Initialized	Retained
WDT	WTCSR	Initialized*[1]	Retained	Retained	Initialized	—	Retained
	WTCNT	Initialized*[1]	Retained	Retained	Initialized	—	Retained
	WRCSR	Initialized*[1]	Retained	Retained	Initialized	—	Retained
SCI (channel 0)	SCSMR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBRR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSCR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTDR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSSR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCRDR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSDCR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSPTR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSMR2_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTBACNT_0	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
SCI (channel 0)	SCRBACNT_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBACOR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
SCI (channel 1)	SCSMR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBRR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSCR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTDR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSSR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCRDR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSDCR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSPTR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSMR2_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTBACNT_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCRBACNT_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBACOR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
SCI (channel 2)	SCSMR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBRR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSCR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTDR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSSR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCRDR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSDCR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSPTR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSMR2_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTBACNT_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCRBACNT_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBACOR_2	Initialized	Retained	Retained	Initialized	Initialized	Retained
SCI (channel 3)	SCSMR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBRR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSCR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTDR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSSR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
SCI (channel 3)	SCRDR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSDCR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSPTR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCSMR2_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCTBACNT_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCRBACNT_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SCBACOR_3	Initialized	Retained	Retained	Initialized	Initialized	Retained
SCIF (channel 4)	SCSMR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCBRR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSCR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTDR_4	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFSR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFRDR_4	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFCR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFDR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSPTR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCLSR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSDCR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTCR_4	Initialized	Retained	Retained	Initialized	Retained	Retained
SCIF (channel 5)	SCSMR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCBRR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSCR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTDR_5	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFSR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFRDR_5	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFCR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFDR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSPTR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCLSR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSDCR_5	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTCR_5	Initialized	Retained	Retained	Initialized	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
SCIF (channel 6)	SCSMR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCBRR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSCR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTDR_6	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFSR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFRDR_6	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFCR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFDR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSPTR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCLSR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSDCR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTCR_6	Initialized	Retained	Retained	Initialized	Retained	Retained
SCIF (channel 7)	SCSMR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCBRR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSCR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTDR_7	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFSR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFRDR_7	Undefined	Retained	Retained	Undefined	Retained	Retained
	SCFCR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFDR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSPTR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCLSR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCSDCR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
	SCFTCR_7	Initialized	Retained	Retained	Initialized	Retained	Retained
IIC3	ICCR1	Initialized	Retained	Retained	Initialized	Retained	Retained
	ICCR2	Initialized	Retained	Retained	Initialized	Retained	Retained
	ICMR	Initialized	Retained	Retained/ Initialized (BC2-BC0)	Initialized	Retained/ Initialized (BC2-BC0)	Retained
	ICIER	Initialized	Retained	Retained	Initialized	Retained	Retained
	ICSR	Initialized	Retained	Retained	Initialized	Retained	Retained
	SAR	Initialized	Retained	Retained	Initialized	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
IIC3	ICDRT	Initialized	Retained	Retained	Initialized	Retained	Retained
	ICDRR	Initialized	Retained	Retained	Initialized	Retained	Retained
	NF2CYC	Initialized	Retained	Retained	Initialized	Retained	Retained
ADC	ADDR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR3	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR4	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR5	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR6	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR7	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADCSR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADCR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR8	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR9	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR10	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR11	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR12	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR13	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR14	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADDR15	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADCSR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADCR_1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADSDR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADSSR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ADTSR_0	Initialized	Retained	Retained	Initialized	Initialized	Retained
PFC	PAIORH	Initialized	Retained	Retained	Initialized	—	Retained
	PAIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PACRH1	Initialized	Retained	Retained	Initialized	—	Retained
	PACRL2	Initialized	Retained	Retained	Initialized	—	Retained
	PACRL1	Initialized	Retained	Retained	Initialized	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
PFC	PAPCRH	Initialized	Retained	Retained	Initialized	—	Retained
	PAPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PBIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PBCRL2	Initialized	Retained	Retained	Initialized	—	Retained
	PBCRL1	Initialized	Retained	Retained	Initialized	—	Retained
	PBPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PCIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PCCRL2	Initialized	Retained	Retained	Initialized	—	Retained
	PCCRL1	Initialized	Retained	Retained	Initialized	—	Retained
	PCPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PDIORH	Initialized	Retained	Retained	Initialized	—	Retained
	PDIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PDCRH2	Initialized*3	Retained	Retained	Retained	—	Retained
	PDCRH1	Initialized*3	Retained	Retained	Retained	—	Retained
	PDCRL2	Initialized*3	Retained	Retained	Retained	—	Retained
	PDCRL1	Initialized*3	Retained	Retained	Retained	—	Retained
	PDPCRH	Initialized	Retained	Retained	Initialized	—	Retained
	PDPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PEIORH	Initialized	Retained	Retained	Initialized	—	Retained
	PEIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PECRH1	Initialized	Retained	Retained	Initialized	—	Retained
	PECRL2	Initialized	Retained	Retained	Initialized	—	Retained
	PECRL1	Initialized	Retained	Retained	Initialized	—	Retained
	PEPCRH	Initialized	Retained	Retained	Initialized	—	Retained
	PEPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PGIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PGCRL1	Initialized*3	Retained	Retained	Retained	—	Retained
	PGPCRL	Initialized*3	Retained	Retained	Retained	—	Retained
	PHIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PHCRL1	Initialized	Retained	Retained	Initialized	—	Retained
	PHPCRL	Initialized	Retained	Retained	Initialized	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
PFC	PJIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PJCRL1	Initialized	Retained	Retained	Initialized	—	Retained
	PJPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PKIORL	Initialized	Retained	Retained	Initialized	—	Retained
	PKCRL1	Initialized	Retained	Retained	Initialized	—	Retained
	PKPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	PLPCRL	Initialized	Retained	Retained	Initialized	—	Retained
	HCPCR	Initialized	Retained	Retained	Initialized	—	Retained
	DRVCR	Initialized	Retained	Retained	Initialized	—	Retained
	PFEXCR	Initialized	Retained	Retained	Initialized	—	Retained
I/O port	PADRH	Initialized	Retained	Retained	Initialized	—	Retained
	PADRL	Initialized	Retained	Retained	Initialized	—	Retained
	PAPRH	Undefined	Retained	Retained	Undefined	—	Retained
	PAPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PBDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PBPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PCDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PCPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PDDRH	Initialized	Retained	Retained	Initialized	—	Retained
	PDDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PDPRH	Undefined	Retained	Retained	Undefined	—	Retained
	PDPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PEDRH	Initialized	Retained	Retained	Initialized	—	Retained
	PEDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PEPRH	Undefined	Retained	Retained	Undefined	—	Retained
	PEPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PFDRL	Undefined	Retained	Retained	Undefined	—	Retained
	PGDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PGPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PHDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PHPRL	Undefined	Retained	Retained	Undefined	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
I/O port	PJDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PJPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PKDRL	Initialized	Retained	Retained	Initialized	—	Retained
	PKPRL	Undefined	Retained	Retained	Undefined	—	Retained
	PLDRL	Undefined	Retained	Retained	Undefined	—	Retained
LVDS (SH72315A only)	LVFRDR	Undefined	Retained	Retained	Undefined	Undefined	Retained
	LVCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVSR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVFCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVFDR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVCCR1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVCCR2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVLCNT	Initialized	Retained	Retained	Initialized	Initialized	Retained
	LVDCNT	Initialized	Retained	Retained	Initialized	Initialized	Retained
RSPI	SPCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SSLP	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPPCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPSR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPDR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPSCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPSSR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPBR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPDCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPCKD	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SSLND	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPND	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPCMD0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPCMD1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPCMD2	Initialized	Retained	Retained	Initialized	Initialized	Retained
	SPCMD3	Initialized	Retained	Retained	Initialized	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
RCAN-ET	MCR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	GSR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	BCR1	Initialized	Retained	Retained	Initialized	Initialized	Retained
	BCR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	IRR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	IMR	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TEC/REC	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TXPR1, TXPR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TXCR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	TXACK0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	ABACK0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	RXPR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	RFPR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	MBIMR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	UMSR0	Initialized	Retained	Retained	Initialized	Initialized	Retained
	MB[0]. CONTROL0H	—	Retained	—	Initialized	—	Retained
	MB[0]. CONTROL0L	—	Retained	—	Initialized	—	Retained
	MB[0]. LAFMH	—	Retained	—	Initialized	—	Retained
	MB[0]. LAFML	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[0]	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[1]	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[2]	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[3]	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[4]	—	Retained	—	Initialized	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
RCAN-ET	MB[0]. MSG_DATA[5]	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[6]	—	Retained	—	Initialized	—	Retained
	MB[0]. MSG_DATA[7]	—	Retained	—	Initialized	—	Retained
	MB[0]. CONTROL1H	Initialized	Retained	Retained	Initialized	Initialized	Retained
	MB[0]. CONTROL1L	Initialized	Retained	Retained	Initialized	Initialized	Retained
	MB[1].	Same as MB[0]					
	MB[2].	Same as MB[0]					
	MB[3].	Same as MB[0]					
	↓	(Ditto)					
	MB[13].	Same as MB[0]					
	MB[14].	Same as MB[0]					
	MB[15].	Same as MB[0]					
TIM32C	TI32CR_0	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32CR_1	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32CR_2	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32SR	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32IER	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32CNT8_0	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32CNT8_1	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32CNT16	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32COR_2	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32STR	Initialized*3	Retained	Retained	Retained	Retained	Retained
	TI32SMR	Initialized*3	Retained	Retained	Retained	Retained	Retained
KEYC	KSCR1	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSCR2	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSFCR	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_0	Initialized*3	Retained	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
KEYC	KSDR_1	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_2	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_3	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_4	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_5	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_6	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_7	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSDR_8	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSIER	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSSR	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSCCR	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSCNT	Initialized*3	Retained	Retained	Retained	Initialized	Retained
	KSCSR	Initialized*3	Retained	Retained	Retained	Initialized	Retained
ROM/FLD	FPMON	Initialized	Retained	Retained	Retained	Retained	Retained
	FMODR	Initialized	Retained	Retained	Retained	Retained	Retained
	FASTAT	Initialized	Retained	Retained	Retained	Retained	Retained
	FAEINT	Initialized	Retained	Retained	Retained	Retained	Retained
	ROMMAT	Initialized	Retained	Retained	Retained	Retained	Retained
	FCURAME	Initialized	Retained	Retained	Retained	Retained	Retained
	FSTATR0	Initialized	Retained	Retained	Retained	Retained	Retained
	FSTATR1	Initialized	Retained	Retained	Retained	Retained	Retained
	FENTRYR	Initialized	Retained	Retained	Retained	Retained	Retained
	FPROTR	Initialized	Retained	Retained	Retained	Retained	Retained
	FRESETR	Initialized	Retained	Retained	Retained	Retained	Retained
	FCMDR	Initialized	Retained	Retained	Retained	Retained	Retained
	FCPSR	Initialized	Retained	Retained	Retained	Retained	Retained
	EEPBCCNT	Initialized	Retained	Retained	Retained	Retained	Retained
	FPESTAT	Initialized	Retained	Retained	Retained	Retained	Retained
	EEPBCSTAT	Initialized	Retained	Retained	Retained	Retained	Retained
	EEPRE0	Initialized	Retained	Retained	Retained	Retained	Retained
	EEPWE0	Initialized	Retained	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Deep Software Standby	Module Standby	Sleep
ROM/FLD	PCKAR	Initialized	Retained	Retained	Retained	Retained	Retained
	FIEBAR	Initialized	Retained	Retained	Retained	Retained	Retained
Power-down mode	STBCR	Initialized	Retained	Retained	Initialized	—	Retained
	STBCR2	Initialized	Retained	Retained	Initialized	—	Retained
	STBCR3	Initialized	Retained	Retained	Initialized	—	Retained
	STBCR4	Initialized	Retained	Retained	Initialized	—	Retained
	STBCR5	Initialized	Retained	Retained	Initialized	—	Retained
	STBCR6	Initialized	Retained	Retained	Initialized	—	Retained
	STBCR7	Initialized*3	Retained	Retained	Retained	—	Retained
	SYSCR1	Initialized	Retained	Retained	Initialized	—	Retained
	SYSCR2	Initialized	Retained	Retained	Initialized	—	Retained
	DPSTBCR	Initialized*3	Retained	Retained	Retained	—	Retained
	DPSWCR	Initialized*3	Retained	Retained	Retained	—	Retained
	SIER	Initialized*3	Retained	Retained	Retained	—	Retained
	SIFR	Initialized*3	Retained	Retained	Retained	—	Retained
	DPSIEGR	Initialized*3	Retained	Retained	Retained	—	Retained
	RSTSR	Initialized*3	Retained	Retained	Retained	—	Retained
H-UDI*4	SDIR	Retained	Retained	Retained	Initialized	Retained	Retained
	SDID	Retained	Retained	Retained	Retained	Retained	Retained

Notes: 1. Retains the previous value after an internal power-on reset by means of the WDT.
2. Bits BN[3:0] are initialized.
3. Initialized only by $\overline{\text{RES}}$ assertion.
Not initialized by an internal reset issued at return from deep standby state, or by an internal power-on reset due to the WDT.
4. Initialized by $\overline{\text{TRST}}$ assertion or in the Test-Logic-Reset state of the TAP controller.

Section 35 Electrical Characteristics

35.1 Absolute Maximum Ratings

Table 35.1 lists the absolute maximum ratings.

Table 35.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage		Vcc	−0.3 to +4.6	V
		PVcc1	−0.3 to +4.6	V
		PVcc2	−0.3 to +4.6	V
PLL power supply voltage		PLLVcc	−0.3 to +4.6	V
Analog power supply voltage		AVcc	−0.3 to +4.6	V
Analog reference voltage		AVref	−0.3 to AVcc +0.3	V
LVDS power supply voltage (SH72315A only)		LVDSVcc	−0.3 to +4.6	V
Input voltage	Analog input pin	VAN	−0.3 to AVcc +0.3	V
	Vcc input pin	Vin	−0.3 to Vcc +0.3	V
	PVcc1 input pin		−0.3 to PVcc1 +0.3	V
	PVcc2 input pin		−0.3 to PVcc2 +0.3	V
	LVDS input pin (SH72315A only)		−0.3 to LVDSVcc +0.3	V
Operating temperature	Consumer specifications	Topr	−20 to +85	°C
	Industrial specifications		−40 to +85	°C
Storage temperature		Tstg	−55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

35.2 Power-On/Power-Off Sequence

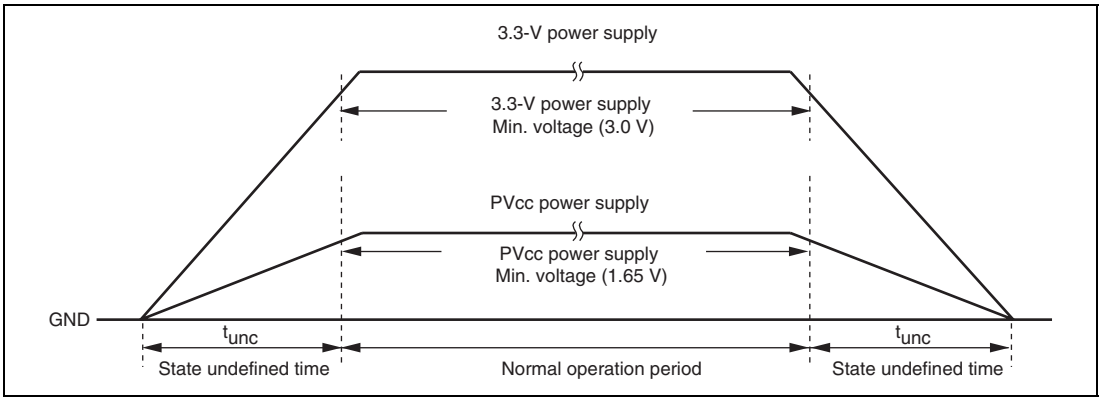


Figure 35.1 Power-On/Power-Off Sequence

Table 35.2 Time for Power-On/Power-Off Sequence

Item	Symbol	Min.	Max.	Unit
State undefined time	t_{unc}	—	100	ms

Note: The 3.3-V power-supply system (Vcc, PLLVcc, AVcc, and LVDSVcc; LVDSVcc is only on the SH72315A) and PVcc power-supply system (PVcc1 and PVcc2) should be turned on and off simultaneously as far as possible. Pin states are undefined over the intervals until the voltages for both power supply systems have reached the minimum voltage after they have been turned on and until the voltages become 0 V after the voltage for either has fallen below the minimum voltage. Since the output- and I/O-pin states and internal states become undefined over these intervals, design the power-supply circuitry so that these intervals become as short as possible. In addition, design the system so that these undefined states do not cause malfunctions of the system as a whole.

35.3 DC Characteristics

Voltage settings of 1.65 to 1.95 V or 3.0 to 3.6 V can be made for PVcc1 and PVcc2. However, when PVcc1 and PVcc2 are set to 1.65 to 1.95 V, I/O buffers with high drivability become a condition of the specification. For changing the I/O buffer drivability, see section 22.1.30, I/O Buffer Driver Control Register (DRVCR).

Table 35.3 DC Characteristics (1) [Common Items]

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		Vcc	3.0	3.3	3.6	V	ZL = 100 Ω
		PVcc1	1.65	1.8	1.95	V	
			3.0	3.3	3.6		
		PVcc2	1.65	1.8	1.95	V	
			3.0	3.3	3.6		
PLL power supply voltage		PLLVcc	3.0	3.3	3.6	V	
Analog power supply voltage		AVcc	3.0	3.3	3.6	V	
LVDS power supply voltage (SH72315A only)		LVDSVcc	3.0	3.3	3.6	V	
Supply current *1 (SH72315A)	Normal operation	Icc*2	—	120	160	mA	Iφ = Mφ = 100 MHz Bφ = Pφ = Aφ = 50 MHz LVDS reception
	Normal operation (writing to FLD)	Icc*3	—	140	200		
	Sleep mode	Isleep*2	—	70	100	mA	
	Software standby mode	Istby*2	—	1.0	10	mA	Ta > 50°C
			—	0.5	2.5	mA	Ta ≤ 50°C
	Deep software standby mode	Idstby*2	—	10	25	µA	Ta > 50°C
			—	50	80	µA	Ta > 50°C RAM retained KEYC/TIM32C operating
			—	5	10	µA	Ta ≤ 50°C
			—	30	40	µA	Ta ≤ 50°C RAM retained KEYC/TIM32C operating

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*1 (SH72315L/ SH72314L)	Normal operation	Icc*2	—	115	155	mA	Iϕ = Mϕ = 100 MHz Bϕ = Pϕ = Aϕ = 50 MHz
	Normal operation (writing to FLD)	Icc*3	—	135	195		
	Sleep mode	Isleep*2	—	65	95	mA	
	Software standby mode	Istby*2	—	1.0	10	mA	Ta > 50°C
			—	0.5	2.5	mA	Ta ≤ 50°C
	Deep software standby mode	Idstby*2	—	10	25	µA	Ta > 50°C
			—	50	80	µA	Ta > 50°C RAM retained KEYC/TIM32C operating
			—	5	10	µA	Ta ≤ 50°C
			—	30	40	µA	Ta ≤ 50°C RAM retained KEYC/TIM32C operating
Input leakage current	Vcc input pin	| Iin |	—	—	1	µA	Vin = 0.5 to Vcc-0.5 V
	PVcc1 input pins (PJ15 to PJ0)						Vin = 0.5 to PVcc1-0.5 V
	PVcc2 input pins (PK7 to PK0)						Vin = 0.5 to PVcc2-0.5 V
	LVDSVcc input pins		—	—	1		Vin = 0.5 to LVDSVcc-0.5 V
Three-state leakage current	Vcc I/O pins, output pins (off state)	| ISTI |	—	—	1	µA	Vin = 0.5 to VCC-0.5 V
	PVcc1 I/O pins (PJ15 to PJ0) (off state)						Vin = 0.5 to PVcc1-0.5 V
	PVcc2 I/O pins (PK7 to PK0) (off state)						Vin = 0.5 to PVcc2-0.5 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply voltage	During A/D conversion	AIcc	—	2	3	mA	Value per A/D converter
	Waiting for A/D conversion		—	2	3	µA	
	Standby		—	2	3	µA	
Analog reference voltage current	During A/D conversion	AIref	—	0.2	0.3	mA	
	Waiting for A/D conversion		—	2	3	µA	
	Standby		—	2	3	µA	
LVDS power supply current (SH72315A only)	During reception	LIcc	—	15	20	mA	
	Waiting for reception		—	15	20	mA	
	Standby		—	2	5	µA	
Input capacitance	All pins	Cin	—	—	20	pF	

Caution: When the A/D converter is not in use, the AVcc, AVss, and AVref pins should not be open. Additionally, in the case of the SH72315A, the LVDSVcc and LVDSVss pins should not be left open-circuit when the LVDS receiver is not in use.

Notes: 1. Supply current values are when all output and pull-up pins are unloaded.

2. Icc, I_{sleep}, I_{stby}, and I_{dstby} represent the total currents consumed in the Vcc, PLLVcc, PVcc1, and PVcc2 systems.

3. Reference value

Table 35.3 DC Characteristics (2) [Except for LVDS-Related Pins]

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, MD1, MD0, FWE, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, EXTAL32	V_{IH}	Vcc – 0.5	—	Vcc + 0.3	V	
	Pins with multiplexed analog functions (PF15 to PF0)		2.2	—	AVcc + 0.3	V	
	Pins with multiplexed LVDS functions (PL5 to PL0) (SH72315A only)		2.2	—	LVDSVcc + 0.3	V	
	PVcc1 pins (PJ15 to PJ0) (excluding Schmitt pins)		PVcc1 × 0.85	—	PVcc1 + 0.3	V	PVcc1 = 1.65 to 1.95 V
			2.2		PVcc1 + 0.3	V	PVcc1 = 3.0 to 3.6 V
	PVcc2 pins (PK7 to PK0) (excluding Schmitt pins)		PVcc2 × 0.85	—	PVcc2 + 0.3	V	PVcc2 = 1.65 to 1.95 V
			2.2	—	PVcc2 + 0.3	V	PVcc2 = 3.0 to 3.6 V
	Input pins other than above (excluding Schmitt pins) (including PL5 to PL0 in case of SH72315L/SH72314L)		2.2	—	Vcc + 0.3	V	
Input low voltage	$\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, MD1, MD0, FWE, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, EXTAL32	V_{IL}	–0.3	—	0.5	V	
	PVcc1 (PJ15 to PJ0) (excluding Schmitt pins)		–0.3	—	PVcc1 × 0.15	V	PVcc1 = 1.65 to 1.95 V
			–0.3	—	0.5	V	PVcc1 = 3.0 to 3.6 V
	PVcc2 (PK7 to PK0) (excluding Schmitt pins)		–0.3	—	PVcc2 × 0.15	V	PVcc2 = 1.65 to 1.95 V
			–0.3	—	0.5	V	PVcc2 = 3.0 to 3.6 V
	KEY31 to KEY0, P7 to P0		–0.3	—	1.15	V	
	Input pins other than above (excluding Schmitt pins)		–0.3	—	0.5	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input characteristics*	Vcc pin	VT^+	Vcc − 0.5	—	—	V	
	PVcc1 pins (PJ15 to PJ0)		PVcc1 × 0.85	—	—	V	PVcc1 = 1.65 to 1.95 V
			PVcc1 − 0.5	—	—	V	PVcc1 = 3.0 to 3.6 V
	PVcc2 pins (PK7 to PK0)		PVcc2 × 0.85	—	—	V	PVcc2 = 1.65 to 1.95 V
			PVcc2 − 0.5	—	—	V	PVcc2 = 3.0 to 3.6 V
	Vcc pin	VT^-	—	—	0.5	V	Vcc pin
	PVcc1 pins (PJ15 to PJ0)		—	—	PVcc1 × 0.15	V	PVcc1 = 1.65 to 1.95 V
			—	—	0.5	V	PVcc1 = 3.0 to 3.6 V
	PVcc2 pins (PK7 to PK0)		—	—	PVcc1 × 0.15	V	PVcc2 = 1.65 to 1.95 V
			—	—	0.5	V	PVcc2 = 3.0 to 3.6 V
	Vcc pin	VT^+–VT^-	Vcc1 × 0.05	—	—	V	
	PVcc1 pins (PJ15 to PJ0)		PVcc1 × 0.02	—	—	V	PVcc1 = 1.65 to 1.95 V
			PVcc1 × 0.05	—	—	V	PVcc1 = 3.0 to 3.6 V
	PVcc2 pins (PK7 to PK0)		PVcc2 × 0.02	—	—	V	PVcc2 = 1.65 to 1.95 V
			PVcc2 × 0.05	—	—	V	PVcc2 = 3.0 to 3.6 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Vcc pin	V_{OH}	2.2	—	—	V	I_{OH} = -1 mA
	PVcc1 pins (PJ15 to PJ0)		PVcc1 × 0.85	—	—	V	PVcc1 = 1.65 to 1.95 V I_{OH} = -0.2 mA
			2.2	—	—	V	PVcc1 = 3.0 to 3.6 V I_{OH} = -1 mA
	PVcc2 pins (PK7 to PK0)		PVcc2 × 0.85	—	—	V	PVcc2 = 1.65 to 1.95 V I_{OH} = -0.2 mA
			2.2	—	—	V	PVcc2 = 3.0 to 3.6 V IOH = -1 mA
	TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS		Vcc – 1.0	—	—	V	Vcc pin I_{OH} = -5 mA

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	Vcc pin	V_{OL}	—	—	0.5	V	I_{OL} = 1.6 mA
	PVcc1 pins (PJ15 to PJ0)		—	—	PVcc1 × 0.15	V	PVcc1 = 1.65 to 1.95 V I_{OL} = 0.2 mA
			—	—	0.5	V	PVcc1 = 3.0 to 3.6 V I_{OL} = 1.6 mA
	PVcc2 pins (PK7 to PK0)		—	—	PVcc2 × 0.15	V	PVcc2 = 1.65 to 1.95 V I_{OL} = 0.2 mA
			—	—	0.5	V	PVcc2 = 3.0 to 3.6 V I_{OL} = 1.6 mA
	TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS		—	—	1.0	V	Vcc pin I_{OL} = 15 mA
	SCL, SDA		—	—	0.5	V	I_{OL} = 3 mA
			—	—	0.6	V	I_{OL} = 8 mA
Input pull-up MOS current	Vcc pin (except for port L)	$-I_P$	–15	–58	–200	µA	Vin = 0 V
	PVcc1 pin (PJ15 to PJ0)		–3	–14	–50	µA	Vin = 0 V PVcc1 = 1.65 to 1.95 V
			–15	–58	–200	µA	Vin = 0 V PVcc1 = 3.0 to 3.6 V
	PVcc2 pin (PK7 to PK0)		–3	–14	–50	µA	Vin = 0 V PVcc2 = 1.65 to 1.95 V
			–15	–58	–200	µA	Vin = 0 V PVcc2 = 3.0 to 3.6 V
	Port L		–5	–11	–24	µA	Vin = 0 V
RAM standby voltage		V_{RAM}	2.7	—	—	V	Vcc

Note: * TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TIC5U to TIC5W, TCLKA to TCLKD, TIOC3AS to TIOC3DS, TIOC4AS to TIOC4DS, TIC5US to TIC5WS, $\overline{\text{POE8}}$ to $\overline{\text{POE0}}$, SCK7 to SCK0, RxD7 to RxD0, IRQ23 to IRQ0, SCL, SDA, TIC1, TIC0, KEY31 to KEY0, P7 to P0, TI32I1A, TI32I0A, TI32I1B, TI32I0B, RSPCK0, MISO0, MOSI0, SSL0, CRx0

Table 35.3 DC Characteristics (3) [LVDS Related-Pins (SH72315A only)]

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Differential input voltage amplitude	VID	200	350	500	mV	ZL = 100 Ω
Input offset voltage	VIS	1.0	1.25	1.5	V	
Input threshold voltage (high)	VIT+	100	—	—	mV	Relative to VIS
Input threshold voltage (low)	VIT-	100	—	—	mV	Relative to VIS
Internal differential input impedance resistor	ZL	75	100	125	Ω	
Internal pull-up resistor	RP	150	400	600	kΩ	

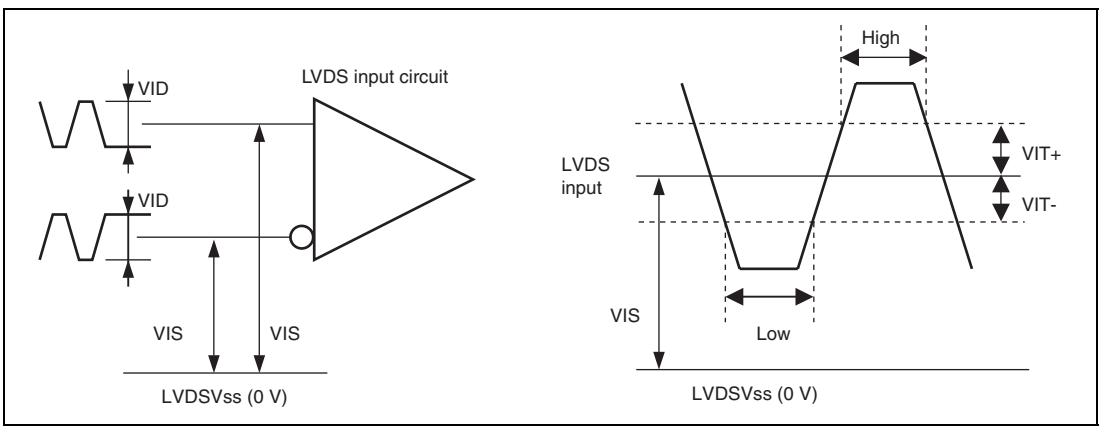


Figure 35.2 DC Characteristics for LVDS Reception

Table 35.4 Permissible Output Currents

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2*	mA
Permissible output low current (total)	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	$-I_{OH}$	—	—	2	mA
Permissible output high current (total)	$\Sigma -I_{OH}$	—	—	25	mA

Note: * TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS (Vcc system): I_{OL} = 15 mA (Max.)/ $-I_{OH}$ = 5 mA (Max.).
SCL and SDA: I_{OL} = 8 mA (Max.).
Of these pins, the number of pins from which current more than 2.0 mA runs evenly should be 3 or less.

Caution: To protect the LSI's reliability, do not exceed the output current values in table 35.4.

35.4 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed. Voltage settings of 1.65 to 1.95 V or 3.0 to 3.6 V can be made for PVcc1 and PVcc2. However, when PVcc1 and PVcc2 are set to 1.65 to 1.95 V, I/O buffers with high drivability become a condition of the specification.

For changing the I/O buffer drivability, see section 22.1.30, I/O Buffer Driver Control Register (DRVCR).

Table 35.5 Maximum Operating Frequency

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU (Iϕ)	f	5	—	100	MHz	
	Internal bus, external bus (Bϕ)		5	—	50		
	Peripheral module (Pϕ)		5	—	50		
	MTU2S (Mϕ)		5	—	100		
	AD (Aϕ)		5	—	50		

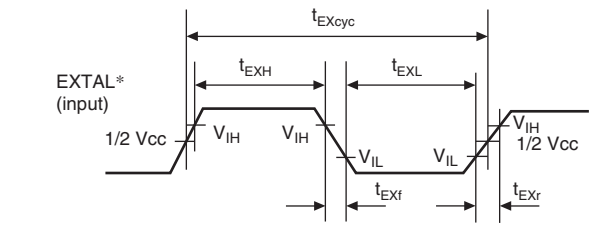
35.4.1 Clock Timing

Table 35.6 Clock Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	12.5	MHz	35.3
EXTAL clock input cycle time	t_{EXcyc}	80	100	ns	
EXTAL clock input low pulse width	t_{EXL}	20	—	ns	
EXTAL clock input high pulse width	t_{EXH}	20	—	ns	
EXTAL clock input rise time	t_{EXr}	—	5	ns	
EXTAL clock input fall time	t_{EXf}	—	5	ns	
CK clock output frequency	f_{OP}	5	50	MHz	35.4
CK clock output cycle time	tcyc	20	200	ns	
CK clock output low pulse width	t_{CKOL}	1/2 tcyc – 7.5	—	ns	
CK clock output high pulse width	t_{CKOH}	1/2 tcyc – 7.5	—	ns	
CK clock output rise time	t_{CKOr}	—	3	ns	
CK clock output fall time	t_{CKOf}	—	3	ns	
EXTAL32 clock input frequency	f_{EX32}	16	62.5	kHz	35.5
EXTAL32 clock input cycle time	$t_{EX32cyc}$	16	62.5	µs	
EXTAL32 clock input low pulse width	t_{EX32L}	8	—	µs	
EXTAL32 clock input high pulse width	t_{EX32H}	8	—	µs	

Item	Symbol	Min.	Max.	Unit	Figure
CK32 clock output frequency	t_{OP32}	16	62.5	kHz	35.6
CK32 clock output cycle time	t_{cyc32}	16	62.5	µs	
CK32 clock output low pulse width (when the external clock is in use)	t_{CK32OL}	1/2tcyc32 – 1	—	µs	
CK32 clock output low pulse width (when the crystal oscillator is in use)	t_{CK32OL}	tcyc32 × 0.6 – 1	tcyc32 × 0.8 – 1	µs	
CK32 clock output high pulse width (when the external clock is in use)	t_{CK32OH}	1/2tcyc32 – 1	—	µs	
CK32 clock output high pulse width (when the crystal oscillator is in use)	t_{CK32OH}	tcyc32 × 0.2 – 1	tcyc32 × 0.4 – 1	µs	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	35.7
Oscillation settling time on return from standby 1	t_{OSC2}	10	—	ms	35.8
Oscillation settling time on return from standby 2	t_{OSC3}	10	—	ms	35.9
EXTAL32 clock oscillation settling time (when the crystal oscillator is in use)	t_{OSC32}	3	—	s	35.10



Note: * When the clock is input on the EXTAL pin.

Figure 35.3 EXTAL Clock Input Timing

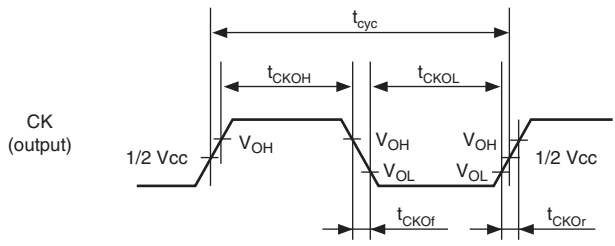
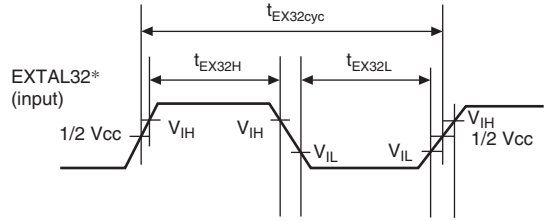


Figure 35.4 CK Clock Output Timing



Note: * When the clock is input on the EXTAL32 pin.

Figure 35.5 EXTAL32 Clock Input Timing

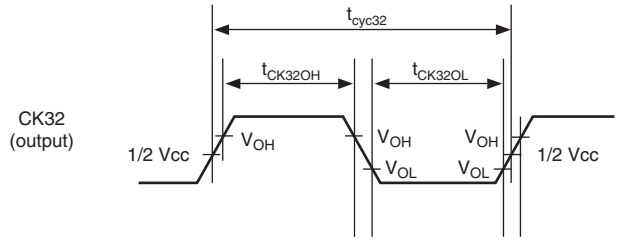
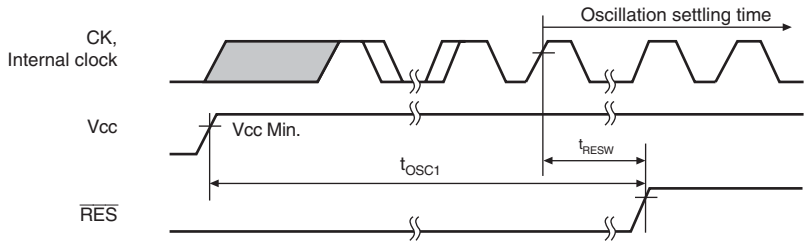


Figure 35.6 CK32 Clock Output Timing



Note: Oscillation settling time when the internal oscillator is used.

Figure 35.7 Power-On Oscillation Settling Time

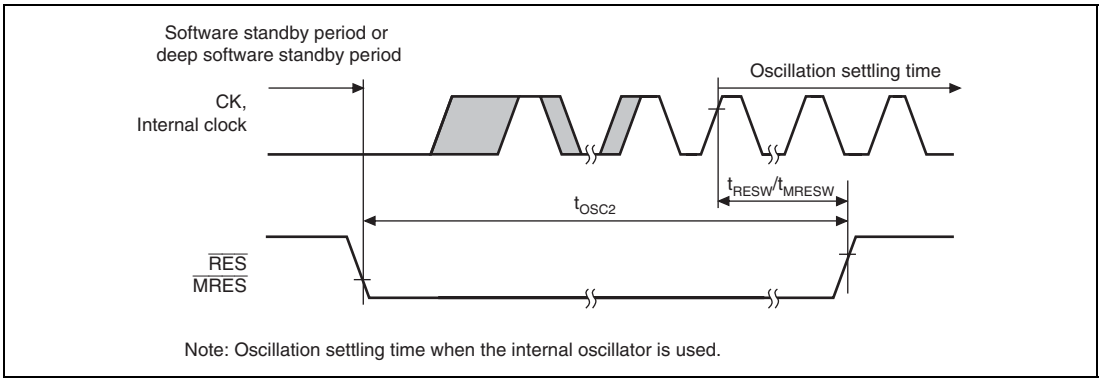


Figure 35.8 Oscillation Settling Time on Return from Standby (Return by Reset)

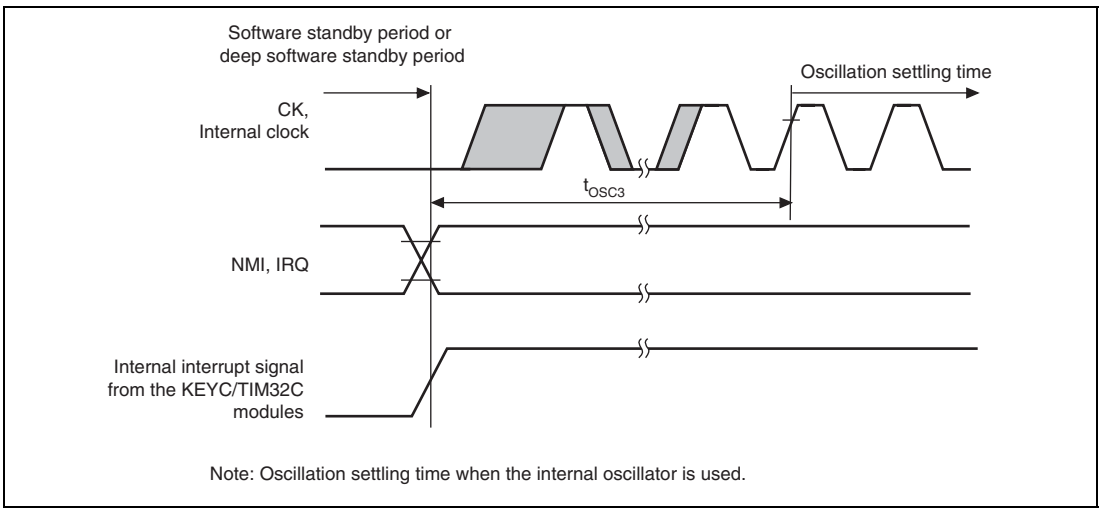


Figure 35.9 Oscillation Settling Time on Return from Standby (Return by an Interrupt)

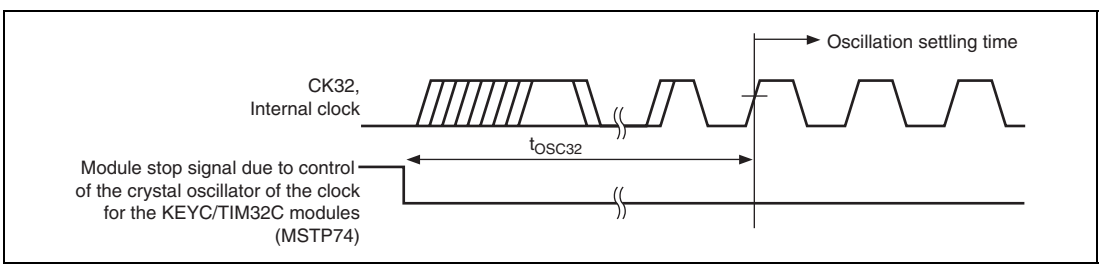


Figure 35.10 EXTAL32 Clock Oscillation Settling Time

35.4.2 Control Signal Timing

Table 35.7 Control Signal Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{RES}}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*[2]*[4]	—	t_{cyc}	35.7, 35.8, 35.11, 35.12
		1.5*[4]	—	µs	
$\overline{\text{RES}}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100	—	µs	
$\overline{\text{RES}}$ setup time*[1]	t_{RESS}	200	—	ns	
$\overline{\text{RES}}$ hold time	t_{RESH}	15	—	ns	
$\overline{\text{MRES}}$ pulse width	t_{MRESW}	20*[3]	—	t_{cyc}	35.8, 35.11, 35.12
$\overline{\text{MRES}}$ setup time*[1]	t_{MRESS}	25	—	ns	
$\overline{\text{MRES}}$ hold time	t_{MRESH}	15	—	ns	
MD1, MD0, FWE setup time	t_{MDS}	20	—	t_{cyc}	35.11
$\overline{\text{BREQ}}$ setup time*[1]	t_{BREQS}	$1/2t_{cyc}$ + 15	—	ns	35.13
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 10	—	ns	
NMI pulse width	t_{NMIW}	20	—	t_{cyc}	35.12
NMI setup time*[1]	t_{NMIS}	100	—	ns	
NMI hold time	t_{NMIH}	10	—	ns	
IRQ23 to IRQ0 pulse width	t_{IRQW}	20	—	t_{cyc}	
IRQ23 to IRQ0 setup time*[1]	t_{IRQS}	35	—	ns	
IRQ23 to IRQ0 hold time	t_{IRQH}	10	—	ns	
$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output delay time	t_{IRQOD}	—	100	ns	35.14
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	$1/2t_{cyc}$ + 20	ns	35.13
Bus-buffer off time	t_{BOFF}	0	100	ns	
Bus buffer on time	t_{BON}	0	100	ns	

Notes: 1. $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, $\overline{\text{BREQ}}$, and IRQ23 to IRQ0 are asynchronous signals. When these setup times are observed, a change of these signals is detected at the clock rising edge. If the setup times are not observed, detection of a signal change may be delayed until the next rising edge of the clock.

2. On return from standby, $t_{RESW} = t_{OSC2}$ (10 ms).
3. On return from standby, $t_{MRESW} = t_{OSC2}$ (10 ms).
4. Input the reset pulse over t_{RESW1} so that all conditions are met.

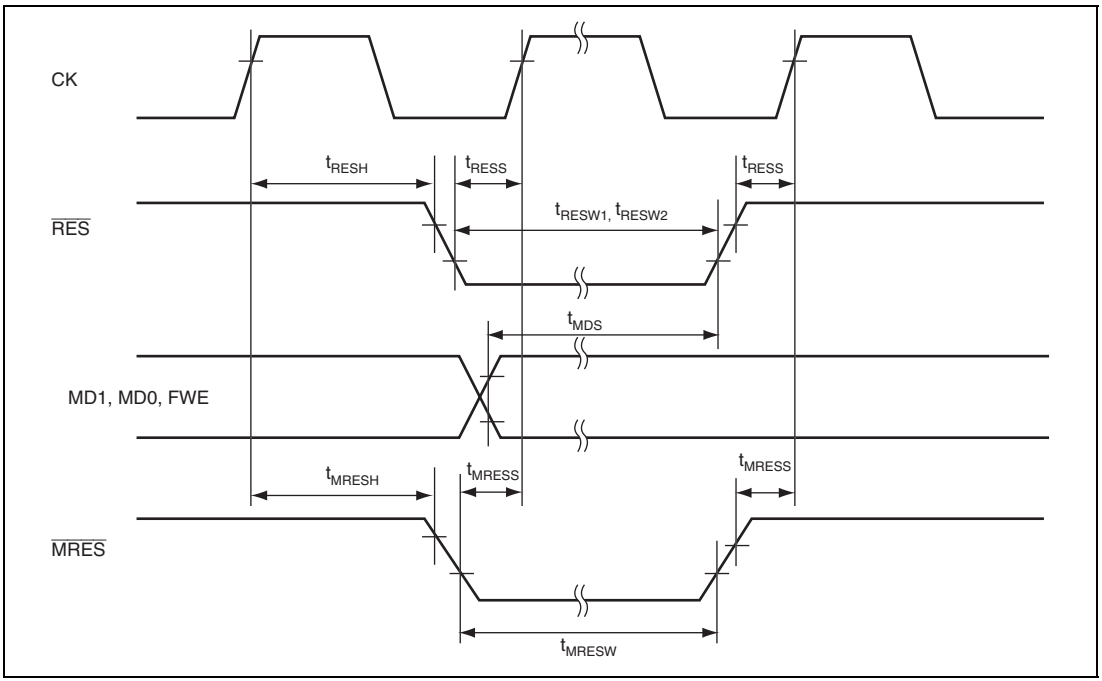


Figure 35.11 Reset Input Timing

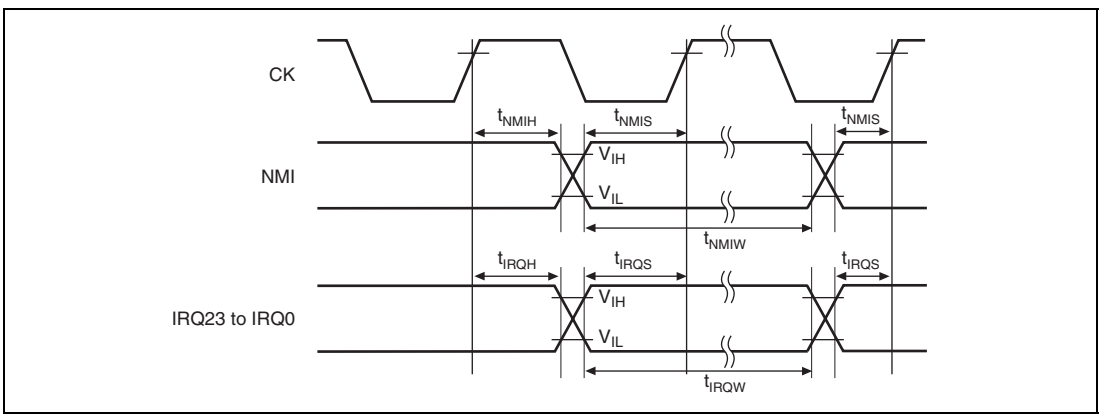


Figure 35.12 Interrupt Signal Input Timing

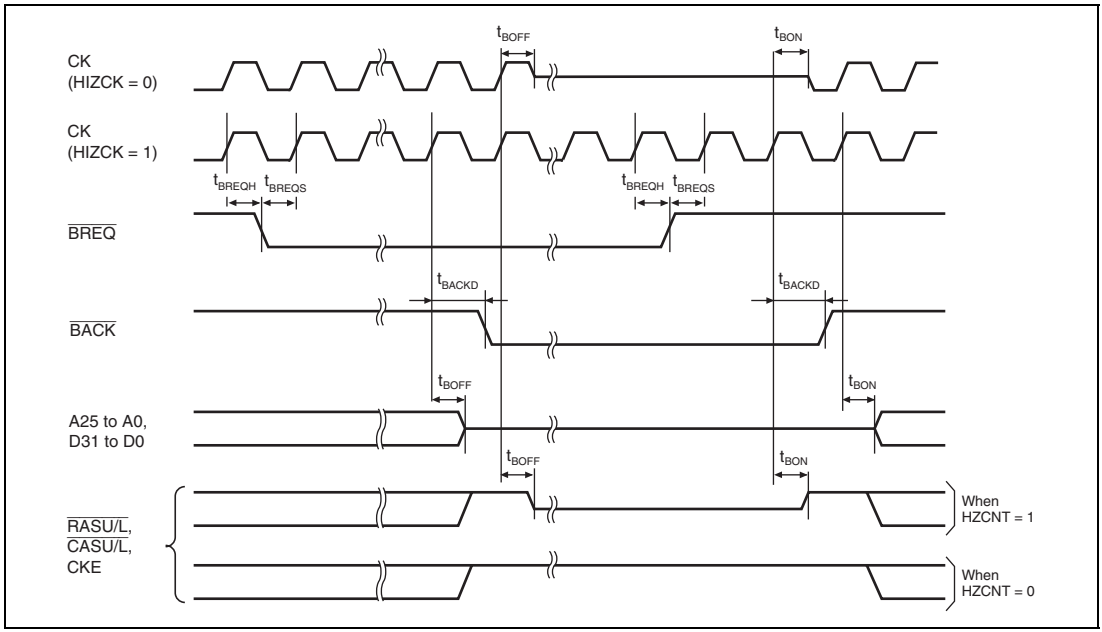


Figure 35.13 Bus Release Timing

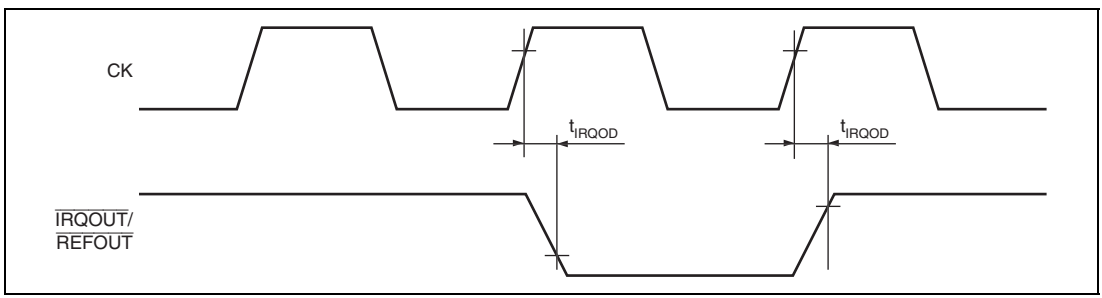


Figure 35.14 Interrupt Signal Output Timing

35.4.3 Bus Timing

Table 35.8 Bus Timing

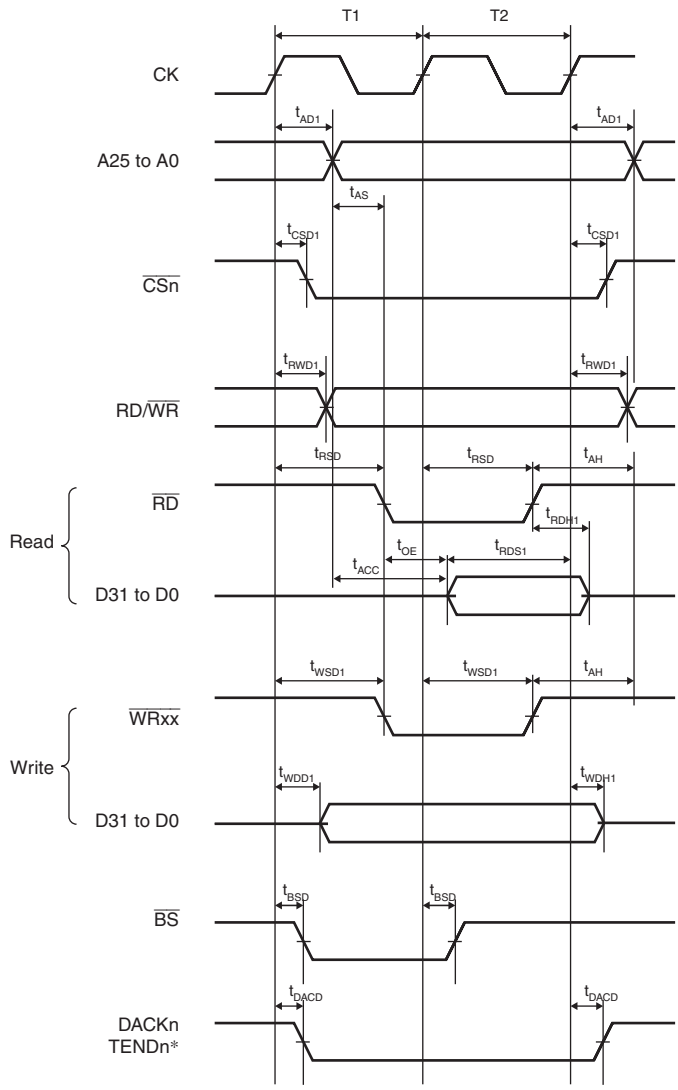
Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Address delay time 1	t_{AD1}	1	18	ns	35.15 to 35.39
Address delay time 2	t_{AD2}	1/2tcyc	1/2tcyc + 18	ns	35.22
Address delay time 3	t_{AD3}	1/2tcyc	1/2tcyc + 18	ns	35.40, 35.41
Address setup time	t_{AS}	0	—	ns	35.15 to 35.18, 35.22
Address hold time	t_{AH}	0	—	ns	35.15 to 35.18
$\overline{BS}$ delay time	t_{BSD}	—	18	ns	35.15 to 35.36, 35.40
$\overline{CS}$ delay time 1	t_{CSD1}	1	18	ns	35.15 to 35.39
$\overline{CS}$ delay time 2	t_{CSD2}	1/2tcyc	1/2tcyc + 18	ns	35.40, 35.41
Read write delay time 1	t_{RWD1}	1	18	ns	35.15 to 35.39
Read write delay time 2	t_{RWD2}	1/2tcyc	1/2tcyc + 18	ns	35.40, 35.41
Read strobe delay time	t_{RSD}	1/2tcyc	1/2tcyc + 18	ns	35.15 to 35.22
Read data setup time 1	t_{RDS1}	1/2tcyc + 14	—	ns	35.15 to 35.21
Read data setup time 2	t_{RDS2}	14	—	ns	35.23 to 35.26, 35.31 to 35.33
Read data setup time 3	t_{RDS3}	1/2tcyc + 14	—	ns	35.22
Read data setup time 4	t_{RDS4}	1/2tcyc + 14	—	ns	35.40
Read data hold time 1	t_{RDH1}	0	—	ns	35.15 to 35.21
Read data hold time 2	t_{RDH2}	2	—	ns	35.23 to 35.26, 35.31 to 35.33
Read data hold time 3	t_{RDH3}	0	—	ns	35.22
Read data hold time 4	t_{RDH4}	1/2tcyc + 5	—	ns	35.40
Read data access time	t_{ACC}*1	tcyc × (n + 1.5) – 33*2	—	—	35.15 to 35.18, 35.20, 35.21
Access time from read strobe	t_{OE}*1	tcyc × (n + 1) – 33*2	—	—	35.15 to 35.18, 35.20, 35.21
Write strobe delay time 1	t_{WSD1}	1/2tcyc	1/2tcyc + 18	ns	35.15 to 35.20

Item	Symbol	Min.	Max.	Unit	Figure
Write strobe delay time 2	t_{WSD2}	—	18	ns	35.21
Write data delay time 1	t_{WDD1}	—	18	ns	35.15 to 35.21
Write data delay time 2	t_{WDD2}	—	18	ns	35.27 to 35.30, 35.35 and 35.36
Write data delay time 3	t_{WDD3}	—	1/2tcyc + 18	ns	35.40
Write data hold time 1	t_{WDH1}	1	—	ns	35.15 to 35.21
Write data hold time 2	t_{WDH2}	1	—	ns	35.27 to 35.30, 35.34 to 35.36
Write data hold time 3	t_{WDH3}	1/2tcyc	—	ns	35.40
$\overline{WAIT}$ setup time	t_{WTS}	1/2tcyc + 18	—	ns	35.16 to 35.22
$\overline{WAIT}$ hold time	t_{WTH}	1/2tcyc + 2	—	ns	35.16 to 35.22
$\overline{RAS}$ delay time 1	t_{RASD1}	1	18	ns	35.23 to 35.34, 35.36 to 35.39
$\overline{RAS}$ delay time 2	t_{RASD2}	1/2tcyc	1/2tcyc + 18	ns	35.40, 35.41
$\overline{CAS}$ delay time 1	t_{CASD1}	1	18	ns	35.23 to 35.39
$\overline{CAS}$ delay time 2	t_{CASD2}	1/2tcyc	1/2tcyc + 18	ns	35.40, 35.41
DQM delay time 1	t_{DQMD1}	1	18	ns	35.23 to 35.36
DQM delay time 2	t_{DQMD2}	1/2tcyc	1/2tcyc + 18	ns	35.40, 35.41
CKE delay time 1	t_{CKED1}	1	18	ns	35.38
CKE delay time 2	t_{CKED2}	1/2tcyc	1/2tcyc + 18	ns	35.41
$\overline{AH}$ delay time	t_{AHD}	1/2tcyc	1/2tcyc + 18	ns	35.19
Multiplexed address delay time	t_{MAD}	—	18	ns	35.19
Multiplexed address hold time	t_{MAH}	1	—	ns	35.19
DACK, TEND delay time	t_{DACD}	—	Depending on the timing of the DMAC	ns	35.15 to 35.36, 35.40

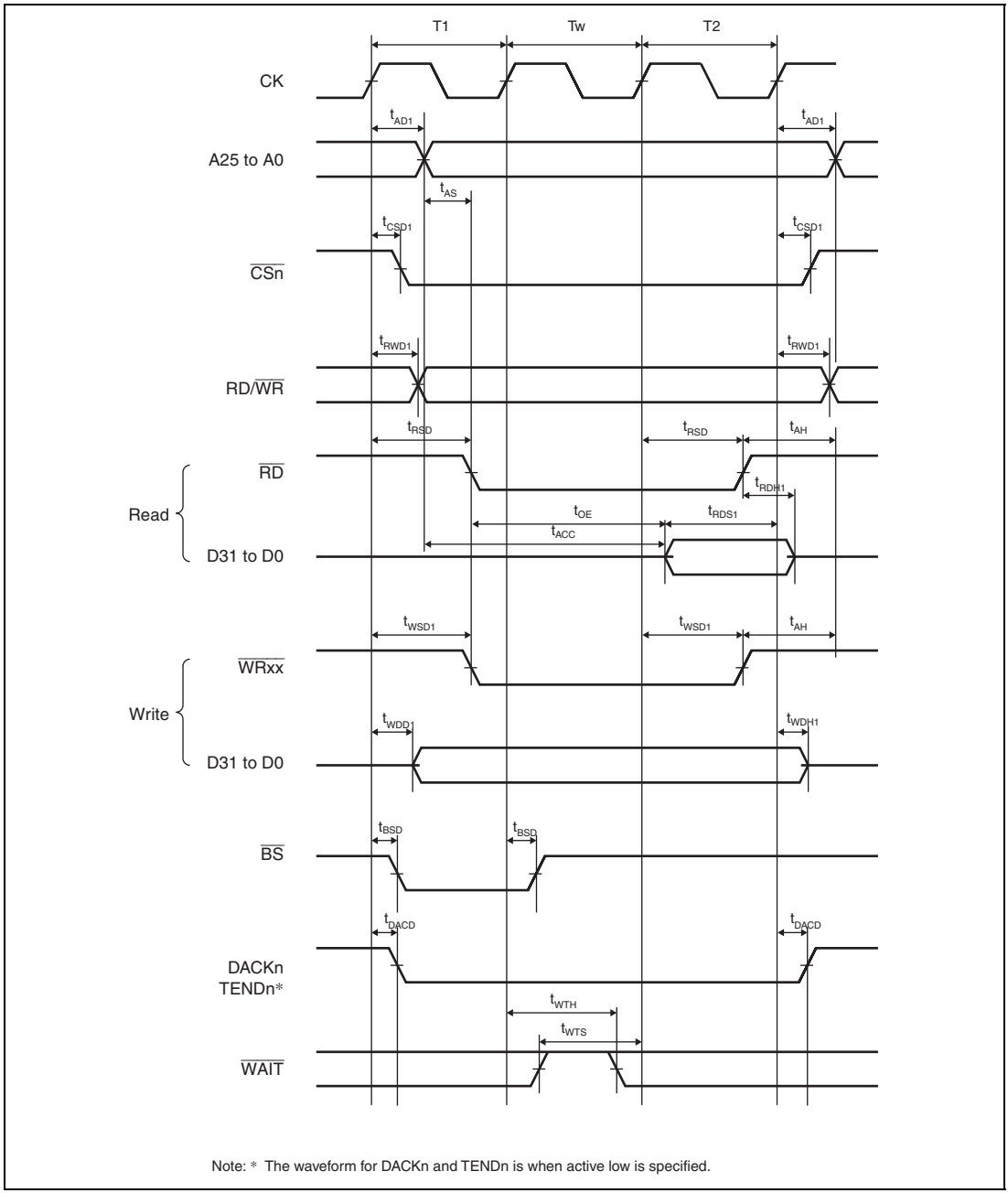
Notes: The maximum value (f_{max}) of Bϕ (external bus clock) depends on the number of wait cycles and the system configuration of your board. 1/2tcyc indicated in minimum and maximum values for the item of delay, setup, and hold times represents a half cycle from the rising edge with a clock. That is, 1/2tcyc describes a reference of the falling edge with a clock.

1. When access-time requirement is satisfied, t_{RDS1} need not be satisfied.
2. n represents the number of wait cycles.



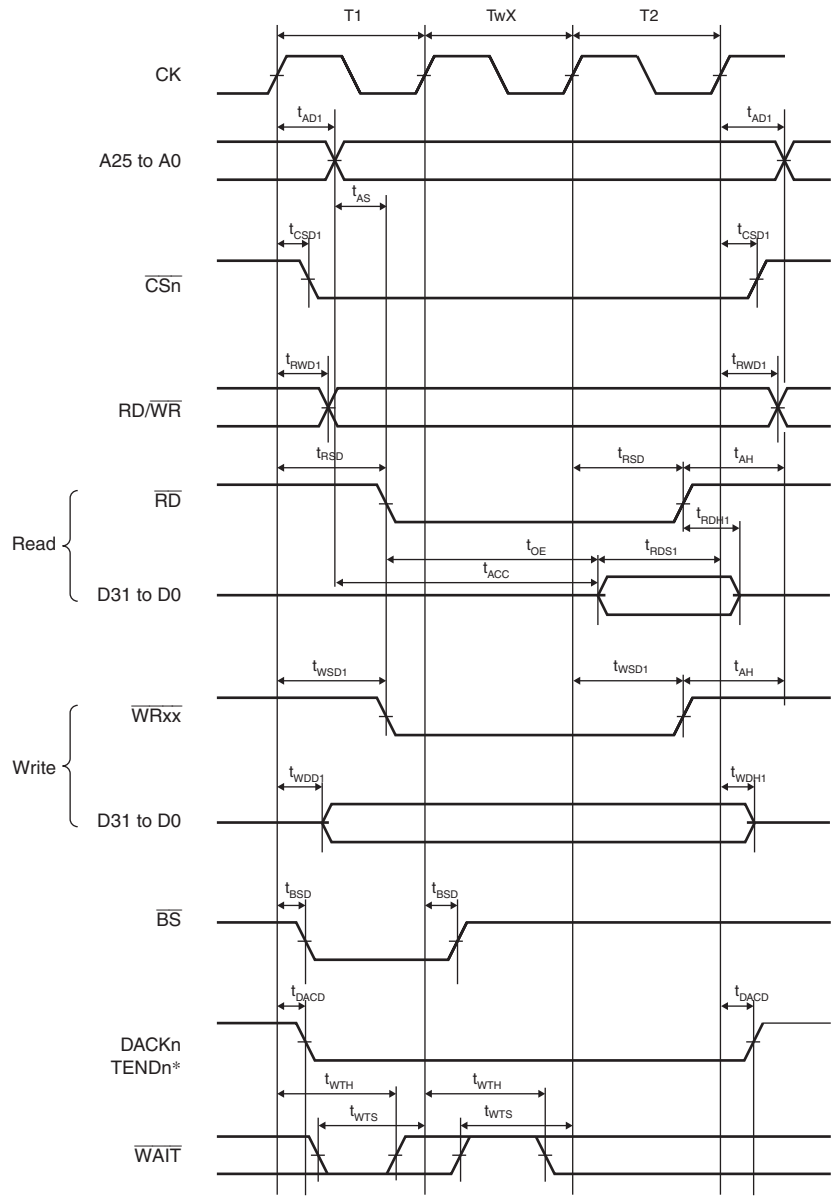
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.15 Basic Bus Timing for Normal Space (No Wait)



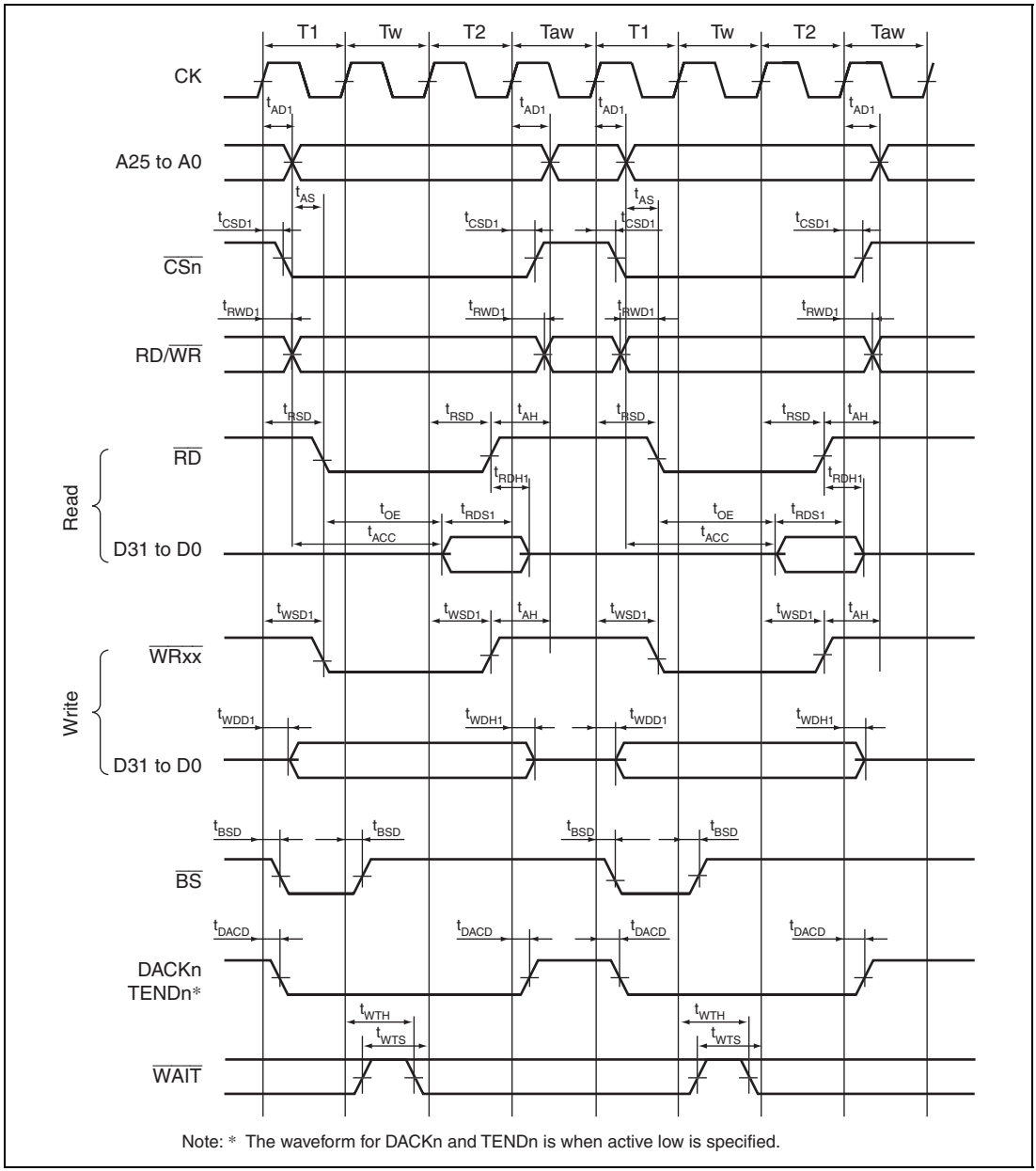
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.16 Basic Bus Timing for Normal Space (One Software Wait Cycle)



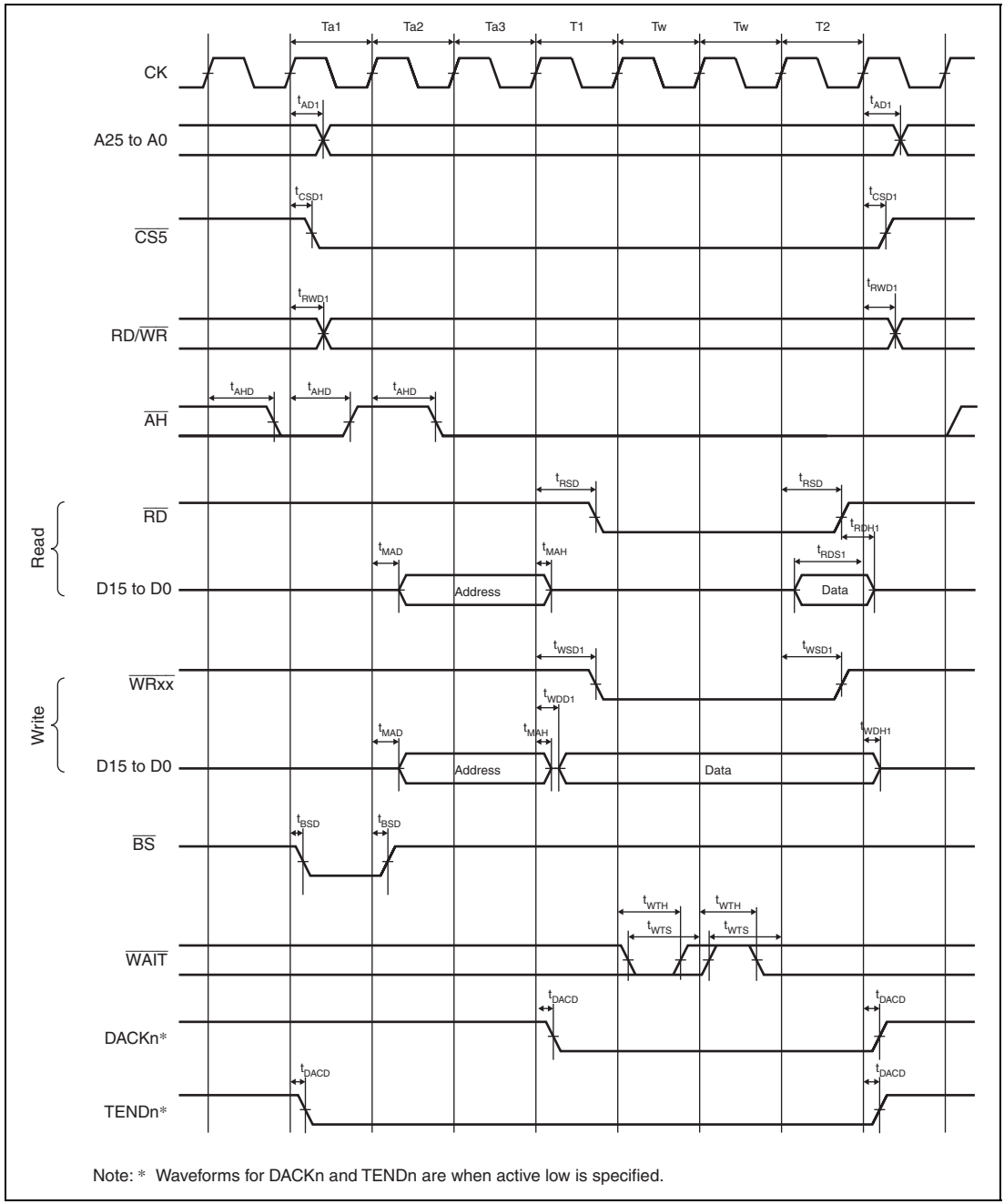
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.17 Basic Bus Timing for Normal Space (One External Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.18 Basic Bus Timing for Normal Space
(One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)



Note: * Waveforms for DACKn and TENDn are when active low is specified.

Figure 35.19 MPX-I/O Interface Bus Cycle
(Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)

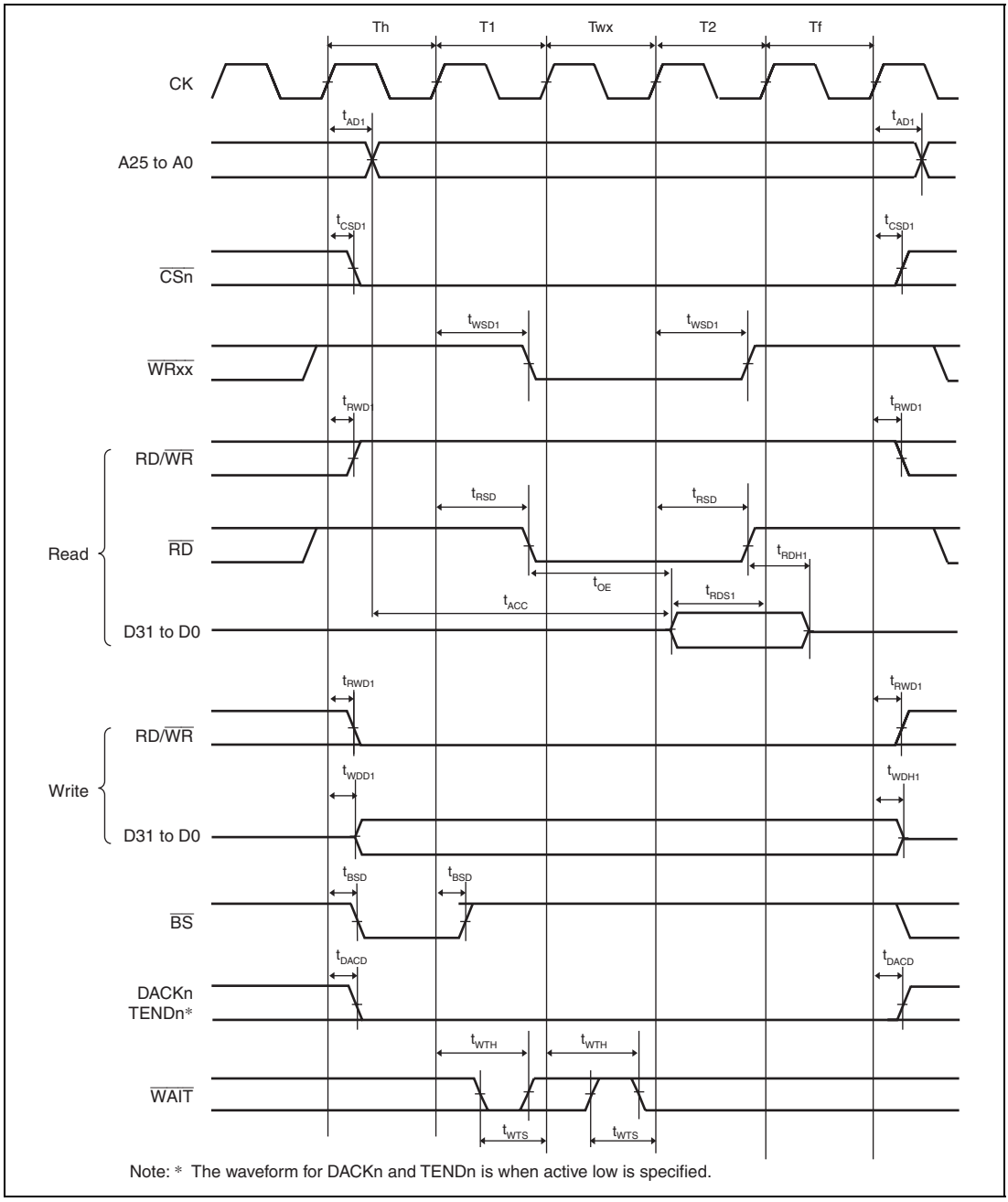
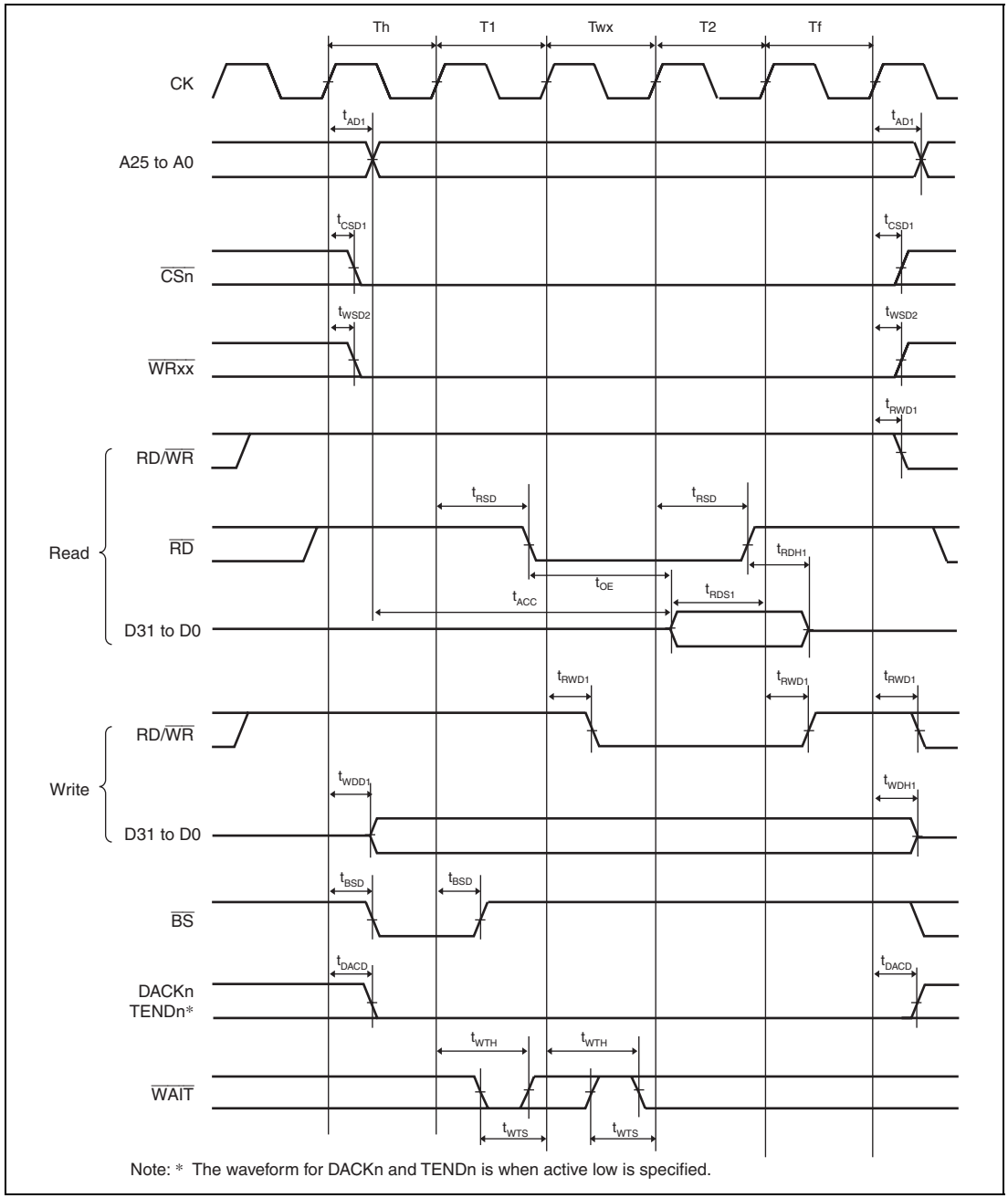
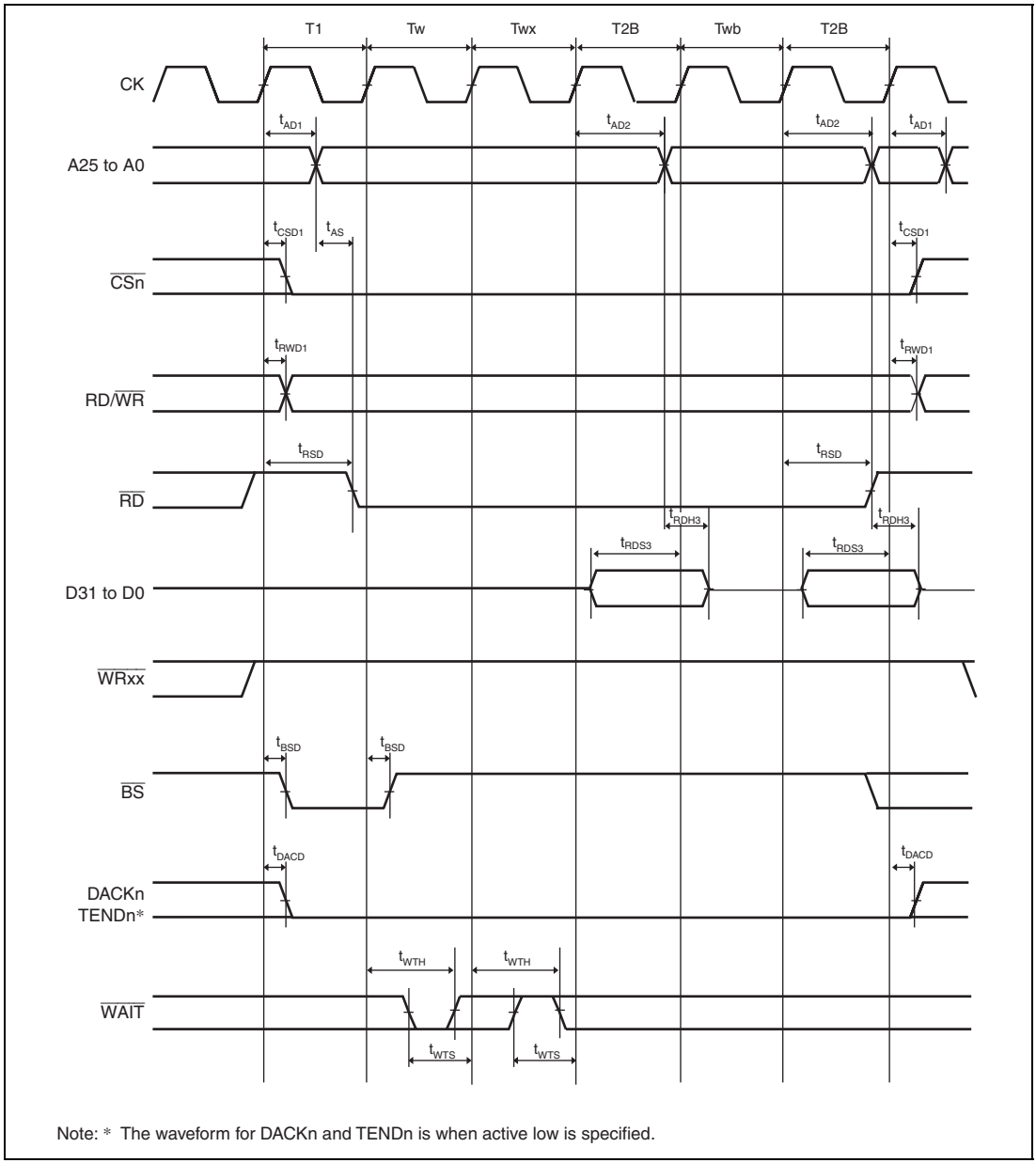


Figure 35.20 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))



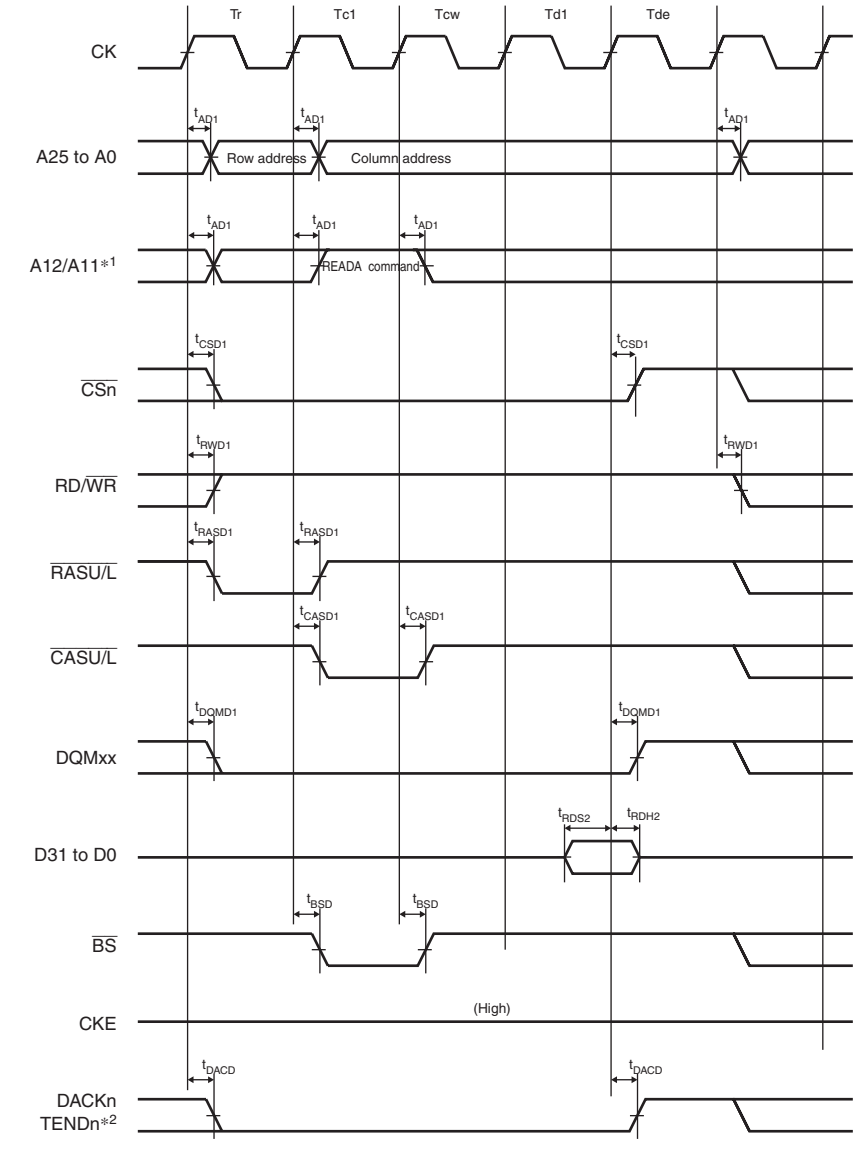
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.21 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))



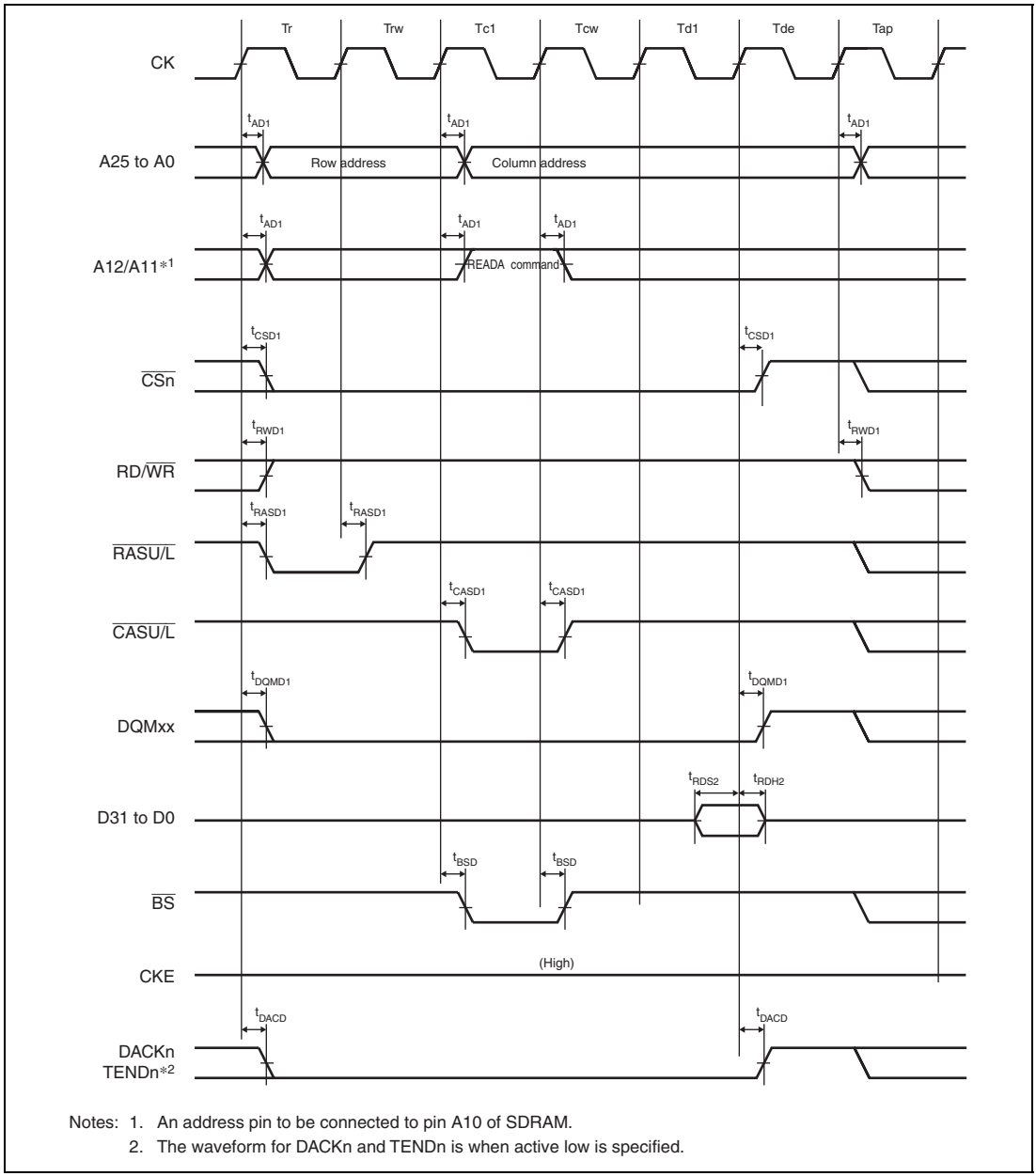
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.22 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two-Cycle Burst)



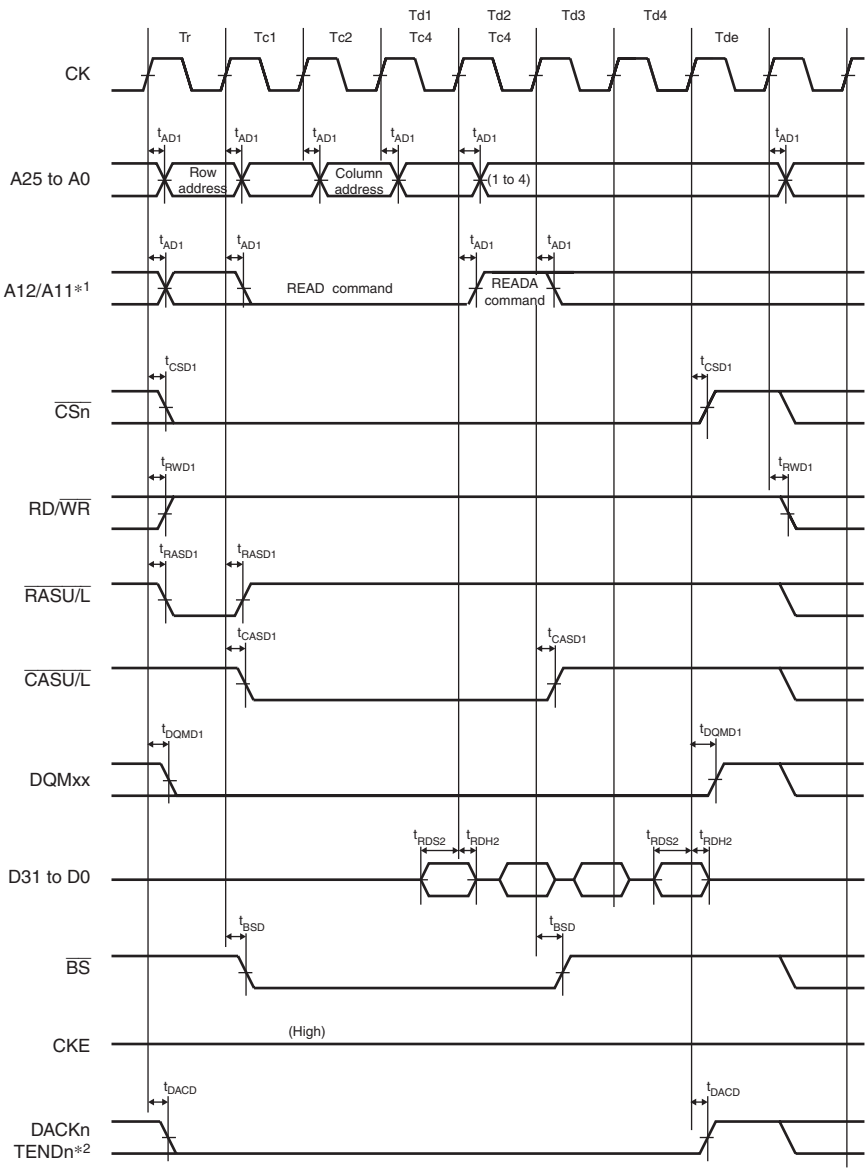
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



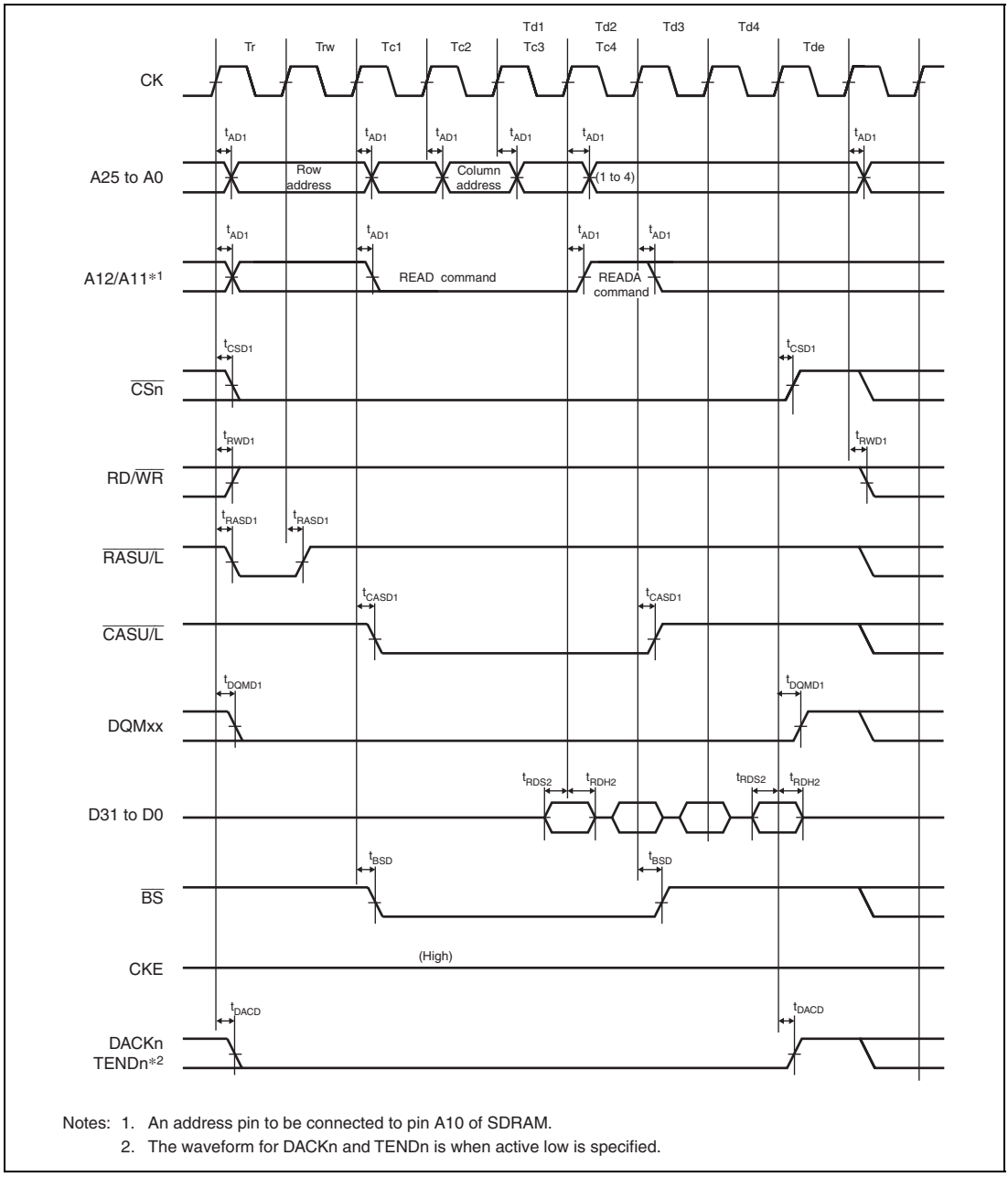
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.24 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



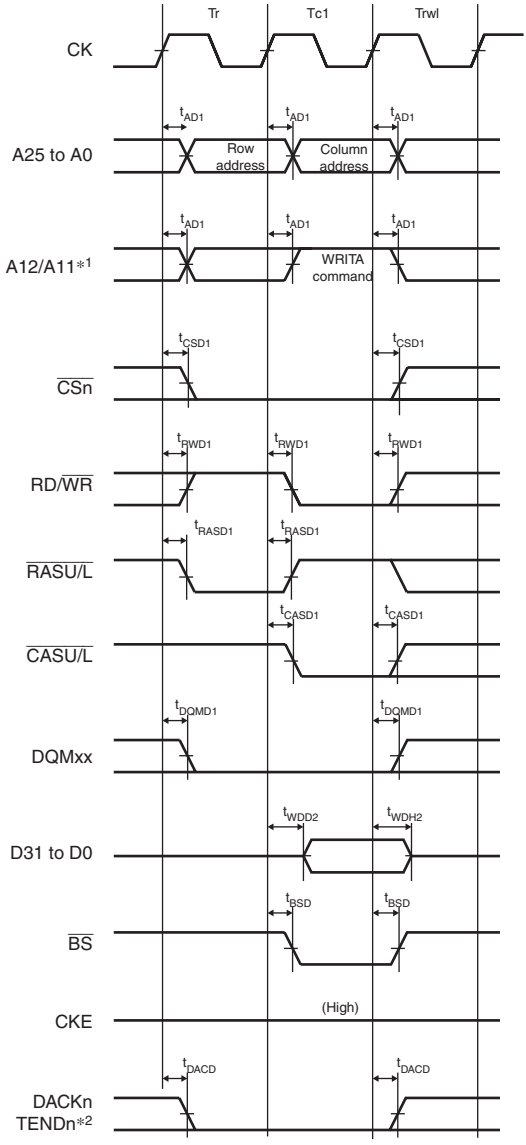
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.25 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



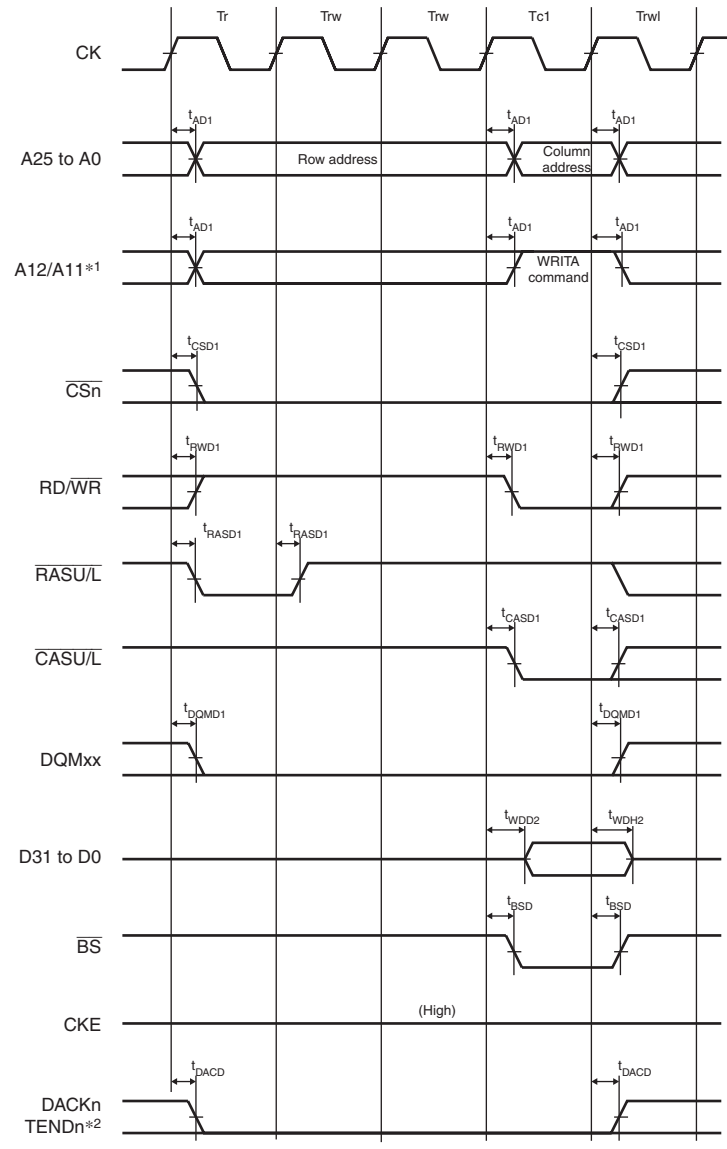
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.26 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



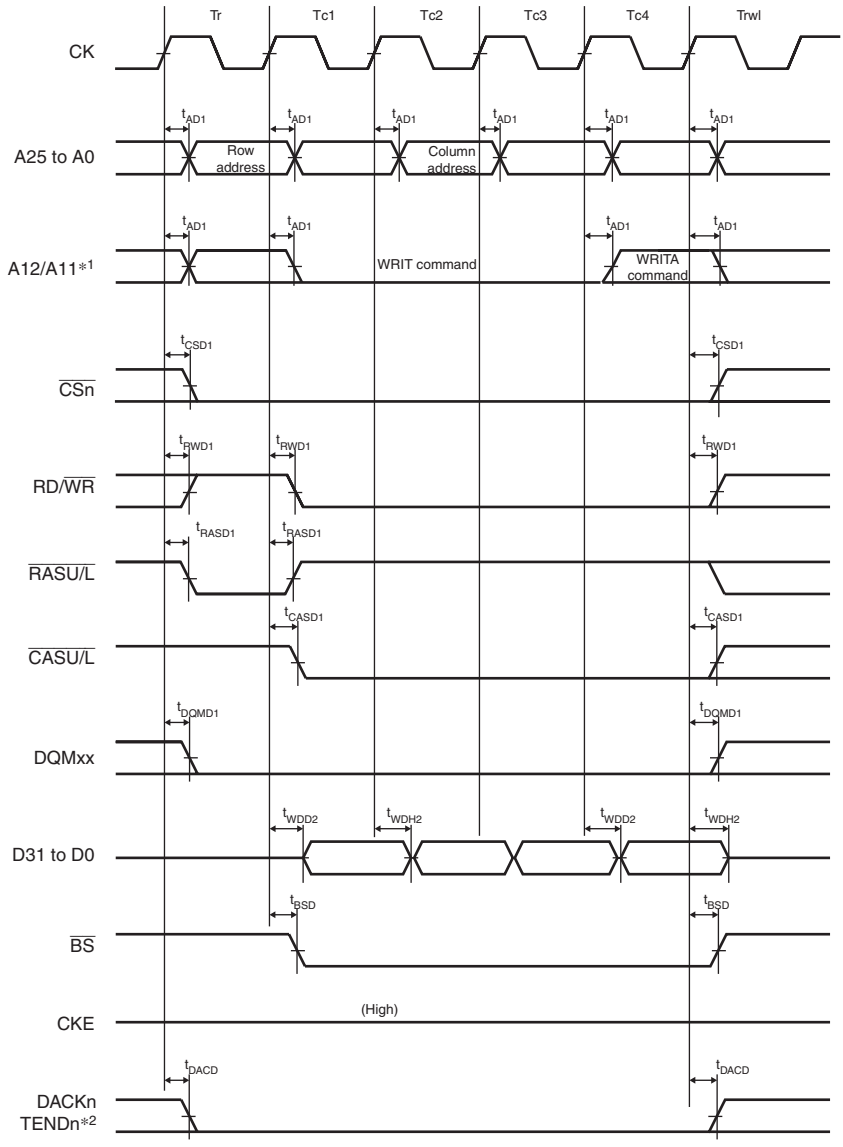
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)



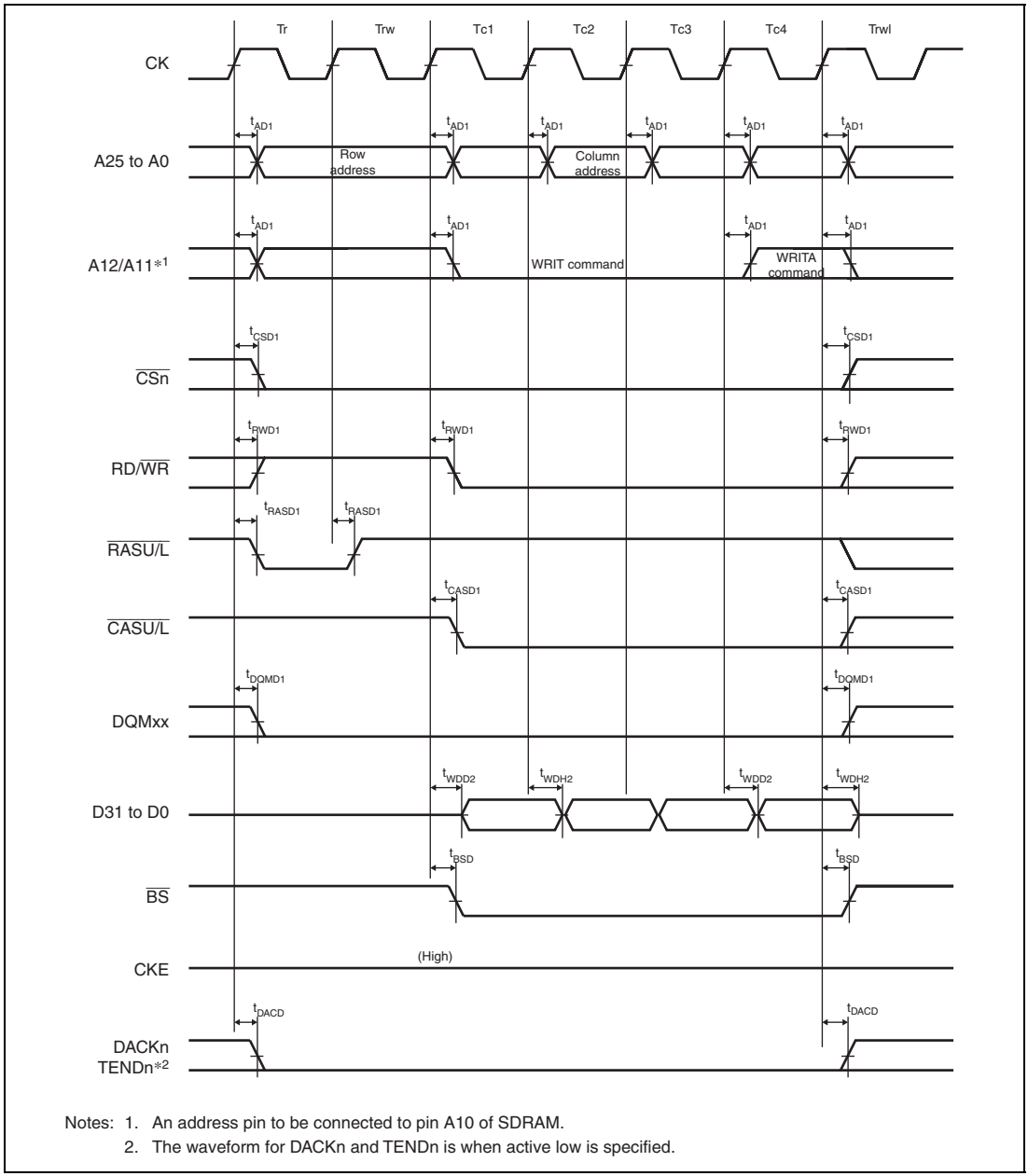
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.28 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



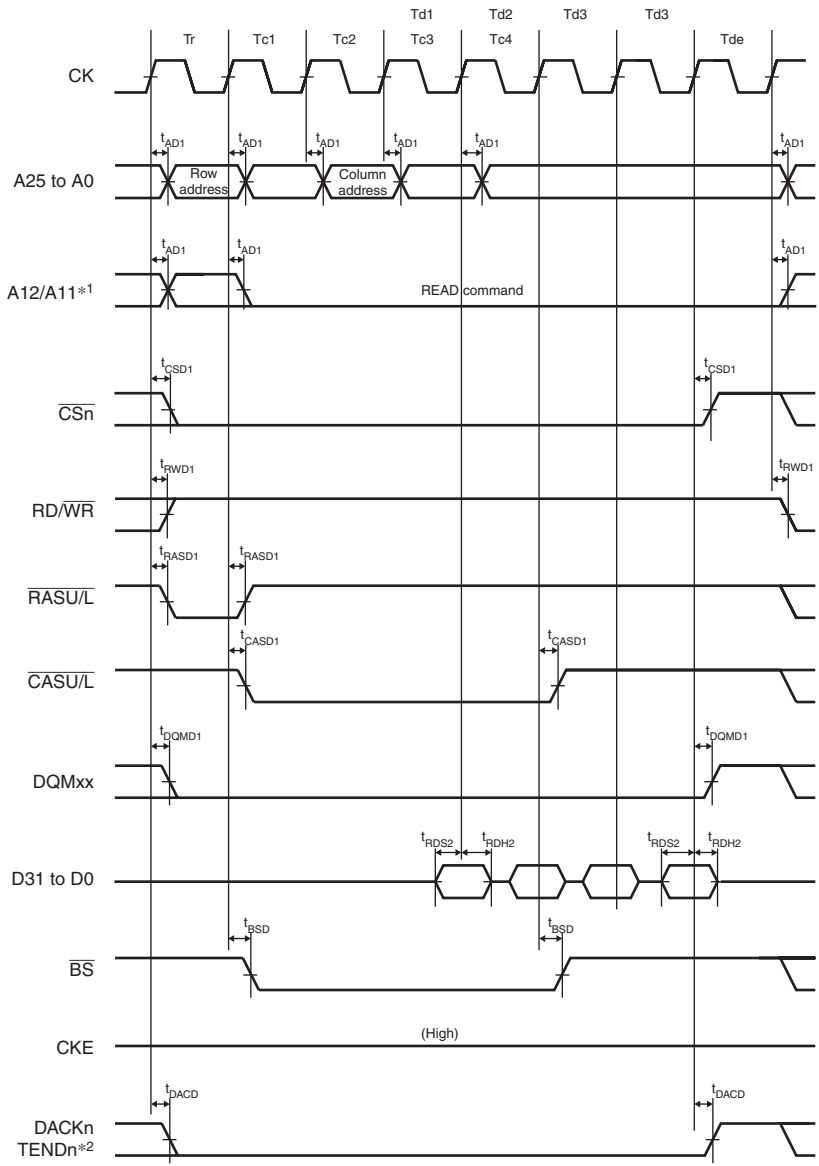
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.29 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



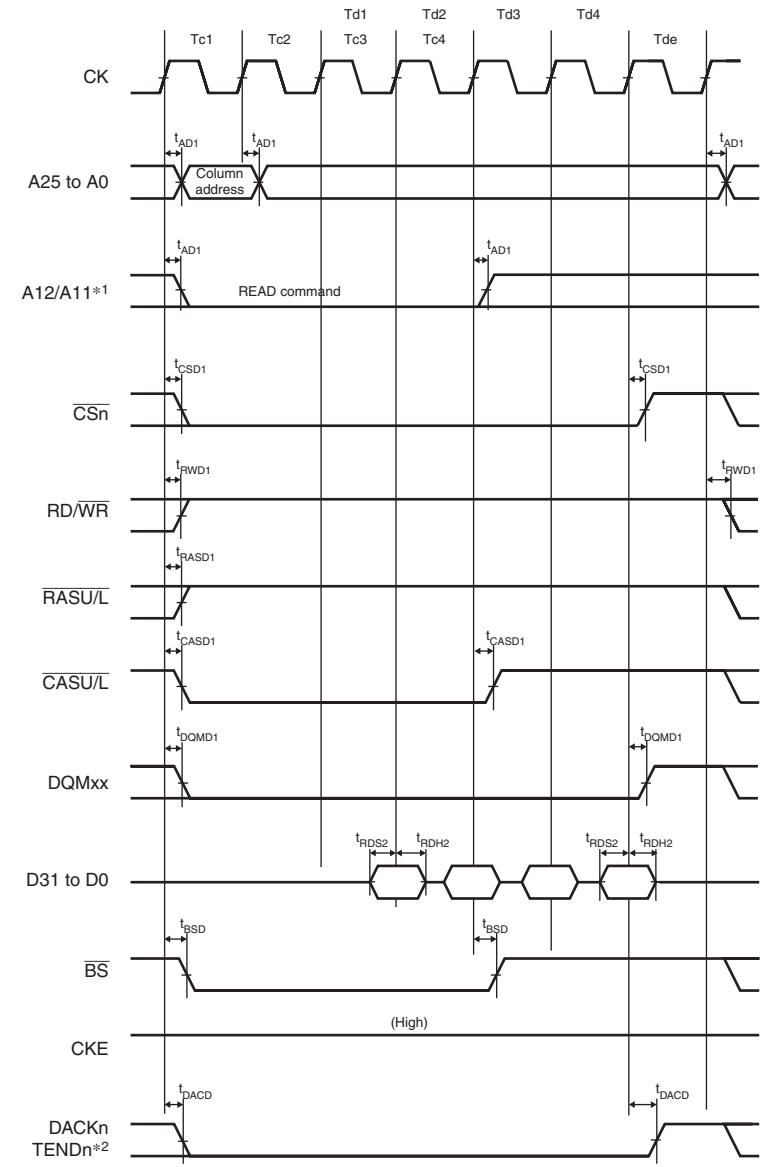
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.30 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



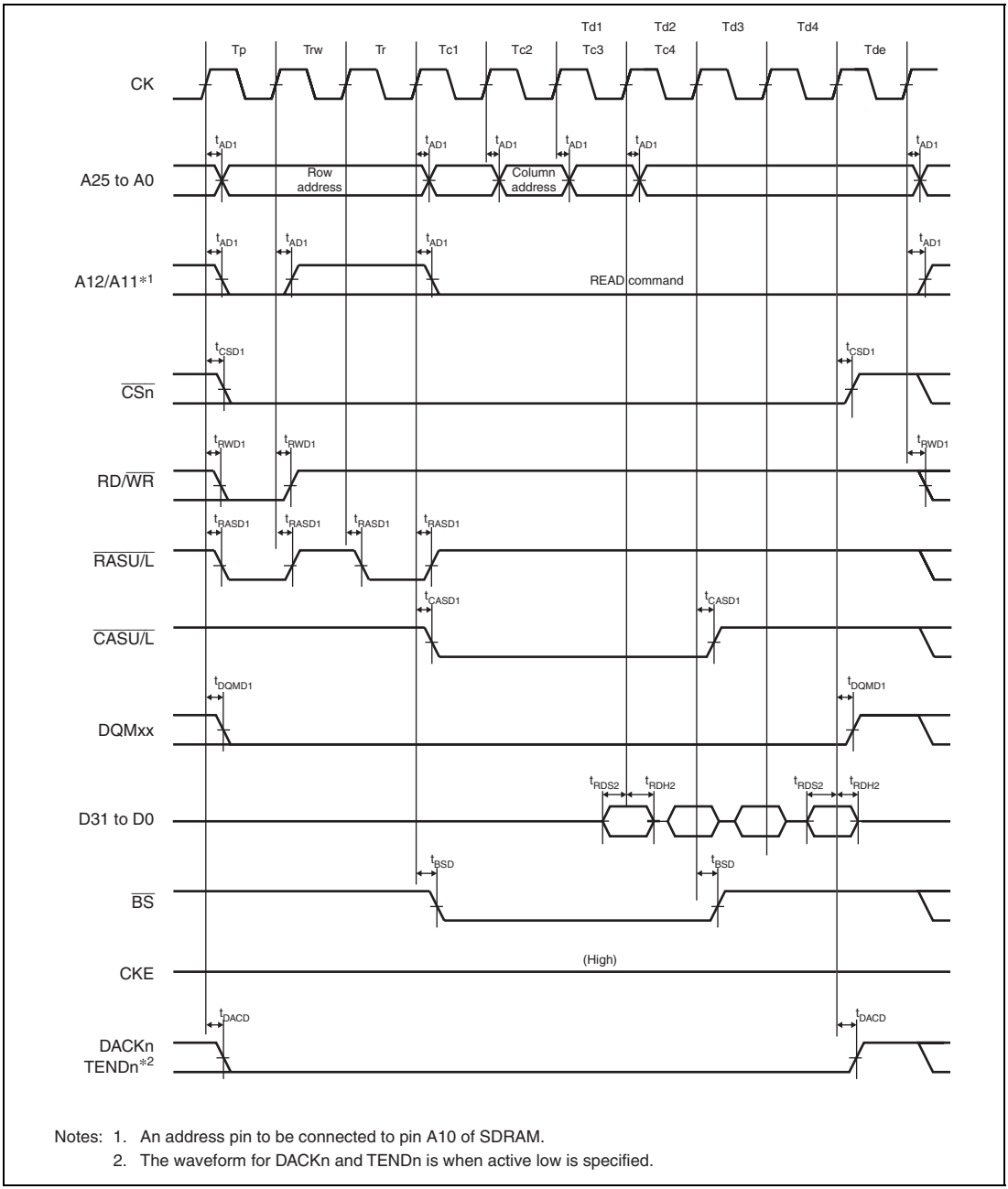
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.31 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)



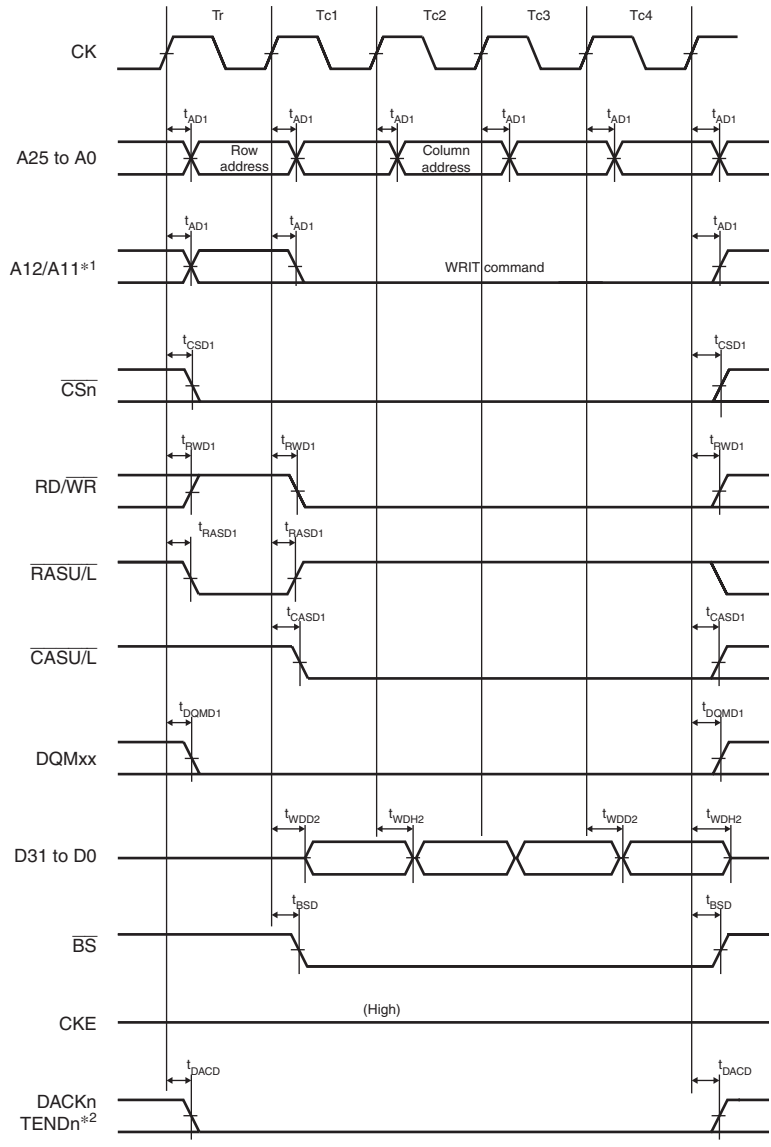
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.32 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)



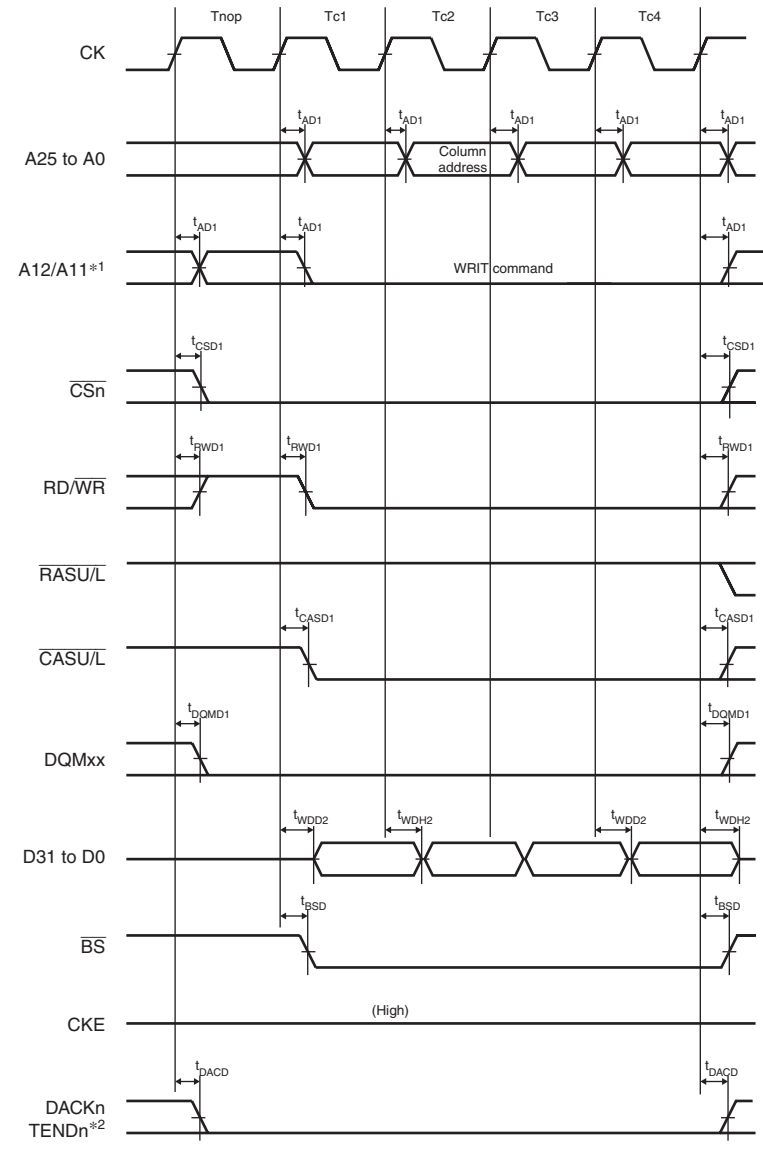
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.33 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency 2, WTRCD = 0 Cycle)



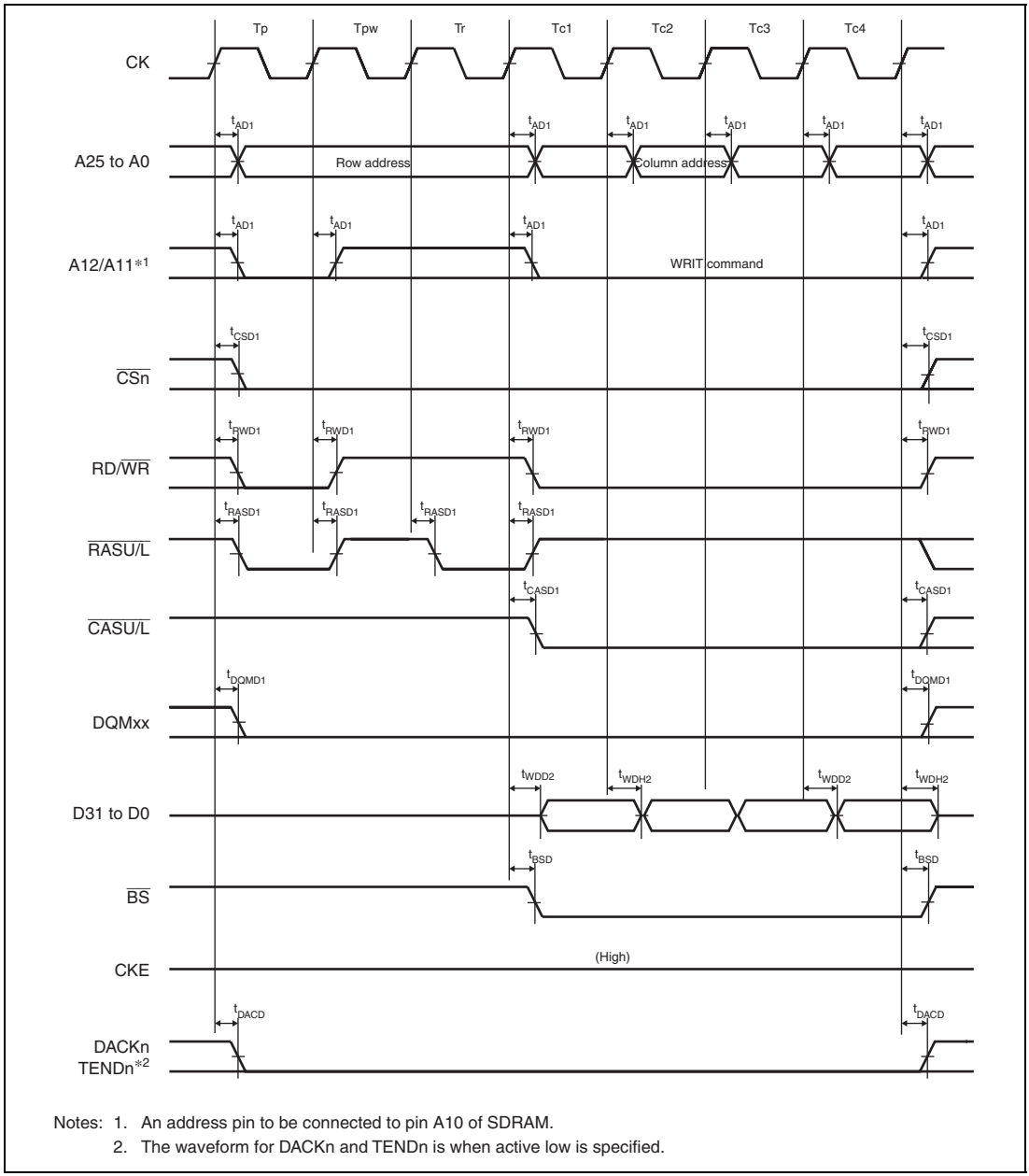
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.34 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles)
(Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



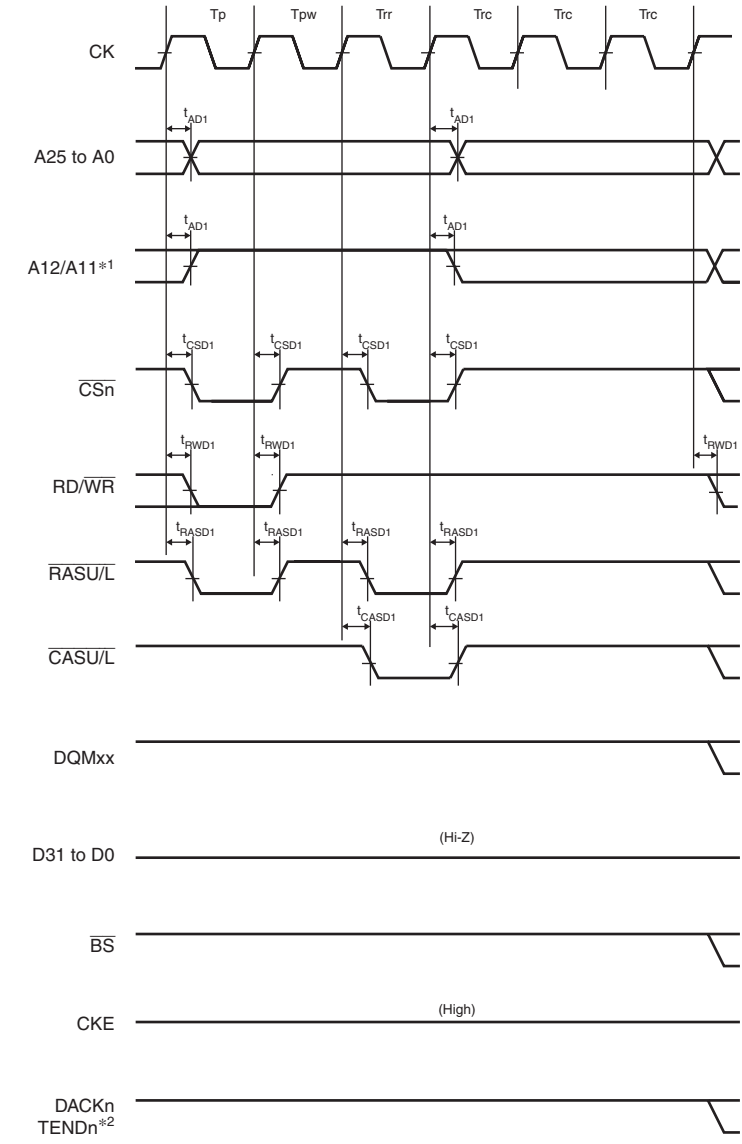
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.35 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)



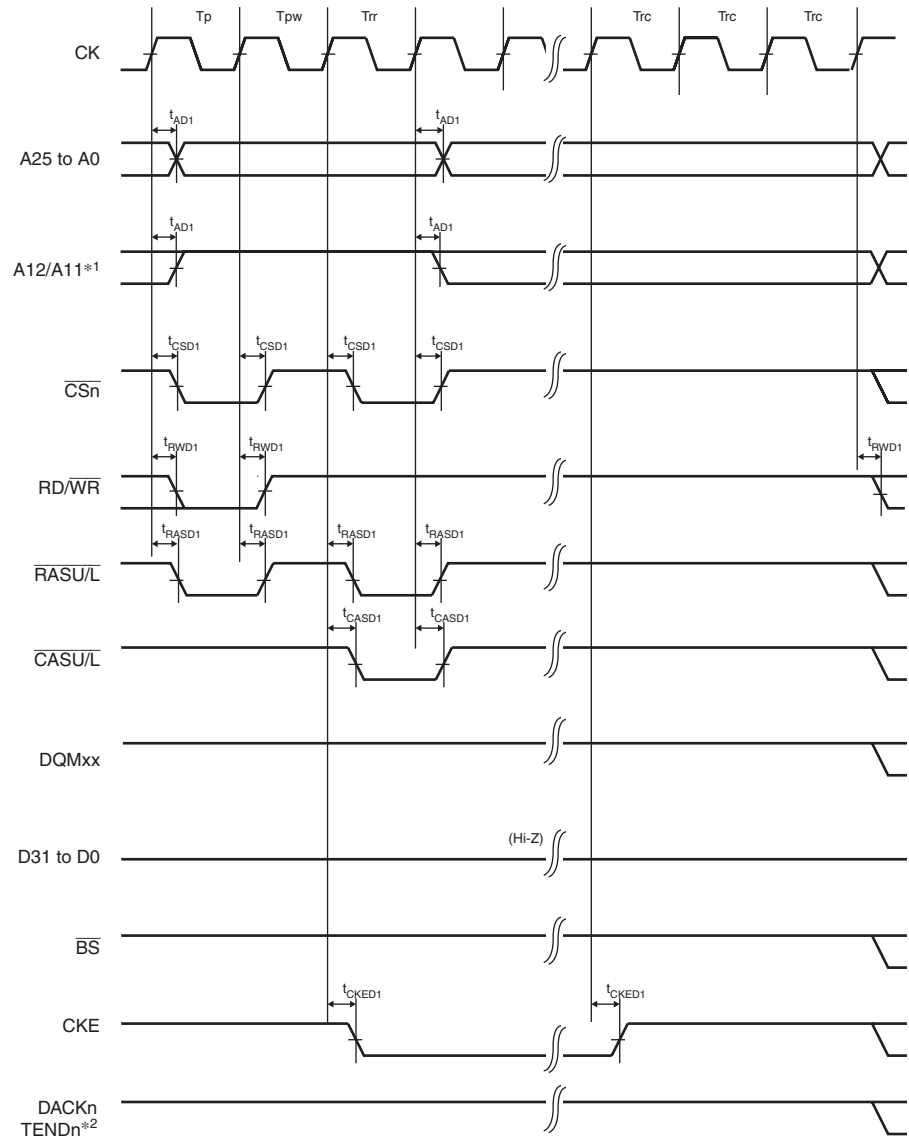
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.36 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)



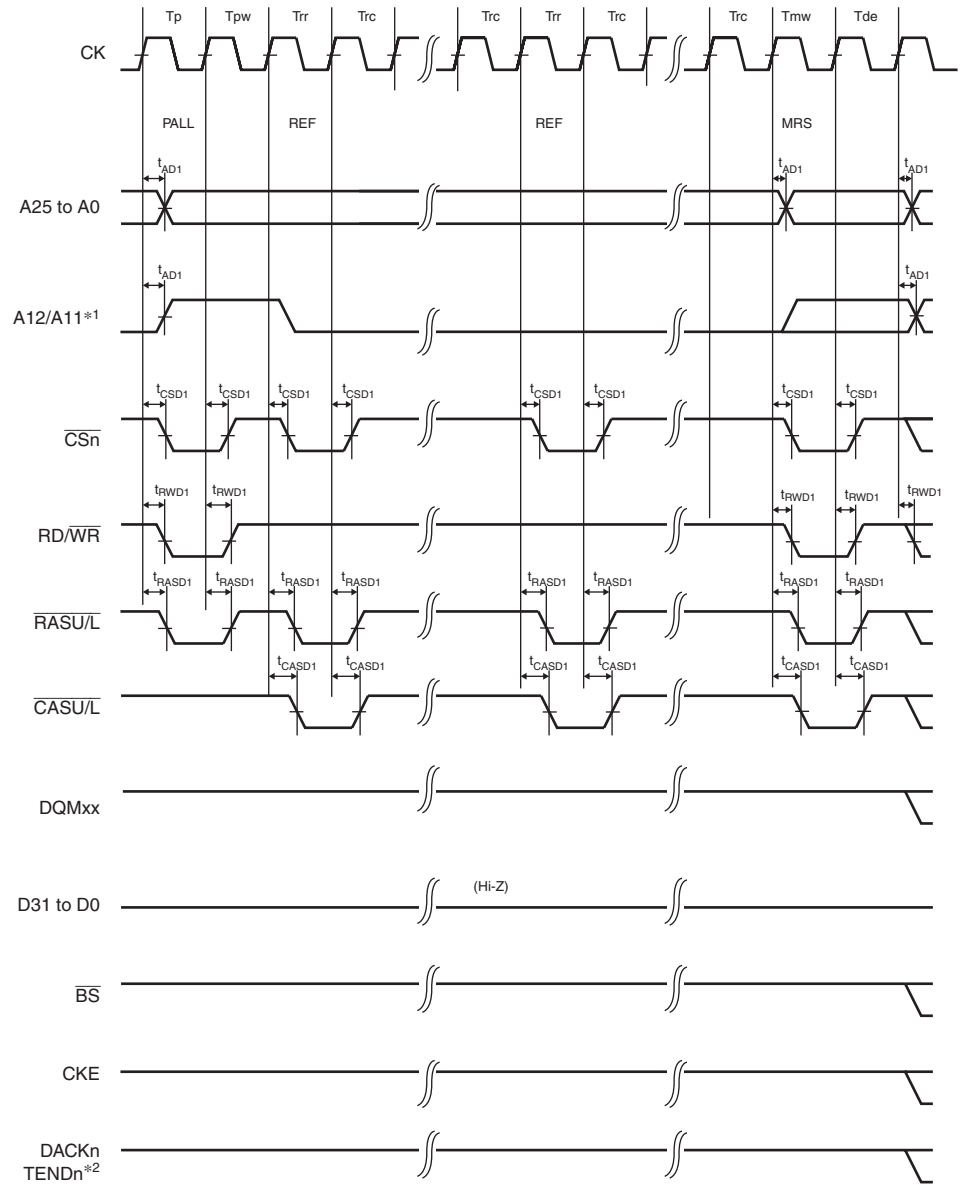
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.37 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



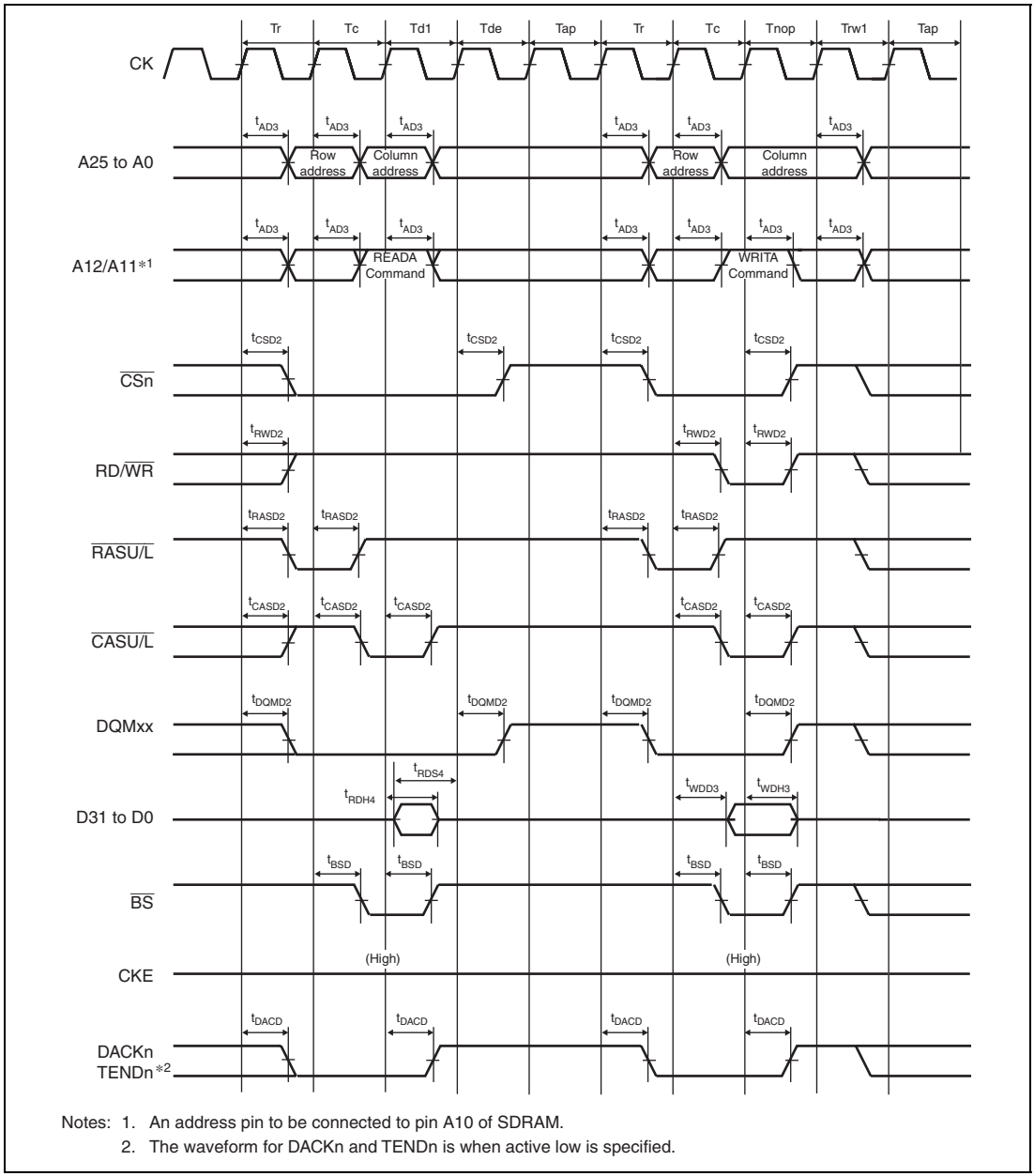
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.38 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



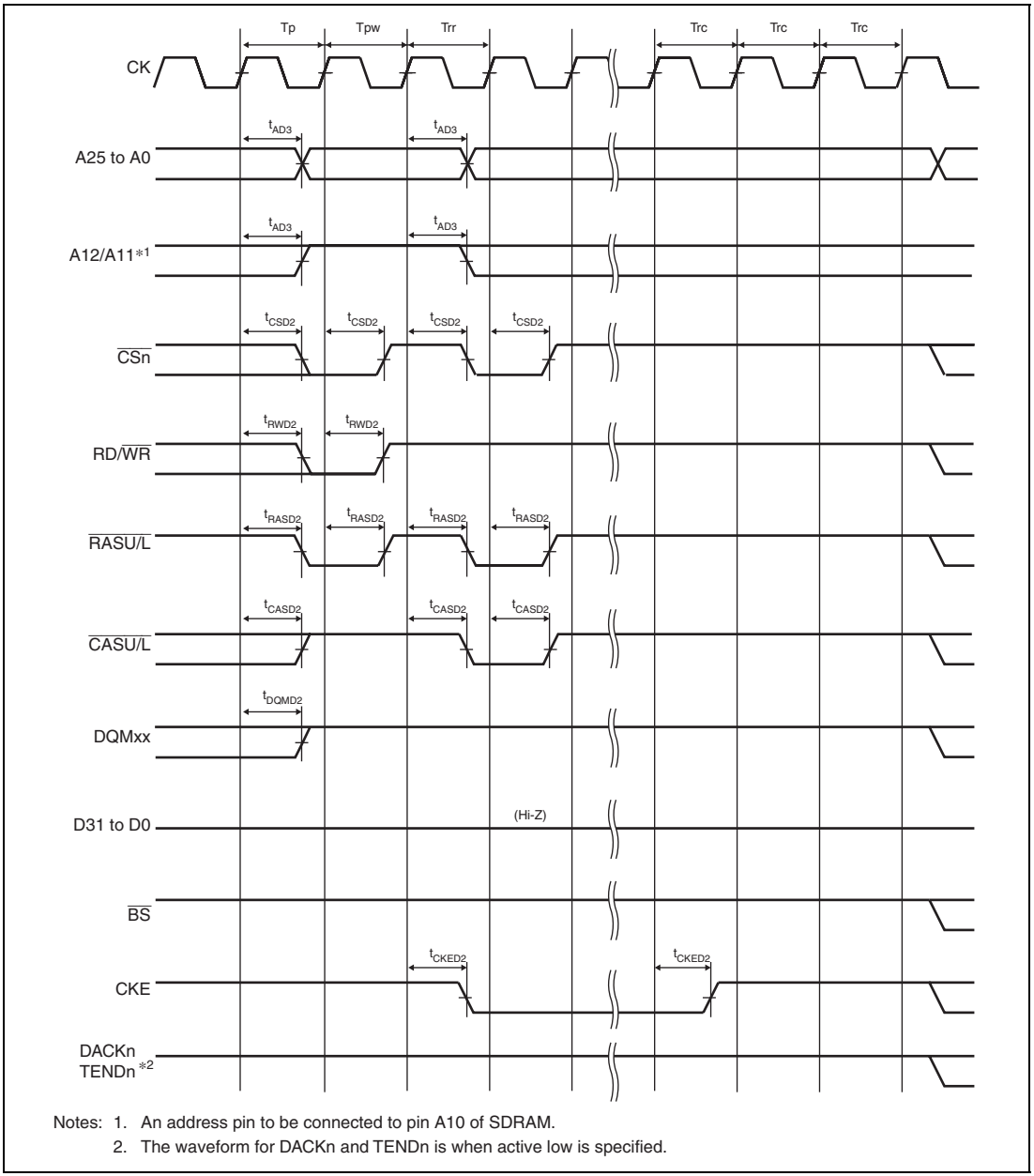
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.39 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.40 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge Mode, TRWL = 2 Cycles)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.41 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)

35.4.4 UBC Trigger Timing

Table 35.9 UBC Trigger Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	50	ns	35.42

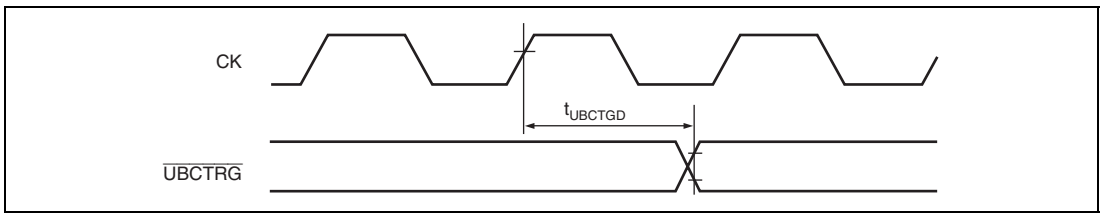


Figure 35.42 UBC Trigger Timing

35.4.5 DMAC Timing

Table 35.10 DMAC Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	20	—	ns	35.43
DREQ hold time	t_{DRQH}	20	—		
DACK, TEND delay time	t_{DACD}	—	20		35.44

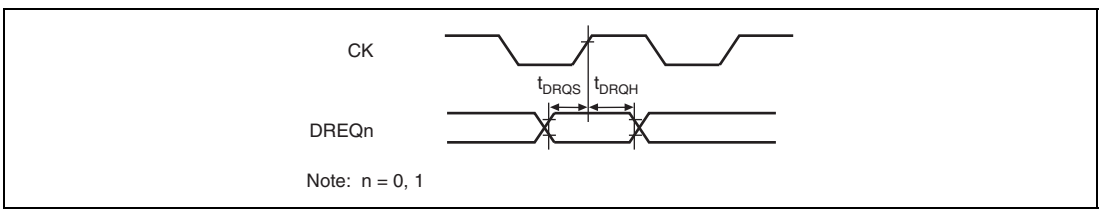


Figure 35.43 DREQ Input Timing

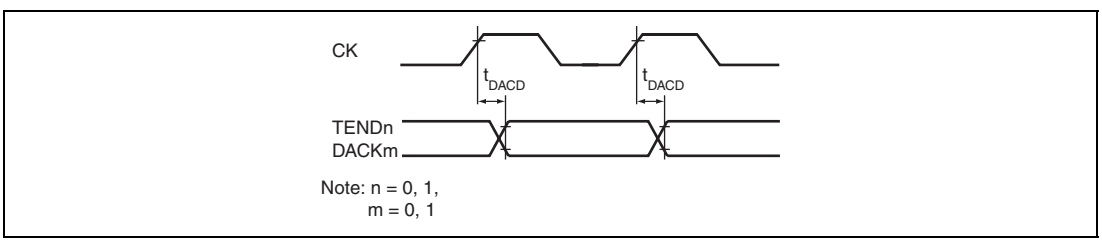


Figure 35.44 DACK, TEND Output Timing

35.4.6 MTU2 Timing

Table 35.11 MTU2 Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	35.45
Input capture input setup time	t_{TICS}	$t_{pcyc}/2 + 20$	—	ns	
Input capture input pulse width (single edge)	t_{TICW}	1.5	—	t_{pcyc}	
Input capture input pulse width (both edges)	t_{TICW}	2.5	—	t_{pcyc}	
Timer input setup time	t_{TCKS}	$t_{pcyc}/2 + 20$	—	ns	35.46
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{pcyc}	
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

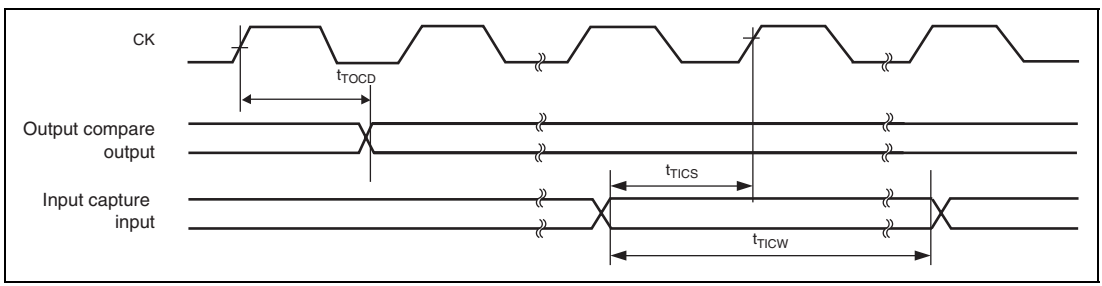


Figure 35.45 MTU2 Input/Output Timing

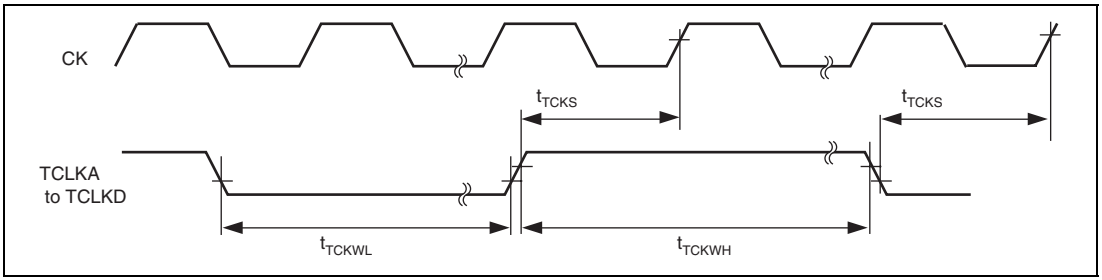


Figure 35.46 MTU2 Clock Input Timing

35.4.7 MTU2S Timing

Table 35.12 MTU2S Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	35.47
Input capture input setup time	t_{TICS}	$t_{mcyc}/2 + 20$	—	ns	
Input capture input pulse width (single edge)	t_{TICW}	1.5	—	t_{mcyc}	
Input capture input pulse width (both edges)	t_{TICW}	2.5	—	t_{mcyc}	

Note: t_{mcyc} indicates MTU2S clock (Mϕ) cycle.

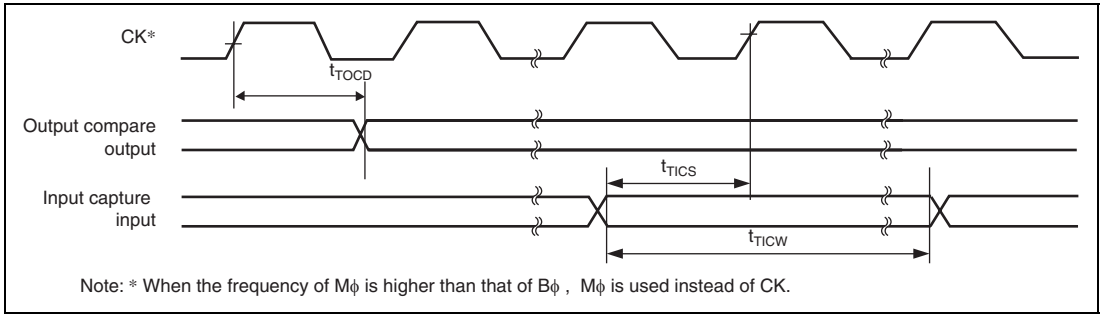


Figure 35.47 MTU2S Input/Output Timing

35.4.8 POE2 Timing

Table 35.13 POE2 Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{POE}}$ input setup time	t_{POES}	50	—	ns	35.48
$\overline{\text{POE}}$ input pulse width	t_{POEW}	1.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

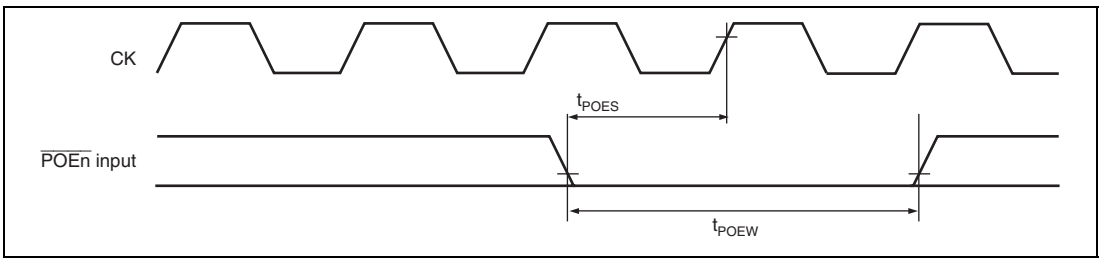


Figure 35.48 POE2 Input Timing

35.4.9 Watchdog Timer Timing

Table 35.14 Watchdog Timer Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	50	ns	35.49

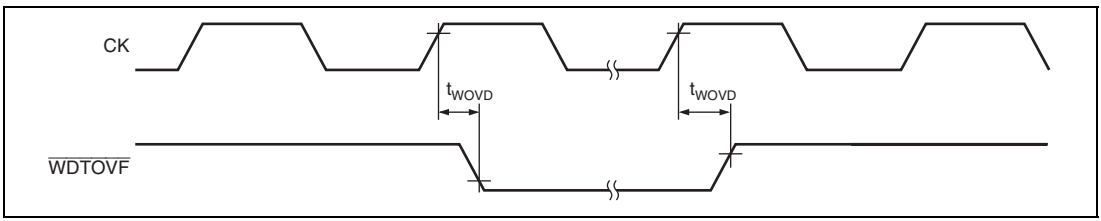


Figure 35.49 Watchdog Timer Timing

35.4.10 A/D Converter Timing

Table 35.15 A/D Converter Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Trigger input setup time	t_{TRGS}	20	—	ns	35.50

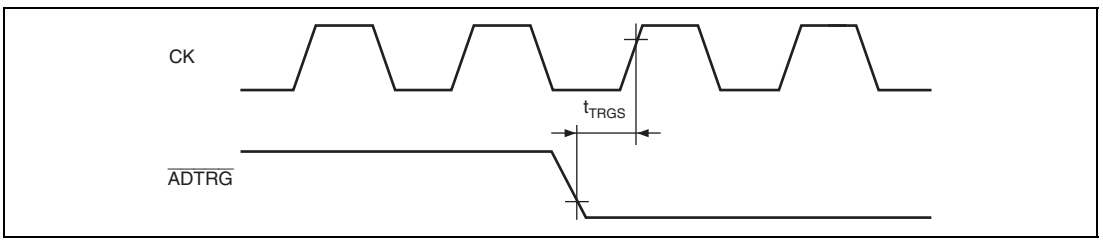


Figure 35.50 A/D Converter External Trigger Input Timing

35.4.11 SCI/SCIF Timing

Table 35.16 SCI/SCIF Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item		Symbol	Min.	Max.	Test Conditions	Unit	Figure
Input clock cycle (asynchronous)		tscyc	4	—		tpcyc	35.51, 35.52
Input clock cycle (clocked synchronous)		tscyc	6	—		tpcyc	
Input clock pulse width		t_{SCKW}	0.4	0.6		tscyc	
Input clock rise time		t_{SCKr}	—	1.5		tpcyc	
Input clock fall time		t_{SCKf}	—	1.5		tpcyc	
Transmit data delay time	Asynchronous	t_{TXD}	—	4tpcyc + 20	V_{CC}/PV_{CC1}/PV_{CC2} = 3.0 to 3.6 V	ns	35.52
			—	4tpcyc + 30	PV_{CC1}/PV_{CC2} = 1.65 to 1.95 V	ns	
Receive data setup time		t_{RXS}	4tpcyc	—		ns	
Receive data hold time		t_{RXH}	4tpcyc	—		ns	
Transmit data delay time	Clock synchronous	t_{TXD}	—	3tpcyc + 20	V_{CC}/PV_{CC1}/PV_{CC2} = 3.0 to 3.6V	ns	
			—	3tpcyc + 30	PV_{CC1}/PV_{CC2} = 1.65 to 1.95 V, Clock-synchronous slave operation	ns	
Receive data setup time		t_{RXS}	3tpcyc + 20	—		ns	
Receive data hold time		t_{RXH}	2tpcyc	—		ns	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

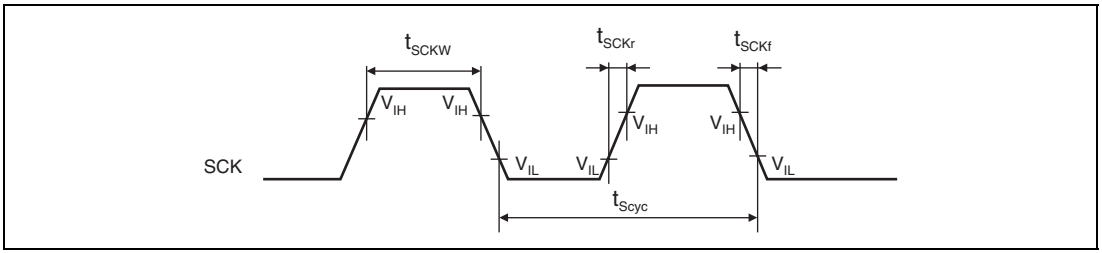


Figure 35.51 SCK Input Clock Timing

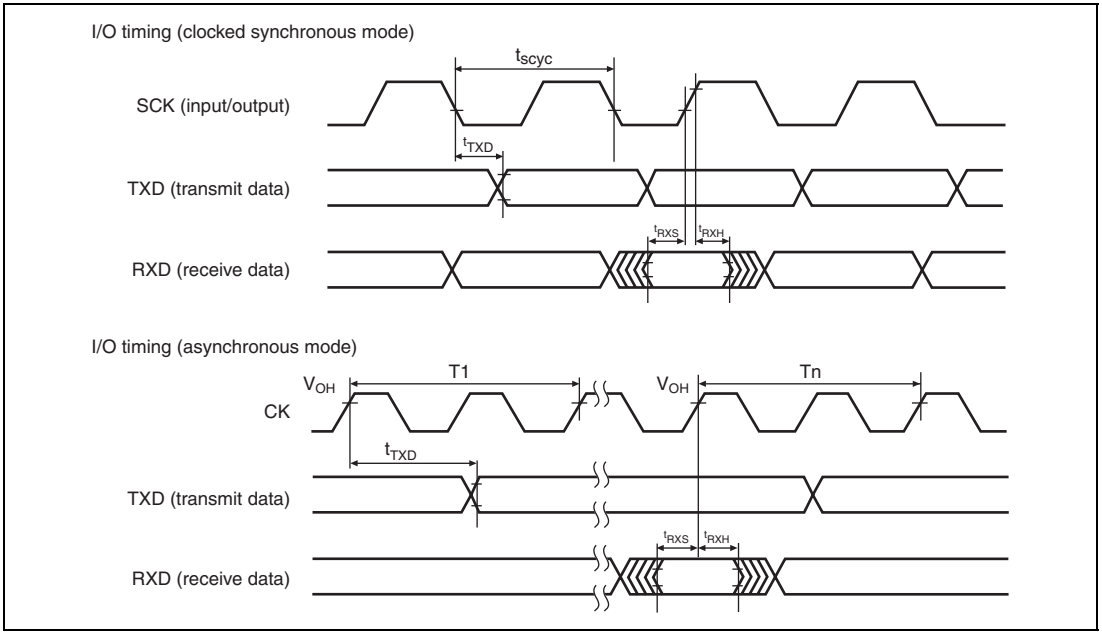


Figure 35.52 Input/Output Timing

35.4.12 I/O Port Timing

Table 35.17 I/O Port Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	50	ns	35.53
Input data setup time	t_{PORTS}	20	—		
Input data hold time	t_{PORTH}	20	—		

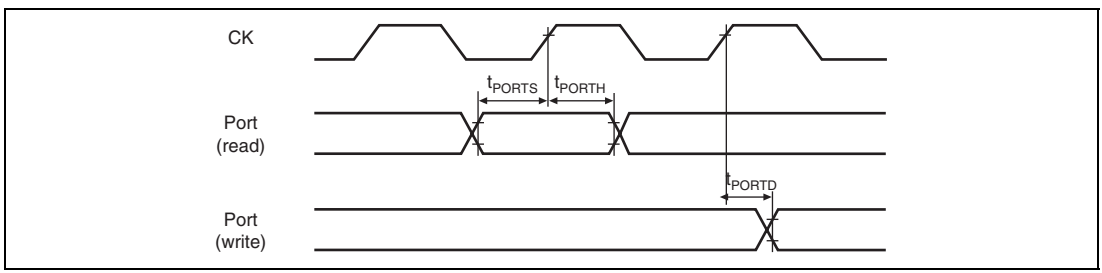


Figure 35.53 I/O Port Timing

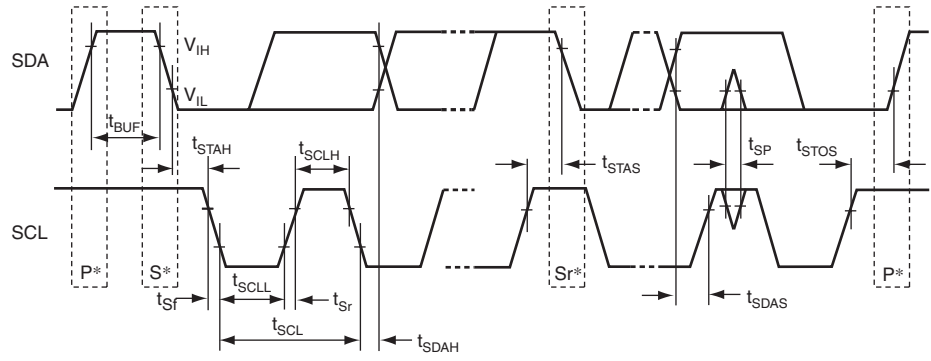
35.4.13 IIC3 Timing

Table 35.18 IIC3 Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

		Specifications			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
SCL input cycle time	t_{SCL}	$12t_{pcyc}$*1 + 600	—	ns	35.54
SCL input high pulse width	t_{SCLH}	$3t_{pcyc}$*1 + 300	—	ns	
SCL input low pulse width	t_{SCLL}	$5t_{pcyc}$*1 + 300	—	ns	
SCL, SDA input rise time	t_{Sr}	—	300	ns	
SCL, SDA input fall time	t_{Sf}	—	300	ns	
SCL, SDA input spike pulse removal time*2	t_{SP}	—	1, 2	t_{pcyc}*1	
SDA input bus free time	t_{BUF}	5	—	t_{pcyc}*1	
Start condition input hold time	t_{STAH}	3	—	t_{pcyc}*1	
Retransmit start condition input setup time	t_{STAS}	3	—	t_{pcyc}*1	
Stop condition input setup time	t_{STOS}	3	—	t_{pcyc}*1	
Data input setup time	t_{SDAS}	$1t_{pcyc}$*1 + 20	—	ns	
Data input hold time	t_{SDAH}	0	—	ns	
SCL, SDA capacitive load	Cb	0	400	pF	
SCL, SDA output fall time*3	t_{Sf}	—	250	ns	

Notes: 1. t_{pcyc} indicates peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.



[Legend]
S: Start condition
P: Stop condition
Sr: Start condition for retransmission

Figure 35.54 IIC3 Input/Output Timing

35.4.14 LVDS Timing (SH72315A only)

Table 35.19 LVDS Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Clock input cycle time	t_{CLKIN}	6.25	50	ns	35.55
Data input cycle time	t_{RXIN}	3.125	25	ns	
Data input setup time	t_{RXS}	0.75	—	ns	
Data input hold time	t_{RXH}	0.75	—	ns	
LVDS reception start settling time	t_{RCV}	5	—	µs	35.56

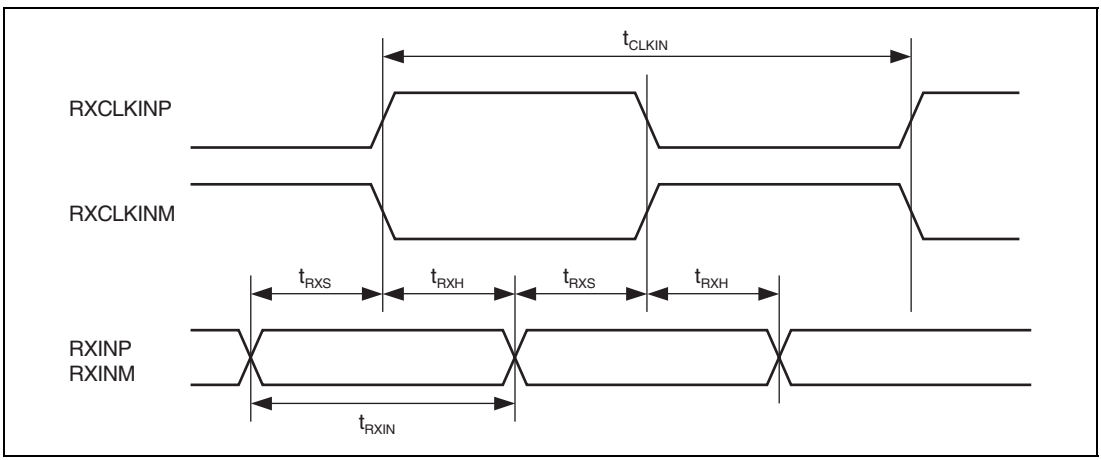


Figure 35.55 LVDS Input Timing

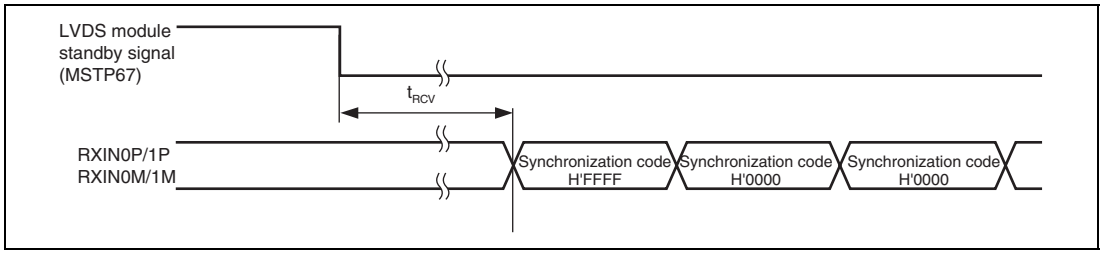


Figure 35.56 LVDS Reception Start Settling Time

35.4.15 Controller Area Network (RCAN-ET) Timing

Table 35.20 Controller Area Network (RCAN-ET) Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Transmit data delay time	t_{CTxD}	—	100	ns	35.57
Receive data setup time	t_{CRxS}	100	—	ns	
Receive data hold time	t_{CRxH}	100	—	ns	

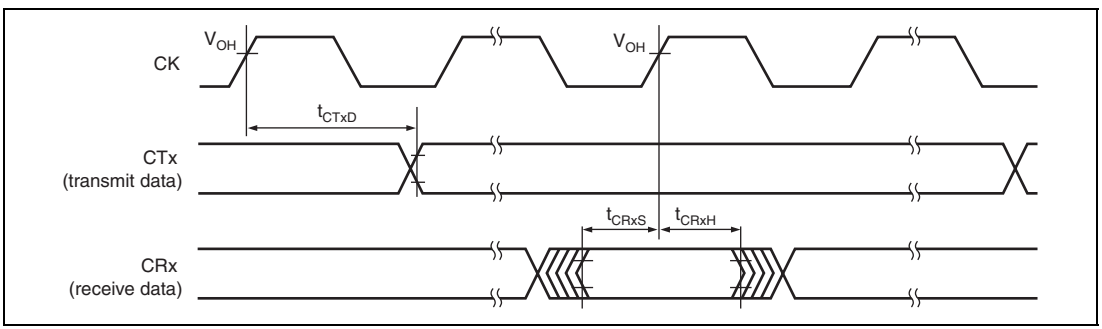


Figure 35.57 RCAN-ET Input/Output Timing

35.4.16 RSPI Timing

Table 35.21 RSPI Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item		Symbol	Min.	Max.	Test Conditions	Unit	Figure
RSPCK clock cycle*1	Master	t_{SPcyc}	2	4096		t_{Pcyc}	35.58
	Slave		8	4096			
RSPCK clock cycle 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 cycle 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*2	Output	t_{SPCKR}, t_{SPCKF}	—	5		ns	
	Input		—	1		t_{Pcyc}	
Data input setup time	Master	t_{SU}	25	—	PV_{CC1} = 3.0 to 3.6 V	ns	35.59 to 35.62
			35	—	PV_{CC1} = 1.65 to 1.95 V		
	Slave		20-2 × t_{Pcyc}	—			
Data input hold time	Master	t_H	0	—		ns	
	Slave		20+2 × t_{Pcyc}	—			
SSL setup time	Master	t_{LEAD}	1	8		t_{SPcyc}	
	Slave		4	—		t_{Pcyc}	
SSL hold time	Master	t_{LAG}	1	8		t_{SPcyc}	
	Slave		4	—		t_{Pcyc}	

Item		Symbol	Min.	Max.	Test Conditions	Unit	Figure
Data output delay time	Master	t_{OD}	—	15		ns	35.59 to 35.62
	Slave		—	$3 \times t_{Pcyc} + 25$	PV_{CC1} = 3.0 to 3.6 V		
			—	$3 \times t_{Pcyc} + 35$	PV_{CC1} = 1.65 to 1.95 V		
Data output hold time	Master	t_{OH}	0	—		ns	
	Slave		0	—			
Continuous transmission delay time	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$		ns	
	Slave		$4 \times t_{Pcyc}$	—			
MOSI, MISO rise/fall time※2	Master	t_{DR}, t_{DF}	—	5		ns	
	Slave		—	1		t_{Pcyc}	
SSL rise/fall time	Master	t_{SSLR}, t_{SSLF}	—	5		ns	
	Slave		—	1		t_{Pcyc}	
Slave access time			—	4		t_{Pcyc}	35.61, 35.62
Slave output release time			—	3		t_{Pcyc}	

Notes: 1. Set t_{Spcyc} so that its value is at least 80 ns.
2. When open drain output is specified, the above timing is not satisfied.

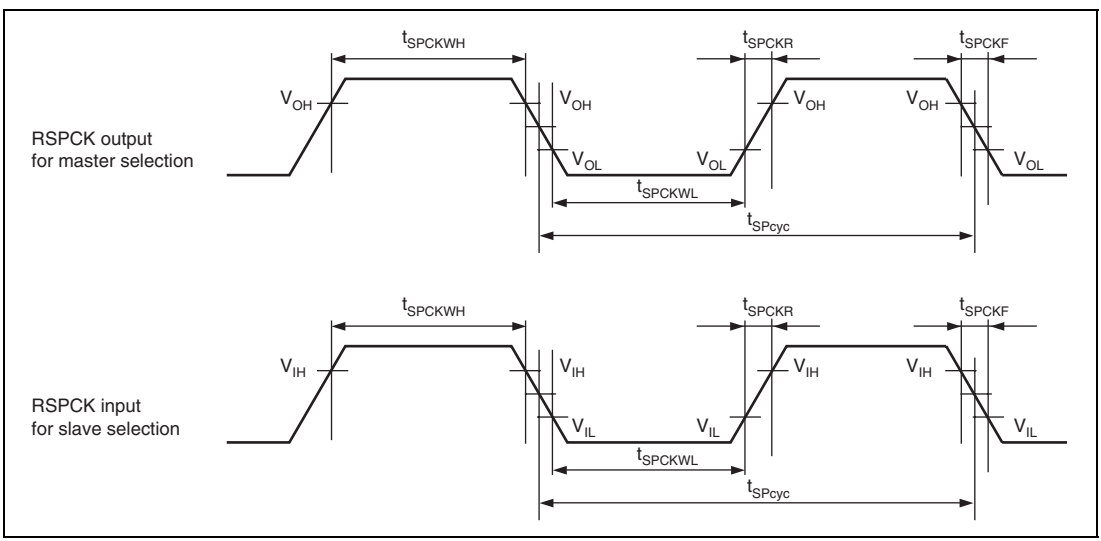


Figure 35.58 RSPI Clock Timing

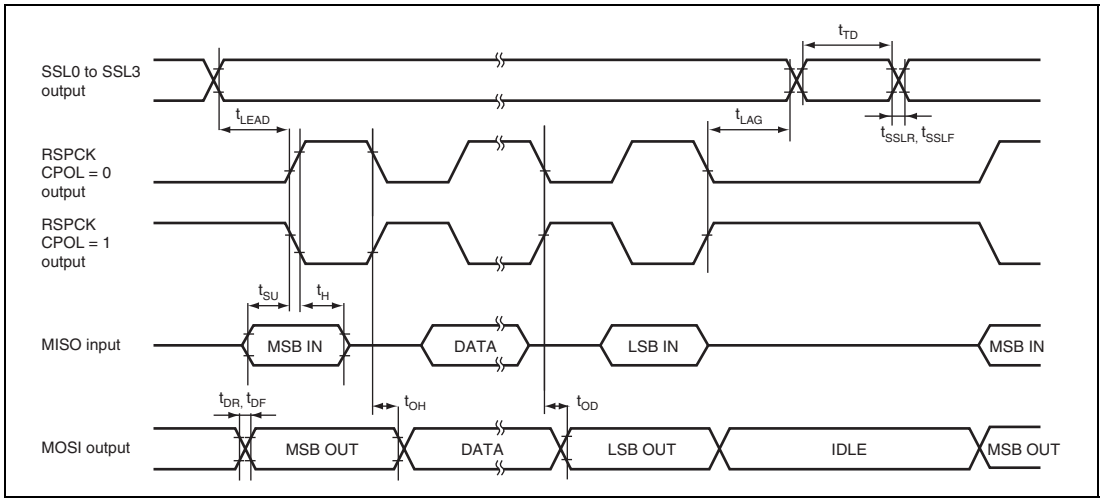


Figure 35.59 RSPI Timing (Master, CPHA = 0)

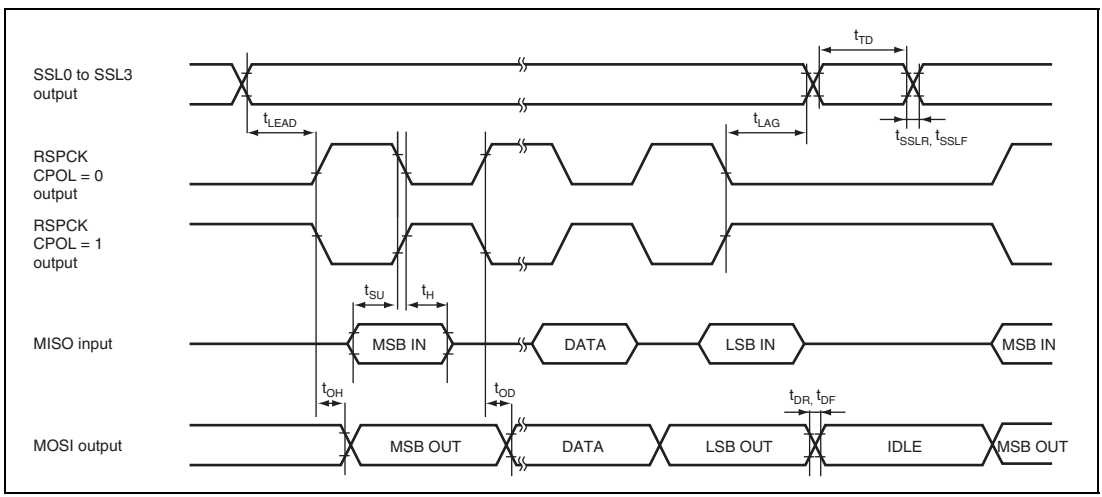


Figure 35.60 RSPI Timing (Master, CPHA = 1)

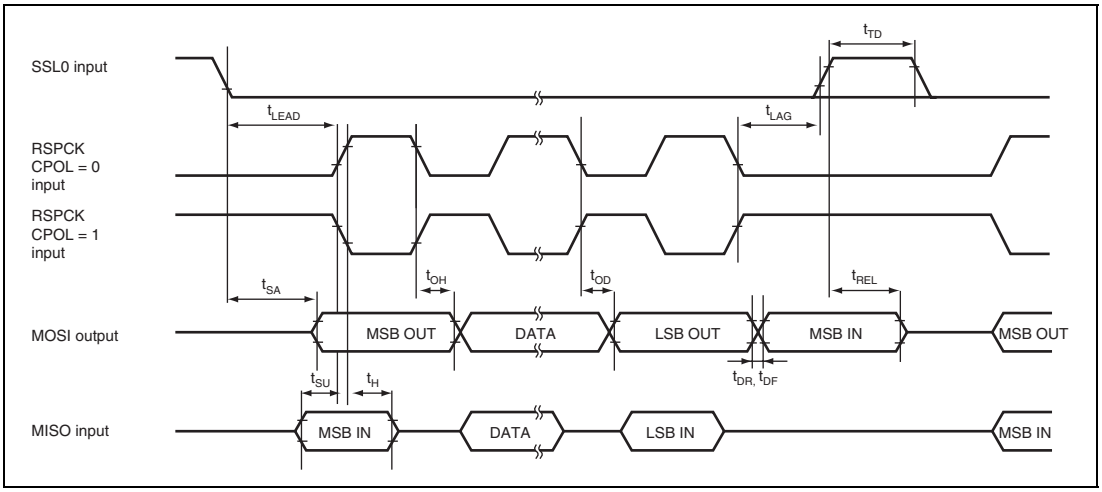


Figure 35.61 RSPI Timing (Slave, CPHA = 0)

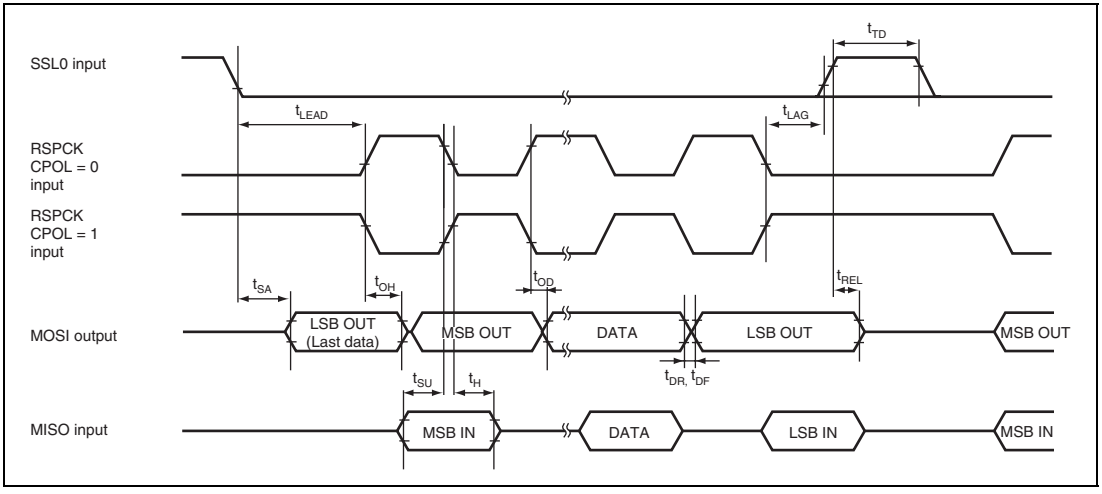


Figure 35.62 RSPI Timing (Slave, CPHA = 1)

35.4.17 CMT2 Timing

Table 35.22 CMT2 Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	35.63
Input capture input setup time	t_{TICS}	50	—	ns	
Input capture input pulse width (single edge)	t_{TICW}	1.5	—	t_{pcyc}	
Input capture input pulse width (both edges)	t_{TICW}	2.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

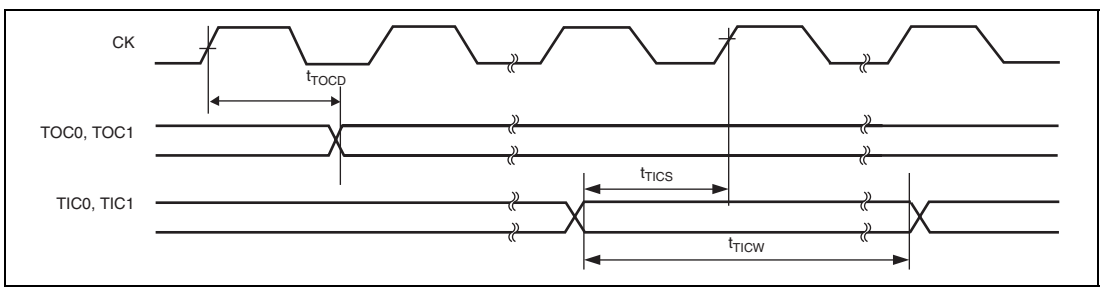


Figure 35.63 CMT2 Input/Output Timing

35.4.18 TIM32C Timing

Table 35.23 TIM32C Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Timer input setup time	t_{TI32S}	500	—	ns	35.64
Timer input pulse width	$t_{TI32WH/L}$	2.5	—	t_{cyc32}	

Note: t_{cyc32} indicates the cycle for the KEYC/TIM32C clock.

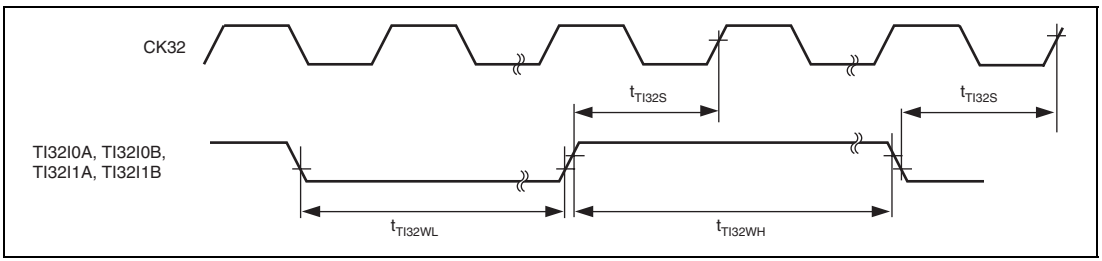


Figure 35.64 TIM32C Input/Output Timing

35.4.19 KEYC Timing

Table 35.24 KEYC Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Key input setup time	t_{KEYS}	500	—	ns	35.65
Key matrix input setup time	t_{PS}	500	—	ns	
Key matrix output delay time	t_{COMD}	—	500	ns	

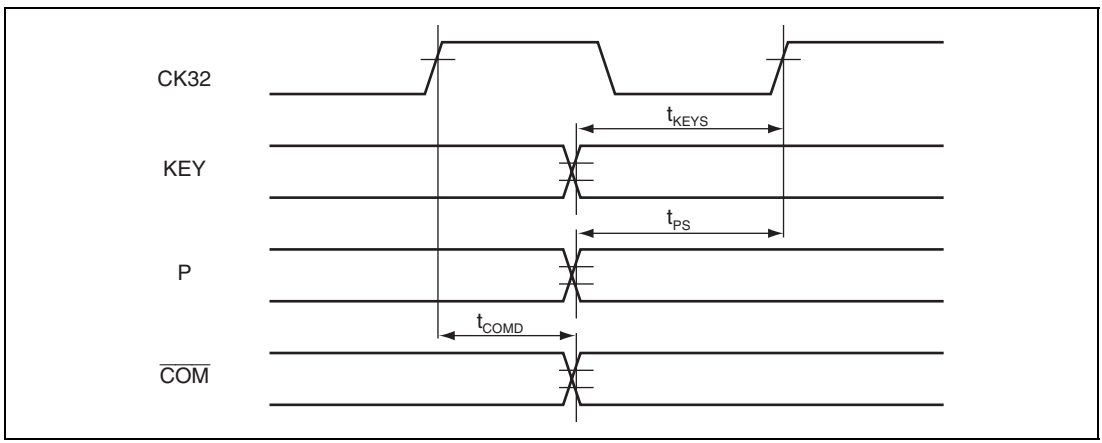


Figure 35.65 KEYC Input/Output Timing

35.4.20 H-UDI Timing

Table 35.25 H-UDI Timing

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50*	—	ns	35.66, 35.67
TCK high pulse width	t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK low pulse width	t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI setup time	t_{TDIS}	20	—	ns	35.67
TDI hold time	t_{TDIH}	20	—	ns	
TMS setup time	t_{TMSS}	20	—	ns	
TMS hold time	t_{TMSH}	20	—	ns	
TDO delay time	t_{TDOD}	—	30	ns	

Note: * This value must exceed the cycle time for the peripheral clock (Pϕ).

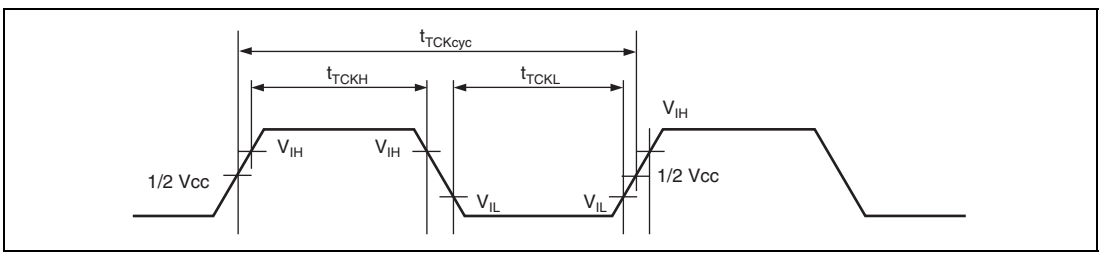


Figure 35.66 TCK Input Timing

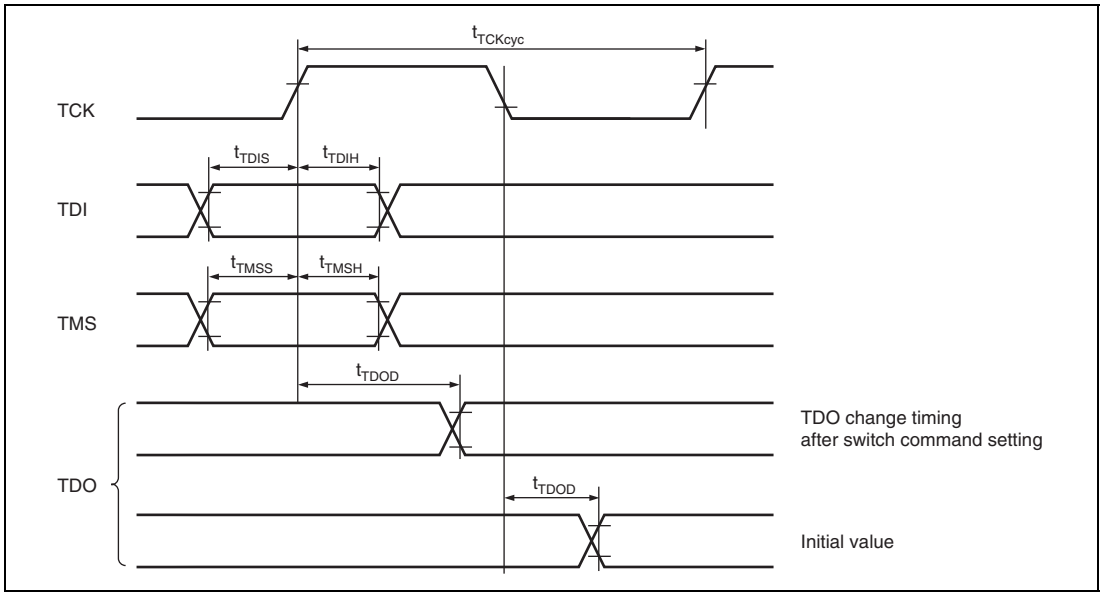
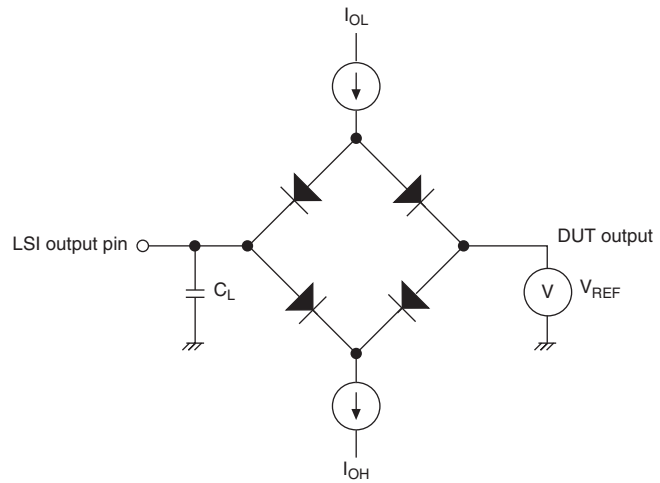


Figure 35.67 H-UDI Data Transmission Timing

35.4.21 AC Characteristics Measurement Conditions

- I/O signal level: V_{IL} (Max.)/V_{IH} (Min.)
- Output signal reference level: Vcc/2, PVcc1/2, PVcc2/2
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement tools. Each pin is set as follows:
20 pF: CK
30 pF: All pins other than the above
2. Test condition for I_{OL}/I_{OH} is −0.2 mA/1.6 mA (Vcc/PVcc1/PVcc2 = 3.0 to 3.6 V) or −0.2 mA/0.2 mA (PVcc1/PVcc2 = 1.65 to 1.95 V).

Figure 35.68 Output Load Circuit

35.5 A/D Converter Characteristics

Table 35.26 A/D Converter Characteristics

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	1.0	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	1.3	kΩ
Nonlinearity error (integral error)	—	—	±3.0*1	LSB
Offset error	—	—	±3.0*1	LSB
Full-scale error	—	—	±3.0*1	LSB
Quantization error	—	—	±0.5*1	LSB
Absolute accuracy	—	—	±4.0*2	LSB

Notes: 1. Reference value
2. This specification is applicable when AVcc = AVref.

35.6 Flash Memory Characteristics

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Table 35.27 Flash Memory Characteristics (1)

Item		Symbol	NPEC < 100 times			100 times ≤ NPEC < 100 times			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Programming time	256 bytes	tP256	—	2	12	—	2.4	14.4	ms	Pϕ = 50 MHz
	8 Kbytes	tP8K	—	45	100	—	54	120	ms	
	32 Kbytes	tP32K	—	180	400	—	216	480	ms	
	64 Kbytes	tP64K	—	360	800	—	432	960	ms	
	128 Kbytes	tP128K	—	720	1600	—	864	1920	ms	
	256 Kbytes	tP256K	—	1440	3200	—	1728	3840	ms	
Erasure time	8 Kbytes	tE8K	—	50	120	—	60	145	ms	
	32 Kbytes	tE32K	—	200	480	—	240	580	ms	
	64 Kbytes	tE64K	—	400	875	—	480	1050	ms	
	128 Kbytes	tE128K	—	800	1750	—	960	2100	ms	
	256 Kbytes	tE256K	—	1600	3500	—	1920	4200	ms	

Table 35.28 Flash Memory Characteristics (2)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Figures
Rewrite erase cycle*1	N_{PEC}	1000*2	—	—	Times	Pϕ = 50 MHz	35.69
Suspend delay time during writing	t_{SPD}	—	—	225	µs	Pϕ = 20 MHz	
		—	—	175	µs	Pϕ = 40 MHz	
		—	—	155	µs	Pϕ = 50 MHz	
First suspend delay time during erasing (in suspension priority mode)	t_{SESD1}	—	—	220	µs	Pϕ = 20 MHz	
		—	—	130	µs	Pϕ = 40 MHz	
		—	—	120	µs	Pϕ = 50 MHz	
Second suspend delay time during erasing (in suspension priority mode)	t_{SESD2}	—	—	1.7	ms	Pϕ = 50 MHz	
Suspend delay time during erasing (in erasing priority mode)	t_{SEED}	—	—	1.7	ms		
Data hold time*3	t_{DDRP}	10	—	—	Years		

Notes: 1. Definition of rewrite/erase cycle:
The rewrite/erase cycle is the number of erasing for each block. When the rewrite/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 256-byte writing is performed 32 times for different addresses in 8-Kbyte block and then the entire block is erased, the rewrite/erase cycle is counted as one. However, writing to the same address for several times as one erasing is not enabled (over writing is prohibited).

2. This indicates the minimum number that guarantees the characteristics after rewriting. (The guaranteed value is in the range from one to the minimum number.)

3. This indicates the characteristic when rewrite is performed within the specification range including the minimum number.

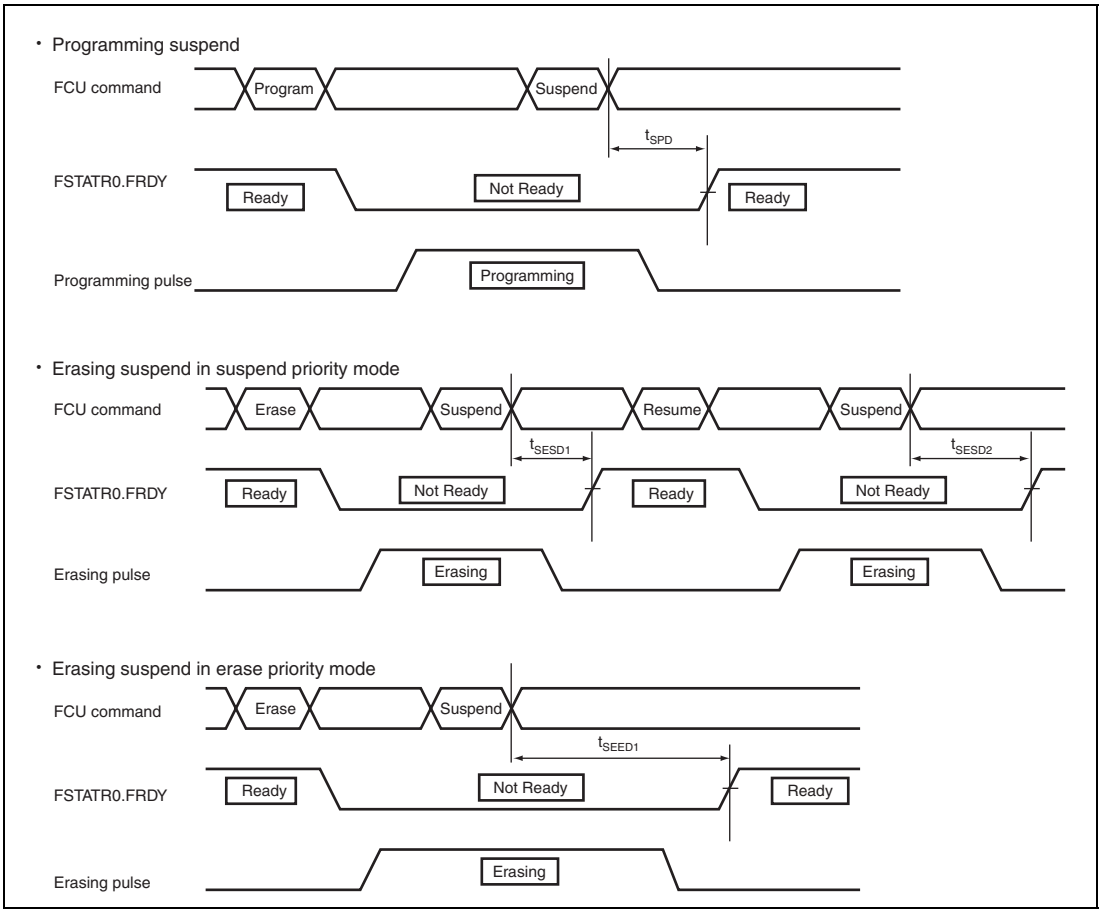


Figure 35.69 Flash Programming/Erasing Suspend Timing

35.7 FLD Characteristics

Table 35.28 FLD (Flash Memory for Data Storage) Characteristics

Conditions: Vcc = PLLVcc = AVcc = LVDSVcc (SH72315A only) = 3.0 to 3.6 V,
PVcc1 = PVcc2 = 1.65 to 1.95 V or 3.0 to 3.6 V, AVref = 3.0 to AVcc,
Vss = PLLVss = AVss = LVDSVss (SH72315A only) = PVss1 = PVss2 = 0 V,
Ta = -20 to +85°C (Consumer specifications),
Ta = -40 to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Figures
Programming time	8 bytes	t_{P8}	—	0.4	2	ms	Pϕ = 50 MHz	
	128 bytes	t_{P128}	—	1	5	ms		
Erasure time	8 Kbytes	t_{E8K}	—	300	900	ms	Pϕ = 50 MHz	
Blank check time	8 bytes	t_{BC8}	—	—	30	µs	Pϕ = 50 MHz	
	8 Kbytes	t_{BC8K}	—	—	2.5	ms		
Rewrite/erase cycle*1		N_{PEC}	1000*2	—	—	Times		
Suspend delay time during writing		t_{SPD}	—	—	225	µs	Pϕ = 20 MHz	35.69
			—	—	175		Pϕ = 40 MHz	
			—	—	155		Pϕ = 50 MHz	
First suspend delay time during erasing (in suspension priority mode)		t_{SESD1}	—	—	220	µs	Pϕ = 20 MHz	
			—	—	130		Pϕ = 40 MHz	
			—	—	120		Pϕ = 50 MHz	
Second suspend delay time during erasing (in suspension priority mode)		t_{SESD2}	—	—	1.7	ms		
Suspend delay time during erasing in erasure priority mode		t_{SEED}	—	—	1.7	ms		
Data hold time*3		t_{DDRP}	10	—	—	Years		

Notes: 1. Definition of rewrite/erase cycle:
The rewrite/erase cycle is the number of erasing for each block. When the rewrite/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 256-byte writing is performed 32 times for different addresses in 8-Kbyte block and then the entire block is erased, the rewrite/erase cycle is counted as one. However, writing to the same address for several times as one erasing is not enabled (over writing is prohibited).
2. This indicates the minimum number that guarantees the characteristics after rewriting. (The guaranteed value is in the range from one to the minimum number.)
3. This indicates the characteristic when rewrite is performed within the specification range including the minimum number.

35.8 Usage Note

35.8.1 Notes on Connecting VCL Capacitors

This LSI includes an internal step-down circuit to automatically reduce the internal power supply voltage to an appropriate level. Between this internal stepped-down power supply (VCL pin) and the VSS pin, a capacitor (0.1 μF) for stabilizing the internal voltage needs to be connected. Connection of the external capacitor is shown in figure 35.70. The external capacitor should be located near the pin. Do not apply any power supply voltage to the VCL pin.

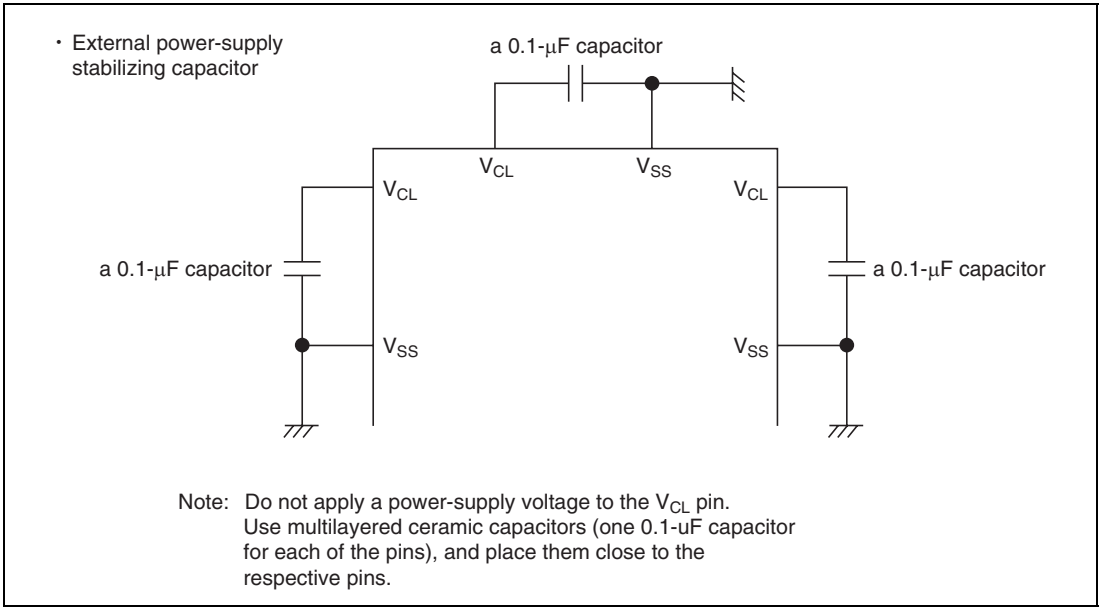


Figure 35.70 Connection of VCL Capacitors

Appendix

A. Pin States

Pin initial states differ according to MCU operating modes. Refer to section 22, Pin Function Controller (PFC), for details.

Table A.1 Pin States

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby※7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Clock	CK	O	O			Z	Z※12	Z	O	Z※12	O	O
	XTAL	O	O				L	L	O	O	O	O
	EXTAL	I	I				I	Z	I	I	I	I
	CK32	O/Z※10	O/Z※10				O/Z※10	O/Z※10	O/Z※10	O/Z※10	O/Z※10	O/Z※10
	XTAL32	O/L※10	O/L※10				O/L※10	O/L※10	O/L※10	O/L※10	O/L※10	O/L※10
	EXTAL32	I/Z※10	I/Z※10				I/Z※10	I/Z※10	I/Z※10	I/Z※10	I/Z※10	I/Z※10
System control	$\overline{\text{RES}}$	I	I				I	I	I	I	I	I
	$\overline{\text{MRES}}$ (PB2)	I	—				I	Z	I	I	I	I
	$\overline{\text{MRES}}$ (PE17)	I	—				I※4	Z	I	I	I※4	I
	$\overline{\text{WDTOVF}}$	O	H※6				H	H	O	O	O	O
	$\overline{\text{BREQ}}$	I	—				Z	Z	I	I	I	I
	$\overline{\text{BACK}}$	O	—				Z	Z	O	L	O	O
Operating mode control	MD0, MD1	I	I				I	I	I	I	I	I
	$\overline{\text{ASEMD0}}$	I	I				I	I	I	I	I	I
	FWE	I	I				I	I	I	I	I	I
Interrupt	NMI	I	I				I	I	I	I	I	I
	IRQ0 to IRQ8 (PA0, PA1, PA6 to PA12, PE0 to PE8)	I	—				I	Z	I	I	I	I
	IRQ0 to IRQ9 (PG0 to PG9)	I	—/I※8				I	I	I	I	I	I

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby※7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Interrupt	IRQ9 (PA13)	I	—				I	Z	I	I	I	I
	IRQ9 (PE9)	I	—				I※5	Z	I	I	I※5	I
	IRQ10	I	—				I	Z	I	I	I	I
	IRQ11 to IRQ15 (PA15 to PA19, PJ1, PJ2, PK0 to PK2)	I	—				I	Z	I	I	I	I
	IRQ11 to IRQ15 (PE11 to PE15)	I	—				I※5	Z	I	I	I※5	I
	IRQ16 to IRQ23 (PC8 to PC15)	I	—				I	Z	I	I	I	I
	$\overline{\text{IRQOUT}}$	O	—				H※1	Z	O	O	O	O
Address bus	A0 to A20	O	O		—		Z※3	Z	O	Z	O	O
	A21 to A25	O	—				Z※3	Z	O	Z	O	O
Data bus	D0 to D15	I/O/Z	Z		—		Z	Z	I/O	Z	I/O	I/O
	D16 to D31	I/O/Z	—	Z	—		Z	Z	I/O	Z	I/O	I/O
Bus control	$\overline{\text{WAIT}}$	I	—				Z	Z	I	Z	I	I
	$\overline{\text{CS0}}$ (PA0)	O	—				Z※3	Z	O	Z	O	O
	$\overline{\text{CS0}}$ (PB7)	O	H		—		Z※3	Z	O	Z	O	O
	$\overline{\text{CS1}}$ to $\overline{\text{CS7}}$	O	—				Z※3	Z	O	Z	O	O
	$\overline{\text{BS}}$	O	—				Z※3	Z	O	Z	O	O
	$\overline{\text{RASU}}$, $\overline{\text{RASL}}$	O	—				Z※2	Z	O	Z※2	O	O
	$\overline{\text{CASU}}$, $\overline{\text{CASL}}$	O	—				Z※2	Z	O	Z※2	O	O
	DQMUU, DQMUL, DQMLU, DQMLL	O	—				Z※3	Z	O	Z	O	O
	$\overline{\text{AH}}$	O	—				Z※3	Z	O	Z	O	O
	RD/$\overline{\text{WR}}$	O	—				Z※3	Z	O	Z	O	O
	$\overline{\text{RD}}$	O	H		—		Z※3	Z	O	Z	O	O
	$\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$	O	—	H	—		Z※3	Z	O	Z	O	O
	$\overline{\text{WRH}}$, $\overline{\text{WRL}}$	O	H		—		Z※3	Z	O	Z	O	O
	$\overline{\text{REFOUT}}$	O	—				H※1	Z	O	O	O	O
	CKE	O	—				Z※2	Z	O	Z※2	O	O

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby*7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
DMAC	DREQ0	I	—				Z	Z	I	I	I	I
	DREQ1 (PD25)	I	—				Z	Z	I	I	I	I
	DREQ1 (PE21)	I	—				I*4	Z	I	I	I*4	I
	DACK0 (PD27), DACK1 (PD26)	O	—				O*1	Z	O	O	O	O
	DACK0 (PE19), DACK1 (PE22)	O	—				Z (MZIZEH in HCPCR = 0)	Z	O	O	O*4	O
							O*1 (MZIZEH in HCPCR = 1)					
	TEND0 (PD23), TEND1 (PD28)	O	—				O*1	Z	O	O	O	O
	TEND0 (PE20), TEND1 (PE23)	O	—				Z (MZIZEH in HCPCR = 0)	Z	O	O	O*4	O
							O*1 (MZIZEH in HCPCR = 1)					
MTU2	TCLKA to TCLKD	I	—				Z	Z	I	I	I	I
	TIOC0A to TIOC0D	I/O	—				K*1	Z	I/O	I/O	I/O	Z
	TIOC1A, TIOC1B	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	TIOC2A, TIOC2B	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	TIOC3A, TIOC3C	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	TIOC3B, TIOC3D	I/O	—				Z (MZIZEL in HCPCR = 0)	Z	I/O	I/O	I/O*5	Z
							K*1 (MZIZEL in HCPCR = 1)					
	TIOC4A to TIOC4D	I/O	—				Z (MZIZEL in HCPCR = 0)	Z	I/O	I/O	I/O*5	Z
							K*1 (MZIZEL in HCPCR = 1)					

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby※7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
MTU2	TIC5U, TIC5V, TIC5W	I	—				Z	Z	I	I	I	I
MTU2S	TIOC3AS, TIOC3CS	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	TIOC3BS, TIOC3DS	I/O	—				Z (MZIZEH in HCPCR = 0) K※1 (MZIZEH in HCPCR = 1)	Z	I/O	I/O	I/O※4	Z
	TIOC4AS to TIOC4DS	I/O	—				Z (MZIZEH in HCPCR = 0) K※1 (MZIZEH in HCPCR = 1)	Z	I/O	I/O	I/O※4	Z
	TIC5US, TIC5VS, TIC5WS	I	—				Z	Z	I	I	I	I
POE2	$\overline{\text{POE0}}$ to $\overline{\text{POE8}}$	I	—				Z	Z	I	I	I	I
SCI	SCK0 (PA2, PB6), SCK1 (PA3, PD0)	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	SCK0 (PE15), SCK1 (PE9)	I/O	—				Z (MZIZEL in HCPCR = 0) K※1 (MZIZEL in HCPCR = 1)	Z	I/O	I/O	I/O※5	I/O
	SCK2, SCK3	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	RXD0 (PA0, PB4)	I	—				Z	Z	I	I	I	I
	RXD0 (PE17)	I	—				I※4	Z	I	I	I※4	I
	RXD1 (PA5, PD2)	I	—				Z	Z	I	I	I	I
	RXD1 (PE11)	I	—				I※5	Z	I	I	I※5	I
	RXD2, RXD3	I	—				Z	Z	I	I	I	I
	TXD0 to TXD3	O	—				O※1	Z	O	O	O	O

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM		Expansion with ROM	Single Chip						
Type	Pin Name	Normal State	16 Bits	32 Bits			Software Standby	Deep Software Standby*7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
SCIF	SCK4	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	SCK5 (PA15, PD10, PJ6)	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	SCK5 (PE12)	I/O	—				Z (MZIZEL in HCPCR = 0)	Z	I/O	I/O	I/O*5	I/O
							K*1 (MZIZEL in HCPCR = 1)					
	SCK6	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	SCK7 (PC13, PD3, PH6)	I/O	—				K*1	Z	I/O	I/O	I/O	I/O
	SCK7 (PE21)	I/O	—				Z (MZIZEH in HCPCR = 0)	Z	I/O	I/O	I/O*4	I/O
							K*1 (MZIZEH in HCPCR = 1)					
	RXD4	I	—				Z	Z	I	I	I	I
	RXD5 (PA13, PD12, PJ8)	I	—				Z	Z	I	I	I	I
	RXD5 (PE14)	I	—				I*5	Z	I	I	I*5	I
	RXD6 (PB11, PC9, PK5), RXD7 (PC15, PD5, PH8)	I	—				Z	Z	I	I	I	I
	RXD6 (PE20), RXD7 (PE23)	I	—				I*4	Z	I	I	I*4	I
	TXD4	O	—				O*1	Z	O	O	O	O
	TXD5 (PA14, PD11, PJ7)	O	—				O*1	Z	O	O	O	O
	TXD5 (PE13)	O	—				Z (MZIZEL in HCPCR = 0)	Z	O	O	O*5	O
							O*1 (MZIZEL in HCPCR = 1)					

Pin Function		Pin State									
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby※7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
SCIF	TXD6 (PB12, PC8, PK4), TXD7 (PC14, PD4, PH7)	O	—				O※1	Z	O	O	O	O
	TXD6 (PE19), TXD7 (PE22)	O	—				Z (MZIZEH in HCPCR = 0)	Z	O	O	O※4	O
							O※1 (MZIZEH in HCPCR = 1)					
IIC3	SCL	I/O	—				Z	Z	I/O	I/O	I/O	I/O
	SDA	I/O	—				Z	Z	I/O	I/O	I/O	I/O
UBC	$\overline{\text{UBCTRG}}$	O	—				O※1	Z	O	O	O	O
A/D converter	AN0 to AN15	I	—				Z	Z	I	I	I	I
	$\overline{\text{ADTRG}}$	I	—				Z	Z	I	I	I	I
CMT2	TIC0, TIC1	I	—				Z	Z	I	I	I	I
	TOC0, TOC1	O	—				O※1	Z	O	O	O	O
TIM32C	TI32I0A, TI32I0B	I	—/I※8				I	I	I	I	I	I
	TI32I1A, TI32I1B	I	—/I※8				I	I	I	I	I	I
LVDS (SH72315A only)	RXCLKINP, RXCLKINM	I	—				Z	Z	I	I	I	I
	RXIN0P, RXIN0M	I	—				Z	Z	I	I	I	I
	RXIN1P, RXIN1M	I	—				Z	Z	I	I	I	I
RSPI	RSPCK0	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	MOSI0	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	MISO0	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	SSL0	I/O	—				K※1	Z	I/O	I/O	I/O	I/O
	SSL1 to SSL3	O	—				O※1	Z	O	O	O	O
RCAN-ET	CRx0	I	—				Z	Z	I	I	I	I
	CTx0	O	—				O※1	Z	O	O	O	O
KEYC	KEY0 to KEY31	I	—/I※8				I	I	I	I	I	I
	P0 to P7	I	—/I※8				I	I	I	I	I	I
	$\overline{\text{COM0}}$ to $\overline{\text{COM7}}$	O	—/O※8				O	O	O	O	O	O

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby*7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
I/O port	PA0 to PA9	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PA10, PA11	I/O	Z	—	Z		K*1	Z	I/O	I/O	I/O	I/O
	PA12 to PA14	I/O	—		Z		K*1	Z	I/O	I/O	I/O	I/O
	PA15	I/O	—			Z	K*1	Z	I/O	I/O	I/O	I/O
	PA16 to PA19	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PB0, PB1	I/O	—		Z		K*1	Z	I/O	I/O	I/O	I/O
	PB2, PB3	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PB4 to PB7	I/O	—		Z		K*1	Z	I/O	I/O	I/O	I/O
	PB8 to PB13	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PC0 to PC15	I/O	—		Z		K*1	Z	I/O	I/O	I/O	I/O
	PD0 to PD15	I/O	—		Z		K*1	Z	I/O	I/O	I/O	I/O
	PD16 to PD31	I/O	Z	—	Z		K*1	Z	I/O	I/O	I/O	I/O
	PE0 to PE3	I/O	Z				K*1	Z	I/O	I/O	I/O	Z
	PE4 to PE8	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PE9	I/O	Z				Z (MZIZEL in HCPCR = 0) K*1 (MZIZEL in HCPCR = 1)	Z	I/O	I/O	I/O*5	Z
	PE10	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PE11 to PE15	I/O	Z				Z (MZIZEL in HCPCR = 0) K*1 (MZIZEL in HCPCR = 1)	Z	I/O	I/O	I/O*5	Z
	PE16	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PE17	I/O	Z				Z (MZIZEH in HCPCR = 0) K*1 (MZIZEH in HCPCR = 1)	Z	I/O	I/O	I/O*4	Z

Pin Function		Pin State										
			Reset State				Power-Down State					
			Expansion without ROM									
Type	Pin Name	Normal State	16 Bits	32 Bits	Expansion with ROM	Single Chip	Software Standby	Deep Software Standby*7	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
I/O port	PE18	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PE19 to PE23	I/O	Z				Z (MZIZEH in HCPCR = 0)	Z	I/O	I/O	I/O*4	Z
							K*1 (MZIZEH in HCPCR = 1)					
	PF0 to PF15	I	Z				Z	Z	I	I	I	I
	PG0 to PG15	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PH0 to PH15	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PJ0 to PJ15	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PK0 to PK7	I/O	Z				K*1	Z	I/O	I/O	I/O	I/O
	PL0 to PL5	I	Z				Z	Z	I	I	I	I
H-UDI*11	$\overline{\text{TRST}}$	PI	PI				Z	Z	PI	PI	PI	PI
	TCK	PI	PI				Z	Z	PI	PI	PI	PI
	TDI	PI	PI				Z	Z	PI	PI	PI	PI
	TMS	PI	PI				Z	Z	PI	PI	PI	PI
	TDO	O/Z*9	O/Z*9				Z	Z	O/Z*9	O/Z*9	O/Z*9	O/Z*9
Emulator*11	AUDCK	Z	Z				Z	Z	Z	Z	Z	Z
	$\overline{\text{AUDSYNC}}$	Z	Z				Z	Z	Z	Z	Z	Z
	AUDATA0 to AUDATA3	Z	Z				Z	Z	Z	Z	Z	Z
	$\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$	—	—				—	—	—	—	—	—

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High-impedance

K: Input pins become high-impedance, and output pins retain their state.

PI: Input (with pull-up resistor)

—: Invalid

$\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$ pin: This pin always has the FWE function in product chip mode.

Pins other than the $\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$ pin: These pins always function as general I/O pins during a reset.

Notes:
1. Output pins become high-impedance when the HIZ bit in the standby control register 3 (STBCR3) is set to 1.
2. Becomes output when the HIZCNT bit in the common control register (CMNCR) is set to 1.
3. Becomes output when the HIZMEM bit in the common control register (CMNCR) is set to 1.
4. Becomes high-impedance when the MZIZEH bit in the high-current port control register (HCPCR) is set to 0.
5. Becomes high-impedance when the MZIZEL bit in the high-current port control register (HCPCR) is set to 0.
6. Becomes input during a power-on reset. Pull-up to prevent erroneous operation. Pull-down with a resistance of at least 1 MΩ as required.
7. Although pulling up as set in the port G pull-up MOS control register (PGPCRL) is still effective while the chip is on deep software standby, settings for pulling up in pull-up MOS control registers other than that for port G (PAPCRH, PAPCRL, PBPCRL, PCPCRL, PDPCRH, PDPCRL, PEPCRH, PEPCRL, PHPCRL, PJPCRL, PKPCRL, PLPCRL) are not.
8. The input state is retained after reset-driven recovery from deep standby.
9. Z when the TAP controller of the H-UDI is in states other than shift-D and shift-IR.
10. Setting the MSTP75 bit in the standby control register 7 (STBCR7) to 1 places the 32-kHz oscillator on module standby.
11. This is the state during operation in product-chip mode (i.e. when $\overline{\text{ASEMD0}}$ = H). Refer to the emulation manual for details on the pin state in ASE mode (i.e. when $\overline{\text{ASEMD0}}$ = L).
12. This becomes an output when the HIZCK bit in the common control register (CMNCR) is set to 1.

B. States of Pins for Bus-Related Signals

Table B.1 States of Pins for Bus-Related Signals (1)

Pin Name		On-Chip ROM Space	On-Chip RAM Space	On-Chip Peripheral Module Space
$\overline{CS0}$ to $\overline{CS7}$		H	H	H
$\overline{BS}$		H	H	H
$\overline{RASU}$, $\overline{RASL}$		H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		H	H	H
DQMLL		H	H	H
$\overline{AH}$		H	H	H
RD/$\overline{WR}$	R	H	H	H
	W	—	H	H
$\overline{RD}$	R	H	H	H
	W	—	H	H
$\overline{WRHH}$	R	H	H	H
	W	—	H	H
$\overline{WRHL}$	R	H	H	H
	W	—	H	H
$\overline{WRH}$	R	H	H	H
	W	—	H	H
$\overline{WRL}$	R	H	H	H
	W	—	H	H
A25 to A0		Address*	Address*	Address*
D31 to D24		High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z
D15 to D8		High-Z	High-Z	High-Z
D7 to D0		High-Z	High-Z	High-Z

[Legend]

R: Read

W: Write

Note: * Indicates the address on the previous access to an external space.

Table B.1 States of Pins for Bus-Related Signals (2)

		External Space (Normal Space)			
			16-Bit Space		
Pin Name		8-Bit Space	Higher-Order Byte	Lower-Order Byte	Word/Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H
DQMUU		H	H	H	H
DQMUL		H	H	H	H
DQMLU		H	H	H	H
DQMLL		H	H	H	H
$\overline{AH}$		H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H
	W	L	L	L	L
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRL}$	R	H	H	H	H
	W	L	H	L	L
A25 to A0		Address	Address	Address	Address
D31 to D24		High-Z	High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z	High-Z
D15 to D8		High-Z	Data	High-Z	Data
D7 to D0		Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (3)

		External Space (Normal Space)						
		32-Bit Space						
Pin Name		Most Significant Byte	2nd Byte	3rd Byte	Least Significant Byte	Higher-Order Word	Lower-Order Word	Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H	H	H	H
DQMUU		H	H	H	H	H	H	H
DQMUL		H	H	H	H	H	H	H
DQMLU		H	H	H	H	H	H	H
DQMLL		H	H	H	H	H	H	H
$\overline{AH}$		H	H	H	H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L
$\overline{RD}$	R	L	L	L	L	L	L	L
	W	H	H	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	L	H	H	H	L	H	L
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	H	L	H	H	L	H	L
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	H	H	L	H	H	L	L
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	H	H	H	L	H	L	L
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D31 to D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23 to D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15 to D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7 to D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (4)

		External Space (SRAM with Byte Selection)		
		16-Bit Space		
Pin name		**Higher-Order Byte**	**Lower-Order Byte**	**Word/Longword**
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid
$\overline{BS}$		L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		H	H	H
DQMLL		H	H	H
$\overline{AH}$		H	H	H
RD/$\overline{WR}$	R	H	H	H
	W	L	L	L
$\overline{RD}$	R	L	L	L
	W	H	H	H
$\overline{WRHH}$	R	H	H	H
	W	H	H	H
$\overline{WRHL}$	R	H	H	H
	W	H	H	H
$\overline{WRH}$	R	L	H	L
	W	L	H	L
$\overline{WRL}$	R	H	L	L
	W	H	L	L
A25 to A0		Address	Address	Address
D31 to D24		High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z
D15 to D8		Data	High-Z	Data
D7 to D0		High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (5)

		External Space (SRAM with Byte Selection)						
		32-Bit Space						
Pin Name		Most Significant Byte	2nd Byte	3rd Byte	Least Significant Byte	Higher-Order Word	Lower-Order Word	Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H	H	H	H
DQMUU		H	H	H	H	H	H	H
DQMUL		H	H	H	H	H	H	H
DQMLU		H	H	H	H	H	H	H
DQMLL		H	H	H	H	H	H	H
$\overline{AH}$		H	H	H	H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L
$\overline{RD}$	R	L	L	L	L	L	L	L
	W	H	H	H	H	H	H	H
$\overline{WRHH}$	R	L	H	H	H	L	H	L
	W	L	H	H	H	L	H	L
$\overline{WRHL}$	R	H	L	H	H	L	H	L
	W	H	L	H	H	L	H	L
$\overline{WRH}$	R	H	H	L	H	H	L	L
	W	H	H	L	H	H	L	L
$\overline{WRL}$	R	H	H	H	L	H	L	L
	W	H	H	H	L	H	L	L
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D31 to D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23 to D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15 to D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7 to D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (6)

		External Space (Burst ROM: Clock Asynchronous)			
			16-Bit Space		
Pin Name		8-Bit Space	Higher-Order Byte	Lower-Order Byte	Word/Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H
DQMUU		H	H	H	H
DQMUL		H	H	H	H
DQMLU		H	H	H	H
DQMLL		H	H	H	H
$\overline{AH}$		H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H
	W	—	—	—	—
$\overline{RD}$	R	L	L	L	L
	W	—	—	—	—
$\overline{WRHH}$	R	H	H	H	H
	W	—	—	—	—
$\overline{WRHL}$	R	H	H	H	H
	W	—	—	—	—
$\overline{WRH}$	R	H	H	H	H
	W	—	—	—	—
$\overline{WRL}$	R	H	H	H	H
	W	—	—	—	—
A25 to A0		Address	Address	Address	Address
D31 to D24		High-Z	High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z	High-Z
D15 to D8		High-Z	Data	High-Z	Data
D7 to D0		Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (7)

		External Space (Burst ROM: Clock Asynchronous)						
		32-Bit Space						
Pin Name		Most Significant Byte	2nd Byte	3rd Byte	Least Significant Byte	Higher-Order Word	Lower-Order Word	Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H	H	H	H
DQMUU		H	H	H	H	H	H	H
DQMUL		H	H	H	H	H	H	H
DQMLU		H	H	H	H	H	H	H
DQMLL		H	H	H	H	H	H	H
$\overline{AH}$		H	H	H	H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{RD}$	R	L	L	L	L	L	L	L
	W	—	—	—	—	—	—	—
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D31 to D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23 to D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15 to D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7 to D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (8)

		External Space (Burst ROM: Clock Synchronous)		
		16-Bit Space		
Pin Name		**Higher-Order Byte**	**Lower-Order Byte**	**Word/Longword**
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid
$\overline{BS}$		L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		H	H	H
DQMLL		H	H	H
$\overline{AH}$		H	H	H
RD/$\overline{WR}$	R	H	H	H
	W	—	—	—
$\overline{RD}$	R	L	L	L
	W	—	—	—
$\overline{WRHH}$	R	H	H	H
	W	—	—	—
$\overline{WRHL}$	R	H	H	H
	W	—	—	—
$\overline{WRH}$	R	H	H	H
	W	—	—	—
$\overline{WRL}$	R	H	H	H
	W	—	—	—
A25 to A0		Address	Address	Address
D31 to D24		High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z
D15 to D8		Data	High-Z	Data
D7 to D0		High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (9)

		External Space (Burst ROM: Clock Synchronous)						
		32-Bit Space						
Pin Name		Most Significant Byte	2nd Byte	3rd Byte	Least Significant Byte	Higher-Order Word	Lower-Order Word	Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H	H	H	H
DQMUU		H	H	H	H	H	H	H
DQMUL		H	H	H	H	H	H	H
DQMLU		H	H	H	H	H	H	H
DQMLL		H	H	H	H	H	H	H
$\overline{AH}$		H	H	H	H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{RD}$	R	L	L	L	L	L	L	L
	W	—	—	—	—	—	—	—
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	—	—	—	—	—	—	—
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D31 to D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23 to D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15 to D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7 to D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

Table B.1 States of Pins for Bus-Related Signals (10)

		External Space (SDRAM)		
		16-Bit Space		
Pin Name		Higher-Order Byte	Lower-Order Byte	Word/Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid*[1]	Valid*[1]	Valid*[1]
$\overline{BS}$		L	L	L
$\overline{RASU}$, $\overline{RASL}$		Valid*[2]	Valid*[2]	Valid*[2]
$\overline{CASU}$, $\overline{CASL}$		Valid*[2]	Valid*[2]	Valid*[2]
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		L	H	L
DQMLL		H	L	L
$\overline{AH}$		H	H	H
RD/$\overline{WR}$	R	H	H	H
	W	L	L	L
$\overline{RD}$	R	H	H	H
	W	H	H	H
$\overline{WRHH}$	R	H	H	H
	W	H	H	H
$\overline{WRHL}$	R	H	H	H
	W	H	H	H
$\overline{WRH}$	R	H	H	H
	W	H	H	H
$\overline{WRL}$	R	H	H	H
	W	H	H	H
A25 to A0		Address	Address	Address
D31 to D24		High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z
D15 to D8		Data	High-Z	Data
D7 to D0		High-Z	Data	Data

[Legend]

R: Read

W: Write

Notes: 1. The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

2. If access was to an address with A25 = 0, $\overline{RASL}$/$\overline{CASL}$ = L, and if access was to an address with A25 = 1, $\overline{RASU}$/$\overline{CASU}$ = L.

Table B.1 States of Pins for Bus-Related Signals (11)

		External Space (SDRAM)						
		32-Bit Space						
Pin Name		Most Significant Byte	2nd Byte	3rd Byte	Least Significant Byte	Higher-Order Word	Lower-Order Word	Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid*1	Valid*1	Valid*1	Valid*1	Valid*1	Valid*1	Valid*1
$\overline{BS}$		L	L	L	L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		Valid*2	Valid*2	Valid*2	Valid*2	Valid*2	Valid*2	Valid*2
$\overline{CASU}$, $\overline{CASL}$		Valid*2	Valid*2	Valid*2	Valid*2	Valid*2	Valid*2	Valid*2
DQMUU		L	H	H	H	L	H	L
DQMUL		H	L	H	H	L	H	L
DQMLU		H	H	L	H	H	L	L
DQMLL		H	H	H	L	H	L	L
$\overline{AH}$		H	H	H	H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L
$\overline{RD}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D31 to D24		Data	High-Z	High-Z	High-Z	Data	High-Z	Data
D23 to D16		High-Z	Data	High-Z	High-Z	Data	High-Z	Data
D15 to D8		High-Z	High-Z	Data	High-Z	High-Z	Data	Data
D7 to D0		High-Z	High-Z	High-Z	Data	High-Z	Data	Data

[Legend]

R: Read

W: Write

Notes: 1. The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

2. If access was to an address with A25 = 0, $\overline{RASL}$/$\overline{CASL}$ = L, and if access was to an address with A25 = 1, $\overline{RASU}$/$\overline{CASU}$ = L.

Table B.1 States of Pins for Bus-Related Signals (12)

		External Space (MPX-I/O)			
			16-Bit Space		
Pin Name		8-Bit Space	Higher-Order Byte	Lower-Order Byte	Word/Longword
$\overline{CS0}$ to $\overline{CS7}$		Valid	Valid	Valid	Valid
$\overline{BS}$		L	L	L	L
$\overline{RASU}$, $\overline{RASL}$		H	H	H	H
$\overline{CASU}$, $\overline{CASL}$		H	H	H	H
DQMUU		H	H	H	H
DQMUL		H	H	H	H
DQMLU		H	H	H	H
DQMLL		H	H	H	H
$\overline{AH}$		H	H	H	H
RD/$\overline{WR}$	R	H	H	H	H
	W	L	L	L	L
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRL}$	R	H	H	H	H
	W	L	H	L	L
A25 to A0		Address	Address	Address	Address
D31 to D24		High-Z	High-Z	High-Z	High-Z
D23 to D16		High-Z	High-Z	High-Z	High-Z
D15 to D8		High-Z	Address/Data	Address	Address/Data
D7 to D0		Address/Data	Address	Address/Data	Address/Data

[Legend]

R: Read

W: Write

Valid: The chip-select signal for the accessed CS space is low, and the other chip-select signals are high.

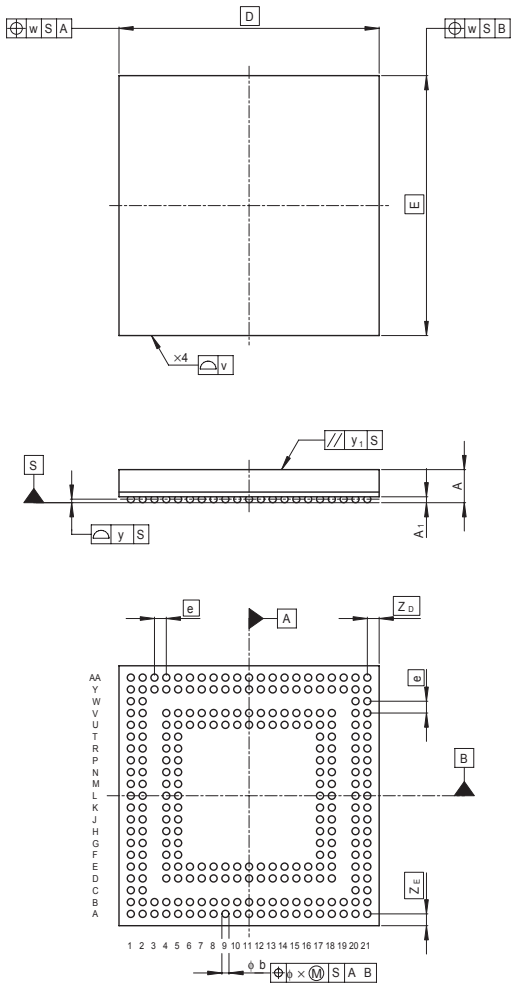
C. Product Code Lineup

Table C.1 Product Code Lineup

Product Type									
Product Name	Classification	ROM Capacity	RAM Capacity (High Speed)	RAM Capacity (for Retention)	LVDS Receive Interface	Application	Operating Temperature	Product Code	Package (Package Code)
SH72315A	F-ZTAT version	1 Mbyte	32 Kbytes	12 Kbytes	Yes	Consumer application	–20 to +85°C	R5F72315ANBG	P-LFBGA1111-256 (PLBG0256KA-B)
						Consumer application	–20 to +85°C	R5F72315ANBA	P-FBGA1717-272 (PRBG0272GA-A)
						Industrial application	–40 to +85°C	R5F72315ADBA	
SH72315L		1 Mbyte	32 Kbytes	12 Kbytes	No	Consumer application	–20 to +85°C	R5F72315LNBG	P-LFBGA1111-256 (PLBG0256KA-B)
						Consumer application	–20 to +85°C	R5F72315LNBA	P-FBGA1717-272 (PRBG0272GA-A)
						Industrial application	–40 to +85°C	R5F72315LDBA	
SH72314L		768 Kbytes	32 Kbytes	12 Kbytes	No	Consumer application	–20 to +85°C	R5F72314LNBG	P-LFBGA1111-256 (PLBG0256KA-B)
						Consumer application	–20 to +85°C	R5F72314LNBA	P-FBGA1717-272 (PRBG0272GA-A)
						Industrial application	–40 to +85°C	R5F72314LDBA	

D. Package Dimensions

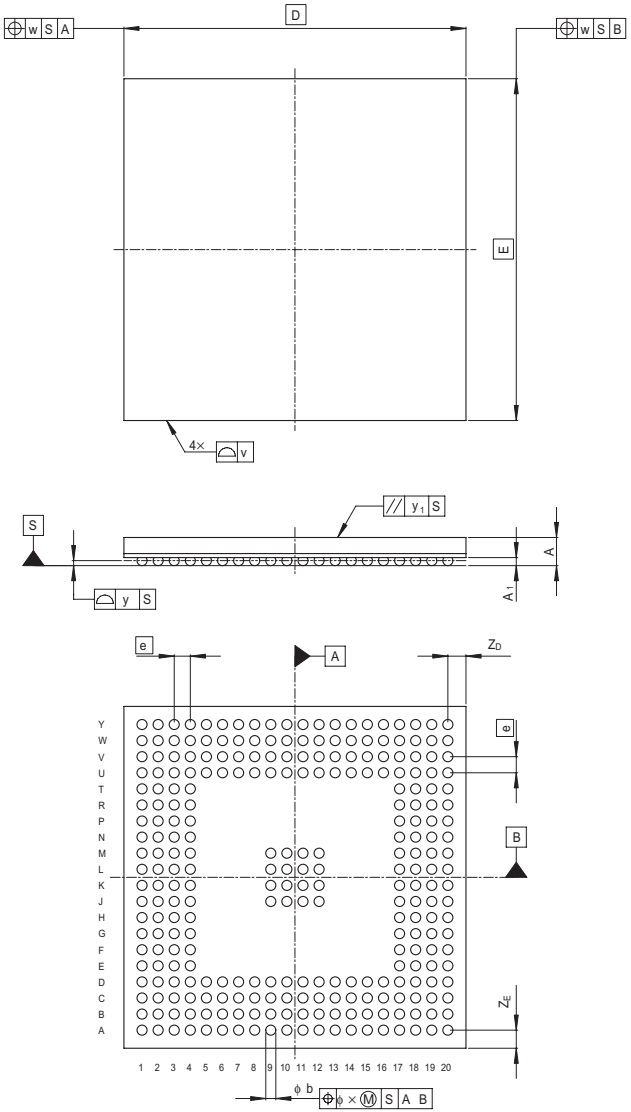
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA256-11x11-0.50	PLBG0256KA-B	—	0.3g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	11.00	—
E	—	11.00	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.20	0.25	0.30
e	—	0.50	—
b	0.25	0.30	0.35
x	—	—	0.05
y	—	—	0.08
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.5	—
Z_E	—	0.5	—

Figure D.1 Package Dimensions (1)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-FBGA272-17x17-0.80	PRBG0272GA-A	—	0.90g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	17.0	—
E	—	17.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.90
A_1	0.35	0.4	0.45
e	—	0.8	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.20
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.9	—
Z_E	—	0.9	—

Figure D.2 Package Dimensions (2)

Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)
1.1 SH7231 Features	1	Amended This LSI has a floating-point unit. It also includes on-chip peripheral functions required for system configuration, such as a large-capacity ROM, a 32-Kbyte high-speed on-chip RAM, 12-Kbyte RAM for data storage, ...
5.4.4 Oscillation Stop Detection Control Register (OSCCR)	127	Deleted **Bit** / **Bit Name** / **Initial Value** / **R/W** 2 / OSCSTOP / 0 / R~~/W~~
10.2 Input/Output Pins	290	Amended **Name** / **I/O** / **Function** WRHL/ DQMUL / Output / Indicates that D23 to D16 are being written to. Connected to the byte select signal when SRAM with byte selection is connected. Functions as the select signals for D23 to D16 when SDRAM is connected.
Table 10.19 Conditions for Determining Number of Idle Cycles	409	Added **No.** / **Condition** / **Description** / **Range** / **Note** (5) / Read data transfer cycle / One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not B'00. / 0 or 1* / One idle cycle is always generated after a read cycle with SDRAM interface. Note: * This is the case for consecutive read operations when the data read are stored in separate registers.
Figure 10.43 Comparison between Estimated Idle Cycles and Actual Value	413	Amended **Condition** / **R → R** [1] or [2] / 0 [3] or [4] / 0 [5] / 1 [6] / 0 [7] / 0 [5] + [6] + [7] / 1

Item	Page	Revision (See Manual for Details)

Table 11.4 DMARS Settings — 454 — Deleted

Peripheral Module	
SCIF_0	TXI0
	RXI0
SCIF_1	TXI1
	RXI1
SCIF_2	TXI2
	RXI2
SCIF_3	TXI3
	RXI3

Table 11.8 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits — 461 — Deleted

CHCR	DMARS		DMA Transfer Request Source	Transfer Source	Transfer Destination
RS[3:0]	MID	RID			
1000	100000	01	SCIF_0 transmit	Any	SCFTDR0
		10	SCIF_0 receive	SCFRDR0	Any
	100001	01	SCIF_1 transmit	Any	SCFTDR1
		10	SCIF_1 receive	SCFRDR1	Any
	100010	01	SCIF_2 transmit	Any	SCFTDR2
		10	SCIF_2 receive	SCFRDR2	Any
	100011	01	SCIF_3 transmit	Any	SCFTDR3
		10	SCIF_3 receive	SCFRDR3	Any

12.3.20 Timer Output Control Register 1 (TOCR1) — 555 — Added and amended

Bit	Bit Name	Description
1	OLSN	Output Level Select N*2*3 This bit selects the negative phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 12.30.
0	OLSP	Output Level Select P*2*3 This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 12.31.

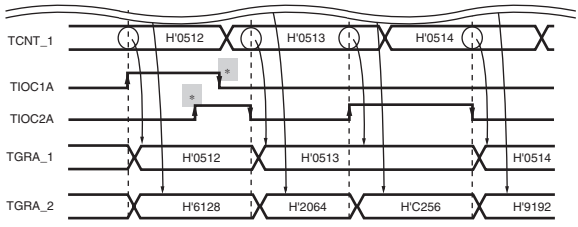
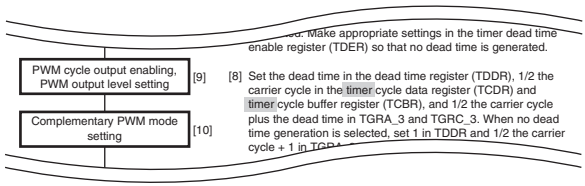
Notes: 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
2. Clearing the TOCS0 bit to 0 makes this bit setting valid.
3. The inverse-phase output is the exact inverse of the positive-phase output unless dead time is generated. When no dead time is generated, only the OLSP setting is valid.

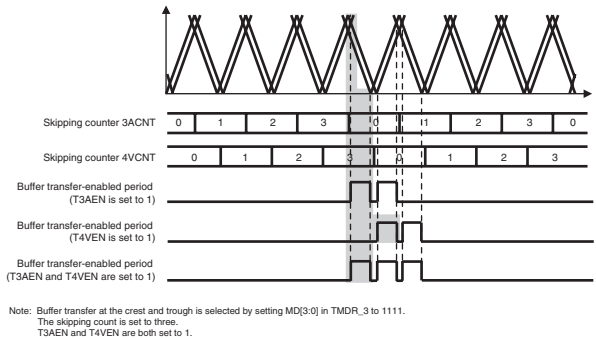
Item	Page	Revision (See Manual for Details)
12.3.21 Timer Output Control Register 2 (TOCR2)	557, 558	Added and amended

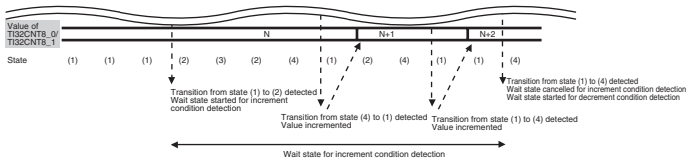
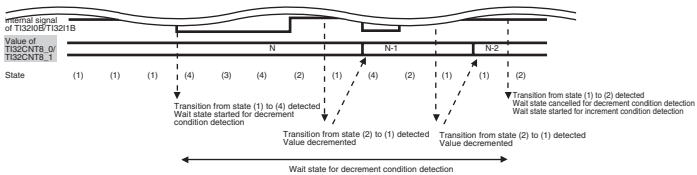
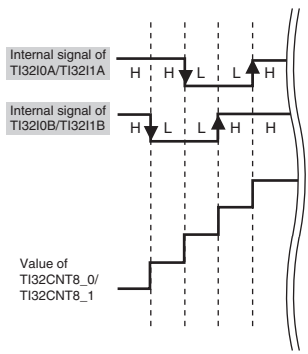
Bit	Bit Name	Description
7, 6	BF[1:0]	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBR to TOCR2. For details, see table 12.32.
5	OLS3N	Output Level Select 3N※1※2 This bit selects the output level on TIOC4D in reset-synchronized PWM mode/complementary PWM mode. See table 12.33.
4	OLS3P	Output Level Select 3P※1※2 This bit selects the output level on TIOC4B in reset-synchronized PWM mode/complementary PWM mode. See table 12.34.
3	OLS2N	Output Level Select 2N※1※2 This bit selects the output level on TIOC4C in reset-synchronized PWM mode/complementary PWM mode. See table 12.35.
2	OLS2P	Output Level Select 2P※1※2 This bit selects the output level on TIOC4A in reset-synchronized PWM mode/complementary PWM mode. See table 12.36.
1	OLS1N	Output Level Select 1N※1※2 This bit selects the output level on TIOC3D in reset-synchronized PWM mode/complementary PWM mode. See table 12.37.
0	OLS1P	Output Level Select 1P※1※2 This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 12.38.

Notes: 1. Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid.
2. The inverse-phase output is the exact inverse of the positive-phase output unless dead time is generated. When no dead time is generated, only the OLSiP setting is valid (i = 1, 2, 3).

12.4.4 Cascaded Operation	586, 587	Added For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). Edge detection as the condition for input capture is the detection of edges in the signal produced by taking the logical OR of the signals on the main and additional pins. For details, refer to (4), Cascaded Operation Example (c).

Item	Page	Revision (See Manual for Details)
Figure 12.23 Cascaded Operation Example (c)	589	Added and amended  Note: * When either of the input pins is at the high level, an edge on the other input pin does not act as an input capture condition.
12.4.5 PWM Modes	591	Amended PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by the cycle register compare match, the output value of each pin is the initial value set in TIOR.
Figure 12.38 Example of Complementary PWM Mode Setting Procedure	609	Amended 
12.4.8 Complementary PWM Mode (2) Outline of Complementary PWM Mode Operation (j) Complementary PWM Mode PWM Output Generation Method	622	Amended A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and compare register. While TCNTS is counting, compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. …… If compare-match **c** occurs first following compare-match **a**, as shown in figure 12.47, compare match **b** is ignored, and the negative phase is turned on by compare-match **d**. This is because turning o of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

Item	Page	Revision (See Manual for Details)
(k) Complementary PWM Mode 0% and 100% Duty Output	627	Amended 100% duty output is performed when the compare register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the compare register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.
Figure 12.78 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period	650	Added  Note: Buffer transfer at the crest and trough is selected by setting MD[3:0] in TMDR_3 to 1111. The skipping count is set to three. T3AEN and T4VEN are both set to 1.
Figure 12.110 TGI Interrupt Timing (Compare Match) (Channel 5)	679	Added Note: The compare match is generated even though TCNT is stopped.
20.4.9 Using the IICRST Bit to Reset I²C Bus Interface 3	1031	Added, this section
Figure 20.20 Sequence for Using the IICRST Bit to Reset I²C Bus Interface 3		Added, the figure
27.1 Features		Deleted ~~• Low-level signal sampling is possible with two-phase counter input pins.~~
27.3.1 Timer 32 Control Registers_0 to _2 (TI32CR_0 to TI32CR_2) (1) Timer 32 Control Registers_0 and _1 (TI32CR_0 and TI32CR_1)	1428	Amended Bit: 7 6 5 4 3 2 1 0 MS - - - - - CKS[1:0] Initial value: 0 0 0 0 0 0 0 0 R/W: R/W R R R R R R/W R/W **Bit** **Bit Name** **Description** 6 to 2 — Reserved These bits are always read as 0. The write value should always be 0.

Item	Page	Revision (See Manual for Details)
Figure 27.2 Example of Procedure for Setting Two-Phase Counter Operation	1440	Deleted [2] Set the operating mode of the two-phase counters using the MS bit, ~~low-level signal sampling condition using the LSE and LSS[1:0] bits,~~ and counter clock using the CKS[1:0] bits in TI32CR.
Figure 27.3 Example of TI32CNT8 Increment Operation in Normal Two-Phase Counter Mode	1441	Added 
Figure 27.4 Example of TI32CNT8 Decrement Operation in Normal Two-Phase Counter Mode	1442	Added 
Figure 27.5 Example of TI32CNT8 Operation in High-Speed Two-Phase Counter Mode	1443	Amended 
Table 29.11 FCU Command Format	1552	Added and amended

Command	Number of Command Cycles*	First Cycle	
		Address	Data

Note: * The numbers of cycles for commands are numbers of cycles of write access to programming/erasure addresses over the peripheral bus (P bus) by the CPU.

Item	Page	Revision (See Manual for Details)
Table 29.12 FCU Modes/States and Acceptable Commands	1556	Added and amended

	Status Read Mode								
Item	Programming/Erasure Processing	Programming processing while erasure is suspended	Programming/Erasure Suspension Processing	Lock Bit Read 2 Processing	Programming-Suspended	Erasure-Suspended	Command-Locked (FRDY = 0)	Command-Locked (FRDY = 1)	Other State
FRDY bit in FSTATR0	0	0	0	0	1	1	0	1	1
SUSRDY bit in FSTATR0	1	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0/1	0/1	0	1	0/1	0/1	0
PRGSPD bit in FSTATR0	0	0	0/1	0/1	1	0	0/1	0/1	0
CMDLK bit in FASTAT	0	0	0	0	0	0	1	1	0
Normal mode transition	×	×	×	×	A	A	×	×	A
Status read mode transition	×	×	×	×	A	A	×	×	A
Lock bit read mode transition	×	×	×	×	A	A	×	×	A
Program	×	×	×	×	×	*	×	×	A
Block erase	×	×	×	×	×	×	×	×	A
P/E suspend	A	×	×	×	×	×	×	×	×
P/E resume	×	×	×	×	A	A	×	×	×
Status register clear	×	×	×	×	A	A	×	A	A
Lock bit read 2	×	×	×	×	A	A	×	×	A
Lock bit program	×	×	×	×	×	*	×	×	A
Peripheral clock notification	×	×	×	×	×	×	×	×	A

Item	Page	Revision (See Manual for Details)
Table 29.13 Error Protection Types	1588	Added

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
Illegal command error	An undefined code has been specified in the first cycle of an FCU command.	1	0	0	0	0
	The value specified in the last of the multiple cycles of an FCU command is not H'D0.	1	0	0	0	0
	The peripheral clock specified in PCKAR is not in the range from 1 to 100 MHz.	1	0	0	0	0
	The command issued during programming or erasure is not a suspend command.	1	0	0	0	0
	A suspend command has been issued during operation that is neither programming nor erasure.	1	0	0	0	0
	A suspend command has been issued in suspended state.	1	0	0	0	0

29.11.10 Items Prohibited during Programming and Erasure	1595	Added • Setting the PCKAR register for a different frequency from that of Pϕ.
29.11.12 Abnormal Ending of Programming or Erasure	1595	Added, this section
30.1 Features	1598	Added The program used for reading can be executed from the on-chip RAM or ROM.
Table 30.7 FCU Command Formats (for FLD only)	1624	Added and amended

Command	Number of Command Cycles*	First Cycle	
		Address	Data

Note: * The numbers of cycles for commands are numbers of cycles of write access to programming/erasure addresses over the peripheral bus (P bus) by the CPU.

30.9.8 Items Prohibited During Programming and Erasure	1640	Added • Setting the PCKAR register for a different frequency from that of Pϕ.
30.9.10 Abnormal Ending of Programming or Erasure	1640	Added, this section

Item	Page	Revision (See Manual for Details)
Table 33.1 Pin Configuration	1696	Amended

Pin Name	Symbol	I/O	Function
ASE mode select pin	$\overline{\text{ASEMD0}}$*4	Input	If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RES}}$ pin is asserted, ASE mode is entered; if a high level is input, product chip mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RES}}$ negation.

Table 33.7 Reset Configuration	1714	Amended

$\overline{\text{ASEMD0}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State

33.1.1 Notes	1720	Amended

2. Fix the $\overline{\text{ASEMD0}}$ pin at high while executing the boundary scan.

34.2 Register Bits	1786	Amended

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
TIM32C	TI32CR_0	MS	—	—	—	—	—	CKS1	CKS0
	TI32CR_1	MS	—	—	—	—	—	CKS1	CKS0
	TI32CR_2	—	—	—	—	CCLR	—	CKS1	CKS0
	TI32SR	CH2F	—	—	—	CH1UF	CH1DF	CH0UF	CH0DF
	TI32IER	CH2IE	—	—	—	CH1UIE	CH1DIE	CH0UIE	CH0DIE

Table 35.6 Clock Timing	1823	Amended

Item	Symbol	Min.	Max.	Unit	Figure
CK clock output rise time	t_{CKOr}	—	3	ns	35.4
CK clock output fall time	t_{CKOf}	—	3	ns	

Item	Page	Revision (See Manual for Details)
Table 35.7 Control Signal Timing	1827, 1828	Added

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{RES}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*2*4	—	tcyc	35.7, 35.8, 35.11, 35.12
		1.5*4	—	µs	
$\overline{RES}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100	—	µs	
$\overline{RES}$ setup time*1	t_{RESS}	200	—	ns	
$\overline{RES}$ hold time	t_{RESH}	15	—	ns	

Notes: 1. $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and IRQ23 to IRQ0 are asynchronous signals. When these setup times are observed, a change of these signals is detected at the clock rising edge. If the setup times are not observed, detection of a signal change may be delayed until the next rising edge of the clock.

2. On return from standby, $t_{RESW} = t_{OSC2}$ (10 ms).
3. On return from standby, $t_{MRESW} = t_{OSC2}$ (10 ms).
4. Input the reset pulse over tRESW1 so that all conditions are met.

Table 35.8 Bus Timing — 1830 — Amended

Item	Symbol	Min.	Max.	Unit	Figure
Read data setup time 1	tRDS1	1/2tcyc + 14	—	ns	35.15 to 35.21
Read data setup time 2	tRDS2	14	—	ns	35.23 to 35.26, 35.31 to 35.33
Read data setup time 3	tRDS3	1/2tcyc + 14	—	ns	35.22
Read data setup time 4	tRDS4	1/2tcyc + 14	—	ns	35.40

Table 35.27 Flash Memory Characteristics (1) — 1886 — Deleted

Item		Symbol	NPEC < 100 times Min.	Typ.	Max.	100 times ≤ NPEC < 100 times Min.	Typ.	Max.	Unit	Test Conditions
Erasure time	8 Kbytes	tE8K	—	50	120	—	60	145	ms	Pϕ = 50 MHz
	32 Kbytes	tE32K	—	200	480	—	240	580	ms	
	64 Kbytes	tE64K	—	400	875	—	480	1050	ms	
	128 Kbytes	tE128K	—	800	1750	—	960	2100	ms	
	256 Kbytes	tE256K	—	1600	3500	—	1920	4200	ms	
	~~12 Kbytes*4~~	~~tE12K~~	~~—~~	~~75~~	~~180~~	~~—~~	~~90~~	~~220~~	~~ms~~	

Table 35.28 Flash Memory Characteristics (2) — 1887 — Deleted

Notes:

~~4. The size of the user boot mat is 32 KB.~~

Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

SH7231 Group User's Manual: Hardware

Publication Date: Rev.1.00 May 20, 2011
Rev.2.00 Mar 19, 2012

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

SH7231 Group

Renesas Electronics Corporation

R01UH0073EJ0200